Cisco Prime IP Express 8.2 User Guide

March 2014

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: OL-31070-01
The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB’s public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

Notwithstanding any other warranty herein, all document files and software of these suppliers are provided “as is” with all faults. Cisco and the above-named suppliers disclaim all warranties, expressed or implied, including, without limitation, those of merchantability, fitness for a particular purpose and noninfringement or arising from a course of dealing, usage, or trade practice.

In no event shall Cisco or its suppliers be liable for any indirect, special, consequential, or incidental damages, including, without limitation, lost profits or loss or damage to data arising out of the use or inability to use this manual, even if Cisco or its suppliers have been advised of the possibility of such damages.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco Prime IP Express 8.2 User Guide
Copyright © 2014 Cisco Systems, Inc. All rights reserved.
## Preface

## CONTENTS

### Part 1: Getting Started

#### Chapter 1: Cisco Prime IP Express Components

- Management Components 1-1
- Simple Network Management 1-2
  - Setting Up the SNMP Server 1-3
  - How Notification Works 1-4
  - Handling SNMP Notification Events 1-5
  - Server Up/Down Traps 1-7
  - Handling SNMP Queries 1-8
  - Integrating Cisco Prime IP Express SNMP into System SNMP 1-9
- Default Ports for Cisco Prime IP Express Services 1-9

#### Chapter 2: Cisco Prime IP Express User Interfaces

- Introduction to the Web-Based User Interfaces 2-1
  - Supported Web Browsers 2-2
  - Access Security 2-2
  - Logging In to the Web UIs 2-2
  - Multiple Users 2-3
  - Changing Passwords 2-4
  - Navigating the Web UIs 2-4
  - Waiting for Page Resolution Before Proceeding 2-4
  - Committing Changes in the Web UIs 2-5
  - Role and Attribute Visibility Settings 2-5
  - Displaying and Modifying Attributes 2-5
    - Grouping and Sorting Attributes 2-5
    - Modifying Attributes 2-6
    - Displaying Attribute Help 2-6
  - Left Navigation Pane and Quick View Icon 2-6
  - Help Pages 2-6
  - Logging Out 2-6
  - Local Cluster Web UI 2-7
<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Basic Main Menu Page</td>
<td>2-7</td>
</tr>
<tr>
<td>Local Advanced Main Menu Page</td>
<td>2-8</td>
</tr>
<tr>
<td>Setting Local User Preferences</td>
<td>2-9</td>
</tr>
<tr>
<td>Configuring Clusters in the Local Web UI</td>
<td>2-9</td>
</tr>
<tr>
<td>Regional Cluster Web UI</td>
<td>2-10</td>
</tr>
<tr>
<td>Command Line Interface</td>
<td>2-10</td>
</tr>
<tr>
<td>Central Configuration Management Server</td>
<td>2-11</td>
</tr>
<tr>
<td>Managing CCM Server</td>
<td>2-11</td>
</tr>
<tr>
<td>Editing CCM Server Properties</td>
<td>2-11</td>
</tr>
<tr>
<td>Bring Your Own Device Web Server</td>
<td>2-12</td>
</tr>
<tr>
<td>Managing BYOD Web Server</td>
<td>2-12</td>
</tr>
<tr>
<td>Editing BYOD Web Server Properties</td>
<td>2-12</td>
</tr>
<tr>
<td>Setting Up BYOD Theme and Content</td>
<td>2-12</td>
</tr>
<tr>
<td>Adding BYOD Themes</td>
<td>2-13</td>
</tr>
<tr>
<td>Adding BYOD Content</td>
<td>2-13</td>
</tr>
<tr>
<td>Global Search in Prime IP Express</td>
<td>2-13</td>
</tr>
</tbody>
</table>

**CHAPTER 3 Server Status Dashboard** 3-1

Opening the Dashboard 3-1

Display Types 3-2

- Tables 3-3
- Line Charts 3-3
- Stacked Area Charts 3-5
- Other Chart Types 3-5

Getting Help for the Dashboard Elements 3-6

Customizing the Display 3-6

Selecting Dashboard Elements to Include 3-7

- Configuring Server Chart Types 3-8

Host Metrics 3-9

- System Metrics 3-9
- JVM Memory Utilization 3-10

DHCP Metrics 3-10

- DHCP Server Request Activity 3-11
- DHCP Server Response Activity 3-11
- DHCP Buffer Capacity 3-12
- DHCP Response Latency 3-13
- DHCP DNS Updates 3-13
- DHCP Address Current Utilization 3-14
Contents

- DHCP Failover Status 3-15
- DHCP General Indicators 3-15
- DHCP Server Lease Data 3-16
- Authoritative DNS Metrics 3-16
  - DNS Outbound Zone Transfers 3-16
  - DNS Inbound Zone Transfers 3-17
  - DNS Network Errors 3-18
  - DNS Related Servers Errors 3-18
  - DNS General Indicators 3-19
  - DNS Queries Per Second 3-19
- Caching DNS Metrics 3-20
  - DNS Queries Type 3-20
  - DNS Queries Responses 3-20
  - DNS Incoming Queries 3-21
  - DNS Recursive Query Time 3-21
  - DNS Caching 3-22
  - Caching DNS General Indicators 3-22
  - DNS Caching Server Queries Per Second 3-23

**CHAPTER 4**

Deploying Cisco Prime IP Express 4-1
  - Target Users 4-1
  - Regional and Local Clusters 4-1
  - Deployment Scenarios 4-2
    - Small-to-Medium-Size LANs 4-3
    - Large Enterprise Network 4-3
  - Configuration and Performance Guidelines 4-5
    - General Configuration Guidelines 4-5
    - Special Configuration Cases 4-6

**PART 2**

Local and Regional Administration

**CHAPTER 5**

Configuring Administrators 5-1
  - Administrators, Groups, and Roles 5-1
    - How Administrators Relate to Groups, and Roles 5-2
    - Administrator Types 5-2
    - Roles, Subroles, and Constraints 5-2
    - Groups 5-5
  - Managing Administrators 5-5
  - Managing Passwords 5-6
Backup and Recovery  8-1

Backing Up Databases  8-1
  Syntax and Location  8-2
  Backup Strategy  8-2
    Using cnr_shadow_backup utility:  8-2
    Setting Automatic Backup Time  8-3
    Performing Manual Backups  8-3
    Using Third-Party Backup Programs with cnr_shadow_backup  8-3
  Database Recovery Strategy  8-4
  Backing Up CNRDB Data  8-5
  Backing Up all CNRDBs using tar or similar tools  8-5
  Recovering CNRDB Data from Damaged Databases  8-6
  Recovering CNRDB Data from Backups  8-7
  Recovering all CNRDBs using tar or Similar Tools  8-8
  Recovering single CNRDB from tar or similar tools  8-9
  Virus Scanning While Running Cisco Prime IP Express  8-9

Troubleshooting Databases  8-9
  Using the cnr_exim Data Import and Export Tool  8-10
  Using the cnrdb_recover Utility  8-11
  Using the cnrdb_verify Utility  8-12
  Using the cnrdb_checkpoint Utility  8-13
  Restoring DHCP Data from a Failover Server  8-13

Address Management

Managing Address Space  9-1
  Address Block Administrator Role  9-1
    Required Permissions  9-1
    Role Functions  9-2
  Viewing Address Space  9-2
  Pulling Replica Address Space from Local Clusters  9-2
  Address Blocks and Subnets  9-3
    Viewing Address Blocks, Subnets, and Address Types  9-4
    Knowing When to Add Address Blocks  9-5
    Adding Address Blocks  9-5
    Delegating Address Blocks  9-6
Contents

Chapter 10
Managing Hosts 10-1
Managing Hosts in Zones 10-1
Adding Additional RRs for the Host 10-2
Editing Hosts 10-2
Removing Hosts 10-3

Chapter 11
Managing Owners and Regions 11-1
Managing Owners 11-1
Managing Regions 11-2
Centrally Managing Owners and Regions 11-2
Pushing and Pulling Owners or Regions 11-3
Pushing Owners or Regions to Local Clusters 11-3
Pulling Owners and Regions from the Replica Database 11-3

Chapter 12
Managing Reports 12-1
ARIN Reports and Allocation Reports 12-1
Managing ARIN Reports 12-1
Managing Point of Contact and Organization Reports 12-2
Creating a Point of Contact Report 12-2
Registering a Point of Contact 12-3
Editing a Point of Contact Report 12-3
Creating an Organization Report 12-4
Registering an Organization 12-4
Editing an Organization Report 12-5
Managing IPv4 Address Space Utilization Reports 12-5
Managing Shared WHOIS Project Allocation and Assignment Reports 12-6
Managing BYOD Reports 12-6
Registered Devices 12-6
Scopes 12-6

Part 4
Domain and Zone Administration
CHAPTER 13

Introduction to the Domain Name System 13-1

How DNS Works 13-1

Domains 13-2
  Learning ExampleCo Address 13-3
  Establishing a Domain 13-3
  Difference Between Domains and Zones 13-3

Nameservers 13-5

Reverse Nameservers 13-6

Authoritative and Caching DNS servers 13-7

High-Availability DNS 13-7

DNS Database 13-7

DNS Views 13-7

About EDNS 13-8

CHAPTER 14

Managing Zones 14-1

Staged and Synchronous Modes 14-1

Creating and Applying Zone Templates 14-2

Managing Primary DNS Servers 14-4
  Configuring Primary Forward Zones 14-4
    Creating Primary Zones 14-5
    Editing Primary Zones 14-7
    Confirming Zone Nameservers 14-8
    Synchronizing Zones 14-8
  Zone Commands 14-8

Importing and Exporting Zone Data 14-9

Adding Primary Reverse Zones 14-11

Adding Reverse Zones as Zones 14-11

Adding Reverse Zones from Subnets 14-12

Getting Zone Counts on the Server 14-13

Managing Secondary Servers 14-13

Adding Secondary Forward Zones 14-13

Enabling Zone Transfers 14-14

Managing DNS ENUM Domain 14-15

Managing DNS ENUM Defaults 14-15

Adding DNS ENUM Domains 14-16

Adding DNS ENUM Numbers 14-17

Pulling and Pushing ENUM Domains 14-17

Pushing ENUM Domains to Local Clusters 14-17
Pulling ENUM Domains from the Replica Database 14-18
Pulling and Pushing ENUM Numbers 14-18
Pushing ENUM Numbers to Local Clusters 14-18
Pulling ENUM Numbers from the Replica Database 14-19

Adding Subzones 14-19
Choosing Subzone Names and Servers 14-20
Creating and Delegating Subzones 14-20
Undelegating Subzones 14-21
Editing Subzone Delegation 14-22

Enabling DNS Updates 14-22
Managing Zone Distributions 14-22
Preparing the Zone Distribution Map 14-23
Creating a Zone Distribution 14-24
Pulling Zone Distributions from Replica Data 14-26

CHAPTER 15
Configuring DNS Views 15-1
DNS Views Processing 15-1
Key Points to Remember When you Work on DNS Views 15-1
Managing DNS Views 15-2
Reorder DNS Views 15-2
Synchronizing DNS Views 15-3
Pushing and Pulling DNS Views 15-3
Pushing DNS Views to Local Clusters 15-3
Pulling DNS Views from Local Clusters 15-4

CHAPTER 16
Managing Resource Records 16-1
Managing Resource Records 16-1
Adding Resource Records 16-2
Protecting Resource Record Sets 16-3
Editing Resource Records 16-4
Removing Resource Records 16-4
Listing Records 16-5
Searching Server-Wide for Records and Addresses 16-5
Filtering Records 16-7
Using Service Location (SRV) Records 16-7
Using NAPTR Records 16-8
Managing Hosts in Zones 16-9
CHAPTER 17

Managing Authoritative DNS Server Properties 17-1

Managing DNS Authoritative Servers 17-1
  Running DNS Authoritative Server Commands 17-1
  Configuring DNS Server Network Interfaces 17-2

Setting DNS Server Properties 17-2
  Setting General DNS Server Properties 17-3
  Specifying Delegation-Only Zones 17-3
  Enabling Round-Robin 17-4
  Enabling Subnet Sorting 17-4
  Enabling Incremental Zone Transfers (IXFR) 17-5
  Restricting Zone Queries 17-5
  Enabling NOTIFY 17-5

Setting Advanced Authoritative DNS Server Properties 17-6
  Setting SOA Time to Live 17-6
  Setting Secondary Refresh Times 17-7
  Setting Secondary Retry Times 17-7
  Setting Secondary Expiration Times 17-8
  Setting Local and External Port Numbers 17-8
  Handling Malicious DNS Clients 17-8
  Tuning DNS Properties 17-9

Troubleshooting DNS Servers 17-9

CHAPTER 18

Managing Caching DNS Server Properties 18-1

Managing DNS Caching Servers 18-1
  Running DNS Caching Server Commands 18-1
  Configuring CDNS Server Network Interfaces 18-2

Defining Forwarders 18-2

Using Exceptions 18-3

Managing DNS64 18-4

Managing DNSSEC 18-5

Setting DNS Caching Server Properties 18-5
  Setting General CDNS Server Properties 18-6
  Specifying Log Settings 18-6
  Specifying Activity Summary Settings 18-7
  Specifying Caching Settings 18-7
  Setting Cache TTLs 18-7
  Defining Root Nameservers 18-8
  Dynamic Allocation of UDP Ports 18-8
### Setting Advanced Caching DNS Server Properties 18-9
- Setting Maximum Memory Cache Sizes 18-9
- Specifying Network Settings 18-10
- Flushing CDNS Cache 18-10
- Detecting and Preventing DNS Cache Poisoning 18-11
  - DNS Cache Poisoning Attacks 18-11
  - Handling DNS Cache Poisoning Attacks 18-11
- Handling Unresponsive Nameservers 18-12
- Caching DNS Domain Redirect 18-12

### Configuring High-Availability DNS Servers 19-1
- HA DNS Processing 19-1
- Configuring an HA DNS Server Pair from Main Server 19-3
- DNS Server Configuration for HA DNS 19-4
- Synchronizing HA DNS Zones 19-5
- HA DNS Configuration Synchronization 19-5
  - Initial Setup Considerations 19-5
  - Migration Procedure 19-6
    - Pre-install Cisco Prime IP Express on the HA DNS backup server 19-6
    - Pre-migration Steps for HA DNS Main Server 19-6
    - Restart Cisco Prime IP Express on the HA DNS Main Server 19-7
    - Copy Cisco Prime IP Express Database Files to HA DNS Backup Server 19-7
    - Reconfigure Cisco Prime IP Express on the HA DNS Backup Server 19-7
    - Configure Cisco Prime IP Express HA DNS on the HA DNS Main Server 19-8
    - Reload the DNS Servers 19-8
- HA DNS Statistics 19-8

### Dynamic Host Administration

### Introduction to Dynamic Host Configuration 20-1
- How DHCP Works 20-1
  - Sample DHCP User 20-2
  - Typical DHCP Administration 20-2
- Leases 20-3
  - Scopes and Policies 20-3
- Cisco Prime IP Express DHCP Implementations 20-4
  - DHCP and IPv6 20-4
  - Virtual Private Networks 20-5
Contents

Subnet Allocation and DHCP Address Blocks 20-5

DNS Update 20-6
  Effect on DNS of Obtaining Leases 20-7
  Effect on DNS of Releasing Leases 20-7
  Effect on DNS of Reacquiring Leases 20-8

DHCP Failover 20-8
  Allocating Addresses Through Failover 20-8

Client-Classes 20-9
  DHCP Processing Without Client-Classes 20-10
  DHCP Processing with Client-Classes 20-11
  Defining Scopes for Client-Classes 20-11
  Choosing Networks and Scopes 20-12

CHAPTER 21

Configuring Scopes and Networks 21-1

Configuring DHCP Servers 21-1
  General Configuration Guidelines 21-1
  Configuring DHCP Server Interfaces 21-2

Defining and Configuring Scopes 21-2
  Creating and Applying Scope Templates 21-3
    Using Expressions in Scope Templates 21-4
    Additional Scope Template Attributes 21-8
  Editing Scope Templates 21-8
  Applying Scope Templates to Scopes 21-8
  Additional Scope Template Attributes 21-8
  Editing Scope Templates 21-8
  Applying Scope Templates to Scopes 21-8
  Cloning a Scope Template 21-9

Creating Scopes 21-9
  Getting Scope Counts on the Server 21-10

Configuring Multiple Scopes 21-11
  Configuring Multiple Scopes for Round-Robin Address Allocation 21-11
  Configuring Multiple Scopes Using Allocation Priority 21-12

Editing Scopes 21-17
  Staged and Synchronous Mode 21-18

Configuring Embedded Policies for Scopes 21-18
Configuring Multiple Subnets on a Network 21-19
Enabling and Disabling BOOTP for Scopes 21-20
Disabling DHCP for Scopes 21-20
Deactivating Scopes 21-21
Setting Scopes to Renew-Only
Setting Free Address SNMP Traps on Scopes
Removing Scopes
  Removing Scopes if Not Reusing Addresses
  Removing Scopes if Reusing Addresses
Managing DHCP Networks
  Listing Networks
  Editing Networks

CHAPTER 22
Configuring Policies and Options 22-1
Configuring DHCP Policies 22-1
  Types of Policies 22-1
  Policy Hierarchy 22-3
  Creating and Applying DHCP Policies 22-3
Cloning a Policy 22-5
Setting DHCP Options and Attributes for Policies 22-5
  Adding Option Values 22-6
  Adding Complex Values for Suboptions 22-7
Creating and Editing Embedded Policies 22-7
Creating DHCP Option Definition Sets and Option Definitions 22-8
  Using Standard Option Definition Sets 22-8
  Creating Custom Option Definitions 22-10
  Creating Vendor-Specific Option Definitions 22-10
Option Definition Data Types and Repeat Counts 22-15
Adding Suboption Definitions 22-16
Importing and Exporting Option Definition Sets 22-17
Pushing Option Definition Sets to Local Clusters 22-17
Pulling Option Definition Sets from Replica Data 22-18
Setting Option Values for Policies 22-18

CHAPTER 23
Managing Leases 23-1
Configuring Leases in Scopes 23-1
  Viewing Leases 23-2
  Lease States 23-2
  Guidelines for Lease Times 23-3
  Importing and Exporting Lease Data 23-4
  Pinging Hosts Before Offering Addresses 23-6
  Deactivating Leases 23-7
  Excluding Leases from Ranges 23-7
Searching Server-Wide for Leases 23-8
Using Client Reservations 23-10
  Differences Between Client Reservations And Lease Reservations 23-13
Creating Lease Reservations 23-13
  DHCPv4 Reservations 23-13
Setting Advanced Lease and Reservation Properties 23-14
  Reserving Currently Leased Addresses 23-15
  Unreserving Leases 23-16
  Extending Reservations to Non-MAC Addresses 23-17
  Forcing Lease Availability 23-18
  Inhibiting Lease Renewals 23-19
  Handling Leases Marked as Unavailable 23-20
  Setting Timeouts for Unavailable Leases 23-20
Running Address and Lease Reports 23-20
  Running Address Usage Reports 23-21
Running IP Lease Histories 23-21
  Enabling Lease History Recording at the Local Cluster 23-22
  Querying IP Lease History 23-22
  Trimming Lease History Data 23-26
Receiving Lease Notification 23-27
  Running Lease Notification Automatically in Linux 23-27
  Running Lease Notification Automatically in Windows 23-28
  Specifying Configuration Files for Lease Notification 23-28
Querying Leases 23-28
  Leasequery Implementations 23-29
  Pre-RFC Leasequery for DHCPv4 23-29
  RFC 4388 Leasequery for DHCPv4 23-30
  Leasequery for DHCPv6 23-31
  Leasequery Statistics 23-32
  Leasequery Example 23-33
DHCP Listener Configuration 23-35
  Lease History Database Compression Utility 23-37
    General Comments on Running cnr_leasehist_compress 23-37
    Running Compression on Linux 23-38
    Running Compression on Windows 23-41
Moving Leases Between Servers 23-42

CHAPTER 24

Advanced DHCP Server Properties 24-1
  Configuring BOOTP 24-1
About BOOTP 24-2
Enabling BOOTP for Scopes 24-2
Moving or Decommissioning BOOTP Clients 24-3
Using Dynamic BOOTP 24-3
BOOTP Relay 24-3
Defining Advanced Server Attributes 24-4
Setting Advanced DHCP Server Attributes 24-4
Deferring Lease Extensions 24-8
Integrating Windows System Management Servers 24-9
Using Extensions to Affect DHCP Server Behavior 24-10
Writing Extensions 24-11
Preventing Chatty Clients by Using an Extension 24-13
Tuning the DHCP Server 24-15
Configuring Virtual Private Networks and Subnet Allocation 24-17
Configuring Virtual Private Networks Using DHCP 24-18
Typical Virtual Private Networks 24-18
Creating and Editing Virtual Private Networks 24-19
VPN Usage 24-21
Configuring DHCP Subnet Allocation 24-22
VPN and Subnet Allocation Tuning Parameters 24-23
Setting DHCP Forwarding 24-24

Chapter 25  Configuring Client-Classes and Clients 25-1
Configuring Client-Classes 25-1
Client-Class Process 25-2
Defining Client-Classes 25-2
Setting Selection Tags on Scopes and Prefixes 25-4
Defining Client-Class Hostname Properties 25-5
Editing Client-Classes and Their Embedded Policies 25-5
Processing Client Data Including External Sources 25-6
Processing Order to Determine Client-Classes 25-7
Processing Order to Determine Selection Tags 25-7
Troubleshooting Client-Classes 25-8
Configuring Clients 25-9
Editing Clients and Their Embedded Policies 25-10
Setting Windows Client Properties 25-11
Allocating Provisional Addresses 25-12
Skipping Client Entries for Client-Classing 25-13
Limiting Client Authentication 25-13
Setting Client Caching Parameters 25-13
Subscriber Limitation Using Option 82 25-14
  General Approach to Subscriber Limitation 25-15
  Typical Limitation Scenario 25-15
  Calculating Client-Classes and Creating Keys 25-15
  Client-Class Lookup Expression Processing 25-16
  Limitation Processing 25-16
  Expression Processing for Subscriber Limitation 25-16
  Configuring Option 82 Limitation 25-17
  Lease Renewal Processing for Option 82 Limitation 25-17
  Administering Option 82 Limitation 25-18
  Troubleshooting Option 82 Limitation 25-18
  Expression Examples 25-19
Configuring Cisco Prime IP Express to Use LDAP 25-19
  About LDAP Directory Servers 25-19
  Adding and Editing LDAP Remote Servers 25-19
  Configuring DHCP Client Queries in LDAP 25-20
  Configuring DHCP LDAP Update and Create Services 25-23
    Lease State Attributes 25-24
    Configuring DHCP to Write Lease States to LDAP 25-25
    Using LDAP Updates 25-26
    Configuring LDAP State Updates 25-26
    Configuring LDAP Entry Creation 25-28
  Troubleshooting LDAP 25-29
    LDAP Connection Optimization 25-29
    Recommended Values for LDAP 25-30

CHAPTER 26
Using Expressions 26-1
  Using Expressions 26-1
  Entering Expressions 26-2
  Creating Expressions 26-3
    Expression Syntax 26-4
    Expression Datatypes 26-4
    Literals in Expressions 26-5
    Expressions Return Typed Values 26-5
    Expressions Can Fail 26-6
    Expression Functions 26-6
    Datatype Conversions 26-21
    Expressions in the CLI 26-22
Expression Examples 26-22
  Limitation Example 1: DOCSIS Cable Modem 26-23
  Limitation Example 2: Extended DOCSIS Cable Modem 26-24
  Limitation Example 3: DSL over Asynchronous Transfer Mode 26-24
Debugging Expressions 26-26

CHAPTER 27
Managing DHCPv6 Addresses 27-1
DHCPv6 Concepts 27-2
  IPv6 Addressing 27-2
  Links and Prefixes 27-2
    Determining Links and Prefixes 27-3
    Generating Addresses 27-4
    Generating Delegated Prefixes 27-5
    Prefix Stability 27-5
    Prefix Allocation Groups 27-6
  DHCPv6 Clients and Leases 27-7
    DHCPv6 Bindings 27-8
    Lease Affinity 27-8
    IPv6 Lease States 27-8
    Lease Life Cycle 27-9
    DHCPv6 Lease Reservations 27-10
    DHCPv6 Client Reservations 27-12
    Searching for Leases 27-12
    Querying Leases for DHCPv6 27-12
  DHCPv6 Policy Hierarchy 27-12
  DHCPv6 Options 27-13

DHCPv6 Configuration 27-13
  Viewing IPv6 Address Space 27-14
  Configuring Links 27-14
    Creating and Editing Link Templates 27-14
    Creating and Editing Links 27-18
  Configuring Prefixes 27-20
    Creating and Editing Prefix Templates 27-20
    Creating and Editing Prefixes 27-25
    Viewing IPv6 Leases 27-29
    Viewing Address Utilization for Prefixes 27-30
  Viewing DHCPv6 Networks 27-32
  Editing DHCPv6 Server Attributes 27-32
  Configuring DHCPv6 Policies 27-32
CHAPTER 28

Managing DHCP Failover 28-1

How DHCP Failover Works 28-1
DHCP Simple Failover 28-2
DHCPv6 Failover 28-3
Setting Up Failover Server Pairs 28-3
  Adding Failover Pairs 28-4
  Synchronizing Failover Pairs 28-5
  Failover Checklist 28-8
Configuring Failover Parameters Based on Your Scenario 28-8
  Setting Backup Percentages 28-8
  Setting the Maximum Client Lead Time 28-10
  Using the Failover Safe Period to Move Servers into PARTNER-DOWN State 28-11
  Setting DHCP Request and Response Packet Buffers 28-13
  Setting Load Balancing 28-13
    Configuring Load Balancing 28-14
Recovering from a DHCP Failover 28-14
  Confirming Failover 28-14
  Monitoring DHCP Failover 28-15
  Failover States and Transitions 28-15
    State Transitions During Integration 28-17
Setting Advanced Failover Attributes 28-19
  Setting Backup Allocation Boundaries 28-20
  DHCPLEASEQUERY and Failover 28-20
Maintaining Failover Server Pair 28-20
  Changing Failover Pair Server Addresses 28-20
  Restarting the Failover Servers 28-21
Restoring a Standalone DHCP Failover Server - Tutorial 28-21
  Background 28-22
  Repair Procedure 28-22
  Reversing the Failover Role on Backup Server 28-23
  Starting with Server A Powered Off 28-23
  Starting with Server A Powered On and DHCP Server Stopped 28-24
Starting with Server A Replaced 28-25
Transferring Current Lease State to Server A 28-26
Repairing Partners to Their Original Roles 28-26
Changing Failover Server Roles 28-27
Establishing Failover Using Standalone Server as Main 28-27
Replacing Servers Having Defective Storage 28-28
Removing Backup Servers and Halting Failover Operation 28-29
Adding Main Servers to Existing Backup Servers 28-29
Configuring Failover on Multiple Interface Hosts 28-29
Troubleshooting Failover 28-29
Monitoring Failover Operations 28-30
Detecting and Handling Network Failures 28-30
Supporting BOOTP Clients in Failover 28-31
Static BOOTP 28-31
Dynamic BOOTP 28-31
Configuring BOOTP Relays 28-32
BOOTP Backup Percentage 28-32

CHAPTER 29
Configuring DNS Update 29-1
DNS Update Process 29-1
Special DNS Update Considerations 29-2
DNS Update for DHCPv6 29-2
DHCPv6 Upgrade Considerations 29-3
Generating Synthetic Names in DHCPv4 and DHCPv6 29-3
Determining Reverse Zones for DNS Updates 29-4
Using the Client FQDN 29-4
Creating DNS Update Configurations 29-5
Creating DNS Update Maps 29-7
Configuring Access Control Lists and Transaction Security 29-8
Access Control Lists 29-9
Configuring Zones for Access Control Lists 29-10
Transaction Security 29-10
Creating TSIG Keys 29-11
Generating Keys 29-11
Considerations for Managing Keys 29-12
Adding Supporting TSIG Attributes 29-12
GSS-TSIG 29-13
Creating GSS-TSIG Configuration 29-14
GSS-TSIG attributes 29-15
Configuring DNS Update Policies 29-15
  Compatibility with Cisco Prime IP Express Releases 29-15
  Creating and Editing Update Policies 29-16
  Defining and Applying Rules for Update Policies 29-16
    Defining Rules for Named Update Policies 29-16
    Applying Update Policies to Zones 29-18
  Confirming Dynamic Records 29-19
  Scavenging Dynamic Records 29-19
  Troubleshooting DNS Update 29-21
  Transitioning to DHCID RR for DHCPv4 29-21
Configuring DNS Update for Windows Clients 29-22
  Client DNS Updates 29-23
  Dual Zone Updates for Windows Clients 29-25
  DNS Update Settings in Windows Clients 29-25
  Windows Client Settings in DHCP Servers 29-25
  SRV Records and DNS Updates 29-26
  Issues Related to Windows Environments 29-28
  Frequently Asked Questions About Windows Integration 29-32
Configuring GSS-TSIG 29-34
  CPIPE DNS Configuration to integrate with AD 29-34
  Primary DNS Server on Linux Integrated to MIT-KDC 29-36
  Troubleshooting GSS-TSIG Configuration 29-37

CHAPTER 30

Using Extension Points 30-1
Using Extensions 30-1
  Creating, Editing, and Attaching Extensions 30-2
  Determining Tasks 30-3
  Deciding on Approaches 30-3
  Choosing Extension Languages 30-4
Language-Independent API 30-4
  Routine Signature 30-4
  Dictionaries 30-5
  Utility Methods in Dictionaries 30-5
  Configuration Errors 30-5
  Communicating with External Servers 30-6
  Recognizing Extensions 30-6
  Multiple Extension Considerations 30-6
Tcl Extensions 30-7
  Tcl Application Program Interface 30-7
Dealing with Tcl errors 30-8
Handling Boolean Variables in Tcl 30-8
Configuring Tcl Extensions 30-8
Init-Entry Extension Point in Tcl 30-9

C/C++ Extensions 30-9
C/C++ API 30-9
Using Types in C/C++ 30-10
Building C/C++ Extensions 30-10
Using Thread-Safe Extensions in C/C++ 30-10
Configuring C/C++ Extensions 30-11
Debugging C/C++ Extensions 30-11
Pointers into DHCP Server Memory in C/C++ 30-11
Init-Entry Entry Point in C/C++ 30-12

DHCP Request Processing Using Extensions 30-12
Enabling DHCPv6 Extensions 30-14
Receiving Packets 30-14
Decoding Packets 30-14
Determining Client-Classes 30-15
Modifying Client-Classes 30-15
Processing Client-Classes 30-15
Building Response Containers 30-16
Determining Networks and Links 30-16
Finding Leases 30-16
Serializing Lease Requests 30-17
Determining Lease Acceptability 30-17
DHCPv6 Leasing 30-19
  DHCPv6 Prefix Usability 30-19
  DHCPv6 Lease Usability 30-19
  DHCPv6 Lease Allocation 30-20
Gathering Response Packet Data 30-20
Encoding Response Packets 30-21
Updating Stable Storage 30-21
Sending Packets 30-21
Processing DNS Requests 30-21
Tracing Lease State Changes 30-22
Controlling Active Leasequery Notifications 30-22

Extension Dictionaries 30-23
Environment Dictionary 30-24
  General Environment Dictionary Data Items 30-25
Modifying Network Address Settings  32-2
Configuring Proxy Server  32-3
Accessing Cisco Prime IP Express Application  32-3
Configurations and Restrictions  32-3

PART 7
Appendices

APPENDIX A
Resource Records  A-1

APPENDIX B
DHCP Options  B-1

Option Descriptions  B-1
RFC 1497 Vendor Extensions  B-1
IP Layer Parameters Per Host  B-3
IP Layer Parameters Per Interface  B-4
Link Layer Parameters Per Interface  B-4
TCP Parameters  B-5
Application and Service Parameters  B-5
DHCPv4 Extension Options  B-8
Microsoft Client Options  B-10
DHCPv6 Options  B-11

Option Tables  B-15
Options by Number  B-15
Options by Cisco Prime IP Express Name  B-20
Option Validation Types  B-26

APPENDIX C
DHCP Extension Dictionary  C-1

Extension Dictionary Entries  C-1
Decoded DHCP Packet Data Items  C-1
Request Dictionary  C-10
Response Dictionary  C-16

Extension Dictionary API  C-26
Tcl Attribute Dictionary API  C-26
Tcl Request and Response Dictionary Methods  C-27
Tcl Environment Dictionary Methods  C-29
DEX Attribute Dictionary API  C-30
DEX Request and Response Dictionary Methods  C-31
DEX Environment Dictionary Methods  C-39
Handling Objects and Options  C-40
Preface

This guide describes configuring Cisco Prime IP Express by using the web-based user interface (web UI) and command line interface (CLI).

Who Should Read This Guide

This guide is designed for network managers who are responsible for maintaining the network Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Simple Network Management Protocol (SNMP) servers. The network manager should be familiar with the following topics:

- Basic concepts and terminology used in internetworking
- Network topology and protocols

How This Guide Is Organized

This guide describes how to become familiar with Cisco Prime IP Express features so that you can use them to administer network addresses. The parts of this guide are described in the following subsections.

Part 1—Getting Started

Part 1 introduces Cisco Prime IP Express, describes the management and protocol components, and describes the user interfaces. This part includes the following chapters:

<table>
<thead>
<tr>
<th>Chapter 1</th>
<th>Cisco Prime IP Express Components</th>
<th>Introduces Cisco Prime IP Express, its deployment scenarios, and some deployment guidelines.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 2</td>
<td>Cisco Prime IP Express User Interfaces</td>
<td>Describes the Cisco Prime IP Express management and protocol components.</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Server Status Dashboard</td>
<td>Describes the Cisco Prime IP Express server status dashboard features and functions.</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Deploying Cisco Prime IP Express</td>
<td>Describes the Cisco Prime IP Express local and regional web UIs and CLIs.</td>
</tr>
</tbody>
</table>
Part 2—Local and Regional Administration

Part 2 describes how to configure administrators, manage the central configuration, and maintain the servers and databases (including backup and recovery). This part includes the following chapters:

Chapter 5  Configuring Administrators  Describes how to configure the local and regional administrators, and provides administration tutorials.

Chapter 6  Managing the Central Configuration  Describes how to manage the central network configuration from the regional cluster.

Chapter 7  Maintaining Servers and Databases  Describes how to maintain the Cisco Prime IP Express servers.

Chapter 8  Backup and Recovery  Describes how to back up or recover the databases.

Part 3—Address Management

Part 3 describes how to manage the IP address space and its hierarchy, hosts, owners and regions, and reports. This part includes the following chapters:

Chapter 9  Managing Address Space  Describes how to manage address space elements known as address blocks and subnets.

Chapter 10  Managing Hosts  Describes how to manage network hosts.

Chapter 11  Managing Owners and Regions  Describes how to manage network owners and regions.

Chapter 12  Managing Reports  Describes how to manage American Registry of Internet Numbers (ARIN) and address allocation reports.
Part 4—Domain and Zone Administration

Part 4 describes how to configure DNS servers, zones, resource records, server attributes, and High Availability (HA) servers. This part includes the following chapters:

- **Chapter 13** Introduction to the Domain Name System
  - Introduces the Domain Name System (DNS) protocol and its Cisco Prime IP Express implementation.

- **Chapter 14** Managing Zones
  - Describes how to manage DNS zones.

- **Chapter 16** Managing Resource Records
  - Describes how to manage DNS resource records (RRs).

- **Chapter 17** Managing Authoritative DNS Server Properties
  - Describes how to set advanced Authoritative DNS server properties.

- **Chapter 18** Managing Caching DNS Server Properties
  - Describes how to set more advanced Caching DNS server properties.

- **Chapter 19** Configuring High-Availability DNS Servers
  - Describes how to configure a High Availability (HA) DNS server.

Part 5—Dynamic Host Administration

Part 5 describes DHCP and how to configure scopes and leases and their several deployments, IPv6 addresses, clients and client-classes, failover, DNS Update, and special processing using extensions. This part includes the following chapters:

- **Chapter 20** Introduction to Dynamic Host Configuration
  - Introduces DHCP and its Cisco Prime IP Express implementation.

- **Chapter 21** Configuring Scopes and Networks
  - Describes how to configure scopes and networks.

- **Chapter 22** Configuring Policies and Options
  - Describes how to configure policies and options.

- **Chapter 23** Managing Leases
  - Describes how to manage leases.

- **Chapter 25** Configuring Client-Classes and Clients
  - Describes how to configure DHCP clients and client-classes.

- **Chapter 26** Using Expressions
  - Describes how to use expressions for DHCP processing.

- **Chapter 27** Managing DHCPv6 Addresses
  - Describes how to manage the DHCPv6 address space.

- **Chapter 28** Managing DHCP Failover
  - Describes how to configure DHCP failover servers.
Part 6—Virtual Appliance

Part 6 describes virtual appliance and how to configure and manage Cisco Prime IP Express virtual appliance. This part includes the following chapters:

- Chapter 31 Introduction to Cisco Prime IP Express Virtual Appliance
- Chapter 32 Managing the Cisco Prime IP Express Virtual Appliance

Part 7—Appendixes, Glossary, and Index

Part 7 includes appendixes that describe DNS RRs, DHCP options, and the DHCP extension dictionary. This part also includes a glossary and an index.

- Appendix A Resource Records
- Appendix B DHCP Options
- Appendix C DHCP Extension Dictionary
- Glossary
- Index

Document Conventions

This guide uses the following documentation conventions.

Formatting

This guide uses the following formatting conventions:
- User input and controls are indicated in **bold**; for example, “enter **1234**” and “click **Modify Scope**.”
- Object attributes are indicated in *italics*; for example, “the *failover-safe-period* attribute.”
• Cross-references to chapters or sections of chapters are indicated in blue type; for example, “see the “Document Conventions” section on page xxxii.”

Navigation and Screens

This guide uses the following navigation and screen display conventions:

• Windows systems use a two-button mouse. To drag and drop an object, click and hold the left mouse button on the object, drag the object to the target location, then release the button.

• Screen displays can differ slightly from those included in this guide, depending on the system or browser you use.

• Web UI Navigation bar labels can have IPv4 and IPv6 variants depending on the administrator role privileges assigned. To simplify procedural instructions, this User Guide uses the most generic versions of the menu bar labels, unless there is a need to be more specific. For example, the Address Space menu label might be rendered as IP v4 and IP v6. The instructions will have the label simply as Address Space.

Callouts

Callouts in the text have the following meaning:

<table>
<thead>
<tr>
<th>Caution</th>
<th>Be careful. The description alerts you to potential data damage or loss.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note</td>
<td>Take note. The description is particularly noteworthy.</td>
</tr>
<tr>
<td>Timesaver</td>
<td>Save time. The description can present a timesaver.</td>
</tr>
<tr>
<td>Tip</td>
<td>Consider this helpful hint. The description can present an optimum action to take.</td>
</tr>
</tbody>
</table>

Product Documentation

We sometimes update the electronic documentation after original publication. Therefore, you should also review the documentation on Cisco.com for any updates.


Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see What’s New in Cisco Product Documentation at: http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html.

Subscribe to What’s New in Cisco Product Documentation, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.
P A R T  1

Getting Started
Cisco Prime IP Express Components

Cisco Prime IP Express provides the tools to configure and control the servers necessary to manage your IP address space. This chapter provides an overview of the management components and concentrates on the Simple Network Management Protocol (SNMP), which are not covered in subsequent parts of this User Guide.

Management Components

Cisco Prime IP Express contains two management components:

- Regional component, consisting of:
  - Web-based user interface (web UI)
  - Command line interface (CLI)
  - Central Configuration Management (CCM)
  - Bring your own device (BYOD)

- Local component, consisting of:
  - Web UI
  - CLI
  - CCM server
  - Authoritative Domain Name System (DNS) server
  - Caching / Recursive Domain Name System (CDNS) server
  - Dynamic Host Configuration Protocol (DHCP) server
  - Simple Network Management Protocol (SNMP) server
  - Management of local address space, zones, scopes, DHCPv6 prefixes and links, and users

Note

We do not recommend configuring both DNS and Caching DNS services in one server.

License management is done from the regional cluster when Cisco Prime IP Express is installed. You must install the regional server first and load all licenses in the regional server. When you install the local cluster, it registers with regional to obtain its license.
The regional CCM server provides central management of local clusters, with an aggregated view of DHCP address space and DNS zones. It provides management of the distributed address space, zones, scopes, DHCPv6 prefixes and links, and users.

The local CCM server provides management of the local address space, zones, scopes, DHCPv6 prefixes and links, and users.

The remainder of this chapter describes the SNMP protocol. The CCM server, web UIs, and CLI are described in Chapter 2, “Cisco Prime IP Express User Interfaces.” The DNS, CDNS and DHCP servers are described in their respective sections of this guide.

## Simple Network Management

The Cisco Prime IP Express Simple Network Management Protocol (SNMP) notification support allows you to query the DHCP and DNS counters, be warned of error conditions and possible problems with the DNS and DHCP servers, and monitor threshold conditions that can indicate failure or impending failure conditions.

Cisco Prime IP Express implements SNMP Trap Protocol Data Units (PDUs) according to the SNMPv2c standard. Each trap PDU contains:

- Generic-notification code, if enterprise-specific.
- A specific-notification field that contains a code indicating the event or threshold crossing that occurred.
- A variable-bindings field that contains additional information about certain events.

Refer to the Management Information Base (MIB) for the details. The SNMP server supports only reads of the MIB attributes. Writes to the attributes are not supported.

The following MIB files are required:

- **Traps**—CISCO-NETWORK-REGISTRAR-MIB.my
- **DNS server**—CISCO-DNS-SERVER-MIB.my

**Note**
The Caching DNS server requires only a subset of the DNS MIB when it is operating. Caching DNS server only supports the **server-start** and **server-stop** notification events.

- **DHCPv4 server**—CISCO-IETF-DHCP-SERVER-MIB.my
- **DHCPv4 server capability**—CISCO-IETF-DHCP-SERVER-CAPABILITY.my
- **DHCPv4 server extensions**—CISCO-IETF-DHCP-SERVER-EXT-MIB.my
- **DHCPv4 server extensions capability**—CISCO-IETF-DHCP-SERVER-EXT-CAPABILITY.my
- **DHCPv6 server**—CISCO-NETREG-DHCPV6-MIB.my (experimental)

**Note**
The MIB, CISCO-NETREG-DHCPV6-MIB is defined to support query of new DHCP v6 related statistics and new DHCP v6 traps.

These MIB files are available in the /misc directory of the Cisco Prime IP Express installation path. The following dependency files are also required:

- **Dependency for DHCPv4 and DHCPv6**—CISCO-SMI.my
• **Additional dependencies for DHCPv6**—INET-ADDRESS-MIB.my

These dependency files are available along with all the MIB files at the following URL:

To get the object identifiers (OIDs) for the MIB attributes, go to the equivalently named .oid file at:

### Related Topics

- Setting Up the SNMP Server, page 1-3
- How Notification Works, page 1-4
- Handling SNMP Notification Events, page 1-5
- Handling SNMP Queries, page 1-8

### Setting Up the SNMP Server

To perform queries to the SNMP server, you need to set up the server properties.

#### Local Basic or Advanced Web UI

<table>
<thead>
<tr>
<th>Step 1</th>
<th>From the <strong>Operate</strong> menu, choose <strong>Manage Servers</strong> under the <strong>Servers</strong> submenu to open the Manage Servers page (see the “Managing Servers” section on page 7-1).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Click the <strong>Local SNMP Server</strong> link to open the Edit Local SNMP Server page.</td>
</tr>
<tr>
<td>Step 3</td>
<td>The <strong>Community string</strong> attribute is the password to access the server. (The community string is a read community string only.) The preset value is <strong>public</strong>.</td>
</tr>
</tbody>
</table>
| Step 4 | You can specify the Log Settings, Miscellaneous Options and Settings, and Advanced Options and Settings:  
  • **trap-source-addr**—Optional sender address to use for outgoing traps.  
  • **server-active**—Determines whether the SNMP server is active for queries. The default value is true. If set to false, the server will run, but is not accessible for queries and does not send out traps.  
  • **cache-ttl**—Determines how long the SNMP caches responds to queries, default to 60 seconds. |
| Step 5 | To manage the SNMP server interfaces in the Advanced mode, click the **Network Interfaces** tab. You can view the default configured network interfaces, and create and edit additional ones. To create and edit them, you must be assigned the server-management subrole of the ccm-admin role. The interface properties are similar to those for the TFTP server. |
| Step 6 | To manage trap recipients for the server:  
  a. Click the **Trap Recipients** tab.  
  b. Enter the name and IP address of a trap recipient (both are required).  
  c. Click **Add Trap Recipient**.  
  d. Repeat for each additional trap recipient.  
  e. To set the port, community string, and agent address for a trap recipient, click its name on the Trap Recipients tab to open the Edit Trap Recipient page, then set the values. |
| Step 7 | Complete the SNMP server setup by clicking **Save**. |
CLI Commands

To set the community string in the CLI so that you can access the SNMP server, use `snmp set community=name`. Use `snmp set trap-source-addr` to set the trap source address. Use `snmp disable server-active` to deactivate the SNMP server and `snmp set cache-ttl=time` to set the cache time-to-live.

To set trap recipients, use `trap-recipient`, in the following syntax to include the IP address:

```
nrcmd> trap-recipient name create ip-addr=ip-addr
```

You can also add the `agent-address`, `community`, and `port-number` values for the trap recipient.

Other SNMP-related commands include `snmp disable server-active` to prevent the server from running when started and the `snmp-interface` commands to configure the interfaces. The `addr-trap` command is described in the “Handling SNMP Notification Events” section on page 1-5.

How Notification Works

Cisco Prime IP Express SNMP notification support allows a standard SNMP management station to receive notification messages from the DHCP and DNS servers. These messages contain the details of the event that triggered the SNMP trap.

Cisco Prime IP Express generates notifications in response to predetermined events that the application code detects and signals. Each event can also carry with it a particular set of parameters or current values. For example, the `free-address-low-threshold` event can occur in the scope with a value of 10% free. Other scopes and values are also possible for such an event, and each type of event can have different associated parameters.

Table 1-1 describes the events that can generate notifications.

<table>
<thead>
<tr>
<th>Event</th>
<th>Notification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address conflict with another DHCP server detected (address-conflict)</td>
<td>An address conflicts with another DHCP server.</td>
</tr>
<tr>
<td>DNS queue becomes full (dns-queue-size)</td>
<td>The DHCP server DNS queue fills and the DHCP server stops processing requests. (This is usually a rare internal condition.)</td>
</tr>
<tr>
<td>Duplicate IP address detected (duplicate-address and duplicate-address6)</td>
<td>A duplicate IPv4 or IPv6 address occurs.</td>
</tr>
<tr>
<td>Duplicate IPv6 prefix detected (duplicate-prefix6)</td>
<td>A duplicate IPv6 prefix occurs.</td>
</tr>
<tr>
<td>Failover configuration mismatch (failover-config-error)</td>
<td>A DHCP failover configuration does not match between partners.</td>
</tr>
<tr>
<td>Caching DNS forwarders not responding (forwards-not-responding)</td>
<td>Forwarding servers stop responding to the Caching DNS server.</td>
</tr>
<tr>
<td>DNS forwarders responding (forwards-responding)</td>
<td>Forwarding servers respond after having been unresponsive.</td>
</tr>
</tbody>
</table>
Simple Network Management

Handling SNMP Notification Events

When Cisco Prime IP Express generates a notification, it transmits a single copy of the notification as an SNMP Trap PDU to each recipient. All events (and scopes or prefixes) share the list of recipients and other notification configuration data, and the server reads them when you initialize the notification.

You can set SNMP attributes in three ways:

- For the DHCP server, which includes the traps to enable and the default free-address trap configuration if you are not specifically configuring traps for scopes or prefixes (or their templates).
- On the scope or prefix (or its template) level by setting the free-address-config attribute.
- For the DNS server, which includes a traps-enabled setting.

To use SNMP notifications, you must specify trap recipients that indicate where trap notifications should go. By default, all notifications are enabled, but you must explicitly define the recipients, otherwise no notifications can go out. The IP address you use is often localhost.

### Table 1-1  SNMP Notification Events (continued)

<table>
<thead>
<tr>
<th>Event</th>
<th>Notification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free-address thresholds (free-address-low and free-address-high; or free-address6-low and free-address6-high)</td>
<td>The high trap when the number of free IPv4 or IPv6 addresses exceeds the high threshold; or a low trap when the number of free addresses falls below the low threshold after previously triggering the high trap.</td>
</tr>
<tr>
<td>High-availability (HA) DNS configuration mismatch (ha-dns-config-error)</td>
<td>An HA DNS configuration does not match between partners.</td>
</tr>
<tr>
<td>HA DNS partner not responding (ha-dns-partner-down)</td>
<td>An HA DNS partner stops responding to the DNS server.</td>
</tr>
<tr>
<td>HA DNS partner responding (ha-dns-partner-up)</td>
<td>An HA DNS partner responds after having been unresponsive.</td>
</tr>
<tr>
<td>DNS masters not responding (masters-not-responding)</td>
<td>Master DNS servers stop responding to the DNS server.</td>
</tr>
<tr>
<td>DNS masters responding (masters-responding)</td>
<td>Master DNS servers respond after having been unresponsive.</td>
</tr>
<tr>
<td>Other server not responding (other-server-down)</td>
<td>A DHCP failover partner, or a DNS or LDAP server, stops responding to the DHCP server.</td>
</tr>
<tr>
<td>Other server responding (other-server-up)</td>
<td>DHCP failover partner, or a DNS or LDAP server, responds after having been unresponsive.</td>
</tr>
<tr>
<td>DNS secondary zones expire (secondary-zone-expired)</td>
<td>A DNS secondary server can no longer claim authority for zone data when responding to queries during a zone transfer.</td>
</tr>
<tr>
<td>Server start (server-start)</td>
<td>The DHCP or DNS server is started or reinitialized.</td>
</tr>
<tr>
<td>Server stop (server-stop)</td>
<td>The DHCP or DNS server is stopped.</td>
</tr>
</tbody>
</table>
The DHCP server provides special trap configurations so that it can send notifications, especially about free addresses for DHCPv4 and DHCPv6. You can set the trap configuration name, mode, and percentages for the low threshold and high threshold. The mode determines how scopes aggregate their free-address levels.

**DHCP v4 Notification**

The DHCP v4 modes and thresholds are (see also the “Handling Deactivated Scopes or Prefixes” section on page 1-6):

- **scope mode**—Causes each scope to track its own free-address level independently (the default).
- **network mode**—Causes all scopes set with this trap configuration (through the scope or scope template `free-address-config` attribute) to aggregate their free-address levels if the scopes share the same `primary-subnet`.
- **selection-tags mode**—Causes scopes to aggregate their free-address levels if they share a primary subnet and have a matching list of selection tag values.
- **low-threshold**—Free-address percentage at which the DHCP server generates a low-threshold trap and re-enables the high threshold. The free-address level for scopes is the following calculation:

  \[
  \frac{100 \times \text{available-nonreserved-leases}}{\text{total-configured-leases}}
  \]

- **high-threshold**—Free-address percentage at which the DHCP server generates a high-threshold trap and re-enables the low threshold.

**DHCP v6 Notification**

The DHCP v6 modes and thresholds are (see also the “Handling Deactivated Scopes or Prefixes” section on page 1-6):

- **prefix mode**—Causes each prefix to track its own free-address level independently.
- **link mode**—Causes all prefixes configured for the link to aggregate their own free-address levels if all prefixes share the same link.
- **v6-selection-tags mode**—Causes prefixes to aggregate their free-address levels if they share a link and have a matching list of selection tag values.
- **low-threshold**—Free-address percentage at which the DHCP server generates a low-threshold trap and re-enables the high threshold. The free-address level for prefixes is the following calculation:

  \[
  \frac{100 \times \text{max-leases} - \text{dynamic-leases}}{\text{max-leases}}
  \]

- **high-threshold**—Free-address percentage at which the DHCP server generates a high-threshold trap and re-enables the low threshold.

**Handling Deactivated Scopes or Prefixes**

A deactivated scope or prefix never aggregates its counters with other scopes or prefixes. For example, if you configure a prefix with `link` or `v6-selection-tags` trap mode, and then deactivate the prefix, its counters disappear from the total count on the aggregation. Any changes to the leases on the deactivated prefix do not apply to the aggregate totals.

Therefore, to detect clients for deactivated scopes or prefixes, you must set the event mode to `scope` or `prefix`, and not to any of the aggregate modes (`network`, `selection-tags`, `link`, or `v6-selection-tags`).
The use case for setting traps on deactivated prefixes, for example, is network renumbering. In this case, you might want to monitor both the new prefixes (as an aggregate, ensuring that you have enough space for all the clients) and old prefixes to ensure that their leases are freed up. You would probably also want to set the high threshold on an old prefix to 90% or 95%, so that you get a trap fired when most of its addresses are free.

Local Basic or Advanced Web UI

Access the SNMP attributes for the DHCP server by choosing Manage Servers from the Operate menu, then click Local DHCP Server in the left pane. You can view the SNMP attributes under SNMP (in Basic mode) or SNMP Settings (in Advanced mode) in the Edit DHCP Server page.

The four lease-enabled values (free-address6-low, free-address6-high, duplicate-address6, duplicate-prefix6) pertain to DHCPv6 only. Along with the traps to enable, you can specify the default free-address trap configuration by name, which affects all scopes and prefixes or links not explicitly configured.

To add a trap configuration, do the following:

---

**Step 1** In Advanced mode, from the Deploy menu choose Traps under the DHCP submenu to access the DHCP trap configurations. The List/Add Trap Configurations page appears.

**Step 2** Click the Add Traps icon in the left pane to open the Add AddrTrapConfig page.

**Step 3** Enter the name, mode, and threshold percentages, then click Add AddrTrapConfig.

---

To edit a trap configuration, do the following:

---

**Step 1** Click the desired trap name in the Traps pane to open the Edit Trap Configuration page.

**Step 2** Modify the name, mode, or threshold percentages.

**Step 3** Click the on option for the enabled attribute to enable the trap configuration.

**Step 4** Click Save for the changes to take effect.

---

To delete a trap configuration, select the trap in the Traps pane and click the Delete icon, then confirm or cancel the deletion.

Regional Basic or Advanced Web UI

In the regional web UI, you can add and edit trap configurations as in the local web UI. You can also pull replica trap configurations and push trap configurations to the local cluster on the List/Add Trap Configurations page.

Server Up/Down Traps

Every down trap must be followed by a corresponding up trap. However, this rule is not strictly applicable in the following scenarios:

1. If a failover partner or LDAP server or DNS server or HA DNS partner is down for a long time, down traps will be issued periodically. An up trap will be generated only when that server or partner returns to service.
Simple Network Management

Chapter 1  Cisco Prime IP Express Components

2. If the DHCP or DNS server is reloaded or restarted, the prior state of the partner or related servers is not retained and duplicate down or up traps can result.

**Note**
Other failover partner or LDAP server or DNS server or HA DNS partner up or down traps occur only to communicate with that partner or server, and therefore may not occur when the other partner or server goes down or returns to service.

**CLI Commands**

To set the trap values for the DHCP server at the local cluster, use `dhcp set traps-enabled=value`. You can also set the `default-free-address-config` attribute to the trap configuration. For example:

```
nrcmd> dhcp set traps-enabled=server-start,server-stop,free-address-low,free-address-high
nrcmd> dhcp set default-free-address-config=v4-trap-config
```

**Note**
If you do not define a `default-free-address-config` (or `v6-default-free-address-config` for IPv6), Cisco Prime IP Express creates an internal, unlisted trap configuration named `default-aggregation-addr-trap-config`. Because of this, avoid using that name for a trap configuration you create.

To define trap configurations for DHCPv4 and DHCPv6, use `addr-trap name create` followed by the `attribute=value` pairs for the settings. For example:

```
nrcmd> addr-trap v4-trap-conf create mode=scope low-threshold=25% high-threshold=30%
nrcmd> addr-trap v6-trap-conf create mode=prefix low-threshold=20% high-threshold=25%
```

**Handling SNMP Queries**

You can use SNMP client applications to query the following MIBs:

- CISCO-DNS-SERVER-MIB.my
- CISCO-IETF-DHCP-SERVER-MIB.my
- CISCO-IETF-DHCP-SERVER-EXT-MIB.my
- CISCO-NETREG-DHCPV6-MIB.my (experimental)

When the SNMP server receives a query for an attribute defined in one of these MIBs, it returns a response PDU containing that attribute value. For example, using the NET-SNMP client application (available over the Internet), you can use one of these commands to obtain a count of the DHCPDISCOVER packets for a certain address:

```
C:\net-snmp5.2.2\bin>snmpget -m ALL -v 2c -c public
192.168.241.39:4444.iso.org.dod.internet.private.enterprises.cisco.ciscoExperiment.ciscoIetfDhcpSrvMIB.ciscoIetfDhcpv4SrvMIBObjects.cDhcpv4Counters.cDhcpv4CountDiscovers
CISCO-IETF-DHCP-SERVER-MIB::cDhcpv4CountDiscovers.0 = Counter32: 0
```

```
C:\net-snmp5.2.2\bin>snmpget -m ALL -v 2c -c public
192.168.241.39:4444 1.3.6.1.4.1.9.10.102.1.3.1
CISCO-IETF-DHCP-SERVER-MIB::cDhcpv4CountDiscovers.0 = Counter32: 0
```
Both commands return the same results. The first one queries the full MIB attribute name, while the second one queries its OID equivalent (which can be less error prone). As previously described, the OID equivalents of the MIB attributes are located in the relevant files at the following URL:


For example, the CISCO-IETF-DHCP-SERVER-MIB.oid file includes the following OID definition that corresponds to the previous query example:

```
cDhcpv4CountDiscovers" "1.3.6.1.4.1.9.10.102.1.3.1"
```

Here are some possible SNMP query error conditions:

- The community string sent in the request PDU does not match what you configured.
- The version in the request PDU is not the same as the supported version (SNMPv2).
- If the object being queried does not have an instance in the server, the corresponding variable binding type field is set to SNMP_NOSUCHINSTANCE. With a GetNext, if there is no next attribute, the corresponding variable binding type field is set to SNMP_ENDOFMIBVIEW.
- If no match occurs for the OID, the corresponding variable binding type field is set to SNMP_NOSUCHOBJECT. With a GetNext, it is set to SNMP_ENDOFMIBVIEW.
- If there is a bad value returned by querying the attribute, the error status in the response PDU is set to SNMP_ERR_BAD_VALUE.

### Integrating Cisco Prime IP Express SNMP into System SNMP

You can integrate the Cisco Prime IP Express SNMP server into the SNMP server, for the system it runs on. The integration can be done in a way where the system will respond to queries for Cisco Prime IP Express MIB entries. On systems using NET-SNMP (and compatible servers) this is done by adding the following entries to the /etc/snmp/snmpd.conf configuration file:

```
view systemview included .1.3.6.1.4.1.9.9
view systemview included .1.3.6.1.4.1.9.10
proxy -v 2c -c public 127.0.0.1:4444 .1.3.6.1.4.1.9.9
proxy -v 2c -c public 127.0.0.1:4444 .1.3.6.1.4.1.9.10
```

The community string `public` and the port number `4444` may have to be replaced if the Cisco Prime IP Express SNMP server has been configured with different values for those settings.

NET-SNMP is commonly available on Linux and other Unix-like systems. On other systems, similar mechanisms may also be available.

### Default Ports for Cisco Prime IP Express Services

Table 1-2 lists the default ports used for the Cisco Prime IP Express services.

<table>
<thead>
<tr>
<th>Port Number</th>
<th>Protocol</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>TCP/UDP</td>
<td>DNS</td>
</tr>
<tr>
<td>53</td>
<td>TCP/UDP</td>
<td>Caching DNS</td>
</tr>
<tr>
<td>Port Number</td>
<td>Protocol</td>
<td>Service</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>67</td>
<td>UDP</td>
<td>DHCP client to server</td>
</tr>
<tr>
<td>67</td>
<td>TCP</td>
<td>Bulk or Active leasequery client to DHCP server</td>
</tr>
<tr>
<td>68</td>
<td>UDP</td>
<td>DHCP server to client</td>
</tr>
<tr>
<td>80</td>
<td>HTTP</td>
<td>BYOD web server client to server web UI</td>
</tr>
<tr>
<td>162</td>
<td>TCP</td>
<td>SNMP traps server to server</td>
</tr>
<tr>
<td>389</td>
<td>TCP</td>
<td>DHCP server to LDAP server</td>
</tr>
<tr>
<td>443</td>
<td>HTTPS</td>
<td>BYOD web server secure client to server web UI</td>
</tr>
<tr>
<td>546</td>
<td>UDP</td>
<td>DHCPv6 server to client</td>
</tr>
<tr>
<td>547</td>
<td>UDP</td>
<td>DHCPv6 client to server</td>
</tr>
<tr>
<td>647</td>
<td>TCP</td>
<td>DHCP failover server to server</td>
</tr>
<tr>
<td>653</td>
<td>TCP</td>
<td>High-Availability (HA) DNS server to server</td>
</tr>
<tr>
<td>1234</td>
<td>TCP</td>
<td>Local cluster CCM server to server</td>
</tr>
<tr>
<td>1244</td>
<td>TCP</td>
<td>Regional cluster CCM server to server</td>
</tr>
<tr>
<td>4444</td>
<td>TCP</td>
<td>SNMP client to server</td>
</tr>
<tr>
<td>5480</td>
<td>HTTPS</td>
<td>Virtual Appliance</td>
</tr>
<tr>
<td>8080</td>
<td>HTTP</td>
<td>Local cluster client to server web UI</td>
</tr>
<tr>
<td>8090</td>
<td>HTTP</td>
<td>Regional cluster client to server web UI</td>
</tr>
<tr>
<td>8443</td>
<td>HTTPS</td>
<td>Local cluster secure client to server web UI</td>
</tr>
<tr>
<td>8453</td>
<td>HTTPS</td>
<td>Regional cluster secure client to server web UI</td>
</tr>
</tbody>
</table>
Cisco Prime IP Express User Interfaces

Cisco Prime IP Express provides a regional and a local web-based user interface (web UI) and a regional and local command line interface (CLI) to manage the CDNS, DNS, DHCP, and Central Configuration Management (CCM) servers:

- Web UI for the regional cluster to access local cluster servers—See the “Regional Cluster Web UI” section on page 2-10.
- Web UI for the local cluster—See the “Local Cluster Web UI” section on page 2-7.
- CLI for the local clusters—Open the CLIContent.html file in the installation /docs directory (see the “Command Line Interface” section on page 2-10).
- CCM servers that provide the infrastructure to support these interfaces—See the “Central Configuration Management Server” section on page 2-11.
- BYOD web server for the regional cluster that provide the infrastructure to support BYOD—See the “Bring Your Own Device Web Server” section on page 2-12.

This chapter describes the Cisco Prime IP Express user interfaces and the services that the CCM servers provide. Read this chapter before starting to configure the Cisco Prime IP Express servers so that you become familiar with each user interface capability.

Introduction to the Web-Based User Interfaces

The web UI provides granular access to configuration data through user roles and constraints. The UI provides quick access to common functions. The web UI granularity is described in the following sections.

Related Topics

- Supported Web Browsers, page 2-2
- Access Security, page 2-2
- Logging In to the Web UIs, page 2-2
- Multiple Users, page 2-3
- Changing Passwords, page 2-4
- Navigating the Web UIs, page 2-4
- Waiting for Page Resolution Before Proceeding, page 2-4
- Committing Changes in the Web UIs, page 2-5
- Role and Attribute Visibility Settings, page 2-5
Supported Web Browsers

The web UI has been tested on Microsoft Internet Explorer 9 and Mozilla Firefox 24 and later. Internet Explorer 8 is not supported.

Access Security

At Cisco Prime IP Express installation, you can choose to configure HTTPS to support secure client access to the web UIs. You must specify the HTTPS port number and provide the keystore at that time. With HTTPS security in effect, the web UI Login page indicates that the “Page is SSL 1 Secure.”

Note
Do not use a dollar sign ($) symbol as part of a keystore password.

Logging In to the Web UIs

You can log into the Cisco Prime IP Express local or regional cluster web UIs either by HTTPS secure or HTTP nonsecure login. After installing Cisco Prime IP Express, open one of the supported web browsers and specify the login location URL in the browser address or netsite field. Login is convenient and provides some memory features to increase login speed.

You can log in using a nonsecure login in two ways:

- On Windows, from the Start menu, choose Start > All Programs > Cisco Prime IP Express 8.2 > Cisco Prime IP Express 8.2 [local | regional] Web UI. This opens the local or regional cluster web UI from your default web browser.

  Note
Open the regional Web UI first and add the licenses for the required services.

- Open the web browser and go to the web site. For example, if default ports were used during the installation, the URLs would be http://hostname:8080 for the local cluster web UI, and http://hostname:8090 for the regional cluster web UI.

This opens the New Product Installation page if no valid license is added at the time of installation. You have to browse and add the valid license. If the license key is acceptable, the Cisco Prime IP Express login page is displayed.

  Note
You can add the licenses only in the regional server. The local has to be registered to the regional at the time of installation to run the desired licensed services.

In the local server, confirm the regional server IP address and port number and also the services you want to run at the time of your first login. Click Register to confirm registration. If the regional server is configured with the required licenses, you will be displayed the login page.

1. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).
Enter the superuser username and password created at the time of installation to log into the Web UI. The password is case-sensitive (See the “Managing Passwords” section on page 5-6). If you already added the valid license and superuser and configured a password at the time of installation, then you can log into the web UI using that username and password.

**Note**
There is no default username or password for login.

**Note**
To prepare for an HTTPS-secured login, see the *Cisco Prime IP Express Installation Guide*.

Depending on how your browser is set up, you might be able to abbreviate the account name or choose it from a drop-down list while setting the username.

To log in, click **Login**.

### Adding License

Cisco will e-mail you one or more license files after you register the Cisco Prime IP Express Product Authorization Key (PAK) on the web according to the Software License Claim Certificate shipped with the product. Cisco administers licenses through a FLEXlm system. Once you have the file or files:

1. Locate the license file or files in a directory (or on the desktop) that is easy to find.
2. On the List/Add Product Licenses page, browse for each file by clicking the **Choose File** button.

**Note**
The List/Add Product Licenses option is only available at the Regional.

3. In the Choose file window, find the location of the initial license file, then click **Open**.
4. If the license key is acceptable, the Add Superuser Administrator page appears immediately.
5. To add further licenses, from **Administration** menu choose **Licenses** under the **User Access** submenu to open the List/Add Product Licenses page. Click **Browse** to open the Choose file window, locate the additional license file, then click **Open**. If the key in the file is acceptable, the key, type, count, and expiration date appear, along with whether it is an evaluation key. If the key is not acceptable, the page shows the license text along with an error message. For the list of license types, see the “Licensing” section on page 5-15.

Above the table of licenses is a License Utilization area that, when expanded, shows the license types along with the total nodes that you can use and those actually used.

If Cisco Prime IP Express is installed as a distributed system, the license management is done from the regional cluster. You will not have the option of adding licenses in local cluster.

### Multiple Users

The Cisco Prime IP Express user interfaces support multiple, concurrent users. If two users try to access the same object record or data, a Modified object error will occur for the second user. If you receive this error while editing user data, do the following:

- **In the web UI**—Cancel the edits and refresh the list. Changes made by the first user will be reflected in the list. Redo the edits, if necessary.
• **In the CLI**—Use the `session cache refresh` command to clear the current edits, before viewing the changes and making further edits. Make changes, if you feel that it is necessary even after the other user’s changes.

### Changing Passwords

Whenever you edit a password on a web UI page, it is displayed as a string of eight dots. The actual password value is never sent to the web browser. So, if you change the password, the field is automatically cleared. You must enter the new password value completely, exactly as you want it to be.

#### Note

The password should not be more than 255 characters long.

For details on changing administrator passwords at the local and regional cluster, see the “Managing Passwords” section on page 5-6.

### Navigating the Web UIs

The web UI provides a hierarchy of pages based on the functionality you desire and the thread you are following as part of your administration tasks. The page hierarchy prevents you from getting lost easily.

#### Caution

Do not use the Back button of the browser. Always use the navigation bar menu, or the **Cancel** button on the page to return to a previous page. Using the browser Back button can cause erratic failures.

A single sign-on feature is available to connect between the regional and local cluster web UIs. The regional cluster web UI pages include the Connect button ( ) in the List/Add Remote clusters page, which you can click to connect to the local cluster associated with the icon. If you have single sign-on privileges to the local cluster, the connection takes you to the related local server management page (or a related page for related server configurations). If you do not have these privileges, the connection takes you to the login page for the local cluster. To return to the regional cluster, local cluster pages have the Return button ( ) on the main toolbar.

#### Note

Navigation bar items can vary based on if you have the role privileges for IPv4 or IPv6. For example, the **Design** menu bar can be **DHCPv4** and **DHCPv6** if you have the ipv6-management subrole of the `addrblock-admin` role assigned.

### Waiting for Page Resolution Before Proceeding

Operations performed in the web UI, such as resynchronizing or replicating data from server clusters, are synchronous in that they do not return control to the browser until the operation is completed. These operations display confirmation messages in blue text. The IE browser displays a wait cursor while the operation is in progress.

#### Tip

Wait for each operation in the web UI to finish before you begin a new operation. If the browser becomes impaired, close the browser, reopen it, then log in again. Some operations like zone distributions can take significant amount of time, so you may have to wait till the operation completes.
Committing Changes in the Web UIs

You do not actually commit the page entries you make until you click **Save** on the page. You can delete items using the delete icon. To prevent unwanted deletions, a Confirm Delete dialog box appears in many cases so that you have a chance to confirm or cancel the deletion.

Role and Attribute Visibility Settings

Click the **username** drop-down list on the top of the main page to modify user preferences, session settings, user permissions, or debug settings.

- To view the user groups and roles for the administrator, select the **User Preferences** option. Superuser is a special kind of administrator. (For details how to set up these administrator roles, see the “Create the Administrators” section on page 5-26.)

- Select **Session Settings** to open the Session Settings dialog, select the mode from the **Session Web UI Mode** drop-down list, and click **Modify Session Settings**. You can also click the drop-down arrow of the Mode icon ( ) to view the list of modes. Select the required mode from the list:
  - **Basic**—Basic user mode (the preset choice).
  - **Advanced**—Advanced user mode that exposes the normal attributes.
  - **Expert**—Expert user mode that exposes a set of attributes that are relevant for fine-tuning or troubleshooting the configuration. In most cases, you would accept the default values for these expert attributes and not change them without guidance from the Cisco Technical Assistance Center (TAC). Each Expert mode attribute is marked with a Warning icon on the configuration pages. Each page is clearly marked as being in Expert mode.

Displaying and Modifying Attributes

Many of the web UI pages, such as those for servers, zones, and scopes, include attribute settings that correspond to those you can set using the CLI. (The CLI name equivalents appear under the attribute name.) The attributes are categorized into groups by their function, with the more prominent attributes listed first and the ones less often configured nearer the bottom of the page.

Grouping and Sorting Attributes

On many Advanced mode web UI pages, you can toggle between showing attributes in groups and in alphabetical order. These pages generally open by default in group view so that you can see the attributes in their respective categories. However, in the case of large numbers of attributes, you might want to see the attributes alphabetized. Click **Show A-Z View** to change the page to show the attributes alphabetically. Click **Show Group View** to change the page to show the attributes in groups. You can also expand or collapse the attribute groups in group view by clicking **Expand All** or **Collapse All**. In Expert mode, the Expert mode attributes are alphabetized separately further down the page under the Visibility=3 heading and are all marked with the Warning icon.
Modifying Attributes

You can modify attribute values and unset those for optional attributes. In many cases, these attributes have preset values, which are listed under the Default column on the page. The explicit value overrides the default one, but the default one is always the fallback. If there is no default value, unsetting the explicit value removes all values for that attribute.

Displaying Attribute Help

For contextual help for an attribute, click the name of the attribute to open a separate popup window.

Left Navigation Pane and Quick View Icon

The Web UI also provides a navigation pane on the left of the main pages. This navigation pane provides access to objects that are added as part of the various categories. You can click the object to edit its properties in the main page.

Each object displayed under a category in the pane also has a Quick View icon associated with it, which is activated when you move the mouse pointer over the object. The Quick View icon expands to open a dialog box that displays the main details about the object, and provides links (if any) to perform the main actions associated with the object.

Help Pages

The web UI provides a separate window that displays help text for each page. The Help pages provide:

- A context-sensitive help topic depending on which application page you have open.
- A clickable and hierarchical Contents and Index, and a Favorites setting, as tabs on a left-hand pane that you can show or hide.
- A Search facility that returns a list of topics containing the search string, ordered by frequency of appearance of the search string.
- Forward and backward navigation through the history of Help pages opened.
- A Print function.
- A Glossary.

Logging Out

Log out of the web UI by clicking the Log Out link. You can find the Log Out link between the username drop-down list and the About link at the top right corner of the application page above the tool bar.
Local Cluster Web UI

The local cluster web UI provides concurrent access to Cisco Prime IP Express user and protocol server administration and configuration. It provides granular administration across servers with permissions you can set on a per element or feature basis. The local cluster web UI is available in three user modes:

- **Basic mode**—Provides a more simplified configuration for the more frequently configured objects, such as DHCP scopes and DNS zones (see the “Local Basic Main Menu Page” section on page 2-7).

- **Advanced mode**—Provides the more advanced configuration method familiar to past users of the Cisco Prime IP Express web UI, with some enhancements (see the “Local Advanced Main Menu Page” section on page 2-8).

- **Expert mode** (marked with the icon)—For details on Expert mode, see the “Role and Attribute Visibility Settings” section on page 2-5.

Change to Basic, Advanced, or Expert mode by clicking the drop-down arrow of the Mode icon ( ) on the toolbar at the top right of the page (see the “Setting Local User Preferences” section on page 2-9).

**Note**

If you change the IP address of your local cluster machine, see the Note in the “Configuring Clusters in the Local Web UI” section on page 2-9.

### Related Topics

- Introduction to the Web-Based User Interfaces, page 2-1
- Regional Cluster Web UI, page 2-10

### Local Basic Main Menu Page

The Basic tab activated on the toolbar at the top right corner of the page implies that you are in Basic user mode. Otherwise, click the drop-down arrow of the Mode icon ( ) to view the list of modes and select Basic.

You can see the submenu items under the navigation bar item by placing the cursor on the main menu. To choose a submenu under a navigation bar item, place the cursor over the navigation bar item. For example, place the cursor on **Operate** to choose the **Manage Servers**.

Also, you can select any submenu under the required navigation bar and then navigate to the required submenu page from the left pane. For example, place the cursor on **Operate**, choose **Schedule Tasks**. You can see List/Add Scheduled Tasks page along with a left pane that has links to Manage Servers, Manage Clusters, Schedule Tasks, and View Change Log. Click the **Manage Servers** link to view the Manage Servers page.

The Local Basic main menu page provides functions with which you can:

- **Open the dashboard to monitor system health**—Open **Operate** menu and click **Dashboard** ( ). See Chapter 3, “Server Status Dashboard.”

- **Set up a basic configuration by using the Setup interview pages**—Click the **Setup** icon at the top, and select the different tabs in the Setup page. See *Cisco Prime IP Express Quick Start Guide* for more details.

- **Administer users, encryption keys**—Place the cursor on **Administration** menu (for user access options) or **Design** menu (for Security > Keys option). See Chapter 5, “Configuring Administrators.”
- **Manage the Cisco Prime IP Express protocol servers**—Place the cursor on **Operate** menu and select **Manage Servers** or **Schedule Tasks** option. See Chapter 7, “Maintaining Servers and Databases.”

- **Manage clusters**—Place the cursor on **Operate** menu and choose **Manage Clusters** option. See “Configuring Server Clusters” section on page 6-2.

- **Configure DHCP**—Place the cursor on **Design** menu and select the options under **DHCP Settings**, **DHCPv4** or **DHCPv6**. See Chapter 21, “Configuring Scopes and Networks.”

- **Configure DNS**—Place the cursor on **Design** menu and select the options under **Cache DNS** and **Auth DNS**. Place the cursor on **Deploy** menu and select the options under **DNS** and **DNS Updates**. See Chapter 14, “Managing Zones.”

- **Manage hosts in zones**—From the **Design** menu, choose **Hosts** under the **Auth DNS** submenu. See Chapter 10, “Managing Hosts.”

- **Go to Advanced mode**—Click **Advanced** in the top right corner of the page. See the “Local Advanced Main Menu Page” section on page 2-8.

### Local Advanced Main Menu Page

To switch to Advanced user mode from the Basic user Main Menu page, click the drop-down arrow of the Mode icon ( ) at the top right of the window to view the list of modes and select **Advanced**. Doing so opens another Main Menu page, except that it shows the Advanced user mode functions. To switch back to Basic mode at any time, click next to the Mode icon at the top right of the window and select **Basic**.

The local Advanced mode Main Menu page includes advanced Cisco Prime IP Express functions that are in addition to the ones in Basic mode:

- **Open the dashboard to monitor system health**—Open **Operate** menu and click **Dashboard** ( ). See Chapter 3, “Server Status Dashboard.”

- **Administer users, groups, roles, regions, access control lists (ACLs), and view change logs**—Place the cursor on **Administration** menu (for user access options), **Design** menu (for ACLs) or **Operate** menu (for change logs). See Chapter 5, “Configuring Administrators.”

- **Manage the Cisco Prime IP Express protocol servers**—Place the cursor on **Operate** menu and select **Manage Servers** or **Schedule Tasks** option. See Chapter 7, “Maintaining Servers and Databases.”

- **Manage clusters**—Place the cursor on **Operate** menu and choose **Manage Clusters** option. See the “Configuring Server Clusters” section on page 6-2.

- **Configure DHCPv4**—Place the cursor on **Design** and select any option under **DHCPv4**. See Chapter 21, “Configuring Scopes and Networks.”

- **Configure DHCPv6**—Place the cursor on **Design** and select any option under **DHCPv6**. See Chapter 27, “Managing DHCPv6 Addresses.”

- **Configure DNS**—Place the cursor on **Design** menu and select the options under **Cache DNS** and **Auth DNS**. Place the cursor on **Deploy** menu and select the options under **DNS** and **DNS Updates**. See Chapter 14, “Managing Zones.”

- **Manage hosts in zones**—From the **Design** menu, choose **Hosts** under the **Auth DNS** submenu. See Chapter 10, “Managing Hosts.”

- **Manage IPv4 address space**—Place the cursor on **Design** and select any option under **DHCPv4**. See Chapter 9, “Managing Address Space.”
• **Configure IPv6 address space**—Place the cursor on **Design** and select any option under **DHCPv6**. See Chapter 27, “Managing DHCPv6 Addresses.”

• **Go to Basic mode**—Click the drop-down arrow of the mode icon ( ) at the top right corner of the page and choose **Basic**. See the “Local Basic Main Menu Page” section on page 2-7.

The Advanced user mode page provides additional functions:

• **View the user role and group data for the logged-in user**—See the “Role and Attribute Visibility Settings” section on page 2-5.

• **Set your preferred session settings**—See the “Role and Attribute Visibility Settings” section on page 2-5.

• **Set server debugging**—You can set debug flags for the protocol servers. Set these values only under diagnostic conditions when communicating with the Cisco Technical Assistance Center (TAC).

• **Change your login administrator password**—See the “Changing Passwords” section on page 2-4.

### Setting Local User Preferences

You can maintain a short list of web UI settings through subsequent user sessions. The only difference between the Basic and Advanced or Expert mode user preference pages is that Advanced and Expert modes have additional columns listing the data types and defaults.

You can edit the user preferences by going to **User Preferences** under the **admin** menu (at the top of the main page). The user preference attributes to set are:

• **Username**—Username string, with a preset value of admin. You cannot modify this field.

• **Web UI list page size**—Adjust the page size by the number of displayed lines in a list; the preset value is 10 lines.

• **Web UI mode**—User mode at startup: Basic, Advanced, or Expert (see the “Role and Attribute Visibility Settings” section on page 2-5). If unset, the mode defaults to the one set in the CCM server configuration (see the “Managing Servers” section on page 7-1).

You can unset the page size and web UI mode values by checking the check box in the **Unset?** column, next to the attribute. After making the user preference settings, click **Modify User Preferences**.

### Configuring Clusters in the Local Web UI

You can define other local Cisco Prime IP Express clusters in the local web UI. The local cluster on the current machine is called the **localhost** cluster. To set up other clusters, choose **Manage Clusters** from **Operate** menu to open the List/Add Clusters page. Note that the **localhost** cluster has the IP address and SCP port of the local machine.

Click the **Add Cluster** icon in the left pane to open the Add Cluster page. At a minimum, you must enter the name and ipaddr of the remote local cluster. You should also enter the admin name and password, along with possibly the SCP port (if not 1234), of the remote cluster. Click **Add Cluster**. To edit a cluster, click the cluster name in the Clusters pane on the left to open the Edit Cluster page. If you want to use secure access mode, select use-ssl as disabled, optional, or required (optional is the preset value; you need the security library installed if you choose required). Make the changes, then click **Save**.
Regional Cluster Web UI

The regional cluster web UI provides concurrent access to regional and central administration tasks. It provides granular administration across servers with permissions you can set on a per element or feature basis. After you log into the application, the Home page appears. Regional cluster administration is described in Chapter 6, “Managing the Central Configuration.”

Related Topics

Introduction to the Web-Based User Interfaces, page 2-1
Local Cluster Web UI, page 2-7

Command Line Interface

Using the Cisco Prime IP Express CLI (the `nrcmd` program), you can control your local cluster server operations. You can set all configurable options, as well as start and stop the servers.

Note

The CLI provides concurrent access, by at most 14 simultaneous users and processes per cluster.

Tip

See the `CLIContents.html` file in the `/docs` subdirectory of your installation directory for details.

The `nrcmd` program for the CLI is located on:

- **Windows**—In the `{install-path}/bin` directory.
- **Linux**—In the `{install-path}/usr/bin` directory.

On a local cluster, once you are in the appropriate directory, use the following command at the prompt:

```
nrcmd -C clustername:port -H username -P password [-L | -R]
```

- `-C`—Cluster name, preset value `localhost`. Specify the port number with the cluster name while invoking `nrcmd` to connect to another cluster. See the preceding example. The port number is optional if the cluster uses the default SCP port—1234 for local and 1244 for regional. Ensure that you include the port number if the port used is not the default one.
- `-N`—Username. You have to enter the username that you created when first logged into the Web UI.
- `-P`—User password. You have to enter the password that you created for the username.
- The local cluster (`-L`) is implied; use `-R` to open the regional cluster CLI.

Tip

For additional command options, see the `CLIGuide.html` file in `/docs`. 
If you change the IP address of your local cluster machine, you must modify the `localhost` cluster to change the address in the `ipaddress` attribute. Do not set the value to 127.0.0.1.

You can also send the output to a file using:

```bash
nrcmd> session log filename
```

For example:

To send the leases on the DHCP server to a file (leases.txt), use the following commands:

```bash
nrcmd> session log leases.txt
nrcmd> lease list
```

To close a previously opened file, use `session log` (no filename). This stops writing the output to any file.

To disconnect from the cluster, use `exit`:

```bash
nrcmd> exit
```

The CLI operates on a coordinated basis with multiple user logins. If you receive a cluster lock message, determine who has the lock and discuss the issue with that person. (See the “Multiple Users” section on page 2-3.)

## Central Configuration Management Server

The CCM servers at the local and regional clusters provide the infrastructure for Cisco Prime IP Express operation and user interfaces. The CCM Server reads, writes, and modifies the Cisco Prime IP Express database (CCM DB). The main purpose of the CCM Server is to store and propagate data from the user to the protocol servers, and from the servers back to the user.

The change set is the fundamental unit of change to a data store. It sends incremental changes to a replicating server and provides an audit log for changes to the data store. Change sets consist of lists of change entries that are groups of one or more changes to a single network object. The web UI provides a view of the change sets for each data store.

## Managing CCM Server

You can view logs and startup logs; edit the server attributes.

To view logs and startup logs, in the local cluster web UI, from the `Operate` menu, choose `Manage Servers` to open the Manage Servers page.

## Editing CCM Server Properties

You can edit the CCM server properties using the Edit CCM Server page.
Local Basic or Advanced Web UI

Step 1 To access the CCM server properties, choose Manage Servers under Operate menu to open the Manage Servers page.
Step 2 Click Local CCM Server in the Manage Servers pane on the left. The Edit Local CCM Server page appears. This page displays all the CCM server attributes.
Step 3 Modify the settings as per your requirement.
Step 4 Click Save to save the CCM server attribute modifications.

Bring Your Own Device Web Server

The BYOD web server at the regional cluster provide the infrastructure for Cisco Prime IP Express BYOD operation. The main purpose of the BYOD Web Server is to authenticate the user against AD and collect the device metadata by registering the device in Cisco Prime IP Express.

Managing BYOD Web Server

You can view logs and startup logs; edit the server attributes.
To view logs and startup logs, in the regional cluster web UI, from the Operate menu, choose Manage Servers under the Server submenu to open the Manage Servers page.

Editing BYOD Web Server Properties

You can edit the BYOD web server properties using the Edit Local BYOD Web Server page.

Regional Basic or Advanced Web UI

Step 1 To access the BYOD web server properties, choose Manage Servers under Operate menu to open the Manage Servers page.
Step 2 Click Local BYOD Web Server in the Manage Servers pane on the left. The Edit Local BYOD Web Server page appears. This page displays all the BYOD web server attributes.
Step 3 Modify the settings as per your requirement.
Step 4 Click Save to save the BYOD web server attribute modifications.

Setting Up BYOD Theme and Content

You can create the content and multiple BYOD themes at the regional cluster which can be applied in BYOD web server interface.
Adding BYOD Themes

Create the BYOD web server themes on the regional cluster using the BYOD Theme page. Apply the created web server theme so that the logo, background, font, and other properties of the BYOD interface are displayed as per the selection.

To add a theme:

Regional Advanced or Expert Web UI

Step 1  From the Deploy menu, choose Theme under the BYOD submenu to open the List/Add Custom Theme page.

Step 2  Click the Add Theme icon in the Theme pane.

Step 3  In the Add Custom Theme window, enter the Theme Name and click Add Custom Theme.

Step 4  In the Edit Custom Theme page, make the necessary changes and click Save.

Adding BYOD Content

Create the BYOD web server contents like login page message, about, terms of services, contact details, and help message on the BYOD Content page of the regional cluster. These content will be used in the BYOD web server interface for the device registration and login pages.

Global Search in Prime IP Express

The Local and Regional Web UI in Prime IP Express also provides a global search functionality for the IP addresses or DNS names available in the local clusters. The search interface element is available at the top right corner of the main page.

Note
To view the search interface element and run the search for IP addresses and DNS names, Cisco Prime IP Express must be licensed with DHCP or DNS, and the DHCP or DNS services must be enabled for the local cluster (in the List/Add Remote Clusters page in Regional Web UI).

Table 2-1 shows the typical search results under different scenarios.

<table>
<thead>
<tr>
<th>You search for...</th>
<th>With active licenses and services for...</th>
<th>Search Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>An IPv4 address</td>
<td>Only DHCP</td>
<td>The closest matching scope, scope lease or scope reservation</td>
</tr>
<tr>
<td>An IPv4 address or a DNS FQDN</td>
<td>Only DNS</td>
<td>The related Zone or Resource Record</td>
</tr>
<tr>
<td>An IPv6 address</td>
<td>Only DHCP</td>
<td>The closest matching prefix, prefix lease or prefix reservation</td>
</tr>
</tbody>
</table>
### Table 2-1 Typical Search Results (continued)

<table>
<thead>
<tr>
<th>You search for...</th>
<th>With active licenses and services for...</th>
<th>Search Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>An IPv6 address or a DNS FQDN</td>
<td>Only DNS</td>
<td>The related Zone or Resource Record</td>
</tr>
<tr>
<td>An IPv4 address, an IPv6 address or a DNS FQDN</td>
<td>Both DHCP and DNS</td>
<td>All of the above, based on the type of address</td>
</tr>
</tbody>
</table>
Server Status Dashboard

The Cisco Prime IP Express server status dashboard in the web user interface (web UI) presents a graphical view of the system status, using graphs, charts, and tables, to help in tracking and diagnosis. These dashboard elements are designed to convey system information in an organized and consolidated way, and include:

- Significant protocol server and other metrics
- Alarms and alerts
- Database inventories
- Server health trends

The dashboard is best used in a troubleshooting desk context, where the system displaying the dashboard is dedicated for that purpose and might be distinct from the systems running the protocol servers. The dashboard system should point its browser to the system running the protocol servers.

You should interpret dashboard indicators in terms of deviations from your expected normal usage pattern. If you notice unusual spikes or drops in activity, there could be communication failures or power outages on the network that you need to investigate.

Related Topics

Opening the Dashboard, page 3-1
Display Types, page 3-2
Customizing the Display, page 3-6
Selecting Dashboard Elements to Include, page 3-7
Host Metrics, page 3-9
DHCP Metrics, page 3-10
Authoritative DNS Metrics, page 3-16
Caching DNS Metrics, page 3-20

Opening the Dashboard

To open the dashboard in local web UI, from the Operate menu, choose Dashboard.
Display Types

Provided you have DHCP and DNS privileges through administrator roles assigned to you, the preset display of the dashboard consists of the following tables (See Figure 3-1 on page 3-2 for an example):

- **System Metrics**—See the “System Metrics” section on page 3-9.
- **DHCP General Indicators**—See the “DHCP General Indicators” section on page 3-15.
- **DNS General Indicators**—See the “DNS General Indicators” section on page 3-19

These are just the preset selections. See the “Selecting Dashboard Elements to Include” section on page 3-7 for other dashboard elements you can select. The dashboard retains your selections from session to session.

### Figure 3-1 Preset Dashboard Elements

Each dashboard element initially appears as a table or a specific chart type, depending on the element:

- **Table**—See the “Tables” section on page 3-3.
- **Line chart**—See the “Line Charts” section on page 3-3.
- **Stacked area chart**—See the “Stacked Area Charts” section on page 3-5.

### General Status Indicators

Note the green box next to each dashboard element name in Figure 3-1. This box indicates that the server sourcing the information is functioning normally. A yellow box indicates that server operation is less than optimum. A red box indicates that the server is down. These indicators are the same as for the server health on the Manage Servers page in the regular web UI.

### Graphic Indicators for Levels of Alert

Graphed lines and stacked areas in the charts follow a standard color and visual coding so that you can immediately determine key diagnostic indicators at a glance. The charts use the following color and textural indicators:

- **High alerts or warnings**—Lines or areas in red, with a hatched texture.
- **All other indicators**—Lines or areas in various other colors distinguish the data elements. The charts do not use green or yellow.
Magnifying and Converting Charts

If Magnified Chart is the selected Chart Link (see Figure 3-5 on page 3-7), you can magnify a chart in a separate window by clicking the chart. In magnified chart view, you can choose an alternative chart type from the one that comes up initially (see the “Other Chart Types” section on page 3-5).

Note

Automatic refresh is turned off for magnified charts (see the “Setting the Polling Interval” section on page 3-6). To get the most recent data, click the Refresh icon next to the word Dashboard at the top left of the page.

To convert a chart to a table, see the “Displaying Charts as Tables” section on page 3-7. You cannot convert tables to a graphic chart format.

Legends

Each chart initially includes a color-coded legend. To turn off the legend display on the main dashboard page, see the “Displaying or Hiding Chart Legends” section on page 3-7. Removing the legend renders the graphic chart size relatively larger, which can be helpful if you have many charts displayed. You cannot remove legends in magnified views.

Tables

Dashboard elements rendered as tables have data displayed in rows and columns. The following dashboard elements are preset to consist of (or include) tables:

- System Metrics
- DHCP DNS Updates
- DHCP Address Current Utilization
- DHCP General Indicators
- DNS General Indicators
- Caching DNS General Indicators

Note

(See Figure 3-1 on page 3-2 for examples.) If you view a table in Expert mode, additional data might appear.

Line Charts

Dashboard elements rendered as line charts can include one or more lines plotted against the x and y axes. The three types of line charts are described in Table 3-1.
### Table 3-1  Line Chart Types

<table>
<thead>
<tr>
<th>Type of Line Chart</th>
<th>Description</th>
<th>Dashboard Elements Rendered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw data line chart</td>
<td>Lines plotted against raw data.</td>
<td>• Java Virtual Machine (JVM) Memory Utilization (Expert mode only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DHCP Buffer Capacity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DHCP Failover Status (two charts)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DNS Network Errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DNS Related Servers Errors</td>
</tr>
<tr>
<td>Delta line chart</td>
<td>Lines plotted against the difference between two sequential raw data.</td>
<td>• DNS Inbound Zone Transfers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DNS Outbound Zone Transfers</td>
</tr>
<tr>
<td>Rate line chart</td>
<td>Lines plotted against the difference between two sequential raw data divided by the sample time between them.</td>
<td>• DHCP Server Request Activity (see Figure 3-2 on page 3-4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DHCP Server Response Activity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DHCP Response Latency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DNS Query Responses</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DNS Forwarding Errors</td>
</tr>
</tbody>
</table>

**Tip**

To get the raw data for a chart that shows delta or rate data, enter Expert mode, set the Chart Link to Data Table (see the “Displaying Charts as Tables” section on page 3-7), then click the chart. The Raw Data table is below the Chart Data table.

### Figure 3-2  Line Chart Example

Detailed View: DHCP Server Request Activity

![Line Chart Example](image-url)
Stacked Area Charts

Dashboard elements rendered as stacked area charts have multiple related metrics plotted as trend charts, but stacked one on top of the other, so that the highest point represents a cumulative value. The values are independently shaded in contrasting colors. (See Figure 3-3 on page 3-5 for an example of the DHCP Server Request Activity chart shown in Figure 3-2 on page 3-4 rendered as a stacked area chart.)

**Figure 3-3 Stacked Area Chart Example**

They are stacked in the order listed in the legend, the left-most legend item at the bottom of the stack and the right-most legend item at the top of the stack.

The dashboard elements that are pre-set to stacked area charts are:

- DHCP Server Request Activity
- DHCP Server Response Activity
- DHCP Response Latency
- DNS Outbound Zone Transfers
- DNS Inbound Zone Transfers

Other Chart Types

The other chart types available for you to choose are:

- **Line**—One of the line charts described in Table 3-1 on page 3-4.
- **Stacked Area**—Charts described in the “Stacked Area Charts” section on page 3-5.
- **Pie**—Shows a single percentage pie chart of the data averaged over the time sampled.
- **Bar**—Multiple related current value metrics plotted side by side as groups of bars that show the actual data sampled.
- **Stacked Bar**—Addition total of the actual samples. This chart shows more distinct data points than the stacked area chart.

**Tip**

Each chart type shows the data in distinct ways and in different interpretations. You can decide which type best suits your needs.
Getting Help for the Dashboard Elements

You can open a help window for each dashboard element by clicking the title of the element.

Customizing the Display

To customize the dashboard display, you can:

- Refresh the data and set an automatic refresh interval.
- Expand a chart and render it in a different format.
- Convert a graphic chart to a table.
- Download data to comma-separated value (CSV) output.
- Display or hide chart legends.
- Configure server chart types.
- Reset to default display

Each chart supports:

- Resizing
- Drag and drop to new cell position
- Minimizing
- Closing

Each chart has a help icon with a description of the chart and a detailed help if you click the chart title.

**Note**
The changes made to the dashboard/chart will persist only if you click **Save** in the Dashboard window.

Refreshing Displays

Refresh each display so that it picks up the most recent polling by clicking the **Refresh** icon.

Setting the Polling Interval

You can set how often to poll for data. Click the **Dashboard Settings** icon in the upper-right corner of the dashboard display. There are four options to set the polling interval of the cached data, which polls the protocol servers for updates. (See **Figure 3-4** on page 3-6.)

**Figure 3-4  Setting the Chart Polling Interval**

You can set the cached data polling (hence, automatic refresh) interval to:

- **Disabled**—Does not poll, therefore does not automatically refresh the data.
- **Slow**—Refreshes the data every 30 seconds.
• **Medium**—Refreshes the data every 20 seconds.
• **Fast** (the preset value)—Refreshes the data every 10 seconds.

### Displaying Charts as Tables

You can choose to display a graphic chart as a table when you magnify the chart by clicking it (see the “Magnifying and Converting Charts” section on page 3-3). At the middle of the top of the dashboard display are the controls for the chart links (see Figure 3-5 on page 3-7).

#### Figure 3-5  Specifying Chart Conversion to Table Format

Click the **Data Table** radio button. When you click the chart itself, it opens as a table. The preset display format is Magnified Chart.

### Exporting to CSV Format

You can dump the chart data to a comma-separated value (CSV) file (such as a spreadsheet) when you magnify the chart by clicking it. In the Chart Link controls at the top of the page (see Figure 3-5 on page 3-7), click the **CSV Export** radio button, then click the chart. A Save As window appears, where you can specify the name and location of the CSV file.

### Displaying or Hiding Chart Legends

You can include or exclude the color-coded legends for charts on the main dashboard page. You might want to remove the legends as you become more familiar with the data and track it on a slightly larger chart display. In the upper-right of the dashboard display are the controls for the legend display (see Figure 3-6 on page 3-7). The preset value is Visible.

#### Figure 3-6  Displaying or Hiding Chart Legends and Selecting Chart

### Selecting Dashboard Elements to Include

You can decide how many dashboard elements you want to display on the page. At times, you might want to focus on one server activity only, such as for the DHCP server, and exclude all other metrics for the other servers. In this way, the dashboard becomes less crowded, the elements are larger and more readable. At other times, you might want an overview of all server activities, with a resulting smaller element display.

You can select the dashboard elements to display from the main Dashboard page by clicking **Chart Selections** in the Dashboard Settings dialog (see Figure 3-6 on page 3-7). Clicking the link opens the Chart Selection page (see Figure 3-7 on page 3-8).
Configuring Server Chart Types

You can set the default chart types on the main dashboard view. You can customize the server charts in the dashboard to display only the specific chart types as default.

To set up default chart type, check the check box corresponding to the **Metrics** chart that you want to display and choose a chart type from the **Type** drop-down list. The default chart types are consistent and shared across different user sessions (see Figure 3-7 on page 3-8).

You can see either the CDNS or DNS Metrics in the Dashboard Settings > Chart Selection page based on the service configured on the server.

The order in which the dashboard elements appear in the Chart Selection list does not necessarily determine the order in which the elements will appear on the page. An algorithm that considers the available space determines the order and size in a grid layout. The layout might be different each time you submit the dashboard element selections.

To change selections, check the check box next to the dashboard element that you want to display. Specific group controls are available in the drop-down list, **Chart Selection**, at the top of the page. To:

- Uncheck all check boxes, choose **None**.
- Revert to the preset selections, choose **Default**. The preset dashboard elements for administrator roles supporting DHCP and DNS are:
  - Host Metrics: System Metrics (see the “Host Metrics” section on page 3-9)
  - DHCP Metrics: General Indicators
  - DNS Metrics: General Indicators
- Select the DHCP metrics only, choose **DHCP** (see the “DHCP Metrics” section on page 3-10).
Select the DNS metrics only, choose **DNS** (see the “Authoritative DNS Metrics” section on page 3-16).

Select the DNS metrics only, choose **CDNS** (see the “Caching DNS Metrics” section on page 3-20).

Select all the dashboard elements, choose **All**.

Click **Save** at the bottom of the page to save your choices, or **Cancel** to cancel the changes.

---

**Host Metrics**

Host metrics comprise two charts:

- **System Metrics**—See the “System Metrics” section on page 3-9.
- **JVM Memory Utilization** (available in Expert mode only)—See the “JVM Memory Utilization” section on page 3-10.

---

**System Metrics**

The System Metrics dashboard element shows the free space on the disk volumes where the Cisco Prime IP Express logs and database directories are located, the date and time of the last server backup, and CPU and memory usage for the various servers. System metrics are available if you choose **Host Metrics: System Metrics** in the Chart Selection list.

The resulting table shows:

- **Logs Volume**—Current free space out of the total space on the disk drive where the logs directory is located, with the equivalent percentage of free space.
- **Database Volume**—Current free space out of the total space on the disk drive where the data directory is located, with the equivalent percentage of free space.
- **Last Good Backup**—Date and time when the last successful shadow database backup occurred (or Not Done if it did not yet occur) since the server agent was last started.
- **CPU Utilization** (in seconds), **Memory Utilization** (in kilobytes), and (in Expert mode only) the **VM Utilization** (in kilobytes) and Process ID (PID) for the:
  - Cisco Prime IP Express server agent
  - CCM server
  - DNS server
  - DHCP server
  - Web server
  - SNMP server
  - DNS caching server
How to Interpret the Data

The System Metrics data shows how full your disk volumes are getting based on the available free space for the Cisco Prime IP Express logs and data volumes. It also shows if you had a last successful backup of the data files and when that occurred. Finally, it shows how much of the available CPU and memory the Cisco Prime IP Express servers are using. The difference in the memory and VM utilization values is:

- **Memory Utilization**—Physical memory that a process uses, or roughly equivalent to the Resident Set Size (RSS) value in UNIX `ps` command output, or to the Task Manager Mem Usage value in Windows: the number of pages the process has in real memory minus administrative usage. This value includes only the pages that count toward text, data, or stack space, but not those demand-loaded in or swapped out.

- **VM Utilization**—Virtual memory that a process uses, or roughly equivalent to the SZ value in UNIX `ps` command output, or to the Task Manager VM Size value in Windows: the in-memory pages plus the page files and demand-zero pages, but not usually the memory-mapped files. This value is useful in diagnosing how large a process is and if it continues to grow.

Troubleshooting Based on the Results

If you notice the free disk space decreasing for the logs or data directory, you might want to consider increasing the disk capacity or look at the programs you are running concurrently with Cisco Prime IP Express.

JVM Memory Utilization

The Java Virtual Machine (JVM) Memory Utilization dashboard element is available only when you are in Expert mode. It is rendered as a line trend chart that traces the Unused Maximum, Free, and Used bytes of JVM memory. The chart is available if you choose **Host Metrics: JVM Memory Utilization** in the Chart Selection list when you are in Expert mode.

How to Interpret the Data

The JVM Memory Utilization data shows how much memory applies to running the dashboard in your browser. If you see the Used byte data spiking, dashboard elements might be using too much memory.

Troubleshooting Based on the Results

If you see spikes in Used memory data, check your browser settings or adjust the polling interval to poll for data less frequently.

DHCP Metrics

These DHCP metric elements are available in the dashboard:

- **DHCP Server Request Activity**—See the “DHCP Server Request Activity” section on page 3-11.
- **DHCP Server Response Activity**—See the “DHCP Server Response Activity” section on page 3-11.
- **DHCP Buffer Capacity**—See the “DHCP Buffer Capacity” section on page 3-12.
- **DHCP Response Latency**—See the “DHCP Response Latency” section on page 3-13.
• DHCP DNS Updates—See the “DHCP DNS Updates” section on page 3-13.
• DHCP Address Current Utilization—See the “DHCP Address Current Utilization” section on page 3-14.
• DHCP Failover Status—See the “DHCP Failover Status” section on page 3-15.
• DHCP General Indicators—See the “DHCP General Indicators” section on page 3-15.
• DHCP Server Lease Data—See the “DHCP Server Lease Data” section on page 3-16

**DHCP Server Request Activity**

The DHCP Server Request Activity dashboard element rendered as a stacked area chart traces the totals in the change rate of incoming DHCP packet activity. The chart is available if you choose **DHCP Metrics: DHCP Server Request Activity** in the Chart Selection list.

The resulting stacked area chart plots the following trends:

- **V4 Discovers**—Number of DHCPv4 discover packets.
- **V4 Requests**—Number of DHCPv4 request packets.
- **V4 Other**—Number of DHCPv4 release, decline, or info-request packets.
- **V4 Lease Queries**—Number of DHCPv4 lease query packets.
- **V6 Solicits**—Number of DHCPv6 solicit packets.
- **V6 Requests/Renews/Rebinds**—Number of DHCPv6 request, renew, and rebind packets.
- **V6 Other**—Number of DHCPv6 release, decline, or information-request packets.
- **V6 Lease Queries**—Number of DHCPv6 lease query packets.
- **Invalid Packets**—Combined number of invalid DHCPv4 and DHCPv6 packets.

**How to Interpret the Data**

The DHCP Server Request Activity data shows the pattern of server traffic based on incoming DHCP requests. The trend should be fairly consistent, with spikes in the number of Invalid packets being a sign that there is some misconfigured data on the network. Note that DHCPv4 and DHCPv6 invalid packet activity is grouped together.

**Troubleshooting Based on the Results**

Check your DHCP server configurations if there is a sudden spike in activity, especially in the number of invalid request packets. Set your server logging to report where the activity is occurring. Spikes or drops in activity can indicate network or power outages that are worth investigating. Spikes in activity can also indicate a faulty client, malicious client activity, or a recovery after a power failure or outage that results in pent-up requests.

**DHCP Server Response Activity**

The DHCP Server Response Activity dashboard element rendered as a stacked area chart traces the totals in the change rate of outgoing DHCP packet activity. The chart is available if you choose **DHCP Metrics: DHCP Server Response Activity** in the Chart Selection list.
The resulting stacked area chart plots the following trends:

- **V4 Offers**—Number of DHCPv4 offer packets.
- **V4 Acks**—Number of DHCPv4 acknowledgment packets.
- **V4 Other Client**—Number of other outgoing DHCPv4 client packets.
- **V4 Lease Queries**—Number of outgoing DHCPv4 lease query packets.
- **V6 Advertisements**—Number of DHCPv6 advertise packets.
- **V6 Replies**—Number of DHCPv6 reply packets.
- **V6 Reconfigures**—Number of DHCPv6 reconfigure packets.
- **V6 Lease Query Replies**—Number of DHCPv6 lease query reply packets.
- **Total Dropped**—Combined number of dropped DHCPv4 and DHCPv6 packets.

**How to Interpret the Data**

The DHCP Server Response Activity data shows the pattern of server traffic to answer DHCP requests. The trend should be fairly consistent, with spikes in the number of Total Dropped packets being a sign that there is some misconfigured data on the network. Note that DHCPv4 and DHCPv6 dropped packet activity is grouped together.

**Troubleshooting Based on the Results**

Check your DHCP server configurations if there is a sudden spike in activity, especially in the number of total dropped response packets. The response activity should match the request activity, except for the normal time shift, and the same diagnostics apply.

**DHCP Buffer Capacity**

The DHCP Buffer Capacity dashboard element rendered as a table shows the number of allocated requests and responses, and a line chart that plots the number of requests and responses in use. The element is available if you choose **DHCP Metrics: DHCP Buffer Capacity** in the Chart Selection list.

The resulting table and line chart plots:

- **Requests in Use**—Trend in the number of in-use request buffers.
- **Responses in Use**—Trend in the number of in-use response buffers.

**How to Interpret the Data**

The DHCP Buffer Capacity data shows the pattern in the use of DHCP request and response buffers. If the buffers begin to increase in an abnormal pattern, there are measures you can take without trying to compensate by increasing the number of allocated buffers.

**Troubleshooting Based on the Results**

If you see increasing and consistent exceeding of the buffer threshold, find the reason why the server is running slowly. Possible reasons include high degrees of logging, slow DHCP extensions or LDAP servers, or overload, such as with chatty clients or frequent rebooting of cable modem termination systems (CMTSs). You might need to increase the buffer sizes.
DHCP Response Latency

The DHCP Response Latency dashboard element rendered as a stacked area chart shows the trend in the response packet latency (the time interval between the request packet and its ensuing response). The chart is available if you choose DHCP Metrics: DHCP Response Latency in the Chart Selection list.

Tip

You must also set the collect-sample-counters DHCP server attribute for this data to display, with the enhanced-sample-counters attribute also set for further granularity. These attribute values are preset. If you are concerned about achieving maximum performance, unset these attributes. (See the “Displaying Statistics” section on page 7-11.)

The resulting stacked area chart plots response latencies at the intervals:

- Less than 50 milliseconds
- 50 to 200 milliseconds
- 200 to 500 milliseconds
- 500 to 1000 milliseconds (note that if the enhanced-sample-counters attribute is not set, all values below 1 second appear in this grouping)
- 1 to 2 seconds
- 2 to 3 seconds
- 3 to 4 seconds
- More than 4 seconds

How to Interpret the Data

The chart shows the trend in response packet latency as an indicator of how long it takes to respond to incoming packets. The gradations in the latency periods are stacked.

Troubleshooting Based on the Results

High response packet latency is similar to high buffer usage for troubleshooting purposes. Look for slow LDAP servers or DHCP extensions, high levels of logging, or disk I/O bottlenecks.

DHCP DNS Updates

The DHCP DNS Updates dashboard element rendered as a table shows the related DNS server and its current state, and how many pending DNS updates are occurring between it and the DHCP server. The table is available if you choose DHCP Metrics: DHCP DNS Updates in the Chart Selection list.

The resulting table shows:

- **Server**—Related DNS server IP address
- **State**—Related DNS server state
- **Pending Updates**—Total number of pending updates
How to Interpret the Data

A high level of pending updates to a specific DNS server indicates that the server is unreachable or unavailable, or its address is wrong.

Troubleshooting Based on the Results

Check into the reachability of the associated DNS servers if the pending update rate spikes, or ensure that the address of the associated server is correct.

DHCP Address Current Utilization

The DHCP Address Current Utilization dashboard element rendered as a table shows the DHCPv4 address utilization (how many assigned addresses exist) for a particular address aggregation, which can be a scope, network, or network plus selection tag. The table is available if you choose DHCP Metrics: DHCP Address Current Utilization in the Chart Selection list.

The resulting table shows:

- **Name**—Aggregation name (or address).
- **In Use**—Number of in-use addresses.
- **Total**—Total number of addresses.
- **Utilization**—Percentage of utilized addresses.
- **Mode** (appears in Expert mode only)—Aggregation mode (scope, network, or selection-tags).

How to Interpret the Data

The chart shows a table with four columns: the scope name, its in-use and total addresses, and the percentage of address utilization based on the previous two columns. The chart is available only if the DHCP server enhanced-sample-counters attribute is enabled.

- If an SNMP trap configuration in scope mode applies, the Name column displays the scope name. Otherwise, it shows the network IP address.

- If traps are not enabled (or if the DHCP server default-free-address-config or v6-default-free-address-config attribute is not set), the network address is appended with an asterisk (*).

- If a selection tag applies, its name is also appended. See the “Handling SNMP Notification Events” section on page 1-5 for details on SNMP traps.

- If you do not define a default-free-address-config (or v6-default-free-address-config) attribute, Cisco Prime IP Express creates an internal, unlisted trap configuration named default-aggregation-addr-trap-config.

  Because of this, do not use the name default-aggregation-addr-trap-config for a trap configuration you create.

Troubleshooting Based on the Results

If the percentage of utilized addresses is high, the addresses reached a saturation point. It might be necessary to reassign addresses from a different scope.
DHCP Failover Status

The DHCP Failover Status dashboard element rendered as two parallel trend charts that show the current and partner server state and the binding updates and acknowledgments sent and received between the two failover partners. The charts are available if you choose **DHCP Metrics: DHCP Failover Status** in the Chart Selection list.

**Note**
The failover status is only for the first failover pair in the related servers list.

The display is a table along with two rate line trend charts that shows the failover status for the first failover pair for the related servers:

- **Local State**—Local DHCP server failover state along with when it occurred.
- **Partner State**—Partner server failover state along with when it occurred.
- **DHCP Failover Status Updates Received**—The first trend chart shows a comparison of the number of binding updates received and binding acknowledgments sent.
- **DHCP Failover Status Updates Sent**—The second trend chart shows a comparison of the number of binding updates sent and binding acknowledgments received.

**How to Interpret the Data**

Along with some state data, the display is split into two line trend charts that are inverses of each other. Each chart compares the binding updates with the acknowledgments. The top chart pairs the binding updates received with the acknowledgments sent; the bottom chart pairs the binding updates sent with the acknowledgments received.

**Troubleshooting Based on the Results**

If the Partner State value is other than 10, check the configuration of the partner server. The updates sent and received data should also be fairly level.

DHCP General Indicators

The DHCP General Indicators dashboard element rendered as a table shows the server state, reload data, and lease counts. The table is available if you choose **DHCP Metrics: DHCP General Indicators** in the Chart Selection list.

The resulting table shows:

- **Server State**—Up or Down (based on whether statistics are available) and its duration.
- **Last Reload**—Date and time of the last server reload.
- **Start Time**—Date and time of the last server process (Cisco Prime IP Express server agent) startup.
- **Total Scopes**—Total number of configured DHCPv4 scopes.
- **V4 Leased Leases**—Number of active DHCPv4 leases, including reservations.
- **V4 Configured Leases**—Number of configured DHCPv4 leases, including reservations and ranges.
- **Total Prefixes**—Number of configured DHCPv6 prefixes.
- **V6 Leased Leases**—Number of active DHCPv6 leases, including reservations and delegated prefixes (which each count as one lease).
• **V6 Allocated Leases**—Number of allocated DHCPv6 leases, including reservations and delegated prefixes (which each count as one lease).

**How to Interpret the Data**

The table indicates the server state, process start time (via the Cisco Prime IP Express server agent), and reload data, and also provides lease statistics. The top set of data compares the DHCPv4 leases actually in effect with those configured; the bottom set of data does the same for DHCPv6 leases.

Time of last reload is important for determining if recent changes to the server configuration occurred from a reload operation. It can also help pinpoint when server changes were last applied, if other indicators show a marked, unexpected behavioral change. Be sure to preserve log files since the last reload.

**Troubleshooting Based on the Results**

A drop or increase in leases might indicate a power or network outage, but it can also indicate a normal variation depending on lease times and usage patterns. The number of scopes or prefixes indicated might also require some evaluation and possible reconfiguration. If the server state is Down, all the DHCP chart indicators show a red status box, so no data will be available. In the case of a server that is down, restart the server.

**DHCP Server Lease Data**

The DHCP Server Lease Data dashboard element, rendered as chart, shows the number of leases per second for the DHCP server. This chart is available if you choose **DHCP Metrics: DHCP Server Lease Data** in the Chart Selection page.

The chart displays:

• **V4 Leases**—Number of IPv4 leases per second.
• **V6 Leases**—Number of IPv6 leases per second.

**Authoritative DNS Metrics**

These authoritative DNS metric elements are available in the dashboard:

• **DNS Outbound Zone Transfers**—See the **“DNS Outbound Zone Transfers”** section on page 3-16.
• **DNS Inbound Zone Transfers**—See the **“DNS Inbound Zone Transfers”** section on page 3-17.
• **DNS Network Errors**—See the **“DNS Network Errors”** section on page 3-18.
• **DNS Related Servers Errors**—See the **“DNS Related Servers Errors”** section on page 3-18.
• **DNS General Indicators**—See the **“DNS General Indicators”** section on page 3-19.
• **DNS Queries Per Second**—See the **“DNS Queries Per Second”** section on page 3-19

**DNS Outbound Zone Transfers**

The DNS Outbound Zone Transfers dashboard element rendered as a stacked area chart tracks the rate of change in full and incremental outbound zone transfer responses, and any associated errors. The chart is available if you choose **DNS Metrics: DNS Outbound Zone Transfers** in the Chart Selection list.
The resulting stacked area chart plots the following trends:

- **Full Responses**—Number of full outbound zone transfers (AXFRs out).
- **Incremental Responses**—Number of incremental outbound zone transfers (IXFRs out).
- **Authorization Errors**—Number of unauthorized (refused) zone transfer requests.
- **Exceed Max Transfers Out**—Number of failed outbound transfers that exceed the maximum limit.
- **Other Errors**—Number of other outbound transfer errors that are not authorization errors.

**How to Interpret the Data**

This chart is useful in gauging if outbound zone transfers to a secondary DNS server are occurring as predicted and if there are any authorizations or failed transfer attempts in the process. The most significant indicator is the trend in the number of outbound zone transfers denied for lack of permission or for not being authorized for the zone.

**Troubleshooting Based on the Results**

Check the primary and secondary server configurations if there are errors or exceeded limits in the outbound zone transfers.

### DNS Inbound Zone Transfers

The DNS Inbound Zone Transfers dashboard element rendered as a stacked area chart tracks the rate of change in full and incremental inbound zone transfer responses, and any associated errors. The chart is available if you choose **DNS Metrics: DNS Inbound Zone Transfers** in the Chart Selection list.

The resulting stacked area chart plots the following trends:

- **Full Responses**—Number of full inbound zone transfers (AXFRs in).
- **Incremental Responses**—Number of incremental inbound zone transfers (IXFRs in).
- **Authorization Errors**—Number of refused responses (xfer-in-auth-errors).
- **Failed Attempts**—Number of failures other than refusals (xfer-failed-attempts).
- **Exceed Max Transfers In**—Number of times that the concurrent inbound transfers reach the maximum limit.

**How to Interpret the Data**

This chart is useful in gauging if inbound zone transfers to a secondary DNS server are occurring as predicted and if there are any authentication or failed transfer attempts in the process. The most significant indicator is the trend in the number of inbound zone transfers denied for lack of permission, for not being authorized for the zone, or for other reasons.

**Troubleshooting Based on the Results**

Check the primary and secondary server configurations if there are errors or exceeded limits in the inbound zone transfers.
DNS Network Errors

The DNS Network Errors dashboard element rendered as a line chart tracks the rate of change in DNS server network errors. The chart is available if you choose DNS Metrics: DNS Network Errors in the Chart Selection list.

The resulting line chart plots the following trends:

- **Query Error Packets/Query Responses**—Ratio of query error packets over responses. Responses consist of:
  - Authoritative
  - Authoritative no-such-name
  - Authoritative no-such-data
  - Nonauthoritative
  - Nonauthoritative no-such-data
  - Requests refused

- **Non Error Dropped Packets/Query Responses**—Ratio of nonerror dropped packets (queries dropped) over responses.

- **Update Errors/Updates**—Ratio of DNS Update errors over total updates.

How to Interpret the Data

This chart indicates query and response errors as an indication of the health of the server.

Troubleshooting Based on the Results

Check the DNS server network configuration if errors are increasing.

DNS Related Servers Errors

The DNS Related Servers Errors dashboard element rendered as a line chart tracks the rate of change in DNS related server errors. The chart is available if you choose DNS Metrics: DNS Related Servers Errors in the Chart Selection list.

The resulting line chart plots the following trends:

- **Referral Timeouts/Referrals**—Ratio of referral timeouts over referrals.

- **Failed Responses/Total Incoming Zone Transfer Requests**—Ratio of failed responses over incoming zone transfer requests.

- **TSIG Errors/TSIG Attempts**—Ratio of transaction signature (TSIG) errors (bad times, keys, or signatures) over total TSIG attempts (successfully received packets).

How to Interpret the Data

This chart indicates the health of connections and data transfers with related DNS servers. All three chart lines can have diagnostic significance.
Troubleshooting Based on the Results

Check the configurations and connectivity of the related servers in HA DNS relationships if errors are increasing.

DNS General Indicators

The DNS General Indicators dashboard element rendered as a table shows the server state, its last and startup reload time, the number of zones per server, and the total resource record (RR) count. The table is available if you choose DNS Metrics: DNS General Indicators in the Chart Selection list.

The resulting table shows:
- **Server State**—Up or Down (based on whether statistics are available), and how long the server has been in this state.
- **Last Reload**—How long since the last server reload.
- **Start Time**—Date and time of the last server process (Cisco Prime IP Express server agent) startup.
- **Total Zones**—Number of configured zones.
- **Total RRs**—Number of resource records.

How to Interpret the Data

The data in this chart shows general server health and operational duration. The objective is to make decisions about the server, such as whether it might be time for another reload, perhaps warranted by the number of configured zones.

Troubleshooting Based on the Results

If the server state is Down, all the DNS chart indicators show a red status box, so no data will be available. In the case of a server that is down, restart the server. The number of zones indicated might also require some evaluation and possible reconfiguration.

DNS Queries Per Second

The DNS Queries Per Second dashboard element, rendered as chart, displays queries per second for the Authoritative DNS server. This chart is available if you choose DNS Metrics: DNS Queries Per Second in the Chart Selection page.
Caching DNS Metrics

These Caching DNS metric elements are available in the dashboard:

- **DNS Queries Type**—See the “DNS Queries Type” section on page 3-20.
- **DNS Queries Responses**—See the “DNS Queries Responses” section on page 3-20.
- **DNS Incoming Queries**—See the “DNS Incoming Queries” section on page 3-21.
- **DNS Recursive Query Time**—See the “DNS Recursive Query Time” section on page 3-21.
- **DNS Caching**—See the “DNS Caching” section on page 3-22.
- **Caching DNS General Indicators**—See the “Caching DNS General Indicators” section on page 3-22.
- **DNS Caching Server Queries Per Second**—See the “DNS Caching Server Queries Per Second” section on page 3-23.

DNS Queries Type

The DNS Queries Type dashboard element rendered as a line chart traces the number queries by type. The chart is available if you choose Caching DNS Metrics: DNS Queries Type in the Chart Selection list.

The resulting line chart plots the following trends:

- **A**—Number of A queries received.
- **AAAA**—Number of AAAA queries received.
- **CNAME**—Number of CNAME queries received.

How to Interpret the Data

This chart shows the number of incoming queries of type A, AAAA, CNAME, PTR, and others.

DNS Queries Responses

The CDNS Query Responses dashboard element rendered as line chart shows the number of responses with NOERROR, NODOMAIN, No Data, Other Errors, Secure, and Unsecure return codes. The display is available if you choose Caching DNS Metrics: DNS Queries Responses in the Chart Selection list.

The resulting line chart plots the following trends:

- **NOERROR**—Number of answers from cache or recursion that result in rcode of NOERROR being returned to client.
- **NXDOMAIN**—Number of answers from cache or recursion that result in rcode of NXDOMAIN being returned to client.
- **NODATA**—Number of answers that result in pseudo rcode of NODATA being returned to client.
- **Other Errors**—Other errors.
- **Secure**—Number of answers that correctly validated.
- **Unsecure**—Number of answers that did not correctly validate.
How to Interpret the Data

This chart shows the following:

- The number of answers to queries, from cache or from recursion, that had the return code NXDOMAIN.
- The number of answers to queries that had the pseudo return code NODATA. This means the actual return code was NOERROR, but additionally, no data was carried in the answer (making what is called a NOERROR/NODATA answer). These queries are also included in the NOERROR number. Common for AAAA lookups when an A record exists, and no AAAA.
- Number of answers that were secure. The answer validated correctly. The AD bit might have been set in some of these answers, where the client signalled (with DO or AD bit in the query) that they were ready to accept the AD bit in the answer.
- Number of answers that did not correctly validate.

In a normal scenario, NOERROR is the successful response code.

Troubleshooting Based on the Results

Check the CDNS server configuration if the errors are increasing.

DNS Incoming Queries

The CDNS Incoming queries by dashboard element rendered as a line chart traces the TCP, IPv6, DNSSEC, EDNS and Total queries. The chart is available if you choose Caching DNS Metrics: DNS Incoming Queries in the Chart Selection list.

The resulting line chart plots the following trends:

- TCP—Total number of queries received over TCP by the CDNS Server.
- IPv6—Total number of queries received over IPv6 by the CDNS Server.
- EDNS—Number of queries with EDNS OPT RR present.
- DNSSec—Number of queries with EDNS OPT RR with DO (DNSSEC OK) bit set.
- Total—Total number of queries received by the CDNS Server.

How to Interpret the Data

This chart shows the number of queries that were made using TCP, IPv6, and DNSSEC towards the CDNS server, number of queries that had an EDNS OPT record present, and the total number of queries received.

DNS Recursive Query Time

The CDNS Queries by Type dashboard element rendered as a line chart traces the average time to complete a recursive query and the median time to complete a query. The table is available if you choose Caching DNS Metrics: DNS Recursive Query Time in the Chart Selection list.

The resulting line chart plots the following trends:

- Average—The average time to complete a recursive query.
- Median—The median time to complete a recursive query.
How to Interpret the Data

Average indicates the time the server took to answer queries that needed recursive processing. Note that the queries that were answered from the cache are not in this average.

Median time indicates the median of the time the server took to answer the queries that needed recursive processing. The median means that 50% of the user queries were answered in less than this time. Because of big outliers (usually queries to non-responsive servers), the average can be bigger than the median.

Troubleshooting Based on the Results

Check the connectivity and configuration for the name servers as forwarders or exception lists for the increasing values of the average and median time.

DNS Caching

The DNS Caching dashboard element rendered as a line chart traces the cache hits and cache misses. The chart is available if you choose Caching DNS Metrics: DNS Caching in the Chart Selection list.

The resulting line chart plots the following trends:
- **Cache Hits**—The total number of queries that were answered from cache.
- **Cache Misses**—The total number of queries that were not found in the cache.

How to Interpret the Data

This chart indicates the number of queries that were successfully answered using a cache lookup against the number of queries that needed recursive processing.

Troubleshooting Based on the Results

If the cache misses are increasing exponentially, check the CDNS logs for errors. Increasing rates of cache misses can indicate that not enough space is available in memory to store the cached queries for more efficient responses.

Caching DNS General Indicators

The Caching DNS General Indicators dashboard element shows the server state, its last and startup reload time, and the total resource record (RR) count. The table is available if you choose Caching DNS Metrics: Caching DNS General Indicators in the Chart Selection list.

The resulting line chart plots the following trends:
- **Server State**—Up or Down (based on whether statistics are available), and how long the server has been in this state.
- **Last Reload**—How long since the last server reload.
- **Start Time**—Date and time of the last server process (Cisco Prime IP Express server agent) startup.
How to Interpret the Data

The data in this chart shows general server health and operational duration. The objective is to make decisions about the server, such as whether it might be time for another reload, perhaps warranted by the number of configured zones.

Troubleshooting Based on the Results

If the server state is Down, all the CDNS chart indicators show a red status box, so no data will be available. In the case of a server that is down, restart the server.

DNS Caching Server Queries Per Second

The DNS Caching Server Queries Per Second dashboard element, rendered as chart, displays queries per second for the Caching DNS server. This chart is available if you choose CDNS Metrics: DNS Caching Server Queries Per Second in the Chart Selection page.
Deploying Cisco Prime IP Express

Cisco Prime IP Express is a full featured, scalable Domain Name System (DNS), and Dynamic Host Configuration Protocol (DHCP) implementation for medium to large IP networks. It provides the key benefits of stabilizing the IP infrastructure and automating networking services, such as configuring clients and provisioning cable modems. This provides a foundation for policy-based networking. Service provider and enterprise users can better manage their networks to integrate with other network infrastructure software and business applications.

Related Topics

Target Users, page 4-1
Regional and Local Clusters, page 4-1
Deployment Scenarios, page 4-2
Configuration and Performance Guidelines, page 4-5

Target Users

Cisco Prime IP Express is designed for:

- **Enterprises**—Helps meet the needs of single- and multisite enterprises (small-to-large businesses) to administer and control network functions. Cisco Prime IP Express automates the tasks of assigning IP addresses and configuring the Transport Control Protocol/Internet Protocol (TCP/IP) software for individual network devices. Forward-looking enterprise users can benefit from class-of-service and other features that help integrate with new or existing network management applications, such as user registration.

Regional and Local Clusters

The regional cluster acts as an aggregate management system for up to a hundred local clusters. Address and server administrators interact at the regional and local clusters through the regional and local web-based user interfaces (web UIs), and local cluster administrators can continue to use the command line interface (CLI) at the local cluster. The regional cluster consists of a Central Configuration Management (CCM) server, Tomcat web server, servlet engine, and server agent (see Chapter 1, “Cisco Prime IP Express Components”). The license management is now done at the regional cluster and hence the local server has to be registered to a regional server to avail the necessary services. See the Overview chapter of Cisco Prime IP Express Installation Guide for more details.
A typical deployment is one regional cluster at a customer network operation center (NOC), the central point of network operations for an organization. Each division of the organization includes a local address management server cluster responsible for managing a part of the network. The System Configuration Protocol (SCP) communicates the configuration changes between the servers.

**Deployment Scenarios**

The Cisco Prime IP Express regional cluster web UI provides a single point to manage any number of local clusters hosting DNS, CDNS, or DHCP servers. The regional and local clusters also provide administrator management so that you can assign administrative roles to users logged in to the application.

This section describes two basic administrative scenarios and the hardware and software deployments for two different types of installations—a small-to-medium local area network (LAN), and a large-enterprise or service-provider network with three geographic locations.

**Related Topics**

Small-to-Medium-Size LANs, page 4-3
Large Enterprise Network, page 4-3
Small-to-Medium-Size LANs

In this scenario, low-end Windows, or Linux servers are acceptable. You can also use systems with EIDE disks, although we recommend you use Ultra-SCSI disks for dynamic DNS update. Figure 4-2 shows a configuration that would be adequate for this network.

Note
Regional server is MUST in deployment for small and medium sized LANs.

Figure 4-2  Small-to-Medium LAN Configuration

Large Enterprise Network

In a large enterprise network serving over 500,000 DHCP clients, use mid-range Windows, or Linux servers. Put DNS and DHCP servers on different systems. Figure 4-3 shows the hardware that would be adequate for this network.

When supporting geographically dispersed clients, locate DHCP servers at remote locations to avoid disrupting local services if wide-area connections fail. Install the Cisco Prime IP Express regional cluster to centrally manage the distributed clusters.
Figure 4-3  Large Enterprise Network Configuration

Site 1
DHCP clients
(on multiple subnets)

Router

Ethernet

Primary DNS server

Secondary DNS server

Services client queries

Main DHCP server

Backup DHCP server

CDNS Server

Regional cluster server

Site 2
DHCP clients
(on multiple subnets)

Router

Primary DNS server

Main DHCP server

Backup DHCP server

CDNS server
Configuration and Performance Guidelines

Cisco Prime IP Express is an integrated DHCP, and DNS server cluster capable of running on a Windows, or Linux workstation or server.

Because of the wide range of network topologies for which you can deploy Cisco Prime IP Express, you should first consider the following guidelines. These guidelines are very general and cover most cases. Specific or challenging implementations could require additional hardware or servers.

Related Topics

General Configuration Guidelines, page 4-5
Special Configuration Cases, page 4-6

General Configuration Guidelines

The following suggestions apply to most Cisco Prime IP Express deployments:

- Configure a separate DHCP server to run in remote segments of the wide area network (WAN).
  Ensure that the DHCP client can consistently send a packet to the server in under a second. The DHCP protocol dictates that the client receive a response to a DHCPDISCOVER or DHCPREQUEST packet within four seconds of transmission. Many clients (notably early releases of the Microsoft DHCP stack) actually implement a two-second timeout.

- In large deployments, separate the secondary DHCP server from the primary DNS server used for dynamic DNS updates.
  Because lease requests and dynamic DNS updates are persisted to disk, server performance is impacted when using a common disk system. So that the DNS server is not adversely affected, run it on a different cluster than the DHCP server.

- Include a time server in your configuration to deal with time differences between the local and regional clusters so that aggregated data at the regional server appears in a consistent way. See the “Polling Lease History Data” section on page 6-13.

- Set DHCP lease times in policies to four to ten days.
  To prevent leases from expiring when the DHCP client is turned off (overnight or over long weekends), set the DHCP lease time longer than the longest period of expected downtime, such as seven days. See Chapter 23, “Managing Leases.”

- Locate backup DNS servers on separate network segments.
  DNS servers are redundant by nature. However, to minimize client impact during a network failure, ensure that primary and secondary DNS servers are on separate network segments.

- If there are high dynamic DNS update rates in the network, configure separate DNS servers for forward and reverse zones.

- Use NOTIFY/IXFR.
  Secondary DNS servers can receive their data from the primary DNS server in two ways: through a full zone transfer (AXFR) or an incremental zone transfer (NOTIFY/IXFR, as described in RFCs 1995 and 1996). Use NOTIFY/IXFR in environments where the name space is relatively dynamic. This reduces the number of records transferred from the primary to the secondary server. See the “Enabling Incremental Zone Transfers (IXFR)” section on page 17-5.
Special Configuration Cases

The following suggestions apply to some special configurations:

- When using dynamic DNS updates for large deployments or very dynamic networks, divide primary and secondary DNS and DHCP servers across multiple clusters.

  Dynamic DNS updates generate an additional load on all Cisco Prime IP Express servers as new DHCP lease requests trigger dynamic DNS updates to primary servers that update secondary servers through zone transfers.

- During network reconfiguration, set DHCP lease renewal times to a small value.

  Do this several days before making changes in network infrastructure (such as to gateway router and DNS server addresses). A renewal time of eight hours ensures that all DHCP clients receive a changed DHCP option parameter within one working day. See Chapter 23, “Managing Leases.”
PART 2

Local and Regional Administration
CHAPTER 5

Configuring Administrators

This chapter explains how to set up network administrators at the local and regional clusters. The chapter also includes local and regional cluster tutorials for many of the administration features.

Related Topics

- Administrators, Groups, and Roles, page 5-1
- External Authentication Servers, page 5-8
- Granular Administration, page 5-12
- Licensing, page 5-15
- License History, page 5-16
- Centrally Managing Administrators, page 5-17
- Local Cluster Management Tutorial, page 5-25
- Regional Cluster Management Tutorial, page 5-32

Administrators, Groups, and Roles

The types of functions that network administrators can perform in Cisco Prime IP Express are based on the roles assigned to them. Local and regional administrators can define these roles to provide granularity for the network administration functions. Cisco Prime IP Express predefines a set of base roles that segment the administrative functions. From these base roles you can define further constrained roles that are limited to administering particular addresses, zones, and other network objects.

The mechanism to associate administrators with their roles is to place the administrators in groups that include these roles.

Related Topics

- How Administrators Relate to Groups, and Roles, page 5-2
- Administrator Types, page 5-2
- Roles, Subroles, and Constraints, page 5-2
- Groups, page 5-5
- Managing Administrators, page 5-5
- Managing Passwords, page 5-6
- Managing Groups, page 5-6
- Managing Roles, page 5-7
How Administrators Relate to Groups, and Roles

There are three administrator objects in Cisco Prime IP Express—administrator, group, and role:

- **Administrator**—An account that logs in and that, through its association with one or more administrator groups, can perform certain functions based on its assigned role or roles. At the local cluster, these functions are administering the local Central Configuration Management (CCM) server and databases, hosts, zones, address space, and DHCP. At the regional cluster, these functions administer the regional CCM server and databases, central configuration, and regional address space. An administrator must be assigned to at least one group to be effective.

  Adding administrators is described in the “Managing Administrators” section on page 5-5.

- **Group**—A grouping of roles. You must associate one or more groups with an administrator, and a group must be assigned at least one role to be usable. The predefined groups that Cisco Prime IP Express provides map each role to a unique group.

  Adding groups is described in the “Managing Groups” section on page 5-6.

- **Role**—Defines the network objects that an administrator can manage and the functions that an administrator can perform. A set of predefined roles are created at installation, and you can define additional constrained roles. Some of the roles include subroles that provide further functional constraints.

  Adding roles is described in the “Managing Roles” section on page 5-7.

Administrator Types

There are two basic types of administrators: superusers and specialized administrators:

- **Superuser**—Administrator with unrestricted access to the web UI, CLI, and all features. This administrator type should be restricted to a few individuals. The superuser privileges of an administrator override all its other roles.

  **Tip**

  You have to create the superuser and password at installation, or when you first log into the web UI.

- **Specialized**—Administrator created by name to fulfill specialized functions, for example, to administer a specific DNS forward or reverse zone, based on the administrator assigned role (and subrole, if applicable). Specialized administrators, like the superuser, require a password, but must also be assigned at least one administrator group that defines the relevant roles. The CLI provides the admin command.

  For an example of creating a local zone or host administrator, see the “Create the Administrators” section on page 5-26.

Roles, Subroles, and Constraints

A license type is associated with each role-subrole combination. A role-subrole is enabled only if that license is available in that cluster.

You can limit an administrator role by applying constraints. For example, you can use the host-admin base role to create a host administrator, named 192.168.50-host-admin, who is constrained to the 192.168.50.0 subnet. The administrator assigned a group that includes this role then logs in with this...
constraint in effect. Adding roles and subroles is described in the “Managing Roles” section on page 5-7.

You can further limit the constraints on roles to read-only access. An administrator can be allowed to read any of the data for that role, but not modify it. However, if the constrained data is also associated with a read-write role, the read-write privilege supersedes the read-only constraints.

An example of adding role constraints is in the “Create a Host Administrator Role with Constraints” section on page 5-30.

The interplay between DNS and host administrator role assignments is such that you can combine an unconstrained dns-admin role with any host-admin role in a group. For example, combining the dns-admin-readonly role and a host-admin role in a group (and naming the group host-rw-dns-ro) provides full host access and read-only access to zones and RRs. However, if you assign a constrained dns-admin role along with a host-admin role to a group and then to an administrator, the constrained dns-admin role takes precedence, and the administrator privileges at login will preclude any host administration.

Certain roles provide subroles with which you can further limit the role functionality. For example, the local ccm-admin or regional-admin, with just the owner-region subrole applied, can manage only owners and regions. By default, all the possible subroles apply when you create a constrained role.

The predefined roles are described in Table 5-1 (local), and Table 5-2 on page 5-4 (regional).

### Table 5-1  Local Cluster Administrator Predefined and Base Roles

<table>
<thead>
<tr>
<th>Local Role</th>
<th>Subroles and Active Functionality</th>
</tr>
</thead>
</table>
| addrblock-admin | Core functionality: Manage address block, subnets, and reverse DNS zones (also requires dns-admin); and notify of scope activity.  
  - ipv6-management: Manage IPv6 prefixes, links, options, leases, and reservations. |
| ccm-admin     | Core functionality: Manage access control lists (ACLs), and encryption keys.  
  - authentication: Manage administrators.  
  - authorization: Manage roles and groups.  
  - owner-region: Manage owners and regions.  
  - database: View database change entries and trim the CCM change sets. |
| cdns-admin    | Core functionality: Manage in-memory cache (flush cache and flush cache name).  
  - security-management: Manage ACLs and DNSSEC configuration.  
  - server-management: Manage DNSSEC configuration, as well as forwarders, exceptions, DNS64, and scheduled tasks, and stop, start, or reload the server. |
| cfg-admin     | Core functionality: Manage clusters.  
  - ccm-management: Manage the CCM server configuration.  
  - dhcp-management: Manage the DHCP server configuration.  
  - dns-management: Manage the DNS server configuration.  
  - cdns-management: Manage Caching DNS server configuration.  
  - snmp-management: Manage the SNMP server configuration. |
### Administrators, Groups, and Roles

**Chapter 5**  Configuring Administrators

**Table 5-1**  *Local Cluster Administrator Predefined and Base Roles (continued)*

<table>
<thead>
<tr>
<th>Local Role</th>
<th>Subroles and Active Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>dhcp-admin</td>
<td>Core functionality: Manage DHCP scopes and templates, policies, clients, client-classes, options, leases, and reservations.</td>
</tr>
<tr>
<td></td>
<td><em>server-management</em>: Manage the DHCP server configuration, failover pairs, LDAP servers, extensions, and statistics.</td>
</tr>
<tr>
<td></td>
<td><em>ipv6-management</em>: Manage IPv6 prefixes, links, options, leases, and reservations.</td>
</tr>
<tr>
<td>dns-admin</td>
<td>Core functionality: Manage DNS zones and templates, resource records, secondary servers, and hosts.</td>
</tr>
<tr>
<td></td>
<td><em>security-management</em>: Manage DNS update policies, ACLs, and encryption keys.</td>
</tr>
<tr>
<td></td>
<td><em>server-management</em>: Manage DNS server configurations and zone distributions, synchronize zones and HA server pairs, and push update maps.</td>
</tr>
<tr>
<td></td>
<td><em>ipv6-management</em>: Manage IPv6 zones and hosts.</td>
</tr>
<tr>
<td>host-admin</td>
<td>Core functionality: Manage DNS hosts. (Note that if an administrator is also assigned a constrained dns-admin role that overrides the host-admin definition, the administrator is not assigned the host-admin role.)</td>
</tr>
</tbody>
</table>

**Table 5-2**  *Regional Cluster Administrator Predefined and Base Roles*

<table>
<thead>
<tr>
<th>Regional Role</th>
<th>Subroles and Active Functionality</th>
</tr>
</thead>
</table>
| central-cfg-admin | Core functionality: Manage clusters and view replica data.  
|                  | *dhcp-management*: Manage DHCP scope templates, policies, client-classes, failover pairs, virtual private networks (VPNs), and options; modify subnets; and replicate data. |
| central-dns-admin | Core functionality: Manage DNS zones and templates, hosts, resource records, and secondary servers; and create subzones and reverse zones.                                                                                       |
|                  | *security-management*: Manage DNS update policies, ACLs, and encryption keys.                                                                                                                                                           |
|                  | *server-management*: Synchronize DNS zones and HA server pairs, manage zone distributions, pull replica zone data, and push update maps.                                                                                          |
| central-host-admin | Core functionality: Manage DNS hosts. (Note that if an administrator is also assigned a constrained central-dns-admin role that overrides the central-host-admin definition, the administrator is not assigned the central-host-admin role.) |
| regional-admin  | Core functionality: Manage licenses and encryption keys.                                                                                                           |
|                  | *authentication*: Manage administrators.                                                                                                                                                                                               |
|                  | *authorization*: Manage roles and groups.                                                                                                                                                                                                 |
|                  | *owner-region*: Manage owners and regions.                                                                                                                                         |
|                  | *database*: View database change entries and trim the CCM change sets.                                                                                                                                                                   |
Groups

Administrator groups are the mechanism used to assign roles to administrators. Hence, a group must consist of one or more administrator roles to be usable. When you first install Cisco Prime IP Express, a predefined group is created to correspond to each predefined role.

Roles with the same base role are combined. A group with an unconstrained dhcp-admin role and a constrained dns-admin role, does not change the privileges assigned to the dns-admin role. For example, if one of the roles is assigned unconstrained read-write privileges, the group is assigned unconstrained read-write privileges, even though other roles might be assigned read-only privileges. Therefore, to limit the read-write privileges of a user while allowing read-only access to all data, create a group that includes the unconstrained read-only role along with a constrained read-write role. (See the “Roles, Subroles, and Constraints” section on page 5-2 for the implementation of host-admin and dns-admin roles combined in a group.)

Managing Administrators

When you first log in, Cisco Prime IP Express will have one administrator—the superuser account. This superuser can exercise all the functions of the web UI and usually adds the other key administrators. However, ccm-admin and regional-admin administrators can also add, edit, and delete administrators.

Creating an administrator requires:

- Adding its name.
- Adding a password.
- Specifying if the administrator should have superuser privileges (usually assigned on an extremely limited basis).
- If not creating a superuser, specifying the group or groups to which the administrator should belong. These groups should have the appropriate role (and possibly subrole) assignments, thereby setting the proper constraints.

Tip

If you accidentally delete all the roles by which you can log into Cisco Prime IP Express (those having superuser, ccm-admin, or regional-admin privileges), you can recover by creating a username/password pair in the install-path/conf/priv/local.superusers file. You must create this file, have write access to it, and include a line in it with the format username password. Use this username and password for the next login session. Note, however, that using the local.superusers file causes reduced security. Therefore, use this file only in emergencies such as when temporarily losing all login access. After you log in, create a superuser account in the usual way, then delete the local.superusers file or its contents. You must create a new administrator account for each individual, to track administrative changes.
Local and Regional Web UI

To add a administrator, do the following:

**Step 1**  From the Administration menu, choose Administrators under the User Access submenu. This opens the List/Add Administrators page (see the “Create the Administrators” section on page 5-26 for an example).

**Step 2**  Click the Add Administrator icon in the Administrators pane, enter a name and password in the Add Admin dialog box, then click Add Admin.

**Step 3**  Choose one or more existing groups from the Groups Available list (or whether the administrator should be a superuser), then click Save.

To edit an administrator, select the administrator in the Administrators pane, then modifying the name, password, superuser status, or group membership on the Edit Administrator page, then click Save. The active group or groups should be in the Selected list.

To delete an administrator, select the administrator in the Administrators pane, click the Delete icon, and then confirm or cancel the deletion.

Managing Passwords

Passwords are key to administrator access to the web UI and CLI. In the web UI, you enter the password on the Login page. In the CLI, you enter the password when you first invoke the nrcmd program. The local or regional CCM administrator or superuser can change any administrator password.

You can prevent exposing a password on entry. In the web UI, logging in or adding a password never exposes it on the page, except as asterisks. In the CLI, you can prevent exposing the password by creating an administrator, omitting the password, then using admin name enterPassword, where the prompt displays the password as asterisks. You can do this instead of the usual admin name set password command that exposes the password as plain text.

Administrators can change their own passwords on clusters. If you want the password change propagated from the regional server to all local clusters, log into the regional cluster. First ensure that your session admin-edit-mode is set to synchronous, and then update your password.

**Note**  The password should not be more than 255 characters long.

Managing Groups

A superuser, ccm-admin, or regional-admin can create, edit, and delete administrator groups. Creating an administrator group involves:

- Adding its name.
- Adding an optional description.
- Choosing associated roles.
Local Advanced and Regional Web UI

To add a group, do the following:

**Step 1** From the Administration menu, choose Groups under the User Access submenu. This opens the List/Add Administrator Groups page (see the “Create a Group to Assign to the Host Administrator” section on page 5-31 for an example).

**Step 2** Click the Add Groups icon in the Groups pane, enter a name and an optional description in the Add Group dialog box, then click Add CCMAdminGroup.

**Step 3** Choose one or more existing roles from the Roles Available list, then click Save.

To edit a group, click the name of the group that you want to edit in the Groups pane to open the Edit Administrator Group page. You can modify the name, description, or role membership in this page. You can view the active roles in the Selected list.

To delete a group, select the group in the Groups pane, click the Delete icon, and then confirm the deletion. Click Cancel in the confirmation window to cancel the deletion.

Managing Roles

A superuser, ccm-admin, or regional-admin administrator can create, edit, and delete administrator roles. Creating an administrator role involves:

- Adding its name.
- Choosing a base role.
- Possibly specifying if the role should be unconstrained, or read-only.
- Possibly adding constraints.
- Possibly assigning groups.

Local Advanced Web UI

**Step 1** From the Administration menu, choose Roles under the User Access submenu. This opens the List/Add Administrator Roles page.

**Step 2** Click the Add Role icon in the Roles pane, enter a name in the Add Roles dialog box and click Add Role.

**Step 3** On the List/Add Administrator Roles page, specify any role constraints, subrole restrictions, or group selections, then click Save.

Regional Advanced Web UI

**Step 1** From the Administration menu, choose Roles under the User Access submenu. This opens the List/Add Administrator Roles page.

**Step 2** Click the Add Role icon in the Roles pane, enter a name, and a base role in the Add Roles dialog box and click Add Role.
Step 3  On the List/Add Administrator Roles page, specify any role constraints, subrole restrictions, or group selections, then click **Save**.

To edit a role, select the role in the Roles pane, then modifying the name or any constraints, subrole restrictions, or group selections on the Edit Administrator Role page. The active subroles or groups should be in the Selected list. Click **Save**.

To delete a role, select the role in the Roles pane, click the **Delete** icon, and then confirm the deletion.

**Note**  You cannot delete the default roles.

### External Authentication Servers

Cisco Prime IP Express includes a RADIUS client component and Active Directory (AD) client component, which are integrated with the authentication and authorization modules of the CCM server. To enable external authentication, you must configure a list of external RADIUS or an AD server at local and regional clusters, and ensure all authorized users are appropriately configured on the respective servers.

When external authentication is enabled, the CCM server handles attempts to log in via the web UI, SDK, or CLI, by issuing a RADIUS request to a RADIUS server or a LDAP request to an AD server that is selected from the configured list. If the corresponding server validates the login request, access is granted, and the CCM server creates an authorized session with the group assignments specified by the RADIUS or the AD server.

**Note**  Any administrators defined in the CCM server’s database are ignored when external authentication is enabled. Attempting to log in with these usernames and passwords will fail. To disable external authentication, you must remove or disable all the configured external servers or change the **auth-type** attribute value to Local.

**Tip**  If all logins fail because the RADIUS or the AD servers are inaccessible or misconfigured, use the local.superusers file to create a temporary username and password. See the “Managing Administrators” section on page 5-5 for more details.

### Configuring an RADIUS External Authentication Server

Cisco Prime IP Express administrators must be assigned to one or more administrator groups to perform management functions. When using a RADIUS server for external authentication, these are set as a vendor specific attribute for each user. Using the Cisco vendor id (9), create the Cisco Prime IP Express groups attribute for each administrator, using the format **cnr:groups=group1,group2,group3**.

For example, to assign an administrator to the built-in groups **dhcp-admin-group** and **dns-admin-group**, enter:

```
cnr:groups=dhcp-admin-group,dns-admin-group
```
To assign superuser access privileges, the reserved group name **superusers** is used. To provide superuser privileges to an administrator, enter:

```
cnr:groups=superusers
```

The superuser privileges override all other groups.

**Note**

You cannot add, delete, or modify external user names and their passwords or groups using Cisco Prime IP Express. You must use the RADIUS server to perform this configuration.

**Related Topics**

- Adding an RADIUS External Configuration Server, page 5-9
- Deleting an RADIUS External Authentication Server, page 5-9
- Pushing and Pulling External Authentication Servers, page 5-20

**Adding an RADIUS External Configuration Server**

To add an external configuration server, do the following:

**Local Advanced and Regional Web UI**

**Step 1**

From the **Administration** menu, choose **RADIUS** under the **External Authentication** submenu. The List/Add RADIUS Server page is displayed.

**Step 2**

Click the **Add RADIUS** icon in the RADIUS pane, enter the name, address of the server you want to configure as the external authentication server, and you can set the key attribute which will be used for communicating with this server in the External Authentication dialog box, and click **Add External Authentication Server**. CCM server uses the key to set the key-secret attribute which is the secret key shared by client and the server.

**Step 3**

To enable the external authentication server, check **enabled** check box of the ext-auth attribute in the Edit Authentication Server page, and then click **Save**.

**Step 4**

Change the **auth-type** attribute to **RADIUS** in the Manage Server page, click **Save** and then restart Cisco Prime IP Express.

**CLI Commands**

To create an external authentication server, use **auth-server name create address [attribute=value ...]** (see the **auth-server** command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions).

**Deleting an RADIUS External Authentication Server**

**Local Advanced and Regional Web UI**

To delete an RADIUS external authentication server, select the server in the RADIUS pane and click the **Delete RADIUS** icon, then confirm the deletion. You can also cancel the deletion by clicking the Close button.
Configuring an AD External Authentication Server

Cisco Prime IP Express administrators must be assigned to one or more administrator groups to perform management functions. The Cisco Prime IP Express groups attribute for each administrator can be created using the format `cnr:groups=group1,group2,group3`.

For example, to assign an administrator to the built-in groups `dhcp-admin-group` and `dns-admin-group`, enter:

```
cnr:groups=dhcp-admin-group,dns-admin-group
```

To assign superuser access privileges, the reserved group name `superusers` is used. To provide superuser privileges to an administrator, enter:

```
cnr:groups=superusers
```

The superuser privileges override all other groups.

A group needs to be created to access CPIPE and add the users to that group. Select an user attribute and provide the group information in the format `cnr:group1,group2,...`

---

**Step 1**
In AD server, create a new group, for example **CPIPE**, with the group scope *Domain Local*.

**Step 2**
Select an user and click **Add** to a group.

**Step 3**
In Enter the Object Names window, select **CPIPE** and click **OK**.

**Step 4**
In AD Server Object windows, select **CPIPE** for the `ad-group-name` attribute and `info` for the `ad-user-attr-map` attribute.

---

**Note**
You cannot add, delete, or modify external user names and their passwords or groups using Cisco Prime IP Express. You must use the AD server to perform this configuration.

---

Configuring Kerbero’s Realm and KDC

For the Cisco Prime IP Express to communicate with the AD server, the Kerbero’s Realm and KDC servers are required. To configure the Kerbero’s Realm and KDC servers in Windows and Linux platforms follow the below examples.

If the IP Express is running on Windows platform (`ksetup`):

To define a KDC entry for a realm:

```
\ksetup /AddKdc <RealmName> [KdcName]
```

For example:

```
\ksetup /AddKdc ECNR.COM tm-chn-ecnr-ad.ecnr.com
```

```
\ksetup /dumpstate
```

```
default realm = partnet.cisco.com (NT Domain)
ECNR.COM:
kdc = tm-chn-ecnr-ad.ecnr.com
Realm Flags = 0x0 No Realm Flags
No user mappings defined.
```
If the IP Express is running on Linux platform the following similar setup has to be configured in `krb5.conf` file:

```bash
default = FILE:/var/log/krb5libs.log
dc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
ticket_lifetime = 1d
default_realm = ECNR.COM
default_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac
dns_lookup_realm = false
dns_lookup_kdc = false
forwardable = true

[realms]
ECNR.COM = {
dc = <kdc server host name>
admin_server = <kdc server host name>
}

[domain_realm]
.ecnr.com = ECNR.COM
ecnr.com = ECNR.COM
```

Related Topics

- Adding an AD External Configuration Server, page 5-11
- Deleting an AD External Authentication Server, page 5-12
- Pushing and Pulling External Authentication Servers, page 5-20

Adding an AD External Configuration Server

To add an external configuration server, do the following:

**Local Advanced and Regional Web UI**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>From the Administration menu, choose Active Directory under the External Authentication submenu. The List/Add Active Directory Server page is displayed.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Click the Add Active Directory icon in the Active Directory pane, enter the name, hostname of the server, domain you want to configure as the external authentication server, and you can set the base domain, LDAP user attribute map, AD group name which will be used for communicating with this server in the External Authentication dialog box, and click Add External Authentication Server.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Change the auth-type attribute to Active Directory in the Manage Server page, click Save and then restart Cisco Prime IP Express.</td>
</tr>
</tbody>
</table>

**CLI Commands**

To create an external authentication server, use `auth-ad-server name create addr [attribute=value ...]` (see the `auth-ad-server` command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions).
Deleting an AD External Authentication Server

Local Advanced and Regional Web UI

To delete an AD external authentication server, select the server in the AD pane and click the Delete Active Directory icon, then confirm the deletion. You can also cancel the deletion by clicking the Close button.

Granular Administration

Granular administration prevents unauthorized users from accidentally making a change on zones, address blocks, subnets, and router interfaces. It also ensures that only authorized users view or modify specific scopes, prefixes, and links. Granular administration constraints administrators to specific set of scopes, prefixes, and links. A constrained administrator can view or make changes to authorized scope, prefix, and link objects only. The CCM server uses owner and region constraints to authorize and filter IPv4 address space objects, and DNS zone related objects (CCMZone, CCMReverseZone, CCMSecondaryZone, CCMRRSet, and CCMHost). The zones are constrained by owners and regions. Owner or region attributes on the CCMSUBnet control access to scopes. Also, owner or region attributes on the Prefix and Link objects control access to prefixes and links.

Local Advanced and Regional Web UI

Step 1 From the Administration menu, choose Roles to open the List/Add Administrator Roles page.
Step 2 Click the Add Role icon in the Roles pane, enter a name for the custom role, for example, my-dhcp, and choose dhcp-admin from the Role drop-down list and click Add Role.
Step 3 Click True or False radio button as necessary, on the Add DHCP Administrator Role page.
Step 4 Choose the required sub roles in the Available field and move them to the Selected field.
Step 5 Click Add Constraint.
   a. On the Add Role Constraint page, modify the fields as necessary.
   b. Click Add Constraint. The constraint must have an index number of 1.
Step 6 Click Save.
The name of the custom role appears on the list of roles in the List/Add Administrator Roles page.

Related Topics

Scope-Level Constraints, page 5-12
Prefix-Level Constraints, page 5-14
Link-Level Constraints, page 5-15

Scope-Level Constraints

A dhcp admin user can view or modify a scope if any of the following conditions is met:
• Owner of the subnet for the scope matches the dhcp-admin owner.
Region of the subnet for the scope matches the region role constraints.
Owner or region of the parent address block matches the dhcp-admin owner or region role constraints. Note that the most immediate parent address block that has owner or region defined takes precedence.

The following conditions are also valid:
- If the matching owner or region constraint is marked as read-only, you can only view the scope.
- If a scope has a primary network defined, the primary subnet and its parent address block owner or region constraints override secondary subnets.
- If no parent subnet or address block defines owner or region constraints, then you can access the scope.
- If you are an unconstrained dhcp-admin user, you can have access to all scopes.

Note
These hierarchical authorization checks for dhcp-admin owner/region constraints are applicable to scopes, subnets, and parent address blocks. Identical hierarchical authorization checks for addrblock-admin owner/region constraints apply to address blocks and subnets. If you have dhcp-admin and the addrblock-admin privileges, you can access address blocks and subnets, if either of the roles allow access.

Examples of Scope-Level Constraints:

Parent CCMAddrBlock 10.0.0.0/8 has owner 'blue' set.
  Scope 'A' has subnet 10.0.0.0/24 has parent CCMSubnet with owner 'red'.
  Scope 'B' has subnet 10.0.1.0/24 has parent CCMSubnet with no owner set.
  Scope 'C' has subnet 10.10.0.0/24 has parent CCMSubnet with owner 'green' and primary-subnet 10.0.0.0/24.
  Scope 'D' has subnet 100.10.0.0/24 has parent CCMSubnet with owner unset, and no parent block.

Scope 'A' owner is 'red'.
Scope 'B' owner is 'blue'.
Scope 'C' owner is 'red'.
Scope 'D' owner is unset. Only unconstrained users can access this scope.

Local Advanced Web UI

To add scopes, do the following:

Step 1 From the Design menu, choose Scopes under the DHCPv4 submenu to open the List/Add DHCP Scopes.

Step 2 Click the Add Scopes icon in the Scopes pane, enter a name, subnet, primary subnet, choose policy, and enter a selection-tag-list in the Add DHCP Scope dialog box.

Step 3 Click Add DHCP Scope. The List/Add DHCP Scopes page appears.

Step 4 Enter values for the fields or attributes as necessary.

Step 5 To unset any attribute value, check the check box in the Unset? column, then click Unset Fields at the bottom of the page.

Step 6 Click Save to add scope, or Revert to cancel the changes.
If you add new scope values or edit existing ones, click **Save** to save the scope object.

**Prefix-Level Constraints**

You can view or modify a prefix, if you have either of the following:

- The ipv6-management subrole of the dhcp-admin, or addrblock-admin role on the local cluster.
- The central-cfg-admin, or regional-addr-admin role on the regional cluster.

You can view or modify a prefix if any of the following conditions is true:

- The owner or region of the parent link matches the owner or region role constraints defined for you.
- The owner or region of this prefix matches the owner or region role constraints defined for you.
- The owner or region of the parent prefix matches the owner or region role constraints defined for you.

You can view or modify a prefix if any of the following conditions is true:

- If the matching owner or region constraint for you is marked as read-only, then you can only view the prefix.
- If the prefix references a parent link, the link owner or region constraints is applicable if the link owner or region constraints set.
- If no parent link or prefix defines any owner or region constraints, then you can access this prefix only if owner or region role constraints are not defined for you.
- If you are an unconstrained user, then you have access to all.

**Examples of Prefix-Level constraints:**

Link ‘BLUE’ has owner ‘blue’ set.
Parent Prefix ‘GREEN’ has owner ‘green’ set.
Prefix ‘A’ has owner ‘red’ set, no parent prefix, and no parent link.
Prefix ‘B’ has owner ‘yellow’ set, parent Prefix ‘GREEN’ and parent link ‘BLUE’.
Prefix ‘C’ has no owner set, parent prefix ‘GREEN’, and no parent link.
Prefix ‘C’ has no owner set, no parent prefix, and no parent link.

Prefix ‘A’ owner is ‘red’.
Prefix ‘B’ owner is ‘blue’.
Prefix ‘C’ owner is ‘green’.
Prefix ‘D’ owner is unset. Only unconstrained users can access this prefix.

**Local Advanced and Regional Web UI**

To view unified v6 address space, do the following:

**Step 1**  
From the **Design** menu, choose **Address Tree** under the **DHCPv6** submenu to open the **DHCP v6 Address Tree** page.

**Step 2**  
View a prefix by adding its name, address, and range, then choosing a DHCP type and possible template (see the “Viewing IPv6 Address Space” section on page 27-14).

**Step 3**  
Choose the owner from the owner drop-down list.

**Step 4**  
Choose the region from the region drop-down list.
Step 5  Click **Add Prefix**. The newly added Prefix appears on the DHCP v6 Address Tree page.

To list or add DHCP prefixes, do the following:

Step 1  From the **Design** menu, choose **Prefixes** under the DHCPv6 submenu to open the List/Add DHCP v6 Prefixes page.

Step 2  Click the **Add Prefixes** icon in the Prefixes pane, enter a name, address, and range for the prefix, then choose the DHCP type and possible template.

Step 3  Choose the owner from the owner drop-down list.

Step 4  Choose the region from the region drop-down list.

Step 5  Click **Add IPv6 Prefix**. The newly added Prefix appears on the List Prefixes page.

---

**Link-Level Constraints**

You can view or modify a link if:

- You are authorized for the ipv6-management subrole of the dhcp-admin or addrblock-admin role on the local cluster, or the central-cfg-admin or regional-addr-admin role on the regional cluster.
- The owner or region of the link matches the owner or region role constraints defined for you.
- No owner or region is defined for the link, and only if no owner or region role constraints are defined for you.

If you are an unconstrained user, then you have access to all links.

The following is an example of Link Level Constraints:

Link 'BLUE' has owner 'blue' set.
Link 'ORANGE' has owner unset.

Link 'BLUE' owner is 'blue'.
Link 'ORANGE' owner is unset. Only unconstrained users can access this link.

---

**Local Advanced and Regional Web UI**

To add links, do the following:

Step 1  From the **Design** menu, choose **Links** under the DHCPv6 submenu to open the List/Add DHCP v6 Links page.

Step 2  Click the **Add Links** icon in the Links pane, enter a name, then choose the link type, and enter a group.

Step 3  Click **Add Link**. The newly added DHCPv6 Link appears on the List/Add DHCP v6 Links page.

---

**Licensing**
Cisco Prime IP Express 8.2 provides separate license for CCM, Authoritative DNS, Caching DNS, DHCP, and IPAM services or for combinations of these services. For more details on the Licensing, see the “License Files” section in the Overview chapter of the Cisco Prime IP Express Installation Guide.

You must have the Central Configuration Management (CCM) license to log into the UI. See the “Logging In to the Web UIs” section on page 2-2 for entering license data the first time you try to log in. You can add the additional service based licenses in the regional server after you log in.

Whenever you log into a regional or local cluster, the overall licensing status of the system is checked. If there are any violations, you will be notified of the violation and the details. This notification is done only once for each user session. In addition, you will be able to see a message on each page indicating the violation.

Regional Web UI

Choose Licenses from Administration > User Access to open the List/Add Product Licenses page. Click Browse to locate the license file, click the file, then click Open. If the license ID in the file is valid, the license key appears in the list of licenses with the message “Successfully added license file "filename." If the ID is not valid, the License field shows the contents of the file and the message “Object is invalid” appears.

The License Utilization section at the top of the page lists the type of license, the number of nodes allowed for the license, and the actual number of nodes used. Expand the section by clicking the plus (+) sign. The license utilization for each licensed service is listed separately in this section.

The Right To Use and the In Use counts are displayed for each licensed service. The Right To Use value will be the aggregation of the counts across all added licenses for that service. The ‘total in use’ value will be the aggregation of the latest utilization numbers obtained from all the local clusters. Only the services having a positive Right to use or In Use count will be listed in this section.

Licenses and usage count of Cisco Prime IP Express will be listed under the section “ip-node”.

The Expert mode attribute lets you specify how often license utilization is collected from all the local clusters. Changes to this setting require a server restart to take effect. You can set this attribute at the Edit CCM Server page. The default value is 4 hours.

CLI Commands

Use license file create to register licenses that are stored in file. The file referenced should include its absolute path or path relative to where you execute the commands. For example:

```
ncmd-R> license "C:\licenses\product.licenses" create
```

Use license list to list the properties of all the created licenses (identified by key), and license listnames to list just the keys. Use license key show to show the properties of a specific license key.

License History

The License History page allows you to view the licenses utilized in the specified time frame.

Regional Web UI

**Step 1** From the Administration menu, choose License History under the User Access submenu.

**Step 2** The View License Utilization History page appears, specify the filter settings in the Set License History Filter attribute.
Click **Filter List** to view the license history for the specified time frame.

### CLI Command

Use `license showUtilHistory --full` view the number of utilized IP nodes against the RTUs (Right-to-Use) (see the `license` command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions).

### Centrally Managing Administrators

As a regional or local CCM administrator, you can:

- Create and modify local and regional cluster administrators, groups, and roles.
- Push administrators, groups, and roles to local clusters.
- Pull local cluster administrators, groups, and roles to the central cluster.

Each of these functions involves having at least one regional CCM administrator subrole defined. Table 5-3 describes the subroles required for these operations.

#### Table 5-3 Subroles Required for Central Administrator Management

<table>
<thead>
<tr>
<th>Central Administrator Management Action</th>
<th>Required Regional Subroles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create, modify, push, pull, or delete administrators</td>
<td>authentication</td>
</tr>
<tr>
<td>Create, modify, push, pull, or delete groups or roles</td>
<td>authorization</td>
</tr>
<tr>
<td>Create, modify, push, pull, or delete groups or roles with associated owners or regions</td>
<td>authorization, owner-region</td>
</tr>
<tr>
<td>Create, modify, push, pull, or delete external authentication servers</td>
<td>authentication</td>
</tr>
</tbody>
</table>

### Related Topics

- Pushing and Pulling Administrators, page 5-17
- Pushing and Pulling External Authentication Servers, page 5-20
- Pushing and Pulling Groups, page 5-22
- Pushing and Pulling Roles, page 5-24

### Pushing and Pulling Administrators

You can push administrators to, and pull administrators from local clusters on the List/Add Administrators page in the regional cluster web UI.

You can create administrators with both local and regional roles at the regional cluster. However, you can push or pull only associated local roles, because local clusters do not recognize regional roles.
Pushing Administrators to Local Clusters

Pushing administrators to local clusters involves choosing one or more clusters and a push mode.

Regional Basic and Advanced Web UI

Step 1 From the Administration menu, choose Administrators.
Step 2 On the List/Add Administrators Page, click the Push All icon in the Administrators pane to push all the administrators listed on the page. This opens the Push Data to Local Clusters dialog box.
Step 3 Choose a push mode by clicking one of the Data Synchronization Mode radio buttons. If you are pushing all the administrators, you can choose Ensure, Replace, or Exact. If you are pushing a single administrator, you can choose Ensure or Replace. In both cases, Ensure is the default mode. You would choose Replace only if you want to replace the existing administrator data at the local cluster. You would choose Exact only if you want to create an exact copy of the administrator database at the local cluster, thereby deleting all administrators that are not defined at the regional cluster.
Step 4 Choose one or more local clusters in the Available field of the Destination Clusters and move it or them to the Selected field.
Step 5 Click Push Data to Clusters.
Step 6 On the View Push Data Report dialog box, view the push details, then click OK to return to the List/Add Administrators page.

Pushing Administrators Automatically to Local Clusters

You can automatically push the new user name and password changes from the regional cluster to the local cluster. To do this, you must enable the synchronous edit mode in the regional cluster. The edit mode is set for the current Web UI session, or set as default for all users is set in the CCM Server configuration.

When synchronous mode is set, all the subsequent changes to user name and password are synchronized with local clusters. You can modify your password on the regional server, and this change is automatically propagated to local clusters.

If you are an admin user, you can make multiple changes to the user credentials on the regional cluster. All these changes are automatically pushed to local clusters.

Regional Basic and Advanced Web UI

Step 1 From the Operate menu, choose Manage Servers under Servers submenu to open the Manage Servers page.
Step 2 Click the Local CCM Server link on the Manage Servers pane to open the Edit CCM Server page.
Step 3 Choose the synchronous radio buttons for the regional edit mode values for admin, dhcp, and dns.
Step 4 Choose the webui mode value from the webui-mode drop-down list.
Step 5 Enter the idle-timeout value.
Step 6 To unset any attribute value, check the check box in the Unset? column, then click Unset Fields at the bottom of the page. To unset the attribute value or to change it, click Save, or Cancel to cancel the changes.

Note Enter values for the attributes marked with asterisks because they are required for CCM server operation. You can click the name of any attribute to open a description window for the attribute.

Connecting to CLI in Regional Mode
You must connect to the CLI in Regional Mode. The -R flag is required for regional mode. To set the synchronous edit mode:

nrcmd-R> session set admin-edit-mode=synchronous

Pulling Administrators from the Replica Database
Pulling administrators from the local clusters is mainly useful only in creating an initial list of administrators that can then be pushed to other local clusters. The local administrators are not effective at the regional cluster itself, because these administrators do not have regional roles assigned to them.

When you pull an administrator, you are actually pulling it from the regional cluster replica database. Creating the local cluster initially replicates the data, and periodic polling automatically updates the replication. However, to ensure that the replica data is absolutely current with the local cluster, you can force an update before pulling the data.

Regional Basic and Advanced Web UI

Step 1 From the Administration menu, choose Administrators under the User Access submenu.
Step 2 On the List/Add Administrators page, click Pull Replica on the Administrators pane. This opens the Select Replica Administrator Data to Pull dialog box.
Step 3 Click the Replicate icon in the Update Replica Data column for the cluster. (For the automatic replication interval, see the “Replicating Local Cluster Data” section on page 6-11.)
Step 4 Choose a replication mode using one of the Mode radio buttons. In most cases, you would leave the default Replace mode enabled, unless you want to preserve any existing administrator properties already defined at the regional cluster by choosing Ensure, or create an exact copy of the administrator database at the local cluster by choosing Exact (not recommended).
Step 5 Click Pull Core Administrators next to the cluster, or expand the cluster name and click Pull Administrator to pull an individual administrator in the cluster.
Step 6 On the Select Replica Admin Data to Pull dialog box, view the change set data, then click OK. You return to the List/Add Administrators page with the pulled administrators added to the list.
Note

If you do not have a regional cluster and would like to copy administrators, roles, or groups from one local cluster to another, you can export them and then reimport them at the target cluster by using the cnr_exim tool (see the “Using the cnr_exim Data Import and Export Tool” section on page 8-10). However, the tool does not preserve the administrator passwords, and you must manually reset them at the target cluster. It is implemented this way to maintain password security. The export command is:

```
cnr_exim -c admin -x -e outputfile.txt
```

Pushing and Pulling External Authentication Servers

You can push all external authentication servers to local cluster or pull the external authentication server data from the local cluster on the List/Add RADIUS Server page or List/Add Active Directory Server page in the regional web UI.

Pushing RADIUS External Authentication Servers

To push external authentication servers to the local cluster, do the following:

**Regional Advanced Web UI**

**Step 1**
From the Administration menu, choose RADIUS under the External Authentication submenu to view the List/Add RADIUS Server page in the regional web UI.

**Step 2**
Click **Push All** icon in the RADIUS pane to push all the external authentication servers listed on the page, or **Push** to push an individual external authentication server. This opens the Push Data to Local Clusters dialog box.

**Step 3**
Choose a push mode using one of the Data Synchronization Mode radio buttons.

- If you are pushing all the external authentication servers, you can choose Ensure, Replace, or Exact.
- If you are pushing a single external authentication server, you can choose Ensure or Replace.

In both the above cases, Ensure is the default mode.

Choose Replace only if you want to replace the existing external authentication server data at the local cluster. Choose Exact only if you want to create an exact copy of the external authentication server data at the local cluster, thereby deleting all external authentication servers that are not defined at the regional cluster.

**Step 4**
Click **Push Data to Clusters**.

Pulling RADIUS External Authentication Servers

To pull the external authentication server data from the local cluster, do the following:
Regional Advanced Web UI

**Step 1** From the Administration menu, choose **RADIUS** under the External Authentication submenu to view the List/Add RADIUS Server page in the regional web UI.

**Step 2** On the List/Add RADIUS Server page, click **Pull Data** on the RADIUS pane. This opens the Select Replica External Authentication Server Data to Pull dialog box.

**Step 3** Click the **Replica** icon in the Update Replica Data column for the cluster. (For the automatic replication interval, see the “Replicating Local Cluster Data” section on page 6-11.)

**Step 4** Choose a replication mode using one of the Mode radio buttons.

*Note* We do not recommend that you create an exact copy of the external authentication server data at the local cluster by choosing **Exact**.

**Step 5** Click **Pull All External Authentication Servers** next to the cluster.

**Step 6** On the Report Pull Replica Authentication servers page, view the pull details, then click **Run**.

On the Run Pull Replica Authentication servers page, view the change set data, then click **OK**. You return to the List/Add Authentication Server page with the pulled external authentication servers added to the list.

---

Pushing AD External Authentication Servers

To push external authentication servers to the local cluster, do the following:

**Regional Advanced Web UI**

**Step 1** From the Administration menu, choose **Active Directory** under the External Authentication submenu to view the List/Add Active Directory Server page in the regional web UI.

**Step 2** Click **Push** to push the external authentication server. This opens the Push Data to Local Clusters dialog box.

**Step 3** Choose a push mode using one of the Data Synchronization Mode radio buttons.

- If you are pushing all the external authentication servers, you can choose **Ensure**, **Replace**, or **Exact**.
- If you are pushing a single external authentication server, you can choose **Ensure** or **Replace**.

  In both the above cases, **Ensure** is the default mode.

  Choose **Replace** only if you want to replace the existing external authentication server data at the local cluster. Choose **Exact** only if you want to create an exact copy of the external authentication server data at the local cluster, thereby deleting all external authentication servers that are not defined at the regional cluster.

**Step 4** Click **Push Data to Clusters**.
Pulling AD Authentication Servers

To pull the external authentication server data from the local cluster, do the following:

Regional Advanced Web UI

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>From the Administration menu, choose Active Directory under the External Authentication submenu to view the List/Add Active Directory Server page in the regional web UI.</td>
</tr>
<tr>
<td>2</td>
<td>On the List/Add Active Directory Server page, click Pull Data on the Active Directory pane. This opens the Select Replica External Authentication Server Data to Pull dialog box.</td>
</tr>
<tr>
<td>3</td>
<td>Click the Replica icon in the Update Replica Data column for the cluster. (For the automatic replication interval, see the “Replicating Local Cluster Data” section on page 6-11.)</td>
</tr>
<tr>
<td>4</td>
<td>Choose a replication mode using one of the Mode radio buttons. Leave the default Replace mode enabled, unless you want to preserve any existing external authentication server properties at the local cluster by choosing Ensure.</td>
</tr>
</tbody>
</table>

Note: We do not recommend that you create an exact copy of the external authentication server data at the local cluster by choosing Exact.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Click Pull All External Authentication Servers next to the cluster.</td>
</tr>
<tr>
<td>6</td>
<td>On the Report Pull Replica Authentication servers page, view the pull details, then click Run. On the Run Pull Replica Authentication servers page, view the change set data, then click OK. You return to the List/Add Authentication Server page with the pulled external authentication servers added to the list.</td>
</tr>
</tbody>
</table>

Pushing and Pulling Groups

Pushing and pulling groups is vital in associating administrators with a consistent set of roles at the local clusters. You can push groups to, and pull groups from, local clusters on the List/Add Administrator Groups page in the regional cluster web UI.

Related Topics

- Pushing Groups to Local Clusters, page 5-22
- Pulling Groups from the Replica Database, page 5-23

Pushing Groups to Local Clusters

Pushing groups to local clusters involves choosing one or more clusters and a push mode.

Regional Basic and Advanced Web UI

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>From the Administration menu, choose Groups under the User Access submenu.</td>
</tr>
</tbody>
</table>
Centrally Managing Administrators

Step 2  On the List/Add Administrator Groups page, click the **Push All** icon on Groups pane to push all the groups listed on the page, or **Push** to push an individual group. This opens the Push Data to Local Clusters dialog box.

Step 3  Choose a push mode using one of the Data Synchronization Mode radio buttons. If you are pushing all the groups, you can choose Ensure, Replace, or Exact. If you are pushing a single group, you can choose Ensure or Replace. In both cases, Ensure is the default mode. You would choose Replace only if you want to replace the existing group data at the local cluster. You would choose Exact only if you want to create an exact copy of the group data at the local cluster, thereby deleting all groups that are not defined at the regional cluster.

Step 4  By default, the associated roles and owners are pushed along with the group. Roles are pushed in Replace mode and owners in Ensure mode. To disable pushing the associated roles or owners, uncheck the respective check box.

Step 5  Choose one or more local clusters in the Available field of the Destination Clusters and move it or them to the Selected field.

Step 6  Click **Push Data to Clusters**.

Step 7  On the View Push Group Data Report page, view the push details, then click **OK** to return to the List/Add Administrator Groups page.

---

### Pulling Groups from the Replica Database

Pulling administrator groups from the local clusters is mainly useful only in creating an initial list of groups that can then be pushed to other local clusters. The local groups are not useful at the regional cluster itself, because these groups do not have regional roles assigned to them.

When you pull a group, you are actually pulling it from the regional cluster replica database. Creating the local cluster initially replicates the data, and periodic polling automatically updates the replication. However, to ensure that the replica data is absolutely current with the local cluster, you can force an update before pulling the data.

### Regional Basic and Advanced Web UI

**Step 1**  From the **Administration** menu, choose **Groups** under the User Access submenu.

**Step 2**  On the List/Add Administrator Groups page, click the **Pull Replica** icon on Groups pane. This opens the Select Replica CCMAdminGroup Data to Pull dialog box.

**Step 3**  Click the **Replica** icon in the Update Replica Data column for the cluster. (For the automatic replication interval, see the “Replicating Local Cluster Data” section on page 6-11.)

**Step 4**  Choose a replication mode using one of the Mode radio buttons. In most cases, you would leave the default Replace mode enabled, unless you want to preserve any existing group properties at the local cluster by choosing Ensure, or create an exact copy of the group data at the local cluster by choosing Exact (not recommended).

**Step 5**  Click **Pull Core Groups** next to the cluster, or expand the cluster name and click **Pull Group** to pull an individual group in the cluster.

**Step 6**  On the Report Pull Replica Groups page, view the pull details, then click **Run**.

**Step 7**  On the Run Pull Replica Groups page, view the change set data, then click **OK**. You return to the List/Add Administrator Groups page with the pulled groups added to the list.
Pushing and Pulling Roles

You can push roles to, and pull roles from, local clusters on the List/Add Administrator Roles page in the regional cluster web UI. You can also push associated groups and owners, and pull associated owners, depending on your subrole permissions (see Table 5-3 on page 5-17).

Related Topics

Pushing Roles to Local Clusters, page 5-24
Pulling Roles from the Replica Database, page 5-24

Pushing Roles to Local Clusters

Pushing administrator roles to local clusters involves choosing one or more clusters and a push mode.

Regional Advanced Web UI

Step 1  From the Administration menu, choose Roles under the User Access submenu.

Step 2  On the List/Add Administrator Roles page, click the Push All icon in the Roles pane to push all the roles listed on the page, or Push to push an individual role. This opens the Push Data to Local Clusters dialog box.

Step 3  Choose a push mode using one of the Data Synchronization Mode radio buttons. If you are pushing all the roles, you can choose Ensure, Replace, or Exact. If you are pushing a single role, you can choose Ensure or Replace. In both cases, Ensure is the default mode. You would choose Replace only if you want to replace the existing role data at the local cluster. You would choose Exact only if you want to create an exact copy of the role data at the local cluster, thereby deleting all roles that are not defined at the regional cluster.

Step 4  By default, the associated groups and owners are pushed along with the role. Groups are pushed in Replace mode and owners in Ensure mode. To disable pushing the associated roles or owners, uncheck the respective check box:

- If you disable pushing associated groups and the group does not exist at the local cluster, a group based on the name of the role is created at the local cluster.
- If you disable pushing associated owners and the owner does not exist at the local cluster, the role will not be configured with its intended constraints. You must separately push the group to the local cluster, or ensure that the regional administrator assigned the owner-region subrole has pushed the group before pushing the role.

Step 5  Choose one or more local clusters in the Available field of the Destination Clusters and move it or them to the Selected field.

Step 6  Click Push Data to Clusters.

Step 7  On the View Push Role Data Report page, view the push details, then click OK to return to the List/Add Administrator Roles page.

Pulling Roles from the Replica Database

Pulling administrator roles from the local clusters is mainly useful only in creating an initial list of roles that can then be pushed to other local clusters. The local roles are not useful at the regional cluster itself.
When you pull a role, you are actually pulling it from the regional cluster replica database. Creating the local cluster initially replicates the data, and periodic polling automatically updates the replication. However, to ensure that the replica data is absolutely current with the local cluster, you can force an update before pulling the data.

**Regional Advanced Web UI**

**Step 1**
From the **Administration** menu, choose **Roles** under the User Access submenu.

**Step 2**
On the List/Add Administrator Roles page, click the **Pull Replica** icon in the **Roles** pane. This opens the Select Replica Administrator Role Data to Pull dialog box.

**Step 3**
Click the **Replicate** icon in the Update Replica Data column for the cluster. (For the automatic replication interval, see the “Replicating Local Cluster Data” section on page 6-11.)

**Step 4**
Choose a replication mode using one of the Mode radio buttons. In most cases, you would leave the default Replace mode enabled, unless you want to preserve any existing role properties at the local cluster by choosing Ensure, or create an exact copy of the role data at the local cluster by choosing Exact (not recommended).

**Step 5**
If you have the owner-region subrole permission, you can decide if you want to pull all the associated owners with the role, which is always in Ensure mode. This choice is enabled by default.

**Step 6**
Click **Pull Core Roles** next to the cluster, or expand the cluster name and click **Pull Role** to pull an individual role in the cluster.

**Step 7**
On the Report Pull Replica Roles page, view the pull details, then click **Run**.

**Step 8**
On the Run Pull Replica Roles page, view the change set data, then click **OK**. You return to the List/Add Administrator Roles page with the pulled roles added to the list.

---

**Local Cluster Management Tutorial**

This tutorial describes a basic scenario on a local cluster of the Example Company. Administrators at the cluster are responsible for users, zone data, DHCP data, address space data, and the servers in general. The task is to set up two zones (example.com and boston.example.com), hosts in the zones, and a subnet. The local cluster must also create a special administrator account so that the regional cluster in San Jose can perform the central configuration and replicate the local cluster administrators and address space at another cluster, as described in the “Regional Cluster Management Tutorial” section on page 5-32.

**Related Topics**

- Administrator Responsibilities and Tasks, page 5-33
- Create the Administrators, page 5-26
- Create the Address Infrastructure, page 5-27
- Create the Zone Infrastructure, page 5-28
- Create a Host Administrator Role with Constraints, page 5-30
- Create a Group to Assign to the Host Administrator, page 5-31
- Test the Host Address Range, page 5-32
Administrator Responsibilities and Tasks

The local cluster administrators have the following responsibilities and tasks:

- **example-cluster-admin**—Created by the superuser:
  - At the Boston cluster, creates the other local administrators (example-zone-admin and example-host-admin).
  - Creates the basic network infrastructure for the local clusters.
  - Constrains the example-host-role to an address range in the boston.example.com zone.
  - Creates the example-host-group (defined with the example-host-role) that the example-zone-admin will assign to the example-host-admin.

- **example-zone-admin**:
  - Creates the example.com and boston.example.com zones, and maintains the latter zone.
  - Assigns the example-host-group to the example-host-admin.

- **example-host-admin**—Maintains local host lists and IP address assignments.

Create the Administrators

For this example, the superuser in Boston creates the local cluster, zone, and host administrators, as described in the “Administrator Responsibilities and Tasks” section on page 5-26.

Local Basic Web UI

- **Step 1** At the Boston local cluster, log in as superuser (usually admin).
- **Step 2** In Basic mode, from the Administration menu, choose Administrators.
- **Step 3** Add the local cluster administrator (with superuser access)—On the List/Add Administrators page:
  - Click the Add Administrators icon in the Administrators pane, enter example-cluster-admin in the Name field.
  - Enter exampleadmin in the Password field, then click Add Admin.
  - Check the Superuser check box.
  - Do not choose a group from the Groups list
  - Click Save.
- **Step 4** Add the local zone administrator on the same page:
  - Click the Add Administrators icon in the Administrators pane, enter example-zone-admin in the Name field, and examplezone in the Password field, then click Add Admin.
  - Multiselect ccm-admin-group, dns-admin-group, and host-admin-group in the Groups drop-down list. The dns-admin-group is already predefined with the dns-admin role to administer DNS zones and servers. The ccm-admin-group guarantees that the example-zone-admin can set up the example-host-admin with a constrained role later on. The host-admin-group is mainly to test host creation in the zone.
  - Click Save.
Add the local host administrator on the same page:

a. Click the Add Administrators icon in the Administrators pane, enter example-host-admin in the Name field, and examplehost in the Password field, then click Add Admin.

b. Do not choose a group at this point. (The example-zone-admin will later assign example-host-admin to a group with a constrained role.)

c. Click Save.

For a description on how to apply constraints to the administrator, see the “Create a Host Administrator Role with Constraints” section on page 5-30.

Create the Address Infrastructure

A prerequisite to managing the zones and hosts at the clusters is to create the underlying network infrastructure. The network configuration often already exists and was imported. However, this tutorial assumes that you are starting with a clean slate.

The local example-cluster-admin next creates the allowable address ranges for the hosts in the boston.example.com zone that will be assigned static IP addresses. These addresses are in the 192.168.50.0/24 subnet with a range of hosts from 100 through 200.

Local Advanced Web UI

Step 1 At the local cluster, log out as superuser, then log in as the example-cluster-admin user with password exampleadmin. Because the administrator is a superuser, all features are available.

Step 2 Click Advanced to go to Advanced mode.

Step 3 Click Design, then Subnets under DHCPv4 submenu.

Step 4 On the List/Add Subnets page, enter the boston.example.com subnet address:

a. Click the Add Subnets icon in the Subnets pane, enter 192.168.50 in the Address field.

b. Choose 24 in the mask drop-down list—This subnet will be a normal Class C network.

c. Leave the Owner, Region, and Address Type fields as is. Add description if desired.

d. Click Add Subnet.

Step 5 Click the 192.168.50.0/24 address to open the Edit Subnet page.

Step 6 In the IP Ranges fields, enter the static address range:

a. Enter 100 in the Start field. Tab to the next field.

b. Enter 200 in the End field.

c. Click Add IP Range. The address range appears under the fields.

Step 7 Click Save.

Step 8 Click Address Space to open the View Unified Address Space page. The 192.168.50.0/24 subnet should appear in the list. If not, click the Refresh icon.
Create the Zone Infrastructure

For this scenario, example-cluster-admin must create the Example Company zones locally, including the example.com zone and its subzones. The example-cluster-admin also adds some initial host records to the boston.example.com zone.

Related Topics

Create the Forward Zones, page 5-28
Create the Reverse Zones, page 5-28
Create the Initial Hosts, page 5-29

Create the Forward Zones

First, create the example.com and boston.example.com forward zones.

Local Basic Web UI

Step 1 At the local cluster, log in as the example-zone-admin user with password examplezone.
Step 2 From the Design menu, choose Forward Zones under the Auth DNS submenu. This opens the List/Add Forward Zones page.
Step 3 Create the example.com zone (tab from field to field):
   a. Click the Add Forward Zone icon in the Forward Zones pane, enter example.com in the Name field.
   b. In the Nameserver FQDN field, enter ns1.
   c. In the Contact E-Mail field, enter hostmaster.
   d. In the Serial Number field, enter the serial number.
   e. Click Add Zone.
Step 4 Create the boston.example.com zone in the same way, using the same values as in the previous steps:
   a. Creating a zone with a prefix added to an existing zone opens the Create Subzone in Parent Zone page, because the zone can be a potential subzone. Because you do want to create this zone as a subzone to example.com, click Create as Subzone on the Create Subzone in Parent Zone page.
   b. Because nameservers are different in each zone, you must create a glue Address (A) record to tie the zones together. Enter 192.168.50.1 in the A record field, then click Specify Glue Records. Then click Report, Run, and Return.
   c. The List/Add Zones page should now list example.com and boston.example.com.
Step 5 Click Advanced, then Show Forward Zone Tree to show the hierarchy of the zones. Return to list mode by clicking Show Forward Zone List.

Create the Reverse Zones

Next, create the reverse zones for example.com and boston.example.com. This way you can add reverse address pointer (PTR) records for each added host. The reverse zone for example.com is based on the 192.168.50.0 subnet; the reverse zone for boston.example.com is based on the 192.168.60.0 subnet.
Local Basic Web UI

Step 1  At the local cluster, you should be logged in as the example-zone-admin user, as in the previous section.

Step 2  From the Design menu, choose Reverse Zones under the Auth DNS submenu.

Step 3  On the List/Add Reverse Zones page, click the Add Reverse Zone icon in the Reverse Zones pane, enter 50.168.192.in-addr.arpa in the Name field. (There is already a reverse zone for the loopback address, 127.in-addr.arpa.)

Step 4  Enter the required fields to create the reverse zone, using the forward zone values:
   a. Nameserver—Enter ns1.example.com. (be sure to include the trailing dot).
   b. Contact E-Mail—Enter hostmaster.example.com. (be sure to include the trailing dot).
   c. Serial Number—Enter the serial number.

Step 5  Click Add Reverse Zone to add the zone and return to the List/Add Reverse Zones page.

Step 6  Do the same for the boston.example.com zone, using 60.168.192.in-addr.arpa as the zone name and the same nameserver and contact e-mail values as in Step 4. (You can cut and paste the values from the table.)

Create the Initial Hosts

As a confirmation that hosts can be created at the Boston cluster, the example-zone-admin tries to create two hosts in the example.com zone.

Local Advanced Web UI

Step 1  As the example-zone-admin user, click Advanced to enter Advanced mode.

Step 2  From the Design menu, choose Hosts under the Auth DNS submenu. This opens the List/Add Hosts for Zone page. You should see boston.example.com and example.com in the Select Zones box on the left side of the window.

Step 3  Click example.com in the list of zones.

Step 4  Add the first static host with address 192.168.50.101:
   a. Enter userhost101 in the Name field.
   b. Enter the complete address 192.168.50.101 in the IP Address(es) field. Leave the IPv6 Address(es) and Alias(es) field blank.
   c. Ensure that the Create PTR Records? check box is checked.
   d. Click Add Host.

Step 5  Add the second host, userhost102, with address 192.168.50.102, in the same way. The two hosts should now appear along with the nameserver host on the List/Add Hosts for Zone page.
Create a Host Administrator Role with Constraints

In this part of the tutorial, the Boston example-cluster-admin creates the example-host-role with address constraints in the boston.example.com zone.

Local Advanced Web UI

Step 1 Log out as the example-zone-admin user and log in as the example-cluster-admin user (with password exampleadmin).

Step 2 Click Advanced to enter Advanced mode.

Step 3 From the Administration menu, choose Roles under User Access submenu to open the List/Add Administrator Roles page.

Step 4 Add the example-host-role:
   a. Click the Add Role icon in the Roles pan to open the Add Roles dialog box.
   b. Enter example-host-role in the Name field.
   c. Click Add Role. The example-host-role should now appear in the list of roles on the List/Add Administrator Roles page.

Step 5 Add the constraint for the role:
   a. Click Add Constraint.
   b. On the Add Role Constraint for Role page, scroll down to Host Restrictions.
   c. For the all-forward-zones attribute, click the false radio button.
   d. For the zones attribute, enter boston.example.com.
   e. For the ipranges attribute, enter the range 192.168.50.101–192.168.50.200.
   f. The zone-regexpr and host-regexpr attribute fields are for entering regular expressions to match zones and hosts, respectively, in regex syntax. (See Table 5-4 for the commonly used regex values.)

Table 5-4 Common Regex Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>. (dot)</td>
<td>Any character (a wildcard). Note that to match a literal dot character (such as in a domain name), you must escape it by using a backslash (), such that .com matches .com.</td>
</tr>
<tr>
<td>\char</td>
<td>Literal character (char) that follows, or the char has special meaning. Used especially to escape metacharacters such as the dot (.) or another backslash. Special meanings include \d to match decimal digits, \D for nondigits, \w for alphanumerics, and \s for whitespace.</td>
</tr>
<tr>
<td>char?</td>
<td>Preceding char once or not at all, as if the character were optional. For example, example.?com matches example.com or examplecom.</td>
</tr>
<tr>
<td>char*</td>
<td>Preceding char zero or more times. For example, ca*t matches ct, cat, and caaat. This repetition metacharacter does iterative processing with character sets (see [charset]).</td>
</tr>
<tr>
<td>char+</td>
<td>Preceding char one or more times. For example, ca+t matches cat and caaat (but not ct).</td>
</tr>
</tbody>
</table>
g. Click Add Constraint. The constraint should have an index number of 1.

Step 6  Click Save.

Create a Group to Assign to the Host Administrator

The Boston example-cluster-admin next creates an example-host-group that includes the example-host-role so that the example-zone-admin can assign this group to the example-host-admin.

Local Advanced Web UI

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>As example-cluster-admin, still in Advanced mode, from the Administration menu, choose Groups submenu to open the List/Add Administrator Groups page.</td>
</tr>
</tbody>
</table>
| Step 2 | Create the example-host-group and assign the example-host-role to it:  
  a. Click the Add Groups icon in the Groups pane, enter example-host-group in the Name field.  
  b. From the Base Role drop-down list, choose example-host-role.  
  c. Click Add Group.  
  d. Add a description such as Group for the example-host-role, then click Save. |
| Step 3 | Log out as example-cluster-admin, then log in as the example-zone-admin user (with password examplezone). |
| Step 4 | As example-zone-admin, assign the example-host-group to the example-host-admin:  
  a. In Basic mode, from the Administration menu, choose Administrators.  
  b. On the List/Add Administrators page, click example-host-admin to edit the administrator.  
  c. On the Edit Administrator page, choose example-host-group in the Available list, then click << to move it to the Selected list.  
  d. Click Save. The example-host-admin should now show the example-host-group in the Groups column on the List/Add Administrators page. |
Test the Host Address Range

The example-host-admin next tests an out-of-range address and then adds an acceptable one.

Local Advanced Web UI

Step 1
At the local cluster, log out as example-zone-admin, then log in as example-host-admin (with password examplehost).

Step 2
Click Advanced to enter Advanced mode.

Step 3
From the Design menu, choose Hosts from the Auth DNS submenu.

Step 4
On the List/Add Hosts for Zone page, try to enter an out-of-range address (note the range of valid addresses in the Valid IP Ranges field):
   a. Enter userhost3 in the Name field.
   b. Deliberately enter an out-of-range address (192.168.50.3) in the IP Address(es) field.
   c. Click Add Host. You should get an error message.

Step 5
Enter a valid address:
   a. Enter userhost103.
   b. Enter 192.168.50.103 in the IP Address(es) field.
   c. Click Add Host. The host should now appear with that address in the list.

Regional Cluster Management Tutorial

This tutorial is an extension of the scenario described in the “Local Cluster Management Tutorial” section on page 5-25. In the regional cluster tutorial, San Jose has two administrators—a regional cluster administrator and a central configuration administrator. Their goal is to coordinate activities with the local clusters in Boston and Chicago so as to create DNS zone distributions, router configurations, and DHCP failover configurations using the servers at these clusters. The configuration consists of:

- One regional cluster machine in San Jose.
- Two local cluster machines, one in Boston and one in Chicago.
- One Cisco uBR7200 router in Chicago.

Related Topics

Administrator Responsibilities and Tasks, page 5-33
Create the Regional Cluster Administrator, page 5-33
Create the Central Configuration Administrator, page 5-33
Create the Local Clusters, page 5-34
Add Zone Management to the Configuration Administrator, page 5-35
Create a Zone for the Local Cluster, page 5-35
Pull Zone Data and Create a Zone Distribution, page 5-36
Create a Subnet and Pull Address Space, page 5-36
Administrator Responsibilities and Tasks

The regional administrators have the following responsibilities and tasks:

- **example-regional-admin**—Created by the superuser at the San Jose regional cluster, who creates the example-cfg-admin.
- **example-cfg-admin**:
  - Defines the Boston and Chicago clusters and checks connectivity with them.
  - Adds a router and modifies a router interface.
  - Pulls zone data from the local clusters to create a zone distribution.
  - Creates a subnet and policy, and pulls address space, to configure DHCP failover pairs in Boston and Chicago.

Create the Regional Cluster Administrator

The regional superuser first creates the example-regional-administrator, defined with groups, to perform cluster and user administration.

**Regional Web UI**

**Step 1** Log into the regional cluster as superuser.

**Step 2** From the Administration menu, choose Administrators to open the List/Add Administrators page for the local cluster version of this page, which is essentially identical.

**Step 3** Click the Add Administrators icon in the Administrators pane, enter example-regional-admin in the Name field, then examplereg in the Password field in the Add Administrator dialog box, then click Add Administrator.

**Step 4** Multiselect central-cfg-admin-group (for cluster administration) and regional-admin-group (for user administration) in the Groups drop-down list.

**Step 5** Click Save.

Create the Central Configuration Administrator

As part of this tutorial, the example-regional-admin next logs in to create the example-cfg-admin, who must have regional configuration and address management capabilities.

**Regional Web UI**

**Step 1** Log out as superuser, then log in as example-regional-admin with password examplereg. Note that the administrator has all but host and address space administration privileges.
Step 2  From the Administration menu, choose Administrators to open the List/Add Administrators page.
Step 3  Click the Add Administrators icon in the Administrators pane, enter example-cfg-admin in the Name field, then cfgadmin in the Password field in the Add Administrator dialog box, then click Add Administrator.
Step 4  Multiselect central-cfg-admin-group and regional-addr-admin-group in the Groups drop-down list.
Step 5  Click Save. The example-cfg-admin now appears with the two groups assigned.

You can also add constraints for the administrator. Click Add Constraint and, on the Add Role Constraint for Role page, choose the read-only, owner, or region constraints, then click Add Constraint.

Create the Local Clusters

The example-cfg-admin next creates the two local clusters for Boston and Chicago.

Regional Web UI

Step 1  Log out as example-regional-admin, then log in as example-cfg-admin with password cfg admin.
Step 2  From the Operate menu, choose Servers from the Manage Clusters submenu to open the List/Add Remote Clusters page.
Step 3  Click the Add Manage Clusters icon in the Manage Clusters pane.
Step 4  On the Add Cluster dialog box, create the Boston cluster based on data provided by its administrator:
   a. Enter Boston-cluster in the name field.
   b. Enter the IP address of the Boston server in the ipaddr field.
   c. Enter example-cluster-admin in the admin field, then exampleadmin in the password field.
   d. Enter in the SCPO-port field the SCP port to access the cluster as set at installation (1234 is the preset value).
   e. Click Add Cluster.
Step 5  Create the Chicago cluster in the same way, except use Chicago-cluster in the name field, enter the remaining values based on data provided by the Chicago administrator, then click Add Cluster. The two clusters should now appear on the List/Add Remote Clusters page.
Step 6  Connect to the Boston cluster. Click the Go Local icon next to Boston-cluster. If this opens the local cluster Manage Servers page, this confirms the administrator connectivity to the cluster. To return to the regional cluster web UI, click the Go Regional icon.
Step 7  Connect to the Chicago cluster to confirm the connectivity in the same way.
Step 8  Confirm that you can replicate data for the two forward zones from the Boston cluster synchronization:
   a. From the Operate menu, choose Replica Data from the Servers submenu.
   b. On the View Replica Class List page, click Boston-cluster in the Select Cluster list.
   c. In the Select Class list, click Forward Zones.
   d. Click the Replicate icon in the Replicate Data column.
Add Zone Management to the Configuration Administrator

Because there are no zones set up at the Chicago cluster, the example-cfg-admin can create a zone at the regional cluster to make it part of the zone distribution. However, the example-regional-admin must first modify the example-cfg-admin to be able to create zones.

Regional Web UI

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Log out as example-cfg-admin, then log in as <strong>example-regional-admin</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>From the <strong>Administration</strong> menu, choose <strong>Administrators</strong>.</td>
</tr>
<tr>
<td>Step 3</td>
<td>On the List/Add Administrators page, click example-cfg-admin from the Administrators pane.</td>
</tr>
<tr>
<td>Step 4</td>
<td>On the Edit Administrator page, click central-dns-admin-group in the Groups Available list, then move it (using &lt;&lt;) to the Selected list. The Selected list should now have central-cfg-admin-group, regional-addr-admin-group, and central-dns-admin-group.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Click <strong>Save</strong>. The change should be reflected on the List/Add Administrators page.</td>
</tr>
</tbody>
</table>

Create a Zone for the Local Cluster

The example-cfg-admin next creates the chicago.example.com zone for the zone distribution with the Boston and Chicago zones.

Regional Web UI

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Log out as example-regional-admin, then log in as <strong>example-cfg-admin</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>From the <strong>Design</strong> menu, choose <strong>Forward Zones</strong> under the <strong>Auth DNS</strong> submenu.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Click the <strong>Add Forward Zones</strong> icon in the <strong>Forward Zones</strong> pane.</td>
</tr>
<tr>
<td>Step 4</td>
<td>On the Add Zone dialog box, enter:</td>
</tr>
<tr>
<td></td>
<td>a. <strong>Name</strong>—chicago.example.com.</td>
</tr>
<tr>
<td></td>
<td>b. <strong>Nameserver FQDN</strong>—ns1.</td>
</tr>
<tr>
<td></td>
<td>c. <strong>Contact E-mail</strong>—hostmaster.</td>
</tr>
<tr>
<td></td>
<td>d. <strong>Nameservers</strong>—ns1 (click <strong>Add Nameserver</strong>).</td>
</tr>
<tr>
<td></td>
<td>e. Click <strong>Add Zone</strong>.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Click the <strong>Reverse Zones</strong> submenu.</td>
</tr>
<tr>
<td>Step 6</td>
<td>On the List Reverse Zones page, create the <strong>60.168.192.in-addr.arpa</strong> reverse zone for the Chicago zone, with the proper attributes set.</td>
</tr>
</tbody>
</table>

e. Click **View Replica Class List**. On the List Replica Forward Zones for Cluster page, you should see the boston.example.com and example.com zones.
Pull Zone Data and Create a Zone Distribution

The example-cfg-admin next pulls zone data from Boston and Chicago and creates a zone distribution.

Regional Web UI

Step 1  As example-cfg-admin, from the Design menu, choose Views under the Auth DNS submenu to view the List/Add Zone Views page.
Step 2  On the List/Add Zone Views page, pull the zone from the replica database:
   a.  Click the Pull Replica icon in the Views pane.
   b.  On the Select Replica Downsize Data to Pull dialog box, leave the Data Synchronization Mode defaulted as Update, then click Report to open the Report Pull Replica Zone Data page.
   c.  Notice the change sets of data to pull, then click Run.
   d.  On the Run Pull Replica Zone Data page, click OK.
Step 3  On the List/Add Zone Views page, notice that the Boston cluster zone distribution is assigned an index number (1) in the Name column. Click the number.
Step 4  On the Edit Zone Views page, in the Primary Server field, click Boston-cluster. (The IP address of the Boston-cluster becomes the first master server in the Master Servers list.)
Step 5  Because we want to make the Chicago-cluster DNS server a secondary server for the Boston-cluster:
   a.  Click Add Server in the Secondary Servers area.
   c.  Click Add Secondary Server.
Step 6  On the Edit Zone Distribution page, in the Forward Zones area, move chicago.example.com to the Selected list.
Step 7  In the Reverse Zones area, move 60.168.192.in-addr.arpa to the Selected list.
Step 8  Click Modify Zone Distribution.

Create a Subnet and Pull Address Space

The example-cfg-admin next creates a subnet at the regional cluster. This subnet will be combined with the other two pulled subnets from the local clusters to create a DHCP failover server configuration.

Regional Web UI

Step 1  As example-cfg-admin, from the Design menu, choose Subnets under the DHCPv4 submenu to open the List/Add Subnets page.
Step 2  Create an additional subnet, 192.168.70.0/24 by clicking the Add Subnets icon in the Subnets pane:
   a.  Enter 192.168.70 (the abbreviated form) as the subnet network address in the Address/Mask field.
   b.  Leave the 24 (255.255.255.0) selected as the network mask.
   c.  Click Add Subnet.
Step 3  Click **Address Space** to confirm the subnet you created.

Step 4  On the View Unified Address Space page, click **Pull Replica Address Space**.

Step 5  On the Select Pull Replica Address Space page, leave everything defaulted, then click **Report**.

Step 6  The Report Pull Replica Address Space page should show the change sets for the two subnets from the clusters. Click **Run**.

Step 7  Click **OK**. The two pulled subnets appear on the List/Add Subnets page.

---

**Push a DHCP Policy**

The example-cfg-admin next creates a DHCP policy, then pushes it to the local clusters.

**Regional Web UI**

Step 1  As example-cfg-admin, from the **Design** menu, choose **Policies** under the **DHCP Settings** submenu.

Step 2  On the List/Add DHCP Policies page, click the **Add Policies** icon in the **Policies** pane.

Step 3  On the Add DHCP Policy dialog box, create a central policy for all the local clusters:

a. Enter **central-policy-1** in the Name field. Leave the Offer Timeout and Grace Period values as is.

b. Enter a lease period. In the DHCP > DHCPv4 > Options drop-down list, choose **dhcp-lease-time [51] (unsigned time)**, then enter 2w (two weeks) for the lease period in the Value field.

c. Click **Add Option**.

d. Click **Add Policy**. The central-policy-1 should appear on the List/Add DHCP Policies page.

Step 4  Push the policy to the local clusters:

a. Select the policy, central-policy-1 and click **Push** button.

b. On the Push DHCP Policy Data to Local Clusters page, leave the Data Synchronization Mode as **Ensure**. This ensures that the policy is replicated at the local cluster, but does not replace its attributes if a policy by that name already exists.

c. Click **Select All** in the Destination Clusters section of the page.

d. Click **<<** to move both clusters to the Selected field.

e. Click **Push Data to Clusters**.

f. View the push operation results on the View Push DHCP Policy Data Report page, then click **OK**.

---

**Create a Scope Template**

The example-cfg-admin next creates a DHCP scope template to handle failover server pair creation.
Regional Web UI

Step 1  As the example-cfg-admin user, from the Design menu, choose Scope Templates under the DHCPv4 submenu.

Step 2  On the List/Add DHCP Scopes page, click the Add Scopes icon in the Scope Templates pane. Enter scope-template-1 in the Name field, then click Add Scope Templates.

Step 3  The template should appear on the List/Add DHCP Scopes page. Set the basic properties for the scope template—Enter or choose the following values in the fields:
   a. Scope Name Expression—To autogenerate names for the derivative scopes, concatenate the example-scope string with the subnet defined for the scope. To do this, enter (concat "example-scope-" subnet) in the field (including the parentheses).
   b. Policy—Choose central-policy-1 in the drop-down list.
   c. Range Expression—Create an address range based on the remainder of the subnet (the second through last address) by entering (create-range 2 100).
   d. Embedded Policy Option Expression—Define the router for the scope in its embedded policy and assign it the first address in the subnet by entering (create-option "routers" (create-ipaddr subnet 1)).

Step 4  Click Save.

Create and Synchronize the Failover Pair

The example-cfg-admin next creates the failover server pair relationship and synchronizes the failover pair. The DHCP server at Boston becomes the main, and the server at Chicago becomes the backup.

Regional Web UI

Step 1  As the example-cfg-admin user, from the Deploy menu, choose Failover under the DHCP submenu.

Step 2  On the List/Add DHCP Failover Pairs page, click the Add Failover Pair icon in the Failover Pairs pane.

Step 3  On the Add DHCP Failover Pair dialog box, enter or choose the following values:
   a. Failover Pair Name—Enter central-fo-pair.
   b. Main Server—Click Boston-cluster.
   c. Backup Server—Click Chicago-cluster.
   d. Scope Template—Click scopetemplate-1.
   e. Click Add Failover Pair.

Step 4  Synchronize the failover pair with the local clusters:
   a. On the List/Add DHCP Failover Pairs page, click the Report icon in the Synchronize column.
   b. On the Report Synchronize Failover Pair page, accept Local Server as the source of network data.
   c. Accept Main to Backup as the direction of synchronization.
   d. Accept the operation Update.
   e. Click Report at the bottom of the page.
g. Click Return.

**Step 5**  
Confirm the failover configuration and reload the server at the Boston cluster:

a. On the List/Add DHCP Failover Pairs page, click the Go Local icon next to Boston-cluster.

b. On the Manage DHCP Server page, click the Reload icon.

c. Click the Go Regional icon at the top of the page to return to the regional cluster.

**Step 6**  
Confirm the failover configuration and reload the server at the Chicago cluster in the same way.
Managing the Central Configuration

This chapter explains how to manage the central configuration at the Cisco Prime IP Express regional cluster.

Related Topics
- Central Configuration Tasks, page 6-1
- Configuring Server Clusters, page 6-2
- Managing DHCP Scope Templates, page 6-15
- Managing DHCP Policies, page 6-16
- Managing DHCP Client-Classes, page 6-17
- Managing Virtual Private Networks, page 6-19
- Managing DHCP Failover Pairs, page 6-20
- Managing Lease Reservations, page 6-21

Central Configuration Tasks

Central configuration management at the regional cluster can involve:

- Setting up server clusters, replicating their data, and polling lease history data from them.
- Setting up routers.
- Managing network objects such as DHCP scope templates, policies, client-classes, options, networks, and virtual private networks (VPNs).
- Managing DHCP failover server pairs.

These functions are available only to administrators assigned the central-cfg-admin role. (The full list of functions for the central-cfg-admin are listed in Table 5-2 on page 5-4.) Note that central configuration management does not involve setting up administrators and checking the status of the regional servers. These functions are performed by the regional administrator, as described in the “Licensing” section on page 5-15 and “Managing Servers” section on page 7-1.
Configuring Server Clusters

Server clusters are groupings of CCM, DNS, CDNS, and DHCP servers at local cluster locations. For example, an organization might have Boston and Chicago clusters of DNS and DHCP servers. A central administrator might want to affect how addresses are allocated at these clusters, or poll lease history data from them. The central administrator might even want to connect to those local clusters, if the required permissions exist, to view changes there or restart the servers.

View the created clusters on the View Tree of Cluster Servers page. To get there, click Clusters. Once the page is populated with clusters, it shows some rich information and provides some useful functions. The Go Local icon allows single sign-on to a local cluster web UI, if an equivalent administrator account exists at the local cluster.

The View Tree of Clusters page might have been populated by manually adding clusters on the List/Add Remote Clusters page, or automatically when adding and synchronizing with routers, which also creates server clusters. The cluster names are links that you can click to edit the cluster information. The resynchronization, replication, and polling functions are described further on in this chapter.

The DHCP server may have the Related Servers icon next to the DHCP server for the cluster. Click this icon to open the List Related Servers for DHCP Server page (see the “Listing Related Servers for DHCP, DNS, and TCP Listener Servers” section on page 6-4). These servers can be DNS, or DHCP failover servers.

Related Topics

Adding Local Clusters, page 6-2
Editing Local Clusters, page 6-3
Listing Related Servers for DHCP, DNS, and TCP Listener Servers, page 6-4
Connecting to Local Clusters, page 6-10
Synchronizing with Local Clusters, page 6-10
Replicating Local Cluster Data, page 6-11
Viewing Replica Data, page 6-11
Deactivating, Reactivating, and Recovering Data for Clusters, page 6-12
Polling Lease History Data, page 6-13
Enabling Lease History Collection, page 6-14

Adding Local Clusters

Adding local clusters to the regional cluster is the core functionality of the central-cfg-admin role. To enable lease history data collection, see the “Polling Lease History Data” section on page 6-13.

The minimum required values to add a cluster are its name, IP address of the machine, administrator username, and password. The cluster name must be unique and its IP address must match that of the host where the CNRDB database is located. Obtain the SCP and HTTP ports, username, and password from the local cluster administrator. The preset value at Cisco Prime IP Express installation for the SCP port is 1234 and the HTTP port is 8080.

You can also set whether you want outbound connections to local servers to be secure by setting the use-ssl attribute to optional or required. It is set to optional by default, and it requires the Cisco Prime IP Express Communications Security Option installed to be effective.
Regional Web UI

From the Operate menu, choose Manage Servers under the Servers submenu. This opens the Manage Servers page. View the local clusters on this page. You can also add server clusters on the List/Add Remote Clusters page. The List/Add Remote Clusters page provide the following functions:

- Connect to a local cluster web UI for local administration.
- Resynchronize with a local cluster to reconcile updates there.
- Pull data over to a regional cluster replica database.
- Query lease history data from a local cluster. This function appears only if you are assigned the regional-addr-admin role with at least the lease-history subrole.

To add a cluster, click the Add Manage Clusters icon in the Manage Clusters pane. This opens the Add Cluster dialog box. For an example of adding a local cluster, see the “Create the Local Clusters” section on page 5-34. Click Add Cluster to return to the List/Add Remote Clusters page.

Local Web UI

You can also manage clusters in the local web UI. See the “Configuring Clusters in the Local Web UI” section on page 2-9 for details.

CLI Commands

To add a cluster, use cluster name create address to give the cluster a name and address and set the important attributes. For example:

nrcmd> cluster example-cluster create 192.168.100.101 admin=admin password=changeme

Note that the administrator must be a superuser to fully synchronize at the local cluster.

Editing Local Clusters

Editing local clusters at the regional cluster is the core functionality of the central-cfg-admin role.

Regional Web UI

To edit a local cluster, click its name on the Manage Clusters pane to open the Edit Remote Cluster page. This page is essentially the same as the List/Add Remote Clusters page, except for an additional attribute unset function. You can choose the service (dhcp, dns, cdns, or none) that you want to run in the local by checking/unchecked the check boxes provided in the Local Services area. Make your changes, then click Save.

Local Web UI

You can also edit clusters in the local web UI. See the “Configuring Clusters in the Local Web UI” section on page 2-9 for details.

CLI Commands

To edit a local cluster, use cluster name set attribute to set or reset the attributes. For example:

nrcmd> cluster Example-cluster set poll-replica-interval=8h
Listing Related Servers for DHCP, DNS, and TCP Listener Servers

If you have related DNS, or DHCP failover servers (see the “Setting Up Failover Server Pairs” section on page 28-3), you can access the attributes for these servers.

Regional Web UI

On the Failover Pairs or HA DNS Server Pair page, click the Manage Failover Servers tab and then click Related Servers tab to open the DHCP Related Server Attributes page. This page shows the communication and failover states the servers are in. Table 6-1 describes the attributes on this page. (For this page to appear, you must be assigned the central-cfg-admin role with the dhcp-management subrole.)

Table 6-1 Attributes for Related Servers

<table>
<thead>
<tr>
<th>Related Server Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related Server Type</td>
<td>Type of related server: DHCP, DNS, or LDAP.</td>
</tr>
<tr>
<td>Related Server IP Address</td>
<td>IP address of the related server. For DHCP failover partners, click this link to open the View Failover Related Server page (see Table 6-2 on page 6-4).</td>
</tr>
<tr>
<td>Communications</td>
<td>State of the communication—None, OK, or Interrupted.</td>
</tr>
<tr>
<td>Requests</td>
<td>Applies to DNS or LDAP related servers only, the number of requests from these servers.</td>
</tr>
<tr>
<td>State</td>
<td>For DHCP failover—None, Startup, Normal, Communications-interrupted, Partner-down, Potential-conflict, Recover, Paused, Shutdown, or Recover-done. For High-Availability (HA) DNS—Send-Update, Probe, or ha-state-unknown. Only the server that is successfully updating can be in Send-Update state. The partner server not sending updates is then always in Probe or unknown state. When the DHCP server comes up if there is no client activity, both DNS servers are often in the unknown state. This changes when the DHCP server tries to do DNS updates.</td>
</tr>
<tr>
<td>Partner Role</td>
<td>For DHCP failover only, the failover role of the partner—Main or Backup.</td>
</tr>
<tr>
<td>Partner State</td>
<td>For DHCP failover only, the partner’s state—None, Startup, Normal, Communications-interrupted, Partner-down, Potential-conflict, Recover, Paused, Shutdown, or Recover-done.</td>
</tr>
<tr>
<td>Update Response Complete</td>
<td>For DHCP failover only, the percentage of completed update responses, valid only if there are outstanding update responses.</td>
</tr>
</tbody>
</table>

Table 6-2 Attributes for DHCP Related Failover Servers

<table>
<thead>
<tr>
<th>Failover Partner Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General attributes</td>
<td></td>
</tr>
<tr>
<td>failover-pair-name</td>
<td>The name of the failover pair object used to manage this server.</td>
</tr>
<tr>
<td>current-time</td>
<td>Current time on the server returning this object.</td>
</tr>
</tbody>
</table>
### Failover Partner Attribute Description

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>comm-state</td>
<td>None, OK, or Interrupted.</td>
</tr>
<tr>
<td>smoothed-time-delta</td>
<td>The time difference between the local server and the partner server. If the local server time is ahead of the partner server time, the attribute value is positive. If the local server time is behind the partner server time, the attribute value is negative. If the servers are not communicating, the last known attribute value is recorded.</td>
</tr>
<tr>
<td>maximum-client-lead-time</td>
<td>Current maximum client lead time (MCLT) on this system.</td>
</tr>
<tr>
<td>sequence-number</td>
<td>Sequence number unique across failover objects, if different from the sequence in the lease, the lease is considered “not up to date” independent of the sf-up-to-date lease flag.</td>
</tr>
<tr>
<td>load-balancing-backup-pct</td>
<td>The current failover load balancing backup percentage. If the backup percentage is zero, failover load balancing is not in use (disabled).</td>
</tr>
</tbody>
</table>

#### Local server information

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>our-ipaddr</td>
<td>IPv4 address of the interface to this server.</td>
</tr>
<tr>
<td>our-ip6address</td>
<td>IPv6 address of the interface to this server.</td>
</tr>
<tr>
<td>role</td>
<td>Failover role of the server returning this object—None, Main, or Backup.</td>
</tr>
<tr>
<td>state</td>
<td>State of the local server—None, Startup, Normal, Communications-interrupted, Partner-down, Potential-conflict, Recover, Paused, Shutdown, or Recover-done.</td>
</tr>
<tr>
<td>start-time-of-state</td>
<td>Time at which the current failover state began.</td>
</tr>
<tr>
<td>start-of-comm-interrupted</td>
<td>Time at which this partner most recently went into communications-interrupted state. This is valid across reloads, while the start-time-of-state never has a time earlier than the most recent server reload.</td>
</tr>
<tr>
<td>est-end-recover-time</td>
<td>Valid if update-request-in-progress is not set to None. If it appears, the time at which the server enters the recover-done state if the update request outstanding is complete. If it does not appear, then the server enters recover-done whenever update-request is completed.</td>
</tr>
<tr>
<td>use-other-available</td>
<td>If false or unset, then this server cannot use other-available leases. If true, then the server can use other-available leases. Valid at all times, but should only be true if in partner-down state.</td>
</tr>
<tr>
<td>use-other-available-time</td>
<td>If, in partner-down state, the use-other-available is false or unset, the time when use-other-available will go to true.</td>
</tr>
<tr>
<td>safe-period-remaining</td>
<td>Duration in seconds remaining in safe-period. If not set to 0, then this server is currently running down a safe period with respect to its partner.</td>
</tr>
<tr>
<td>load-balancing-local-hba</td>
<td>The current hash bucket assignment of the local server, usually shown as a range of the hash bucket numbers. (See RFC 3074.)</td>
</tr>
<tr>
<td>request-buffers-in-use</td>
<td>The number of failover request buffers the DHCP server is using at the time the statistics are calculated.</td>
</tr>
<tr>
<td>decaying-max-request-buffers-in-use</td>
<td>The maximum number of failover request buffers that have recently been in use.</td>
</tr>
<tr>
<td>request-buffers-allocated</td>
<td>The number of request buffers that the server has allocated to support the failover capability.</td>
</tr>
</tbody>
</table>
### Configuring Server Clusters

#### Table 6-2 Attributes for DHCP Related Failover Servers (continued)

<table>
<thead>
<tr>
<th>Failover Partner Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>connection-start-time</code></td>
<td>The time at which the most recent connection started. This value is set whenever a connection is started, and it not cleared when a connection ended.</td>
</tr>
<tr>
<td><code>connection-end-time</code></td>
<td>The time at which the most recent connection ended. This value is set whenever a connection is ended, and it not cleared when a new connection starts.</td>
</tr>
</tbody>
</table>

#### Partner server information

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ipaddr</code></td>
<td>IP address of the partner server.</td>
</tr>
<tr>
<td><code>ip6address</code></td>
<td>IPv6 address of the partner server.</td>
</tr>
<tr>
<td><code>partner-role</code></td>
<td>Failover role of the partner of the server returning this object—None, Main, or Backup.</td>
</tr>
<tr>
<td><code>partner-state</code></td>
<td>Last known state which the partner end of the failover relationship is in—None, Startup, Normal, Communications-interrupted, Partner-down, Potential-conflict, Recover, Paused, Shutdown, or Recover-done.</td>
</tr>
<tr>
<td><code>start-time-of-partner-state</code></td>
<td>Time at which the partner current failover state began.</td>
</tr>
<tr>
<td><code>est-partner-end-recover-time</code></td>
<td>If the <code>partner-state</code> is Recover, an estimated prediction of when the partner will time out its MCLT and finish being in recover state.</td>
</tr>
<tr>
<td><code>last-comm-ok-time</code></td>
<td>Time at which this server last found communications to be OK.</td>
</tr>
<tr>
<td><code>load-balancing-partner-hba</code></td>
<td>The current hash bucket assignment of the partner server, usually shown as a range of the hash bucket numbers. (See RFC 3074.)</td>
</tr>
<tr>
<td><code>partner-vendor-major-version</code></td>
<td>The vendor ID major version from the partner server.</td>
</tr>
<tr>
<td><code>partner-vendor-minor-version</code></td>
<td>The vendor ID minor version from the partner server.</td>
</tr>
</tbody>
</table>

#### Update requests sent to partner

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>update-request-outstanding</code></td>
<td>If None or unset, then the server does not have an update request queued for its partner. If not set to None, then it does have an update request queued for its failover partner. Valid values are None, Update, and Update-all.</td>
</tr>
<tr>
<td><code>update-request-start-time</code></td>
<td>Time at which any <code>update-request-outstanding</code> request was started.</td>
</tr>
<tr>
<td><code>update-request-done-time</code></td>
<td>Time at which the last of any update request completed.</td>
</tr>
<tr>
<td><code>v6-update-response-in-progress</code></td>
<td>The type and origin of the response.</td>
</tr>
<tr>
<td><code>v6-update-response-percent-complete</code></td>
<td>The percent complete of the current IPv6 update response.</td>
</tr>
<tr>
<td><code>v6-update-response-start-time</code></td>
<td>The time that the IPv6 update response mentioned in <code>v6-update-response-in-progress</code> was started.</td>
</tr>
<tr>
<td><code>v6-update-response-done-time</code></td>
<td>The time that the most recent IPv6 update response sent an update done to the partner server.</td>
</tr>
</tbody>
</table>
### Table 6-2  Attributes for DHCP Related Failover Servers (continued)

<table>
<thead>
<tr>
<th>Failover Partner Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Update requests processed for partner</strong></td>
<td></td>
</tr>
<tr>
<td><code>update-response-in-progress</code></td>
<td>If this server is processing an update response, gives information about the type and origin of the response.</td>
</tr>
<tr>
<td><code>update-response-percent-complete</code></td>
<td>If <code>update-response-outstanding</code> appears, the percent complete of the current update response.</td>
</tr>
<tr>
<td><code>update-response-start-time</code></td>
<td>Time that the update response mentioned in <code>update-response-in-progress</code> was started.</td>
</tr>
<tr>
<td><code>update-response-done-time</code></td>
<td>Time that the most recent update response sent an update done to the partner server.</td>
</tr>
</tbody>
</table>

### Load Balancing Counters

<table>
<thead>
<tr>
<th>Load Balancing Counters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>load-balancing-processed-requests</code></td>
<td>The number of server processed requests, both IPv4 and IPv6, subject to load balancing. This counter includes only the requests made after the latest transition of server to normal state.</td>
</tr>
<tr>
<td><code>load-balancing-dropped-requests</code></td>
<td>The number of server dropped requests, both IPv4 and IPv6, subject to load balancing. This counter includes only the requests made after the latest transition of server to normal state.</td>
</tr>
<tr>
<td><code>load-balancing-processed-total</code></td>
<td>The number of server processed requests, both IPv4 and IPv6, subject to load balancing. This counter includes the requests since this server was last started or reloaded.</td>
</tr>
<tr>
<td><code>load-balancing-dropped-total</code></td>
<td>The number of server dropped requests, both IPv4 and IPv6, subject to load balancing. This counter includes the requests since this server was last started or reloaded.</td>
</tr>
</tbody>
</table>

### Binding Update or Ack Counters (this connection)

<table>
<thead>
<tr>
<th>Binding Update or Ack Counters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>binding-updates-sent</code></td>
<td>The number of binding update (BNDUPD) messages sent to the failover partner.</td>
</tr>
<tr>
<td><code>binding-acks-received</code></td>
<td>The number of binding acknowledgement (BNDACK) messages received from the failover partner.</td>
</tr>
<tr>
<td><code>binding-updates-received</code></td>
<td>The number of binding update (BNDUPD) messages received from the failover partner.</td>
</tr>
<tr>
<td><code>binding-acks-sent</code></td>
<td>The number of binding acknowledgement (BNDACK) messages sent to the failover partner.</td>
</tr>
<tr>
<td><code>v6-binding-updates-sent</code></td>
<td>The number of IPv6 binding updates (BNDUPD6) messages received from the failover partner since the start of the most recently established connection.</td>
</tr>
<tr>
<td><code>v6-binding-acks-received</code></td>
<td>The number of IPv6 binding acknowledgements (BNDACK6) messages received from the failover partner since the start of the most recently established connection.</td>
</tr>
<tr>
<td><code>v6-binding-updates-received</code></td>
<td>The number of IPv6 binding updates (BNDUPD6) messages received from the failover partner since the start of the most recently established connection.</td>
</tr>
</tbody>
</table>
### Table 6-2 Attributes for DHCP Related Failover Servers (continued)

<table>
<thead>
<tr>
<th>Failover Partner Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>v6-binding-acks-sent</td>
<td>The number of IPv6 binding acknowledgements (BNDACK6) messages sent to the failover partner since the start of the most recently established connection.</td>
</tr>
</tbody>
</table>

**Binding Update/Ack Counters Totals**

<table>
<thead>
<tr>
<th>Binding Updates/Ack Counters Totals</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>binding-updates-sent-total</td>
<td>The number of IPv4 binding updates (BNDUPD) messages sent to the failover partner since the most recent statistics reset.</td>
</tr>
<tr>
<td>binding-acks-received-total</td>
<td>The number of IPv4 binding acknowledgements (BNDACK) messages received from the failover partner since the most recent statistics reset.</td>
</tr>
<tr>
<td>binding-updates-received-total</td>
<td>The number of IPv4 binding updates (BNDUPD) messages received from the failover partner since the most recent statistics reset.</td>
</tr>
<tr>
<td>binding-acks-sent-total</td>
<td>The number of IPv4 binding acknowledgements (BNDACK) messages sent to the failover partner since the most recent statistics reset.</td>
</tr>
<tr>
<td>v6-binding-updates-sent-total</td>
<td>The number of IPv6 binding updates (BNDUPD6) messages sent to the failover partner since the most recent statistics reset.</td>
</tr>
<tr>
<td>v6-binding-acks-received-total</td>
<td>The number of IPv6 binding acknowledgements (BNDACK6) messages received from the failover partner since the most recent statistics reset.</td>
</tr>
<tr>
<td>v6-binding-updates-received-total</td>
<td>The number of IPv6 binding updates (BNDUPD6) messages received from the failover partner since the most recent statistics reset.</td>
</tr>
<tr>
<td>v6-binding-acks-sent-total</td>
<td>The number of IPv6 binding acknowledgements (BNDACK6) messages sent to the failover partner since the most recent statistics reset.</td>
</tr>
</tbody>
</table>

**Flow Control Counters (this connection)**

<table>
<thead>
<tr>
<th>Flow Control Counters (this connection)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>current-binding-updates-in-flight</td>
<td>The current number of binding updates (both IPv4 and IPv6) that are currently in-flight (sent).</td>
</tr>
<tr>
<td>current-binding-updates-queued</td>
<td>The current number of binding updates (both IPv4 and IPv6) that are queued at present.</td>
</tr>
<tr>
<td>maximum-binding-updates-in-flight</td>
<td>The maximum number of binding updates (both IPv4 and IPv6) that were in-flight (sent) at one time.</td>
</tr>
<tr>
<td>maximum-binding-updates-queued</td>
<td>The maximum number of binding updates (both IPv4 and IPv6) that were queued at one time.</td>
</tr>
<tr>
<td>last-binding-update-sent-time</td>
<td>The time the last binding update (either IPv4 or IPv6) was sent.</td>
</tr>
<tr>
<td>last-binding-ack-received-time</td>
<td>The time the last IPv4 or IPv6 binding acknowledgement (whether NAKed or not) was received.</td>
</tr>
<tr>
<td>last-binding-update-received-time</td>
<td>The time the last binding update (either IPv4 or IPv6) was received.</td>
</tr>
<tr>
<td>last-binding-ack-sent-time</td>
<td>The time the last IPv4 or IPv6 binding acknowledgement (whether NAKed or not) was sent.</td>
</tr>
</tbody>
</table>
### Table 6-3 Attributes for DNS Related Failover Servers

<table>
<thead>
<tr>
<th>Failover Partner Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General attributes</strong></td>
<td></td>
</tr>
<tr>
<td><code>current-time</code></td>
<td>Current time on the server returning this object.</td>
</tr>
<tr>
<td><code>ipaddr</code></td>
<td>IP address</td>
</tr>
<tr>
<td><code>comm-state</code></td>
<td>None.</td>
</tr>
<tr>
<td><code>dns-server-state</code></td>
<td>PROBE.</td>
</tr>
<tr>
<td><code>probe-polling-event-id</code></td>
<td>Zero.</td>
</tr>
<tr>
<td><code>requests</code></td>
<td>Zero.</td>
</tr>
<tr>
<td><strong>HA DNS Configuration information</strong></td>
<td></td>
</tr>
<tr>
<td><code>ha-dns-role</code></td>
<td>STANDALONE-DNS.</td>
</tr>
<tr>
<td><code>dns-timeout</code></td>
<td>Number of milliseconds that the DHCP server will wait for a response from the DNS server for a dynamic dns update, before retrying dynamic dns update.</td>
</tr>
<tr>
<td><code>max-dns-retries</code></td>
<td>Number of times that the DHCP server will try to send dynamic updates to a DNS server.</td>
</tr>
<tr>
<td><code>ha-dns-failover-timeout</code></td>
<td>Maximum time period, in seconds, the DHCP server will wait for a reply from a DNS server, before the DHCP will failover to use next DNS Server to perform the dynamic-update. Default value is 30 seconds.</td>
</tr>
<tr>
<td><code>ha-dns-probe-timeout</code></td>
<td>If cnr-ha-dns is enabled, DHCP server will use this timer to co-ordinate and reduce latency in failing over between HA-DNS servers, when HA-DNS servers are in COMMUNICATION-INTERRUPTED state or SYNCHRONIZING. Default value is 3 seconds.</td>
</tr>
<tr>
<td><code>ha-dns-probe-retry</code></td>
<td>If cnr-ha-dns is enabled, DHCP server will use this retry count and ha-dns-probe-timeout to co-ordinate and reduce latency in failing over between HA-DNS servers, when HA-DNS servers are in COMMUNICATION-INTERRUPTED state or SYNCHRONIZING. Default value is 1 retry attempt.</td>
</tr>
<tr>
<td><strong>Current HA DNS State Information</strong></td>
<td></td>
</tr>
<tr>
<td><code>ha-dns-state</code></td>
<td>State of HA-DNS Servers interaction.</td>
</tr>
<tr>
<td><code>last-ha-dns-state</code></td>
<td>Failover role of the partner of the server returning this object—None, Main, or Backup.</td>
</tr>
<tr>
<td><code>last-ha-dns-state-change-time</code></td>
<td>Time at which the failover role was last changed.</td>
</tr>
<tr>
<td><code>last-reply-received-time</code></td>
<td>Time at which the last reply was received.</td>
</tr>
<tr>
<td><code>last-ha-dns-role-switch-time</code></td>
<td>Time at which the failover role was changed from one state to another.</td>
</tr>
</tbody>
</table>

### Table 6-4 Attributes for TCP Listener Related Servers

<table>
<thead>
<tr>
<th>Failover Partner Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General attributes</strong></td>
<td></td>
</tr>
<tr>
<td><code>comm-state</code></td>
<td>None.</td>
</tr>
</tbody>
</table>
To refresh the data on the Related Server tab, click **Refresh Data**.

On the Related Server tab, if the partner is in the Communications-interrupted failover state, you can click **Set Partner Down** in association with an input field for the partner-down date setting. This setting is initialized to the value of the `start-of-communications-interrupted` attribute. (In Normal web UI mode, you cannot set this date to be an earlier value than the initialized date. In Expert web UI mode, you can set this value to any date.) After clicking **Set Partner Down**, you return to the List Related Servers for DHCP Server page to view the result of the partner-down action. Never set both partners to Partner Down mode.

To return from the List Related Servers for DHCP Server page or View Failover Related Server page, click **Return**.

**CLI Commands**

To list the related servers for a DHCP server, use `dhcp getRelatedServers`.

**Connecting to Local Clusters**

In the web UI, if you have an equivalent administrator account at the local cluster, you can single sign-on to the local cluster Manage Servers page by clicking the **Connect** icon on the List/Add Remote Clusters page. To return to the regional cluster web UI, click the **Return** icon at the top right corner of the local cluster page. If you do not have an equivalent account at the local cluster, the Connect icon opens the local cluster login page.

**Synchronizing with Local Clusters**

Synchronization is configuring regional and local clusters so that they can work together in a unified fashion. When you synchronize:

1. The list of local servers are copied to the regional cluster.
2. A shared secret is established between the regional and local clusters for single sign-on.

### Table 6-4 Attributes for TCP Listener Related Servers

<table>
<thead>
<tr>
<th>Failover Partner Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>current-connections</td>
<td>Zero</td>
</tr>
<tr>
<td><code>ipaddr</code></td>
<td>IP address.</td>
</tr>
<tr>
<td><code>ip6addr</code></td>
<td>IPv6 address.</td>
</tr>
<tr>
<td>name</td>
<td><code>foobar</code> string (w/o null terminator).</td>
</tr>
<tr>
<td>port</td>
<td>Port number.</td>
</tr>
<tr>
<td><code>rejected-connections</code></td>
<td>Zero.</td>
</tr>
<tr>
<td><code>total-connections</code></td>
<td>Zero.</td>
</tr>
</tbody>
</table>
Synchronization occurs once when you create a local cluster at the regional cluster. However, changes might occur at the local cluster periodically, requiring you to re-synchronize with it. For example, you might change the username and password used to make local connections. Resynchronization does not happen automatically—you must click the Resynchronize icon next to the cluster name on the List/Add Remote Clusters page. The result is a positive confirmation for success or an error message for a failure.

When you upgrade the local cluster, you should also resynchronize the cluster. For synchronization to be effective, the user account specified for the local cluster must be a superuser. If you get a synchronization error message, check the local cluster to ensure that it is running properly.

**Note**

When you resynchronize clusters at the regional cluster, an automatic reinitialization of replica data occurs. The result is that for larger server configurations, resynchronization might take several minutes. The benefit, however, is that you do not need a separate action to update the replica data.

**Replicating Local Cluster Data**

Replication is copying the configuration data from a local server to the regional cluster replica database. Replication needs to occur before you can pull DHCP object data into the regional server database. During replication:

1. The current data from the local database is copied to the regional cluster. This usually occurs once.
2. Any changes made in the master database since the last replication are copied over.

Replication happens at a given time interval. You can also force an immediate replication by clicking the Replicate icon on the List/Add Remote Clusters page.

You can set the automatic replication interval on the Add Server Cluster page, or adjust it on the Edit Server Cluster page, using the poll-replica-interval attribute. This interval is preset at four hours. You can also set the fixed time of day to poll replica data by using the poll-replica-offset attribute; its default value is zero hours (no offset).

**Caution**

If the replica database is corrupted in any way, the regional CCM server will not start. If you encounter this problem, stop the regional service, remove (or move) the replica database files located in the install-path/regional/data/replica directory (and the log files in the /logs subdirectory), then restart the regional server. Doing so recreates the replica database without any data loss.

**Viewing Replica Data**

In the web UI, you can view the replica data cached in the replica database at the regional cluster by choosing View Replica Data from Servers submenu under the Operate menu. This opens the View Replica Class List page.

**Regional Web UI**

Select the:

1. Cluster in the Select Cluster list.
2. Object class in the Select Class list.
3. Replicate the data for the cluster and class chosen. Click the Replicate Data for Cluster button.
4. View the replica data. Click **View Replica Class List**, which opens a List Replica Data for Cluster page for the cluster and specific class of object you choose. On this page, you can:

- Click the name of an object to open a View page at the regional cluster. Return to the List Replica page by clicking **Return to object List**.

  **Note**  The List Replica Address Blocks and List Replica Subnets pages do not provide this function. To view the address blocks or subnets for the local cluster, use the **Go Local** icon.

- Click the **Connect** icon to go to the List page for the object at the local cluster. Return to the List Replica object page by clicking the **Return** icon.

Click **Return** on the List Replica Data for Cluster page to return to the View Replica Class List page.

**Deactivating, Reactivating, and Recovering Data for Clusters**

Deactivating a cluster might be necessary if you suspect that a hard disk error occurred where configuration data could have been lost. You can deactivate the cluster, remedy the problem, recover cluster data from the replica database, then reactivate the cluster. This saves you from having to delete and then recreate the cluster with all of its data lost in the process.

Deactivating, reactivating, and recovering the data for a cluster is available only in the web UI, and you must be an administrator assigned the central-config-admin role.

Data that is not recovered (and that you need to manually restore) includes:

- Contents of the **cnr.conf** file (see the “Modifying the cnr.conf File” section on page 7-23)
- Web UI configuration files
- Unprotected DNS resource records
- Administrator accounts

  **Note**  If the local secret db is lost, the old references are no longer valid, even though they are restored. To recover your passwords, you have to use central management for your admins, and then push them to your local clusters. Routers, since they have their own secrets, also need to be centrally managed and then should be re-pushed. For the local cluster partner objects, running the sync from regional will create valid objects, but the old cluster objects may need to be deleted first.

- Lease history
- Extension scripts

  **Note**  Restoring the data to a different IP address requires some manual reconfiguration of such things as DHCP failover server pair and High-Availability (HA) DNS server pair addresses.
Regional Web UI

Deactivate a cluster by clicking the Deactivate button for the cluster. This immediately changes the button to Reactivate to show the status of the cluster. Deactivating a cluster disables deleting, synchronizing, replicating data, and polling lease history. These operations are not available while the cluster is deactivated.

Deactivating the cluster also displays the Recover icon in the Recover Data column of the cluster. Click this icon to recover the replica data. This opens a separate “in process” status window that prevents any operations on the web UI pages while the recovery is in process. As soon as the recovery is successful, the disabled functions are again enabled and available.

To reactivate the cluster, click the Reactivate button to change back to the Deactivate button and show the status as active.

Polling Lease History Data

Lease history data is automatically collected at any regional cluster where these feature is enabled for the DHCP server or failover pair. The default polling interval to update the regional databases is 4 hours. You can poll the servers by clicking the Lease History icon on the List/Add Remote Clusters page. For this manual polling, if the server is in a failover relationship, data is only retrieved for the subnets where the server is the main.

If you have address space privileges (you are assigned the regional-addr-admin role with at least the lease-history subrole), you can query the lease history data by choosing Current Utilization or Lease History from Operate menu (see the “Running IP Lease Histories” section on page 23-21).

Related Topics

Polling Process, page 6-13
Adjusting the Polling Intervals, page 6-13

Polling Process

When the regional cluster polls the local cluster for lease history, it first requests all available data up to the current time. This time is recorded in the history databases, and subsequent polls request only new data from this time forward. All times are stored relative to each local cluster time, adjusted for that cluster time zone.

If the times on each server are not synchronized, you might observe odd query results. For example, if the regional cluster time lags behind that of a local cluster, the collected history might be in the future relative to the time range queries at the regional cluster. If so, the result of the query would be an empty list. Data merged from the several clusters could also appear out of sequence, because of the different time skews between local clusters. This type of inconsistency would make it difficult to interpret trends. To avoid these issues, using a network time service for all clusters is strongly recommended.

Adjusting the Polling Intervals

You can adjust the automatic polling interval for lease history, along with other attributes. These attributes are set in three places at the regional cluster, with the following priority:

1. **Cluster**—These values override the server-wide settings, unless they are unset, in which case the server values are used. The cluster values are set when adding or editing the cluster. In the CLI, set the attributes listed in Table 6-5, using the `cluster` command.
2. **Regional CCM server** (the preset polling interval is 4 hours)—This is set on the Edit CCM Server page, accessible by clicking **Servers**, then the Local CCM Server link. In the CLI, set the attributes listed in Table 6-5 using the `ccm` command.

![Note]
If lease history collection is not explicitly turned on at the local cluster DHCP server (see the “Enabling Lease History Collection” section on page 6-14), no data is collected, even though polling is on by default.

<table>
<thead>
<tr>
<th>Table 6-5 Subnet Utilization and Lease History Polling Regional Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attribute Type</strong></td>
</tr>
<tr>
<td>Polling interval—How often to poll data</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Retry interval—How often to retry after an unsuccessful polling</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Offset—Hour of the day to guarantee polling</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The polling offset attribute ensures that polling occurs at a specific hour of the day, set as 24-hour time, in relation to the polling interval. For example, if you set the interval to 4h and the offset to 6h (6 A.M.), the polling occurs at 2 A.M., 6 A.M., 10 A.M., 2 P.M., 6 P.M., and 10 P.M. each day.

**Enabling Lease History Collection**

**Step 1** Configure the local cluster DHCP server with scopes and address ranges so that clients have requested leases.

**Step 2** Explicitly enable lease history data collection. The DHCP server attributes to set are:

- **ip-history**—Enable or disable the lease history database for v4-only (DHCPv4), v6-only (DHCPv6), or both.
- **ip-history-max-age**—Limit on the age of the history records (preset to 4 weeks).

In the CLI, set the attributes using the `dhcp set ip-history=<value> (v4-only, v6-only, both, or disable)` command.

**Step 3** If in staged dhcp edit mode, reload the local cluster DHCP server.

**Step 4** At the regional cluster, create the cluster that includes this DHCP server.

**Step 5** In the regional web UI, go to the Lease History Settings section of the List/Add Remote Clusters page.

**Step 6** Set the attributes in Table 6-5 on page 6-14.

**Step 7** Click **Save**.

**Step 8** On the List/Add Remote Clusters page, click the **Replica** icon next to the cluster name.

**Step 9** Click the **Lease History** icon for the cluster involved to obtain the initial set of lease history data. This data is refreshed automatically at each polling interval.
Managing DHCP Scope Templates

Scope templates apply certain common attributes to multiple scopes. These common attributes include a scope name based on an expression, policies, address ranges, and an embedded policy options based on an expression. The scope templates you add or pull from the local clusters are visible on the List/Add DHCP Scope Templates page (choose Scope Templates from the Design > DHCPv4 menu).

For details on creating and editing scope templates, and applying them to scopes, see the “Creating and Applying Scope Templates” section on page 21-3. The regional cluster web UI has the added feature of pushing scope templates to local clusters and pulling them from local clusters.

Related Topics

Pushing Scope Templates to Local Clusters, page 6-15
Pulling Scope Templates from Replica Data, page 6-16

Pushing Scope Templates to Local Clusters

You can push the scope templates you create from the regional cluster to any of the local clusters. In the web UI, go to the List/Add DHCP Scope Templates page, and do any of the following:

- if you want to push a specific template to a cluster, select the scope template from the Scope Templates pane on the left, and click Push (at the top of the page). This opens the Push DHCP Scope Template page.
- If you want to push all of the available scope templates, click the Push All icon at the top of the Scope Templates pane. This opens the Push Data to Local Clusters page.

Regional Web UI

The Push DHCP Scope Template page and Push Data to Local Clusters page identify the data to push, how to synchronize it with the local cluster, and the cluster or clusters to which to push it. The data synchronization modes are:

- **Ensure** (preset value)—Ensures that the local cluster has new data without affecting any existing data.
- **Replace**—Replaces data without affecting other objects unique to the local cluster.
- **Exact**—Available for “push all” operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the local cluster.

Choose the destination cluster or clusters in the Available field and move it or them to the Selected field.

Tip

The synchronization mode and cluster choice settings are persistent for the duration of the current login session, so that they are in effect each time you access this page, unless you change them.

After making these choices, click Push Data to Clusters. This opens the View Push Scope Template Data Report page.
Pulling Scope Templates from Replica Data

You may choose to pull scope templates from the replica data of the local clusters instead of explicitly creating them. (You may first want to update the policy replica data by clicking the Replicate icon next to the cluster name.) To pull the scope templates in the regional web UI, click the Pull Replica icon at the top of the Scope Templates pane.

Regional Web UI

The Select Replica DHCP Scope Template Data to Pull page shows a tree view of the regional server replica data for the local clusters’ scope templates. The tree has two levels, one for the local clusters and one for the scope templates in each cluster. You can pull individual scope templates from the clusters, or you can pull all of their scope templates. To pull individual scope templates, expand the tree for the cluster, then click Pull Scope Template next to its name. To pull all the scope templates from a cluster, click Pull All Scope Templates.

To pull the scope templates, you must also choose a synchronization mode:

- Ensure—Ensures that the regional cluster has new data without affecting any existing data.
- Replace (preset value)—Replaces data without affecting other objects unique to the regional cluster.
- Exact—Available for “pull all” operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the regional cluster.

Managing DHCP Policies

Every DHCP server must have one or more policies defined for it. Policies define lease duration, gateway routers, and other configuration parameters, in what are called DHCP options. Policies are especially useful if you have multiple scopes, because you need only define a policy once and apply it to the multiple scopes.

For details on creating and editing DHCP policies, and applying them to scopes, see the “Configuring DHCP Policies” section on page 22-1. The regional cluster web UI has the added feature of pushing policies to, and pulling them from, the local clusters.

Related Topics

- Pushing Policies to Local Clusters, page 6-16
- Pulling Policies from Replica Data, page 6-17

Pushing Policies to Local Clusters

You can also push the policies you create from the regional cluster to any of the local clusters. In the regional web UI, go to List/Add DHCP Policies page, and do any of the following:

- If you want to push a specific policy to a cluster, select the policy from the Policies pane on the left, and click Push (at the top of the page).
- If you want to push all the policies, click the Push All icon at the top of the Policies pane.
Regional Web UI

The Push DHCP Policy Data to Local Clusters page identifies the data to push, how to synchronize it with the local cluster, and the cluster or clusters to which to push it. The data synchronization modes are:

- **Ensure** (preset value)—Ensures that the local cluster has new data without affecting any existing data.
- **Replace**—Replaces data without affecting other objects unique to the local cluster.
- **Exact**—Available for push-all operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the local cluster.

Choose the destination cluster or clusters in the Available field and move it or them to the Selected field. Then click **Push Data to Clusters** to open the View Push Policy Data Report page.

Tip

The synchronization mode and cluster choice settings are persistent for the duration of the current login session, so that they are in effect each time you access this page, unless you change them.

Pulling Policies from Replica Data

You may choose to pull policies from the replica data of the local clusters instead of explicitly creating them. (In the regional web UI, you may first want to update the policy replica data by clicking the **Replicate** icon next to the cluster name). To pull the policies, click the **Pull Replica** icon at the top of the Policies pane.

Regional Web UI

The Select Replica DHCP Policy Data to Pull page shows a tree view of the regional server replica data for the local clusters’ policies. The tree has two levels, one for the local clusters and one for the policies in each cluster. You can pull individual policies from the clusters, or you can pull all of their policies. To pull individual policies, expand the tree for the cluster, then click **Pull Policy** next to its name. To pull all the policies from a cluster, click **Pull All Policies**.

To pull all the policies, you must also choose a synchronization mode:

- **Ensure**—Ensures that the regional cluster has new data without affecting any existing data.
- **Replace** (preset value)—Replaces data without affecting other objects unique to the regional cluster.
- **Exact**—Available for “pull all” operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the regional cluster.

Managing DHCP Client-Classes

Client-classes provide differentiated services to users that are connected to a common network. You can group your user community based on administrative criteria, and then ensure that each user receives the appropriate class of service. Although you can use the Cisco Prime IP Express client-class facility to control any configuration parameter, the most common uses are for:

- **Address leases**—How long a set of clients should keep its addresses.
- **IP address ranges**—From which lease pool to assign clients addresses.
- **DNS server addresses**—Where clients should direct their DNS queries.
• **DNS hostnames**—What name to assign clients.
• **Denial of service**—Whether unauthorized clients should be offered leases.

For details on creating and editing client-classes, see Chapter 25, “Configuring Client-Classes and Clients.” The regional cluster web UI has the added feature of pushing client-classes to, and pulling them from, the local clusters.

**Related Topics**

- Pushing Client-Classes to Local Clusters, page 6-18
- Pulling Client-Classes from Replica Data, page 6-18

**Pushing Client-Classes to Local Clusters**

You can also push the client-classes you create from the regional cluster to any of the local clusters. In the Regional web UI, go to the List/Add DHCP Client Classes page, and do any of the following:

- If you want to push a specific client-class to a cluster in the web UI, select the client-class from the Client Classes pane on the left, and click **Push** (at the top of the page). This opens the Push DHCP Client Class page.
- If you want to push all the client-classes, click the **Push All** icon at the top of the Client Classes pane. This opens the Push Data to Local Clusters page.

**Regional Web UI**

The Push DHCP Client Class page and Push Data to Local Clusters page identifies the data to push, how to synchronize it with the local cluster, and the cluster or clusters to which to push it. The data synchronization modes are:

- **Ensure** (preset value)—Ensures that the local cluster has new data without affecting any existing data.
- **Replace**—Replaces data without affecting other objects unique to the local cluster.
- **Exact**—Available for “push all” operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the local cluster.

Choose the destination cluster or clusters in the Available field and move it or them to the Selected field. Then click **Push Data to Clusters** to open the View Push Client-Class Data Report page.

**Tip**

The synchronization mode and cluster choice settings are persistent for the duration of the current login session, so that they are in effect each time you access this page, unless you change them.

**Pulling Client-Classes from Replica Data**

You may choose to pull client-classes from the replica data of the local clusters instead of explicitly creating them. (In the web UI, you might first want to update the client-class replica data by clicking the **Replicate** icon next to the cluster name.) To pull the client-classes, click the **Pull Replica** icon at the top of the Client Classes pane.
Regional Web UI

The Select Replica DHCP Client-Class Data to Pull page shows a tree view of the regional server replica data for the local clusters’ client-classes. The tree has two levels, one for the local clusters and one for the client-classes in each cluster. You can pull individual client-classes from the clusters, or you can pull all of their client-classes. To pull individual client-classes, expand the tree for the cluster, then click Pull Client-Class next to its name. To pull all the client-classes from a cluster, click Pull All Client-Classes.

To pull the client-classes, you must also choose a synchronization mode:

- **Ensure**—Ensures that the regional cluster has new data without affecting any existing data.
- **Replace** (preset value)—Replaces data without affecting other objects unique to the regional cluster.
- **Exact**—Available for “pull all” operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the regional cluster.

Managing Virtual Private Networks

A virtual private network (VPN) is a specialized address space identified by a key. A VPN allows address overlap in a network, because the addresses are distinguished by separate keys. Most IP addresses exist in the global address space outside of a VPN. You can create regional VPNs only if you are an administrator assigned the dhcp-management subrole of the central-cfg-admin role.

For details on creating and editing VPNs, and applying them to various network objects, see the “Configuring Virtual Private Networks Using DHCP” section on page 24-18. The regional web UI has the added feature of pushing VPNs to local clusters and pulling them from local clusters.

Related Topics

- Pushing VPNs to Local Clusters, page 6-19
- Pulling VPNs from Replica Data, page 6-20

Pushing VPNs to Local Clusters

You can push the VPNs you create from the regional cluster to any of the local clusters. In the Regional web UI, go to the List/Add VPNs page, and do any of the following:

- If you want to push a specific VPN to a cluster in the web UI, select the VPN from the VPNs pane on the left, and click Push (at the top of the page). This opens the Push VPN page.
- If you want to push all the VPNs, click the Push All icon at the top of the VPNs pane. This opens the Push Data to Local Clusters page.

Regional Web UI

The Push VPN page and Push Data to Local Clusters page identify the data to push, how to synchronize it with the local cluster, and the cluster or clusters to which to push it. The data synchronization modes are:

- **Ensure** (preset value)—Ensures that the local cluster has new data without affecting any existing data.
- **Replace**—Replaces data without affecting other objects unique to the local cluster.
Managing DHCP Failover Pairs

With DHCP failover, a backup DHCP server can take over for a main server if the latter comes off the network for any reason. You can use failover to configure two servers to operate as a redundant pair. If one server is down, the other server seamlessly takes over so that new DHCP clients can get, and existing clients can renew, their addresses. Clients requesting new leases need not know or care about which server responds to their lease request. These clients can obtain leases even if the main server is down.

In the regional web UI, you can view any created failover pairs on the List/Add DHCP Failover Pairs page. To access this page, click DHCP, then Failover. This functionality is available only to administrators who are assigned the dhcp-management subrole of the central-cfg-admin role.

For details on creating and editing failover pairs, see the “Setting Up Failover Server Pairs” section on page 28-3. The regional cluster web UI has the added feature of pulling addresses from local clusters to create the failover pairs.

To pull the address space for a failover pair, you must have regional-addr-admin privileges.

Tip

The synchronization mode and cluster choice settings are persistent for the duration of the current login session, so that they are in effect each time you access this page, unless you change them.

Pulling VPNs from Replica Data

Instead of explicitly creating VPNs, you can pull them from the local clusters. (In the regional web UI, you may first want to update the VPN replica data by clicking the Replica icon next to the cluster name.) To pull the replica data, click the Pull Replica icon at the top of the VPNs pane on the left, to open the Select Replica VPN Data to Pull page.

This page shows a tree view of the regional server replica data for the local clusters’ VPNs. The tree has two levels, one for the local clusters and one for the VPNs in each cluster. You can pull individual VPNs or you can pull all of them. To pull individual VPNs, expand the tree for the cluster, then click Pull VPN next to its name. To pull all the VPNs, click Pull All VPNs.

To pull the VPNs, you must choose a synchronization mode:

- **Ensure**—Ensures that the regional cluster has new data without affecting any existing data.
- **Replace** (preset value)—Replaces data without affecting other objects unique to the regional cluster.
- **Exact**—Available for “pull all” operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the regional cluster.

Managing DHCP Failover Pairs

With DHCP failover, a backup DHCP server can take over for a main server if the latter comes off the network for any reason. You can use failover to configure two servers to operate as a redundant pair. If one server is down, the other server seamlessly takes over so that new DHCP clients can get, and existing clients can renew, their addresses. Clients requesting new leases need not know or care about which server responds to their lease request. These clients can obtain leases even if the main server is down.

In the regional web UI, you can view any created failover pairs on the List/Add DHCP Failover Pairs page. To access this page, click DHCP, then Failover. This functionality is available only to administrators who are assigned the dhcp-management subrole of the central-cfg-admin role.

For details on creating and editing failover pairs, see the “Setting Up Failover Server Pairs” section on page 28-3. The regional cluster web UI has the added feature of pulling addresses from local clusters to create the failover pairs.

To pull the address space for a failover pair, you must have regional-addr-admin privileges.

Tip

The synchronization mode and cluster choice settings are persistent for the duration of the current login session, so that they are in effect each time you access this page, unless you change them.
Managing Lease Reservations

You can push lease reservations you create from the regional cluster to any of the local clusters. In the regional cluster web UI, go to the List/Add DHCPv4 Reservations page or List/Add DHCPv6 Reservations page, and click the Push All icon in the Reservations pane on the left. Note that you cannot push individual reservations. If the cluster pushed to is part of a DHCP failover configuration, pushing a reservation also pushes it to the partner server.

Related Topics

DHCPv4 Reservations, page 6-21
DHCP v6 Reservations, page 6-21

DHCPv4 Reservations

To create DHCPv4 reservations, the parent subnet object must exist on the regional server. If there are pending reservation edits at regional, these can be pushed to the subnet local cluster or failover pair. If the subnet has never been pushed, the parent scope is added to the local cluster or pair.

Once a subnet is pushed to a local cluster or pair, reservations are pushed to that cluster or pair. To move the scopes and subnet to another local cluster or failover pair, the subnet must first be reclaimed.

DHCPv6 Reservations

To create DHCPv6 reservations, the parent prefix must exist on the regional server. When there are pending reservation or prefix changes, you can push the updates to the local cluster.

Once a prefix is pushed to a local cluster, it can only update that local cluster. To move the prefix to another local cluster, it must first be reclaimed.
Regional Web UI

The ensuing page identifies the data to push, how to synchronize it with the local cluster, and the cluster or clusters to which to push it. The data synchronization modes are:

- **Ensure**—Ensures that the local cluster has new data without affecting any existing data.
- **Replace** (preset value)—Replaces data without affecting other objects unique to the local cluster.
- **Exact**—Available for “push all” operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the local cluster.

Choose the destination cluster or clusters in the Available field and move it or them to the Selected field.

Tip

The synchronization mode and cluster choice settings are persistent for the duration of the current login session, so that they are in effect each time you access this page, unless you change them.

After making these choices, click **Push Data to Clusters**. This opens the View Push Reservations Data Report page. Click **OK** on this page.

You can also pull the replica address space on the List/Add DHCP v6 Reservations page, and opt whether to omit reservations when doing so. You should use this option only to reduce processing time when you are sure that there are no pending changes to reservations to merge. To omit reservations for the pull, check the **Omit Reservations?** check box, then click **Pull Data**.

See the “Managing DHCPv6 Addresses” section on page 27-1.
Maintaining Servers and Databases

This chapter explains how to administer and control your local and regional server operations.

Related Topics

Managing Servers, page 7-1
Scheduling Recurring Tasks, page 7-3
Logging Server Events, page 7-4
Log Files, page 7-6
View Change Log, page 7-7
Dynamic Update on Server Log Settings, page 7-8
Monitoring and Reporting Server Status, page 7-9
Running Data Consistency Rules, page 7-21
Troubleshooting, page 7-23

Managing Servers

If you are assigned the server-management subrole of the ccm-admin role, you can manage the Cisco Prime IP Express servers as follows:

- **Start**—Load the database and start the server.
- **Stop**—Stop the server.
- **Reload**—Stop and restart the server. (Note that you do not need to reload the server for all RR updates, even protected RR updates. For details, see Chapter 29, “Configuring DNS Update”.)
- **Check statistics**—See the “Displaying Statistics” section on page 7-11.
- **View logs**—See the “Searching the Logs” section on page 7-5.
- **Manage interfaces**—See the specific protocol pages for how to manage server interfaces.

Starting and stopping a server is self-explanatory. When you reload the server, Cisco Prime IP Express performs three steps—stops the server, loads configuration data, and restarts the server. Only after you reload the server does it use your changes to the configuration.

Note

The CDNS, DNS, DHCP, and SNMP servers are enabled by default to start on reboot. You can change this using [server] type enable or disable start-on-reboot in the CLI.
Local Basic or Advanced and Regional Web UI

You can manage the protocol servers in the following ways depending on if you are a:

- **Local or regional cluster administrator**—Choose **Manage Servers** from the **Operate** menu to open the Manage Servers page.

  The local and regional cluster web UI access to server administration is identical, even though the available functions are different. As a regional administrator, you can check the state and health of the regional CCM server, server agent. However, you cannot stop, start, reload, or view statistics, logs, or interfaces for them.

  At the local cluster, to manage the DHCP, DNS, CDNS, or SNMP servers, select the server in the Manage Servers pane and do any of the following:
  - Click the Statistics tab to view statistics for the server. (See the “Displaying Statistics” section on page 7-11.)
  - Click the Log tab in the View Log column to view the log messages for the server. (See the “Logging Server Events” section on page 7-4.)
  - Click the Start Server button to start the server.
  - Click the Stop Server button to stop the server.
  - Click the Restart Server button to reload the server.

- **Local cluster DNS administrator**—Choose **DNS Server** from the **Deploy** menu to open the Manage DNS Authoritative Server page.

  Along with the Statistics, Startup Logs, Logs, HA DNS Server Status, Start Server, Stop Server, and Restart Server functions, you can also perform other functions when you click the Commands button to open the DNS Commands dialog box.

  The server command functions are:
  - **Forcing all zone transfers** (see the “Enabling Zone Transfers” section on page 14-14)—Click the Run icon. This is the equivalent of `dns forceXfer secondary` in the CLI.
  - **Scavenging all zones** (see the “Scavenging Dynamic Records” section on page 29-19)—Click the Run icon. This is the equivalent of `dns scavenge` in the CLI.

- **Local cluster Caching DNS server**—Choose **CDNS Server** from the **Deploy** menu to open the Manage DNS Caching Server page.

  Along with the Statistics, Startup Logs, Logs, Start Server, Stop Server, and Restart Server functions, you can also perform other functions when you click Commands button to open the CDNS Commands dialog box.

  In Advanced and Expert modes, you can flush Caching CDNS cache and flush the resource records. Click the Commands button to execute the commands.

- **Local cluster DHCP administrator**—Click **DHCP Server** from the **Deploy** menu to open the Manage DHCP Server page.

  Along with the Statistics, Startup Logs, Logs, Start Server, Stop Server, and Restart Server functions, you can also perform other functions when you click the Commands button to open the DHCP Server Commands dialog box.

  This page provides the Get Leases with Limitation ID feature, to find clients that are associated through a common limitation identifier (see the “Administering Option 82 Limitation” section on page 25-18). Enter at least the IP address of the currently active lease in the IP Address field, then
click the Run icon. You can also enter the limitation ID itself in the form nn:nn:nn or as a string ("nnnn"), in which case the IP address becomes the network in which to search. This function is the equivalent of `dhcp limitationList ipaddress limitation-id show` in the CLI.

**CLI Commands**

In the CLI, the regional cluster allows CCM server management only:

- To start the server, use `server type start` (or simply `type start`; for example, `dhcp start`).
- To stop the server, use `server type stop` (or simply `type stop`; for example, `dhcp stop`). If stopping the server, it is advisable to save it first using the `save` command.
- To reload the server, use `server type reload` (or simply `type reload`; for example, `dhcp reload`). Cisco Prime IP Express stops the server you chose, loads the configuration data, and then restarts the server.
- To set or show attributes for the server, use `[server] type set attribute=value` or `[server] type show.
  For example:
  ```
  nrcmd> ccm set ipaddr=192.168.50.10
  ```

**Scheduling Recurring Tasks**

In Basic and Advanced user mode in the local cluster web UI, you can schedule a number of recurring tasks. These tasks are:

- Reloading the DHCP server.
- Reloading the DNS server.
- Synchronizing DHCP failover server pairs:
  - If in staged dhcp edit mode, reload the main DHCP server.
  - Synchronize the failover configuration to the backup DHCP server.
  - If in staged dhcp edit mode, reload the backup DHCP server.
- Synchronizing High-Availability (HA) DNS server pairs:
  - If in staged dhcp edit mode, reload the main DNS server.
  - Synchronize the HA DNS configuration to the backup DNS server.
  - If in staged dhcp edit mode, reload the backup DNS server.
- Synchronizing zone distribution maps:
  - If in staged dhcp edit mode, reload the main DNS server.
  - If in staged dhcp edit mode, reload the backup HA DNS server.
  - Synchronize the zone distribution maps.
  - If in staged dhcp edit mode, reload the secondary DNS server or servers.
Logging Server Events

When you start Cisco Prime IP Express, it automatically starts logging Cisco Prime IP Express system activity. Cisco Prime IP Express maintains all the logs by default on:

- **Windows**—`install-path\logs`
- **Linux**—`install-path/logs` (to view these logs, use the `tail -f` command)

**Tip**

To avoid filling up the Windows Event Viewer and preventing Cisco Prime IP Express from running, in the Event Log Settings, check the **Overwrite Events as Needed** box. If the events do fill up, save them to a file, then clear them from the Event Log.

---

**Note**

The DNS server startup and background loading slows down when HA is enabled before the HA DNS server communicates to its partner. You need to allow the HA DNS server to communicate with its partner before reloading or restarting the DNS server.

---

Local Basic or Advanced Web UI

To set up one or more of these recurring server tasks:

**Step 1**
From the **Operate** menu, choose **Schedule Tasks** to open the List/Add Scheduled Tasks page.

**Step 2**
Click the **Add Scheduled Tasks** icon in the Scheduled Tasks pane on the left to open the Add Scheduled Task page.

**Step 3**
Enter values in the appropriate fields:

a. Name of the scheduled task. This can be any identifying text string.

b. Pull down from the available list of task types, which are:

- **dhcp-reload**—Reloads the DHCP server
- **dns-reload**—Reloads the DNS server
- **cdns-reload**—Reloads the Caching DNS server
- **sync-dhcp-pair**—Synchronizes the DHCP failover server pair
- **sync-dns-pair**—Synchronizes the HA DNS failover server pair
- **sync-zd-map**—Synchronizes zone distribution maps
- **sync-dns-update-map**—Synchronizes DNS update maps

c. Indicate the time interval for the scheduled task, such as 60m or 4w2d.

**Step 4**
Click **Add Scheduled Task**.

**Step 5**
If you click the name of the task on the List/Add Scheduled Tasks page, on the Edit Scheduled Task page you can view (in the Task Status section) the last status or the list of last errors (if any) that occurred during the task execution. Click **Run Now** to run the task immediately.
Local Basic or Advanced and Regional Web UI

Server logging is available in the web UI when you open the Manage Servers page for a server (see the “Managing Servers” section on page 7-1), then click the Log icon in the View Log column for the server. This opens the Log for Server page. The log is in chronological order with the page with the latest entries shown first. If you need to see earlier entries, click the left arrow at the top or bottom of the page.

Related Topics

- Searching the Logs, page 7-5
- Logging Format and Settings, page 7-5

Searching the Logs

The web UI provides a convenient way to search for entries in the activity and startup log files. You can locate specific message text, log message IDs, and message timestamps using a regular expression string entry. When you click the Log icon in the View Log or View Startup Log column on the Manage Servers page (or one of the specific server pages), this opens a Log for Server page. In the text field next to the Search icon at the top or bottom of the page, enter the search string in the regular expression syntax. (For example, entering name? searches for occurrences of the string name in the log file.) Click the Search icon to view the results of log search.

Click the name of the log message, which opens the Log for Server page with the full message text. To view the full message text, click the name of the log message. Change between Table and Text view by clicking the Log icon. Click Close on the Log Search Result page to close the browser window.

Logging Format and Settings

The server log entries include the following categories:

- **Activity**—Logs the activity of your servers.
- **Info**—Logs standard operations of the servers, such as starting up and shutting down.
- **Warning**—Logs warnings, such as invalid packets, user miscommunication, or an error in a script while processing a request.
- **Error**—Logs events that prevent the server from operating properly, such as out of memory, unable to acquire resources, or errors in configuration.

**Note**

Warnings and errors go to the Event Viewer on Windows (see the Tip on page 7-4). For a description of the log messages for each server module, see the install-path/docs/msgid/MessageIdIndex.html file.
Local Basic or Advanced and Regional Web UI

You can affect which events to log. For example, to set the logging for the local cluster DNS and DHCP server:

- **DNS**—From the Deploy > DNS menu, choose DNS Server to open the Manage DNS Server page. Click the name of the server to open the Edit DNS Server page. Expand the Log Settings section to view the log settings. Make changes to the attributes as desired, click Save, then reload the server. (See Table 17-2 on page 17-10 for the log settings to maximize DNS server performance.)

- **DHCP**—From the Deploy > DHCP menu, choose DHCP Server to open the Manage DHCP Server page. Click the name of the server to open the Edit DHCP Server page. Expand the Log Settings section to view the log settings. Make changes to the attributes as desired, click Save, then reload the server. (See Table 24-3 on page 24-16 for the log settings to maximize DHCP server performance.)

**CLI Commands**

Use `dns set log-settings`, and `dhcp set log-settings` for the respective servers.

## Log Files

Table 7-1 describes the Cisco Prime IP Express log files in the `install-path/logs` directory.

**Table 7-1 Log Files in .../logs Directory**

<table>
<thead>
<tr>
<th>Component</th>
<th>File in /logs Directory</th>
<th>Local/Regional</th>
<th>Logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>install_cnr_log</td>
<td>Both</td>
<td>Installation process</td>
</tr>
<tr>
<td>Upgrade</td>
<td>ccm_upgrade_status_log</td>
<td>Both</td>
<td>Upgrade process</td>
</tr>
<tr>
<td></td>
<td>dns_upgrade_status_log</td>
<td>Local</td>
<td>Upgrade process</td>
</tr>
<tr>
<td>Server agent</td>
<td>agent_server_1_log</td>
<td>Both</td>
<td>Server agent starts and stops</td>
</tr>
<tr>
<td>Port check</td>
<td>checkports_log</td>
<td>Both</td>
<td>Network ports</td>
</tr>
<tr>
<td>DNS server</td>
<td>name_dns_1_log</td>
<td>Local</td>
<td>DNS activity</td>
</tr>
<tr>
<td></td>
<td>dns_startup_log</td>
<td>Local</td>
<td>DNS startup activity</td>
</tr>
<tr>
<td>CDNS server</td>
<td>cdns_log</td>
<td>Local</td>
<td>CDNS activity</td>
</tr>
<tr>
<td></td>
<td>cdns_startup_log</td>
<td>Local</td>
<td>CDNS startup activity</td>
</tr>
<tr>
<td>DHCP server</td>
<td>name_dhcp_1_log</td>
<td>Local</td>
<td>DHCP activity</td>
</tr>
<tr>
<td></td>
<td>dhcp_startup_log</td>
<td>Local</td>
<td>DHCP startup activity</td>
</tr>
<tr>
<td>SNMP server</td>
<td>cnrsnmp_log</td>
<td>Local</td>
<td>SNMP activity</td>
</tr>
<tr>
<td>CCM database</td>
<td>config_ccm_1_log</td>
<td>Both</td>
<td>CCM configuration, starts, stops</td>
</tr>
<tr>
<td></td>
<td>ccm_startup_log</td>
<td></td>
<td>CCM startup activity</td>
</tr>
<tr>
<td>Web UI</td>
<td>cnrwebui_log</td>
<td>Both</td>
<td>Web UI state</td>
</tr>
<tr>
<td>Tomcat/web UI</td>
<td>catalina.date.log.txt</td>
<td>Both</td>
<td>CCM database for Tomcat server and web UI (Because new files are created daily, periodically archive old log files.)</td>
</tr>
<tr>
<td></td>
<td>jsui_log.date.txt</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>cnrwebui_access_log.date.txt</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Each component can generate a number of log files, each with a preconfigured maximum size of 1 MB. The first log file name has the _log suffix. When this file reaches its maximum size, it gets the .01 version extension appended to its name and a new log file is created without the version extension. Each version extension is incremented by one for each new file created. When the files reach their configured maximum number, the oldest file is deleted and the next oldest assumes its name. The usual maximum number is four for the DNS, and DHCP servers.

Cisco Prime IP Express also has server_startup_log files. This applies to the CCM, DHCP, and DNS servers. These files log the start up and shut down phases of the server (the information is similar to the normal log file information). Server startup log files are useful in diagnosing problems that have been reported when the server was last started.

The number of these start-up logs is fixed at four for a server, and the size is fixed at one MB per server.

Note Some user commands can create User authentication entries in the Server Agent log because of separate connections to the cluster. Do not interpret these as a system security violation by another user.

CLI Commands

You can check the configured maximums for the DNS, and DHCP servers using [server] type serverLogs show in the CLI, which shows the maximum number (nlogs) and size (logsize) of these protocol server log files. You can adjust these parameters using [server] type serverLogs set nlogs=value and [server] type serverLogs set logsize=value. You cannot adjust these maximums for any of the other log files.

Note A change to the server logs will not take effect until you restart Cisco Prime IP Express.

View Change Log

In the web UI, you can view the change logs and tasks associated with configurations you make.

Local Basic and Advanced Web UI

From the Operate menu, choose Change Log. To view the change log, you must be assigned the database subrole of the ccm-admin or regional-admin role:

- The View Change Log page shows all the change logs, sorted by DBSN name. To get to the bottom of the list, click the right arrow at the bottom left of the page. Click the DBSN number of the change log entry to open a View Change Set page for it.

On the View Change Log page, you can filter the list, manually trim it, and save it to a file. You can filter the list by:

- Start and end dates
- Administrator who initiated the changes
- Configuration object class
- Specific object
- Object identifier (ID), in the format OID-00:00:00:00:00:00:00:00
- Server
Dynamic Update on Server Log Settings

The DHCP and the DNS servers register the changes on the server logs only during the server configuration, which happens during a reload. Reloading the servers is time consuming. Cisco Prime IP Express allows the DHCP and DNS servers to register the changes to log settings, without a reload.

Local Basic or Advanced Web UI

To dynamically update DHCP server log settings, do the following:

Step 1  From the **Deploy > DHCP** menu, choose **DHCP Server**. The Manage DHCP Server page appears.
Step 2  Click the name of the DHCP server in the left pane to open the Edit DHCP Server page.
Step 3  Modify the log settings as desired.
Step 4  Click **Save** at the bottom of the page. The new log settings are applied to the DHCP server. The Manage DHCP Server page is displayed with an updated page refresh time.

To dynamically update DNS server log settings, do the following:

Step 1  From the **Deploy > DNS** menu, choose **DNS Server**. This opens the Manage DNS Server page.
Step 2  Click the name of the DNS server in the left pane to open the Edit DNS Server page.
Step 3  Modify the log settings as desired.
Step 4  Click **Save** at the bottom of the page. The new log settings are applied to the DNS server. The Manage DNS Server page is displayed with an updated page refresh time.

**Note**
If the dhcp-edit-mode or dns-edit-mode is set to synchronous, and if the server running, the change in server log settings is communicated to the server.

To dynamically update the DHCP or DNS server log settings using the CLI, you must have the appropriate edit-mode set to synchronous. After changing the server log settings, use the `save` command to save the settings.

For example:

```
nrcmd>session set dhcp-edit-mode=synchronous
nrcmd>dhcp set log-settings=new-settings
nrcmd>save
```
Monitoring and Reporting Server Status

Monitoring the status of a server involves checking its:

- State
- Health
- Statistics
- Log messages
- Address usage
- Related servers (DNS and DHCP)
- Leases (DHCP)

Related Topics

Server States, page 7-9
Displaying Health, page 7-10
Displaying Statistics, page 7-11
Displaying IP Address Usage, page 7-17
Displaying Related Servers, page 7-18
Displaying Leases, page 7-20

Server States

All Cisco Prime IP Express protocol servers (DNS, DHCP, and SNMP) pass through a state machine consisting of the following states:

- **Loaded**—First step after the server agent starts the server (transitional).
- **Initialized**—Server was stopped or fails to configure.
- **Unconfigured**—Server is not operational because of a configuration failure (transitional).
- **Stopped**—Server was administratively stopped and is not running (transitional).
- **Running**—Server is running successfully.

The two essential states are initialized and running, because the server transitions through the states so quickly that the other states are essentially invisible. Normally, when the server agent starts the server, it tells the server to be up. The server process starts, sets its state to loaded, then moves up to running. If you stop the server, it walks down the states to initialized, and if you restart, it moves up to running again. If it fails to configure for some reason, it drops back to initialized, as if you had stopped it.

There is also an exiting state that the server is in very briefly when the process is exiting. The user interface can also consider the server to be disabled, but this rarely occurs and only when there is no server process at all (the server agent was told not to start one).
Displaying Health

You can display aspects of the health of a server, or how well it is running. The following items can decrement the server health, so you should monitor their status periodically. For the:

- Server agent (local and regional clusters)
- CCM server (local and regional clusters)
- DNS server (local cluster):
  - Configuration errors
  - Memory
  - Disk space usage
  - Inability to contact its root servers
- Caching DNS server (local cluster)
- DHCP server (local cluster):
  - Configuration errors
  - Memory
  - Disk space usage
  - Packet caching low
  - Options not fitting in the stated packet limit
  - No more leases available

Server Health Status

The server health status varies from the value 0 to 10. The value 0 means the server is not running and 10 means the server is running. Some of the servers report only 0 or 10, and not anything in between. When a server reports a value from 1 to 9, it means that it detected conditions that indicate possible problems. It has nothing to do with the actual performance of the server. So, if the health of the server is a value from 1 to 9, the server log files need to be reviewed to see what errors were logged.

Note

Depending on the level of activity and the size and number of log files, the condition that reduced the server health might not be visible in the log files. It is important to review the log files, but the servers do not log all the conditions that reduce the server health.

The following conditions can reduce the DHCP server health:

- Configuration errors (occurs when the server is getting started or restarting)
- When the server detects out-of-memory conditions
- When packet receive failures occur
- When packets are dropped because the server is out of request or response buffers
- When the server is unable to construct a response packet

Tip

Health values range from 0 (the server is not running) to 10 (the highest level of health). It is recommended that the health status can be ignored, with the understanding that zero means server is not running and greater than zero means server is running. On Linux, you can run the `cnr_status` command,
in the install-path/usrbin/ directory, to see if your local cluster server is running. For more information on how to check whether the local cluster server is running, see the Cisco Prime IP Express Installation Guide.

Local Basic or Advanced and Regional Web UI

From the Operate menu, select Manage Servers. Check the Manage Servers page for the state and health of each server.

CLI Commands

Use [server] type getHealth. The number 10 indicates the highest level of health, 0 that the server is not running.

Displaying Statistics

To display server statistics, the server must be running.

Local Basic or Advanced and Regional Web UI

Go to the Manage Servers page, click the name of the server in the left pane, then click the Statistics tab, if available. On the Server Statistics page, click the name of the attribute to get popup help.

The DHCP, DNS, and CDNS statistics are each divided into two groups of statistics. The first group is for total statistics and the second group is for sample statistics. The total statistics are accumulated over time. The sample statistics occur during a configurable sample interval. The names of the two categories vary per server and per user interface, and are identified in Table 7-2.

<table>
<thead>
<tr>
<th>Server</th>
<th>User Interface</th>
<th>Total Statistics (Command)</th>
<th>Sample Statistics (Command)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP</td>
<td>Web UI</td>
<td>Total Statistics</td>
<td>Activity Summary</td>
</tr>
<tr>
<td></td>
<td>CLI</td>
<td>Total Counters since the start of the last DHCP server process (dhcp getStats)</td>
<td>Sampled counters since the last sample interval (dhcp getStats sample)</td>
</tr>
<tr>
<td>DNS</td>
<td>Web UI</td>
<td>Total Statistics</td>
<td>Sample Statistics</td>
</tr>
<tr>
<td></td>
<td>CLI</td>
<td>Total Counters since the start of the last server process (dns getStats)</td>
<td>Sampled counters since the last sample interval (dns getStats sample)</td>
</tr>
<tr>
<td>CDNS</td>
<td>Web UI</td>
<td>Total Statistics</td>
<td>Sample Statistics</td>
</tr>
<tr>
<td></td>
<td>CLI</td>
<td>Total Counters since the start of the last server process (cdns getStats total)</td>
<td>Sampled counters since the last sample interval (cdns getStats sample)</td>
</tr>
</tbody>
</table>
To set up the sample counters, you must activate either the `collect-sample-counters` attribute for the server or a `log-settings` attribute value called `activity-summary`. You can also set a `log-settings` value for the sample interval for each server, which is preset to 5 minutes. The `collect-sample-counters` attribute is preset to true for the DNS server, but is preset to false for the DHCP server. For example, to enable the sample counters and set the interval for DHCP, set the following attributes for the DHCP server:

- Enable `collect-sample-counters` (`dhcp enable collect-sample-counters`)
- Set `log-settings` for `activity-summary` (`dhcp set log-settings=activity-summary`)
- Set `activity-summary-interval` to 5m (`dhcp set activity-summary-interval=5m`)

**CLI Commands**

In the CLI, if you use `[server] type getStats`, the statistics are encoded in curly braces followed by sets of digits, as described in Table 7-3 on page 7-13 for DNS, Table 7-5 on page 7-17 for DHCP. Reset the counters and total statistic by using `dhcp resetStats`, `dns resetStats`, or `cdns resetStats`.

**Related Topics**

- DNS Statistics, page 7-12
- CDNS Statistics, page 7-14
- DHCP Statistics, page 7-16

**DNS Statistics**

The DNS server statistics in the web UI appear on the DNS Server Statistics page, click on the statistic’s name to read its description. You can refresh the DNS Server Statistics.

The DNS server statistics that you can view are:

- **Attribute**—Displays server statistics such as server identifier, recursive service, process uptime, time since reset, and so on.

**Total Statistics**

- Performance Statistics—Displays the total statistics of the DNS Server performance.
- Query Statistics—Displays the total statistics of the queries.
- Security Statistics—Displays the total statistics of the security.
- Error Statistics—Displays the total statistics of the errors.
- Max Counter Statistics—Displays the total statistics of the maximum number of concurrent threads, RRs, DNS update latency, concurrent packets, and so on.
- HA Statistics—Displays the total statistics of the HA DNS Server.
- IPv6 Statistics—Displays the total statistics of the IPv6 packets received and sent.

**Sample Statistics**

- Performance Statistics—Displays the sample statistics about the DNS Server performance.
- Query Statistics—Displays the sample statistics about the queries.
- Security Statistics—Displays the sample statistics about the security.
- Error Statistics—Displays the sample statistics about the errors.
- HA Statistics—Displays the sample statistics about the HA DNS Server.
- IPv6 Statistics—Displays the sample statistics about the IPv6 packets received and sent.

**Note**

To get the most recent data, click the **Refresh Server Statistics** icon at the top left corner of the page.

The CLI **dns getStats** command has the following options:

```
dns getStats [performance | query | errors | security | maxcounters | ha | ipv6 | all] {total | sample}
```

The **dns getStats all** command is the most commonly used.

```
nrcmd> dns getStats
nrcmd> dns getStats
100 Ok
(1) 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

**Table 7-3 DNS Statistics**

<table>
<thead>
<tr>
<th>Digit</th>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>id</td>
<td>Implementation ID (release and build information).</td>
</tr>
<tr>
<td>2</td>
<td>config-recurs</td>
<td>Recursion services—(1) available, (2) restricted, (3) unavailable.</td>
</tr>
<tr>
<td>3</td>
<td>config-up-time</td>
<td>Time (in seconds) elapsed since the last server startup.</td>
</tr>
<tr>
<td>4</td>
<td>config-reset-time</td>
<td>Time (in seconds) elapsed since the last server reset (restart).</td>
</tr>
<tr>
<td>5</td>
<td>config-reset</td>
<td>Status or action to reinitializes any name server state—If using the (2) reset action, reinitializes any persistent name server state; the following are read-only statuses: (1) other—server in some unknown state, (3) initializing, or (4) running.</td>
</tr>
<tr>
<td>6</td>
<td>counter-auth-ans</td>
<td>Number of queries answered authoritatively.</td>
</tr>
<tr>
<td>7</td>
<td>counter-auth-no-names</td>
<td>Number of queries returning authoritative no such name responses.</td>
</tr>
<tr>
<td>8</td>
<td>counter-auth-no-data-resps</td>
<td>Number of queries returning authoritative no such data (empty answer) responses. (Deprecated statistics)</td>
</tr>
<tr>
<td>9</td>
<td>counter-non-auth-datas</td>
<td>Number of queries answered nonauthoritatively (cached). (Deprecated statistics)</td>
</tr>
<tr>
<td>10</td>
<td>counter-non-auth-no-datas</td>
<td>Number of queries answered nonauthoritatively with no data.</td>
</tr>
<tr>
<td>11</td>
<td>counter-referrals</td>
<td>Number of queries forwarded to other servers.</td>
</tr>
<tr>
<td>12</td>
<td>counter-errors</td>
<td>Number of responses answered with errors (RCODE values other than 0 or 3).</td>
</tr>
<tr>
<td>13</td>
<td>counter-rel-names</td>
<td>Number of requests received for names of only one label (relative names).</td>
</tr>
<tr>
<td>14</td>
<td>counter-req-refusals</td>
<td>Number of refused queries.</td>
</tr>
<tr>
<td>15</td>
<td>counter-req-unparses</td>
<td>Number of unparseable requests.</td>
</tr>
<tr>
<td>16</td>
<td>counter-other-errors</td>
<td>Number of aborted requests due to other errors.</td>
</tr>
<tr>
<td>17</td>
<td>total-zones</td>
<td>Total number of configured zones.</td>
</tr>
</tbody>
</table>
CDNS Statistics

The CDNS server statistics in the web UI appear on the DNS Caching Server Statistics page, click on the name of the statistics to read its description. You can refresh the CDNS Server Statistics.

**Table 7-4 CDNS Statistics**

<table>
<thead>
<tr>
<th>Digit</th>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1}</td>
<td>name</td>
<td>Name identifying the DNS Caching Server.</td>
</tr>
<tr>
<td>2</td>
<td>time-current</td>
<td>The current time given by the CDNS Server.</td>
</tr>
<tr>
<td>3</td>
<td>time-up</td>
<td>The amount of time the server has been up and running.</td>
</tr>
<tr>
<td>4</td>
<td>time-elapsed</td>
<td>The elapsed since last statistics poll.</td>
</tr>
<tr>
<td>5</td>
<td>queries-total</td>
<td>Total number of queries received by the CDNS Server.</td>
</tr>
<tr>
<td>6</td>
<td>queries-over-tcp</td>
<td>Total number of queries received over TCP by the CDNS Server.</td>
</tr>
<tr>
<td>7</td>
<td>queries-over-ipv6</td>
<td>Total number of queries received over TCP by the CDNS Server.</td>
</tr>
<tr>
<td>8</td>
<td>queries-with-edns</td>
<td>Number of queries with EDNS OPT RR present.</td>
</tr>
<tr>
<td>9</td>
<td>queries-with-edns-do</td>
<td>Number of queries with EDNS OPT RR with DO (DNSSEC OK) bit set.</td>
</tr>
<tr>
<td>10</td>
<td>queries-type-A</td>
<td>Number of A queries received.</td>
</tr>
<tr>
<td>11</td>
<td>queries-type-AAAA</td>
<td>Number of AAAA queries received.</td>
</tr>
<tr>
<td>12</td>
<td>queries-type-CNAME</td>
<td>Number of CNAME queries received.</td>
</tr>
<tr>
<td>13</td>
<td>queries-type-PTR</td>
<td>Number of PTR queries received.</td>
</tr>
<tr>
<td>14</td>
<td>queries-type-NS</td>
<td>Number of NS queries received.</td>
</tr>
<tr>
<td>15</td>
<td>queries-type-SOA</td>
<td>Number of SOA queries received.</td>
</tr>
<tr>
<td>16</td>
<td>queries-type-MX</td>
<td>Number of MX queries received.</td>
</tr>
<tr>
<td>17</td>
<td>queries-type-DS</td>
<td>Number of DS queries received.</td>
</tr>
<tr>
<td>18</td>
<td>queries-type-DNSKEY</td>
<td>Number of DNSKEY queries received.</td>
</tr>
<tr>
<td>19</td>
<td>queries-type-RRSIG</td>
<td>Number of RRSIG queries received.</td>
</tr>
<tr>
<td>21</td>
<td>queries-type-NSEC</td>
<td>Number of NSEC queries received.</td>
</tr>
<tr>
<td>22</td>
<td>queries-type-NSEC3</td>
<td>Number of NSEC3 queries received.</td>
</tr>
<tr>
<td>23</td>
<td>queries-type-other</td>
<td>Number of queries received of type 256+.</td>
</tr>
<tr>
<td>24</td>
<td>queries-with-flag-QR</td>
<td>Number of incoming queries with QR (query response) flag set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>These queries are dropped.</td>
</tr>
<tr>
<td>25</td>
<td>queries-with-flag-AA</td>
<td>Number of incoming queries with AA (auth answer) flag set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>These queries are dropped.</td>
</tr>
<tr>
<td>26</td>
<td>queries-with-flag-TC</td>
<td>Number of incoming queries with TC (truncation) flag set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>These queries are dropped.</td>
</tr>
<tr>
<td>27</td>
<td>queries-with-flag-RD</td>
<td>Number of incoming queries with RD (recursion desired) flag set.</td>
</tr>
<tr>
<td>28</td>
<td>queries-with-flag-RA</td>
<td>Number of incoming queries with RA (recursion available) flag set.</td>
</tr>
<tr>
<td>29</td>
<td>queries-with-flag-Z</td>
<td>Number of incoming queries with Z flag set.</td>
</tr>
<tr>
<td>30</td>
<td>queries-with-flag-AD</td>
<td>Number of incoming queries with AD flag set.</td>
</tr>
<tr>
<td>31</td>
<td>queries-with-flag-CD</td>
<td>Number of incoming queries with CD flag set.</td>
</tr>
</tbody>
</table>
### Table 7-4 CDNS Statistics (continued)

<table>
<thead>
<tr>
<th>Digit</th>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>queries-failing-acl</td>
<td>Number of queries being dropped or refused due to ACL failures.</td>
</tr>
<tr>
<td>33</td>
<td>cache-hits</td>
<td>The total number of queries that were answered from cache.</td>
</tr>
<tr>
<td>34</td>
<td>cache-misses</td>
<td>The total number of queries that were not found in the cache.</td>
</tr>
<tr>
<td>35</td>
<td>cache-prefetches</td>
<td>Number of prefetches performed.</td>
</tr>
<tr>
<td>36</td>
<td>requestlist-total</td>
<td>The total number of queued requests waiting for recursive replies.</td>
</tr>
<tr>
<td>37</td>
<td>requestlist-total-user</td>
<td>The total number of queued user requests waiting for recursive replies.</td>
</tr>
<tr>
<td>38</td>
<td>requestlist-total-system</td>
<td>The total number of queued system requests waiting for recursive replies.</td>
</tr>
<tr>
<td>39</td>
<td>requestlist-total-average</td>
<td>The average number of requests on the request list.</td>
</tr>
<tr>
<td>40</td>
<td>requestlist-total-max</td>
<td>The maximum number of requests on the request list.</td>
</tr>
<tr>
<td>41</td>
<td>requestlist-total-overwritten</td>
<td>The number of requests on the request list that were overwritten by newer entries.</td>
</tr>
<tr>
<td>42</td>
<td>requestlist-total-exceeded</td>
<td>The number of requests dropped because the request list was full.</td>
</tr>
<tr>
<td>43</td>
<td>recursive-replies-total</td>
<td>The total number of recursive queries replies.</td>
</tr>
<tr>
<td>44</td>
<td>recursive-time-average</td>
<td>The average time to complete a recursive query.</td>
</tr>
<tr>
<td>45</td>
<td>recursive-time-median</td>
<td>The median time to complete a recursive query.</td>
</tr>
<tr>
<td>46</td>
<td>mem-process</td>
<td>An estimate of the memory in bytes of the CDNS process.</td>
</tr>
<tr>
<td>47</td>
<td>mem-cache</td>
<td>Memory in bytes of RRSets cache.</td>
</tr>
<tr>
<td>48</td>
<td>mem-query-cache</td>
<td>Memory in bytes of incoming query message cache.</td>
</tr>
<tr>
<td>49</td>
<td>mem-iterator</td>
<td>Memory in bytes used by the CDNS iterator module.</td>
</tr>
<tr>
<td>50</td>
<td>mem-validator</td>
<td>Memory in bytes used by the CDNS validator module.</td>
</tr>
<tr>
<td>51</td>
<td>answers-with-NOERROR</td>
<td>Number of answers from cache or recursion that result in rcode of NOERROR being returned to client.</td>
</tr>
<tr>
<td>52</td>
<td>answers-with-NXDOMAIN</td>
<td>Number of answers from cache or recursion that result in rcode of NXDOMAIN being returned to client.</td>
</tr>
<tr>
<td>53</td>
<td>answers-with-NODATA</td>
<td>Number of answers that result in pseudo rcode of NODATA being returned to client.</td>
</tr>
<tr>
<td>54</td>
<td>answers-with-other-errors</td>
<td>Number of answers that result in pseudo rcode of NODATA being returned to client.</td>
</tr>
<tr>
<td>55</td>
<td>answers-secure</td>
<td>Number of answers that correctly validated.</td>
</tr>
<tr>
<td>56</td>
<td>answers-unsecure</td>
<td>Number of answers that did not correctly validate.</td>
</tr>
<tr>
<td>57</td>
<td>answers-rrset-unsecure</td>
<td>Number of RRSets marked as bogus by the validator.</td>
</tr>
<tr>
<td>58</td>
<td>answers-unwanted</td>
<td>Number of replies that were unwanted or unsolicited. High values could indicate spoofing threat.</td>
</tr>
<tr>
<td>59</td>
<td>reset-time</td>
<td>Reports the most recent time the stats were reset (i.e. <code>cdns resetStats</code> in nrcmd).</td>
</tr>
</tbody>
</table>
Table 7-4  
CDNS Statistics (continued)

<table>
<thead>
<tr>
<th>Digit</th>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>sample-time</td>
<td>Reports the time the server collected the last set of sample statistics.</td>
</tr>
<tr>
<td>61</td>
<td>sample-interval</td>
<td>Reports the sample interval used by the server when collecting sample statistics.</td>
</tr>
</tbody>
</table>

**DHCP Statistics**

The DHCP server statistics in the web UI appear on the DHCP Server Statistics page, click on the statistic’s name to read its description.

The DHCP server statistics details are available for:

- **Attribute**—Displays the server statistics such as server start time, server reload time, server up time, and statistics reset time.
- **Total Statistics**—Displays the total statistics of the scopes, request buffers, response buffers, packets and so on.
- **Lease Counts (IPv4)**—Displays the sample statistics of the IPv4 lease counts such as active leases, configured leases, reserved leases, and reserved active leases.
- **Packets Received (IPv4)**—Displays the sample statistics of the IPv4 packets received.
- **Packets Sent (IPv4)**—Displays the sample statistics of the IPv4 packets sent.
- **Packets Failed (IPv4)**—Displays the statistics of the failed IPv4 packets.

The Additional Attributes are:

- **Failover Statistics**—Displays the statistics of the DHCP failover server.
- **IPv6 Statistics**—Displays the statistics of the IPv6 prefixes configured, timed-out IPv6 offer packets and so on.
- **Lease Counts (IPv6)**—Displays the statistics of the IPv6 lease counts of active leases, configured leases, reserved leases, and reserved active leases.
- **Packets Received (IPv6)**—Displays the statistics of the IPv6 packets received.
- **Packets Sent (IPv6)**—Displays the statistics of the IPv6 packets sent.
- **Packets Failed (IPv6)**—Displays the statistics of the failed IPv6 packets.

Additional Attributes also includes Top Utilized Aggregations.

The CLI `dhcp getStats` command has the following options:

```
dhcp getStats [[all | server [,] failover [,] dhcpv6] [total | sample]
```

The `dhcp getStats all` command is the most commonly used.

```
nrcmd> dhcp getStats
```

**Note**

To get the most recent data, click the Refresh Server Statistics icon at the top left of the page.

The CLI `dhcp getStats` command has the following options:

```
dhcp getStats [[all | server [,] failover [,] dhcpv6] [total | sample]
```
The \texttt{dhcp getStats all} command is the most commonly used. The \texttt{dhcp getStats} command without this option returns the statistics in a single line of positional values in the following format (Table 7-5 shows how to read these values):

```
100 Ok
(1) 2 3 4 5 6 7 8
```

<table>
<thead>
<tr>
<th>Digit</th>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>start-time-str</td>
<td>Date and time of last server reload, as a text string.</td>
</tr>
<tr>
<td>2</td>
<td>total-discovers</td>
<td>Number of DISCOVER packets received.</td>
</tr>
<tr>
<td>3</td>
<td>total-requests</td>
<td>Number of REQUEST packets received.</td>
</tr>
<tr>
<td>4</td>
<td>total-releases</td>
<td>Number of RELEASED packets received.</td>
</tr>
<tr>
<td>5</td>
<td>total-offers</td>
<td>Number of OFFER packets sent.</td>
</tr>
<tr>
<td>6</td>
<td>total-acks</td>
<td>Number of acknowledgement (ACK) packets sent.</td>
</tr>
<tr>
<td>7</td>
<td>total-naks</td>
<td>Number of negative acknowledgement (NAK) packets sent.</td>
</tr>
<tr>
<td>8</td>
<td>total-declines</td>
<td>Number of DECLINE packets received.</td>
</tr>
</tbody>
</table>

### Displaying IP Address Usage

Displaying IP address usage gives an overview of how clients are currently assigned addresses.

**Local Advanced and Regional Web UI**

You can look at the local or regional cluster address space, or generate a lease history report at the regional cluster, to determine IP address usage. These functions are available in both web UIs in the Design > DHCPv4 menu, if you have address space privileges at the local or regional cluster.

You can determine the current address space utilization by clicking the View icon in the Current Usage column for the unified address space, address block, and subnet (see the “Viewing Address Utilization for Address Blocks, Subnets, and Scopes” section on page 9-10). You can also get the most current IP address utilization by querying the lease history (see the “Querying Leases” section on page 23-28). In the latter case, the regional CCM server references the appropriate DHCP server directly. To ensure this subnet-to-server mapping, you must update the regional address space view so that it is consistent with the relevant local cluster. Do this by pulling the replica address space, or reclaming the subnet to push to the DHCP server (see the “Reclaiming Subnets” section on page 9-9). Also ensure that the particular DHCP server is running.
You can generate an IP address usage report using the `report` command. The command has the following syntax:

```
report [column-separator=string | dhcpv4 | dhcp-only | dhcpv6 | file=outputfile | vpn=name
```

The column-separator specifies the character string that separates the report columns (the preset value is the space character). If you want to include more than one space, precede them with the backslash (\) escape character (enclosed in quotation marks). You can specify DHCPv4 or DHCPv6 addresses (`dhcp-only` is the same as `dhcpv4`). Not specifying the VPN returns the addresses in the current VPN only.

### Displaying Related Servers

Cisco Prime IP Express displays the relationship among servers in a DNS zone distribution or a DHCP failover configuration. In the web UI, you can view a related servers page when you click the **Related Servers** icon on various pages. You can use the display of related servers to diagnose and monitor misconfigured or unreachable servers.

### Related Topics

- Monitoring Remote Servers Using Persistent Events, page 7-18
- DNS Zone Distribution Servers, page 7-19
- DHCP Failover Servers, page 7-20

### Monitoring Remote Servers Using Persistent Events

To service clients that require updates to DNS and LDAP related servers, the DHCP server uses a persistent event algorithm to ensure updates to related servers when a related server is temporarily unavailable. In addition, the algorithm prevents a misconfigured or offline DNS server from using up all the available update resources.

At startup, the DHCP server calculates the number of related servers in the configuration that require persistent events. A preconfigured Maximum Pending Events attribute (an Expert mode attribute that specifies the number of in-memory events that is preset to 40,000) is divided by the number of servers to obtain a limit on the number of events permitted for each remote server. This calculation covers related DNS and LDAP servers (DHCP failover does not use persistent storage for events). The DHCP server uses this calculation to issue log messages and take the action described in Table 7-6 on page 7-18. The table shows a hypothetical case of a DHCP server with four related DNS servers each having a limit of 10K events.

<table>
<thead>
<tr>
<th>Event Reached</th>
<th>DHCP Server Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>50% of the calculated per-server limit (Maximum Pending Events value divided by the number of total related servers); for example, 5K events on a related server out of a total of 40K maximum pending events</td>
<td>Issues an INFO log message every 2 minutes, as long as the limits are exceeded: The queue of events for the name remote server at address has x events, and has reached the info limit of y/2 events out of an upper limit of y events per remote server. The remote server may be misconfigured, inoperable, or unreachable.</td>
</tr>
</tbody>
</table>
Chapter 7  Maintaining Servers and Databases

Monitoring and Reporting Server Status

Table 7-6  Persistent Event Algorithm (continued)

<table>
<thead>
<tr>
<th>Event Reached</th>
<th>DHCP Server Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% of the calculated per-server limit and less than 50% of the Maximum</td>
<td>Issues a WARNING log message every 2 minutes, as long as the limits are exceeded: The queue of events for the name remote server at address has x events, has exceeded the limit of y events per remote server, but is below the limit of z total events in memory. The remote server may be misconfigured, inoperative, or unreachable.</td>
</tr>
<tr>
<td>Pending Events value; for example, 10K events on a related server, with fewer</td>
<td>10K total maximum pending events</td>
</tr>
<tr>
<td>than 10K total maximum pending events</td>
<td></td>
</tr>
<tr>
<td>100% of the calculated per-server limit and 50% or more of the Maximum</td>
<td>Issues an ERROR log message every 2 minutes, as long as the limits are exceeded: The queue of events for the name remote server at address has x events, and has grown so large that the server cannot continue to queue new events to the remote server. The limit of y events per remote server and z/2 total events in memory has been reached. This and future updates to this server will be dropped. The current eventID n is being dropped.</td>
</tr>
<tr>
<td>Pending Events value; for example, 10K events on a related server, with 20K</td>
<td>20K total maximum pending events</td>
</tr>
<tr>
<td>total maximum pending events</td>
<td></td>
</tr>
<tr>
<td>100% of the Maximum Pending Events value; for example, 40K events across all</td>
<td>Issues an ERROR log message: The queue of pending events has grown so large that the server cannot continue to queue new events. The queue’s size is z, and the limit is z. The server drops all subsequent events with all related servers.</td>
</tr>
<tr>
<td>related servers</td>
<td></td>
</tr>
</tbody>
</table>

SNMP traps and DHCP server log messages also provide notification that a related server is unreachable.

DNS Zone Distribution Servers

A DNS zone distribution simplifies creating multiple zones that share the same secondary server attributes. You can view and set the primary and secondary DNS servers in a zone distribution.

Local Basic or Advanced Web UI

From the Deploy > DNS menu, click Zone Distribution. This opens the List Zone Distributions page. The local cluster allows only one zone distribution, the default. Click this zone distribution name to open the Edit Zone Distribution page, which shows the authoritative and secondary servers in the zone distribution.

Regional Web UI

From the Deploy > DNS menu, choose Zone Distributions. This opens the List/Add Zone Distributions page. The regional cluster allows creating more than one zone distribution. Click the zone distribution name to open the Edit Zone Distribution page, which shows the primary, authoritative, and secondary servers in the zone distribution.
CLI Commands

Create a zone distribution using `zone-dist name create primary-cluster`, then view it using `zone-dist list`. For example:

```
nrcmd> zone-dist distr-1 create Boston-cluster
nrcmd> zone-dist list
```

DHCP Failover Servers

Related servers in a DHCP failover pair relationship can show the following information:

- **Type**—Main or backup DHCP server.
- **Server name**—DNS name of the server.
- **IP address**—Server IP address in dotted octet format.
- **Requests**—Number of outstanding requests, or two dashes if not applicable.
- **Communication status**—OK or INTERRUPTED.
- **Cluster state**—Failover state of this DHCP server.
- **Partner state**—Failover state of its partner server.

For details on DHCP failover implementation, see Chapter 28, “Managing DHCP Failover.”

Local Basic or Advanced Web UI

From the **Deploy > DHCP** menu, choose **Failover Pairs**. The List/Add DHCP Failover Pairs page shows the main and backup servers in the failover relationship.

CLI Commands

Use `dhcp getRelatedServers` to display the connection status between the main and partner DHCP servers. If there are no related servers, the output is simply `100 Ok`.

Displaying Leases

After you create a scope, you can monitor lease activity and view lease attributes.

Local Basic or Advanced Web UI

From the **Design > DHCPv4** menu, choose **Scopes**; or from the **Design > DHCPv6** menu, choose **Prefixes** (in Advanced mode). On the List/Add DHCP Scopes or List/Add DHCPv6 Prefixes page, click the **View** icon in the Leases column to open the List DHCP Leases for Scope or List DHCP Leases for Prefix page.

Regional Web UI

From the **Operate > Reports > DHCPv4** menu or **Operate > Reports > DHCPv6** menu, choose **Lease History**. Set the query parameters, then click **Query Lease History**. (See the “Querying Leases” section on page 23-28.)
Running Data Consistency Rules

Using consistency rules, you can check data inconsistencies such as overlapping address ranges and subnets. You can set data consistency rules at the regional and local clusters.

The table on the List Consistency Rules page explains these rules. Check the check box next to the rule that you want to run.

The List Consistency Rules page includes functions to select all rules and clear selections. You can show the details for each of the rule violations as well as view the output. The rule selections you make are persistent during your user session.

Local Basic or Advanced and Regional Web UI

To run consistency rules, do the following:

**Step 1** From the Operate > Reports menu, choose Consistency Reports.

The List Consistency Rules page appears.

**Step 2** Check the check boxes for each of the listed consistency rules that you want to apply.

- To select all the rules, click the Select All Rules link.
- To clear all selections, click the Clear Selection link.

**Step 3** Click Run Rules.

The Consistency Rules Violations page appears. The rules are categorized by violation type.

- To show details for the violations, click the Show Details link.
- To show the output, click the page icon.

**Step 4** Click Return to return to the List Consistency Rules page.

CLI Tool

Use the cnr_rules consistency rules tool from the command line to check for database inconsistencies. You can also use this tool to capture the results of the rule in a text or XML file.

The cnr_rules tool is located at:

- **Windows**—..\bin\cnr_rules.bat
- **Linux**—../usrbin/cnr_rules

To run the cnr_rules tool, enter:

```bash
> cnr_rules -N username -P password [options]
```

- **–N username**—Authenticates using the specified username.
- **–P password**—Authenticates using the specified password.
- **options**—Describes the qualifying options for the tool, as described in Table 7-7. If you do not enter any options, the command usage appears.
### Table 7-7  `cnr_rules` Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>--list</td>
<td>Lists the available consistency rules.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> The list of available commands is tailored to the permissions of</td>
<td><code>cnr_rules -N admin -P changeme -list</code></td>
</tr>
<tr>
<td></td>
<td>the administrator specified in the value of the --N option.</td>
<td></td>
</tr>
<tr>
<td>--run [rule-match]</td>
<td>Run the available rules. Optionally, you can run a subset of the available</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rules by applying a case-insensitive rule-match string.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Tip</strong> To match a string containing spaces, enclose the string using</td>
<td><code>cnr_rules -N admin -P changeme -run *router interface*</code></td>
</tr>
<tr>
<td></td>
<td>double-quotations (&quot;). For example:</td>
<td></td>
</tr>
<tr>
<td>--details</td>
<td>Includes details of the database objects that violate consistency rules in</td>
<td><code>cnr_rules -N admin -P changeme -run DNS --details</code></td>
</tr>
<tr>
<td></td>
<td>the results.</td>
<td></td>
</tr>
<tr>
<td>--xml</td>
<td>Generates rule results in an XML file.</td>
<td><code>cnr_rules -N admin -P changeme -run --xml</code></td>
</tr>
<tr>
<td>--path classpath</td>
<td>Changes the Java classpath that is searched to locate the available</td>
<td></td>
</tr>
<tr>
<td></td>
<td>consistency rules (optional).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In order to run a new, custom consistency rule, you can use this option.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You must get the support of a support engineer to do this.</td>
<td></td>
</tr>
</tbody>
</table>

You can redirect the output of any of these preceding commands to another file. Use the following syntax to capture the rule results in a:

- **Text file:**
  ```
  > cnr_rules -N username -P password -run --details > filename.txt
  ```

- **XML file:**
  ```
  > cnr_rules -N username -P password -run --xml > filename.xml
  ```
Troubleshooting

The following sections describe troubleshooting the configuration and the DNS, and DHCP servers.

Related Topics

Immediate Troubleshooting Actions, page 7-23
Modifying the cnr.conf File, page 7-23
Troubleshooting Server Failures, page 7-25
Linux Troubleshooting Tools, page 7-25
Using the TAC Tool, page 7-26

Immediate Troubleshooting Actions

When facing a problem, it is crucial not to cause further harm while isolating and fixing the initial problem. Here are things to do (or avoid doing) in particular:

- Have 512 MB or more of memory and 2.5 GB or more of a data partition.
- Do not reboot a cable modem termination system (CMTS).
- Enable DHCP failover.
- Do not reload, restart, or disrupt Cisco Prime IP Express with failover resynchronization in progress.

Modifying the cnr.conf File

Cisco Prime IP Express uses the cnr.conf file for basic configuration parameters. This file is normally located in the install-path/conf directory. Cisco Prime IP Express creates the file during installation and processes it line by line.

You can edit this file if configuration parameters change. Note that during normal operation, you would not want to change the values. However, certain conditions might require you to modify certain values, such as when you move the data files for disk space reasons.

Table 7-7 cnr_rules Options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--interactive</td>
<td>Runs the tool in an interactive session.</td>
</tr>
</tbody>
</table>

Example:

```
> cnr_rules -N admin -P changeme -run -interactive
RuleEngine [type ? for help] > ?
Commands:
  load <class>   // load the specified rule class
  run <rule-match> // run rules matching a string, or '*' for all
  list           // list rules by name
  xml            // toggle xml mode
  detail         // toggle detail mode (non-xml only)
  quit           // quit RuleEngine
```
The format of the `cnr.conf` file consists of parameter name-value pairs, one per line; for example, for a Windows local cluster installation:

```plaintext
cnr.rootdir=C:\Program Files\IP Express\Local
cnr.ccm-port=1234
cnr.cisco-gss-appliance-integration=n
cnr.datadir=C:\IPExpress\Local\data
cnr.java-home=C:\Program Files\Java\jre1.5.0_12
cnr.logdir=C:\IPExpress\Local\logs
cnr.https-port=8443
cnr.tempdir=C:\IPExpress\Local\temp
cnr.http-port=8080
cnr.ccm-mode=local
cnr.ccm-type=cnr
cnr.http-enabled=y
cnr.https-enabled=n
cnr.keystore-file=C:\
cnr.keystore-password=unset
cnr.backup-time=23:45
```

Directory paths must be in the native syntax for the operating system. The format allows the use of colons (:) in directory paths, but not as name-value pair separators; it does not allow line continuation or embedded unicode characters. Other modifications to the file might include the location of the log directory (see the “Log Files” section on page 7-6) or the time `cnr_shadow_backup` backups should occur (see the “Setting Automatic Backup Time” section on page 8-3).

In rare cases, you might want to modify the file; for example, to exclude certain data from daily backups due to capacity issues. To do this, you need to add the appropriate settings manually.

⚠️ **Caution**

We recommend that you use the default settings in this file. If you must change these settings, do so only in consultation with the Cisco Technical Assistance Center (TAC) or the Cisco Prime IP Express development team.

The following settings are supported:

- **cnr.backup-dest**—Specify the destination to place backed up databases. Defaults to `cnr.datadir` if not specified.
- **cnr.backup-dbs**—Provide a comma-separated list of the databases you want to backup. For a local cluster the default is `ccm, dhcp, dns, mcd`. For a regional cluster it is `ccm, lease6hist, leasehist, subnetutil, replica`.
- **cnr.backup-files**—Provide a comma-separated list of files and the complete path to the files that you want copied as part of the backup. Files are copied to `cnr.backup-dest`.
- **cnr.dbrecover-backup**—Specify whether to run db recover and db verify on a backed up Oracle Berkeley database. The default is true. This setting is used for daily backups only. Manual backups ignore this setting. Disabling the automatic operation means that you must run the operation manually, preferably on a separate machine, or at a time when the Cisco Prime IP Express servers are relatively idle.
- **cnr.daily-backup**—Specify whether to run the daily back up. The default is true.
Troubleshooting Server Failures

The server agent processes (nwreglo and nwregregion) normally detect server failures and restart the server. You can usually recover from the failure and the server is not likely to fail again immediately after restarting. On rare occasions, the source of the server failure prevents the server from successfully restarting, and the server fails again as soon as it restarts. In such instances, perform the following steps:

**Step 1** If the server takes a significantly long time to restart, stop and restart the server agent. On:

- **Windows**:
  
  ```
  net stop nwreglo or nwregregion
  net start nwreglo or nwregregion
  ```

- **Linux**:
  
  ```
  /etc/rc.d/init.d/nwreglo stop or nwregregion stop
  /etc/rc.d/init.d/nwreglo start
  ```

**Step 2** Keep a copy of all the log files. Log files are located in the `install-path/logs` directory on Linux, and the `install-path/logs` folder on Windows. The log files often contain useful information that can help isolate the cause of a server failure.

**Step 3** Use the TAC tool, as described in the “Using the TAC Tool” section on page 7-26, or save the core or user.dmp file, if one exists, depending on the operating system:

- **Windows**—The user.dmp file is located in the system directory, which varies depending on the Windows system. Search for this file and save a renamed copy.

- **Linux**—The core file is located in the `install-path`. Save a renamed copy of this file that Cisco Prime IP Express does not overwrite.

**Step 4** On Windows, use the native event logging application to save the System and Application event logs to files. You can do this from the Event Viewer. These event logs often contain data that helps debug Cisco Prime IP Express server problems. For a description of the log messages for each server module, see the `install-path/docs/msgid/MessageIdIndex.html` file.

Linux Troubleshooting Tools

You can also use the following commands on Linux system to troubleshoot Cisco Prime IP Express. To:

- See all Cisco Prime IP Express processes:
  
  ```
  ps -leaf | grep nwr
  ```

- Monitor system usage and performance:
  
  ```
  top
  vmstat
  ```

- View login or boot up errors:
  
  - On Linux—`grep /var/log/messages`

- View the configured interfaces and other network data:
  
  ```
  ifconfig -a
  ```
Using the TAC Tool

There may be times when any amount of troubleshooting steps will not resolve your problem and you have to resort to contacting the Cisco Technical Assistance Center (TAC) for help. Cisco Prime IP Express provides a tool so that you can easily assemble the server or system error information, and package this data for TAC support engineers. This eliminates having to manually assemble this information with TAC assistance. The resulting package from this tool provides the engineers enough data so that they can more quickly and easily diagnose the problem and provide a solution.

The `cnr_tactool` utility is available in the bin directory of the Windows, and `usrbin` directory of the UNIX or Linux, installation directories. Execute the `cnr_tactool` utility:

```bash
> cnr_tactool -N username -P password [-d output-directory]
```

The output directory is optional and normally is the temp directory of the installation directories (in the `/var` path on Linux). If you do not supply the username and password on the command line, you are prompted for them:

```bash
> cnr_tactool
username:
password:
[processing messages....]
```

The tool generates a packaged tar file whose name includes the date and version. The tar file contains all the diagnostic files.
Backup and Recovery

This chapter explains how to maintain the Cisco Prime IP Express databases.

Related Topics
- Backing Up Databases, page 8-1
- Troubleshooting Databases, page 8-9

Backing Up Databases

Because the Cisco Prime IP Express databases do a variety of memory caching and can be active at any time, you cannot rely on third-party system backups to protect the database. They can cause backup data inconsistency and an unusable replacement database.

For this purpose, Cisco Prime IP Express provides a shadow backup utility, cnr_shadow_backup. Once a day, at a configurable time, Cisco Prime IP Express takes a snapshot of the critical files. This snapshot is guaranteed to be a consistent view of the databases.

Related Topics
- Syntax and Location, page 8-2
- Backup Strategy, page 8-2
- Database Recovery Strategy, page 8-4
- Backing Up CNRDB Data, page 8-5
- Backing Up all CNRDBs using tar or similar tools, page 8-5
- Recovering CNRDB Data from Damaged Databases, page 8-6
- Recovering CNRDB Data from Backups, page 8-7
- Recovering all CNRDBs using tar or Similar Tools, page 8-8
- Recovering single CNRDB from tar or similar tools, page 8-9
- Virus Scanning While Running Cisco Prime IP Express, page 8-9
- Virus Scanning While Running Cisco Prime IP Express, page 8-9
Syntax and Location

Be sure to understand that the notation "/data/db" in the following sections refers to directories in the Cisco Prime IP Express product data location path, depending on the operating system:

- **Windows**—"/data" means the data directory, which by default is C:\CiscoPrimeIPExpress\{Local | Regional}\data.
- **Linux**—"/data" means the data directory, which by default is /var/nwreg2/{local | regional}/data.

Cisco Prime IP Express database utility programs mentioned in the following sections are located in the "/bin" directory, which you run as its full path name:

- **Windows**—"/bin/program" means the program file in the bin directory, which by default is C:\Program Files\Cisco Prime IP Express\{Local | Regional}\bin\program for a 32-bit OS and C:\Program Files (x86)\Cisco Prime IP Express\{Local | Regional}\bin\program for a 64-bit OS.
- **Linux**—"/bin/program" means the program file in the bin directory, which by default is /opt/nwreg2/local/usrbin/program or /opt/nwreg2/regional/usrbin/program.

**Note**
Use only the approved utilities for each type of database. In Windows, if you want to run the utility from outside the installed path, you must set the CNR_HOME environment variable.

Backup Strategy

The backup strategy involves either:

- Making CCM perform a nightly shadow backup for you (See the “Setting Automatic Backup Time” section on page 8-3) and using the shadow backups for permanent backup and then doing an explicit backup - either using the cnr_shadow_backup utility and backing up the backup files (*.bak DBs) or

  Shutting down Cisco Prime IP Express and performing a backup using TAR or other similar tools.

Using cnr_shadow_backup utility:

Use the cnr_shadow_backup utility to back up the following databases:

- **CNRDB databases**—...data/dhcp, ...data/dns/csetdb, ...data/dns/rrdb, ...data/cdns, ...data/leasehist, ...data/lease6hist, ...data/subnetutil, ...data/mcd, ...data/replica, and ...data/ccm/ndb

**Note**
If you change the location of the data directory, you must edit the cnr.conf file, which is located in ../conf (see the “Modifying the cnr.conf File” section on page 7-23). Change the cnr.datadir variable to the full path to the data directory. For example, the following is the default value on Windows:

```
cnr.datadir=C:\\IPExpress\\{Local | Regional}\\data
```

The most basic component of a backup strategy is the daily shadow backup. When problems occur with the operational database, you might need to try recovering based on the shadow backup of the previous day. Therefore, you must recognize and correct any problems that prevent a successful backup.
Chapter 8     Backup and Recovery

Backing Up Databases

The most common problem is disk space exhaustion. To get a rough estimate of disk space requirements, take the size of the ../data directory and multiply by 10. System load, such as usage patterns, application mix, and the load on Cisco Prime IP Express itself, may dictate that a much larger reserve of space be available.

You should regularly archive existing shadow backups (such as to tape, other disks, or other systems) to preserve them for possible future recovery purposes.

Caution

Using a utility on the wrong type of database other than the one recommended can cause database corruption. Use only the utilities indicated. Also, never use the database utilities on the operational database, only on a copy.

Related Topics

Setting Automatic Backup Time, page 8-3
Performing Manual Backups, page 8-3
Using Third-Party Backup Programs with cnr_shadow_backup, page 8-3

Setting Automatic Backup Time

You can set the time at which an automatic backup should occur by editing the cnr.conf file (in ../conf). Change the cnr.backup-time variable to the hour and minute of the automatic shadow backup, in 24-hour HH:MM format, then restart the server agent. For example, the following is the preset value:

cnr.backup-time=23:45

Performing Manual Backups

You can also initiate a manual backup with the cnr_shadow_backup utility, which requires root privileges. Enter the cnr_shadow_backup command at the prompt to perform the backup.

Note

To restore DHCP data from a failover partner that is more up to date than a backup, see the “Restoring DHCP Data from a Failover Server” section on page 8-13.

Using Third-Party Backup Programs with cnr_shadow_backup

You should avoid scheduling third-party backup programs while cnr_shadow_backup is operating. Third-party backup programs should be run either an hour earlier or later than the cnr_shadow_backup operation. As described in the “Setting Automatic Backup Time” section on page 8-3, the default shadow backup time is daily at 23:45.

Configure third-party backup programs to skip the Cisco Prime IP Express operational database directories and files, and to back up only their shadow copies.

The operational files are listed in the “Backup Strategy” section on page 8-2. On Linux, Cisco Prime IP Express also maintains lock files in the following directories:

- Cisco Prime IP Express server processes—/var/nwreg2/local/temp/np_destiny_trampoline or /var/nwreg2/regional/temp/np_destiny_trampoline
The lock files are recreated during a reboot. These files are important while a system is running. Any maintenance process (such as virus scanning and archiving) should exclude the temporary directories, operational database directories, and files.

Windows does not maintain lock files, but uses named-pipes instead.

**Database Recovery Strategy**

Cisco Prime IP Express uses the CNRDB database. Table 8-1 lists the types of CNRDB database that must be backed up and recovered.

<table>
<thead>
<tr>
<th>Subdirectory</th>
<th>Cluster</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mcd</td>
<td>local</td>
<td>CNRDB</td>
<td>MCD change log data. Only exists for databases where there is MCD change log history that has not been trimmed.</td>
</tr>
<tr>
<td>ccm</td>
<td>local, regional</td>
<td>CNRDB</td>
<td>Central Configuration Management database. Stores local centrally managed cluster and the SNMP server data.</td>
</tr>
<tr>
<td>dns</td>
<td>local</td>
<td>CNRDB</td>
<td>DNS database. Stores zone state information, names of protected RRs, and zone configuration data for the DNS server.</td>
</tr>
<tr>
<td>cdns</td>
<td>local</td>
<td>CNRDB</td>
<td>Caching DNS database. Stores the initial DNSSEC root trust anchor and root hints.</td>
</tr>
<tr>
<td>dhcp</td>
<td>local</td>
<td>CNRDB</td>
<td>DHCP database. Stores lease state data for the DHCP server.</td>
</tr>
<tr>
<td>dhcpeventstore</td>
<td>local</td>
<td>CNRDB</td>
<td>Queue that Cisco Prime IP Express maintains to interact with external servers, such as for LDAP and DHCPv4 DNS Update interactions. Recovery is not necessary.</td>
</tr>
<tr>
<td>replica</td>
<td>regional</td>
<td>CNRDB</td>
<td>Stores replica data for the local clusters.</td>
</tr>
<tr>
<td>lease6hist</td>
<td>regional</td>
<td>CNRDB</td>
<td>DHCPv6 lease history database.</td>
</tr>
<tr>
<td>leasehist</td>
<td>regional</td>
<td>CNRDB</td>
<td>DHCPv4 lease history database.</td>
</tr>
</tbody>
</table>

The general approach to recovering a Cisco Prime IP Express installation is:

1. Stop the Cisco Prime IP Express server agent.
2. Restore or repair the data.
3. Restart the server agent.
4. Monitor the server for errors.

After you are certain that you executed a successful database recovery, always manually execute the cnr_shadow_backup utility to make a backup of the current configuration and state.
Backing Up CNRDB Data

In the case of the CNRDB databases, the cnr_shadow_backup utility copies the database and all log files to a secondary directory in the directory tree of the installed Cisco Prime IP Express product. For:

- **DHCP**—The operational database is in the .../data/dhcp/ndb and .../data/dhcp/clientdb directories, with the log files in the .../data/dhcp/ndb/logs directory. The shadow copies are in the .../data/dhcp.bak/ndb directory.

- **DNS**—The operational database is in the .../data/dns/rrdb directory. The important operational components are the High-Availability (HA) DNS is in the .../data/dns/hadb directory, with log files in the .../data/dns/hadb/logs directory. The shadow copies are in the .../data/dns.bak directory.

- **CDNS**—The operational database is in the .../data/cdns directory. The shadow copies are in the .../data/cdns.bak directory.

- **CCM**—The operational database and log files are in the .../data/ccm/ndb directory. The shadow copies are in the .../data/ccm.bak directory.

- **MCD change log**—The operational database and log files are in the .../data/mcd/ndb directory. The shadow copies are in the .../data/mcd.bak directory. MCD Change Log database may not exist if there are no change log entries. Also, the database is deleted when the MCD change log history is trimmed or when there is no MCD change log data to begin with.

- **Lease history**—The operational database and log files are in the .../data/leasehist and .../data/lease6hist directories. The shadow copies are in the .../data/leasehist.bak and .../data/lease6hist.bak directories.

- **Replica**—The operational database and log files are in the .../data/replica directory.

The actual file naming convention is:

- **Database**—dhcp.ndb and dns.ndb.
- **Log files**—log.00000000001 through log.9999999999. The number of files varies with the rate of change to the server. There are typically only a small number. The specific filename extensions at a site vary over time as the database is used. These log files are not humanly readable.

Backing Up all CNRDBs using tar or similar tools

This section describes the procedure for backing up all Cisco Prime IP Express databases using tar or similar tools.

**Step 1** Shut down Cisco Prime IP Express.

Backups cannot be done using tar or similar tools if Cisco Prime IP Express is running.

**Step 2** Back up the entire data directory and subdirectories:

```
> /var/nwreg2/local/data or /var/nwreg2/regional/data
> /opt/nwreg2/*/conf
```

**Step 3** Restart Cisco Prime IP Express when the backup is complete.
Chapter 8  Backup and Recovery

Backing Up Databases

Note
Technically the backups do not need to include the *.bak directories (and subdirectories of those directories) as those contain nightly shadow backups. However, unless your available storage space is severely limited, we recommend a full backup of the entire data directory (and subdirectories) including the shadow backups.

Recovering CNRDB Data from Damaged Databases

This section describes a procedure that recovers any or all CNRDB type databases. Depending on the event that caused the database corruption, you can restore the database to a healthy state by using the current data. This is the best option. Always attempt recovery on a copy of the database file and associated log files, never on the operational files. This is a simple file copy operation, distinct from a shadow backup. Also, never attempt a recovery while Cisco Prime IP Express is running.

In most cases, you can use the log files that accompany the databases (such as the DHCP log files in .../data/dhcp/nb/logs) to repair a failed server database. You can do so because the log files journal all database activity. You should never move, rename, or delete these log files, even after successfully completing a recovery. In fact, the recovery process uses copies rather than the originals of these files.

Caution
It is possible to damage the CNRDB database files without the damage being immediately obvious. Such damage could occur if you (a) inappropriately delete log files; (b) mix pre- and post-recovery database and log files; or (c) attempt to recover database files currently in use by an application. For the CNRDB database, use the cnrdb_archive, cnrdb_recover, and cnrdb_verify utilities.

Use the cnrdb_recover utility (see the “Using the cnrdb_recover Utility” section on page 8-11), included in the Cisco Prime IP Express product distribution, for database recovery. Use this tool with care. You should never use it directly on an operational database, or on files another application is concurrently accessing. On a successful database recovery, do not intermingle the recovered files (database file and log files) with files from another source, such as the operational database or shadow backups. Recovered database files acquire state information that make them incompatible with older database files.

Step 1
Stop the Cisco Prime IP Express server agent. This stops all the protocol servers. Ensure that enough disk space is available for a copy of the database files, plus a 15% safety margin.

Step 2
Create a backup directory, named backup, outside the Cisco Prime IP Express installation tree. On Windows, this could be C:/temp/backup; on Linux, this could be /tmp/backup.

As a precaution, a copy of the directory tree of the current database that you are going to repair will get copied here automatically.

Step 3
Copy the database subdirectories you want to restore under .../data to the backup directory. For example, to recover the DHCP database, recursively copy the .../data/dhcp directory and its subdirectories to /tmp/backup:

```bash
> cp -rp /var/nwreg2/local/data/dhcp /tmp/backup
```

When the copy is completed, double check that the database file and all log files were copied correctly. Do not allow these files to be modified in any way. Do not run any utilities or servers on these files.
Note

The log files must not be copied or moved while trying to repair the database because the "set_lg_dir logs" in the DB_CONFIG file provides the information as to where the log files are located (in the log subdirectory). This enables the CNRDB utilities to find the log files without your having to copy or move the log files to any location. The relative path is used in the DB_CONFIG file so that it is easier to move the directories around.

Step 4

Repair the database:

a. From the database file directory, run the `cnrdb_recover` program, using the `-c` and `-v` options. It is helpful to use `-v` in that it displays output in the absence of errors (see the “Using the `cnrdb_recover` Utility” section on page 8-11). For example:

```
> cd /var/nwreg2/local/data/dhcp/ndb
> /opt/nwreg2/local/bin/cnrdb_recover -v -c
db_recover: Finding last valid log LSN: file: 1 offset 95181
db_recover: Recovery starting from [1][28]
db_recover: Recovery complete at Mon June 19 18:44:15 2006
db_recover: Maximum transaction ID 800000009 Recovery checkpoint [1][95229]
db_recover: Recovery complete at Mon Jun 19 18:44:15 2006
db_recover: Maximum transaction ID 80000000 Recovery checkpoint [1][95529]
```

b. Run the `cnrdb_verify` utility for each of the servers. There is no output if the verification is successful (see the “Using the `cnrdb_verify` Utility” section on page 8-12). For example:

```
> cd /var/nwreg2/local/data/dhcp/ndb
> /opt/nwreg2/local/bin/cnrdb_verify dhcp.ndb
```

c. Optionally, for additional confidence, run the `cnrdb_archive` utility:

```
  `cnrdb_archive` -l—Lists all log files
  `cnrdb_archive` -s—Lists the database file
```

d. If there are any indications that an error occurred, proceed to restore the database from a backup, as described in the “Recovering CNRDB Data from Backups” section on page 8-7.

Step 5

For DHCP only, delete the files in the `../data/dhcpeventstore` directory.

Step 6

Restart Cisco Prime IP Express.

Recovering CNRDB Data from Backups

If there are any indications, such as server log messages or missing data, that database recovery was unsuccessful, you may need to base a recovery attempt on the current shadow backup (in the Cisco Prime IP Express installation tree). To do this:

Step 1

Stop the Cisco Prime IP Express server agent.

Step 2

Move the operational database files to a separate temporary location.

Step 3

Copy each `../data/name.bak` directory to `../data/name`; for example, copy `../data/ccm.bak` to `../data/ccm`.

Step 4

Repair the database:

a. From the database file directory, run the `cnrdb_recover` program, using the `-c` and `-v` options. It is helpful to use `-v` in that it displays output in the absence of errors (see the “Using the `cnrdb_recover` Utility” section on page 8-11). For example:

```
> cd /var/nwreg2/local/data/dhcp/ndb
> /opt/nwreg2/local/bin/cnrdb_recover -v -c
db_recover: Finding last valid log LSN: file: 1 offset 95181
db_recover: Recovery starting from [1][28]
db_recover: Recovery complete at Mon June 19 18:44:15 2006
db_recover: Maximum transaction ID 800000009 Recovery checkpoint [1][95229]
db_recover: Recovery complete at Mon Jun 19 18:44:15 2006
db_recover: Maximum transaction ID 80000000 Recovery checkpoint [1][95529]
```

b. Run the `cnrdb_verify` utility for each of the servers. There is no output if the verification is successful (see the “Using the `cnrdb_verify` Utility” section on page 8-12). For example:

```
> cd /var/nwreg2/local/data/dhcp/ndb
> /opt/nwreg2/local/bin/cnrdb_verify dhcp.ndb
```

c. Optionally, for additional confidence, run the `cnrdb_archive` utility:

```
  `cnrdb_archive` -l—Lists all log files
  `cnrdb_archive` -s—Lists the database file
```

d. If there are any indications that an error occurred, proceed to restore the database from a backup, as described in the “Recovering CNRDB Data from Backups” section on page 8-7.

Step 5

For DHCP only, delete the files in the `../data/dhcpeventstore` directory.

Step 6

Restart Cisco Prime IP Express.

Recovering CNRDB Data from Backups

If there are any indications, such as server log messages or missing data, that database recovery was unsuccessful, you may need to base a recovery attempt on the current shadow backup (in the Cisco Prime IP Express installation tree). To do this:

Step 1

Stop the Cisco Prime IP Express server agent.

Step 2

Move the operational database files to a separate temporary location.

Step 3

Copy each `../data/name.bak` directory to `../data/name`; for example, copy `../data/ccm.bak` to `../data/ccm`.
Chapter 8      Backup and Recovery

Backing Up Databases

Step 4   Rename the files.

The CNRDB database maintains centrally managed configuration data that is synchronized with the server configuration databases.

Step 5   Create a new data directory and then untar or recover the backed up directory.

We recommend that you run the DB directory and recovery tools to ensure that the databases are good.

Note   Ensure that the logs subdirectory is present in the same directory or the logs path is mentioned in the DB_CONFIG file.

Step 6   Restart the server agent.

Note   If the recovery fails, perhaps because the current shadow backup is simply a copy of corrupted files, use the most recent previous shadow backup. This illustrates the need to regularly archive shadow backups. You cannot add operational log files to older shadow backup files. All data added to the database since the shadow backup was made will be lost.

After a successful database recovery, initiate an immediate backup and archive the files using the cnr_shadow_backup utility (see the “Performing Manual Backups” section on page 8-3).

Recovering all CNRDBs using tar or Similar Tools

This section describes the procedure for recovering all Cisco Prime IP Express databases using tar or similar tools.

Step 1   Shut down Cisco Prime IP Express. Run /etc/init.d/nwreglocal stop to ensure that Cisco Prime IP Express is down.

Step 2   Rename the active data directory (such as mv data old-data).

Note   You must have sufficient disk space for twice the size of the data directory (and all the files in it and its subdirectories). If you do not have sufficient disk space, move the active data directory to another drive.

Step 3   Create a new data directory and then untar or recover the backed up directory.

We recommend that you run the CNRDB directory and recovery tools to ensure that the databases are good.

Step 4   Start Cisco Prime IP Express.
Note

Technically the restores do not need to include the *.bak directories (and subdirectories of those directories) as those contain nightly shadow backups. However, unless your available storage space is severely limited, we recommend a full restore of the entire data directory (and subdirectories) including the shadow backups.

Recovering single CNRDB from tar or similar tools

This section describes the procedure for recovering single database using tar or similar tools.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Shut down Cisco Prime IP Express. Run <code>/etc/init.d/nwreglocal stop</code> to ensure that Cisco Prime IP Express is down.</td>
</tr>
<tr>
<td>2</td>
<td>Rename the active data directory (such as mv data old-data).</td>
</tr>
</tbody>
</table>

Note

You must have sufficient disk space for twice the size of the data directory (and all the files in it and its subdirectories). If you do not have sufficient disk space, move the active data directory to another drive.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Create a new data directory and then untar or recover only the files in that directory (and its subdirectories) from the backup. We recommend that you run the CNRDB integrity and recovery tools to ensure that the CNRDB are good.</td>
</tr>
<tr>
<td>4</td>
<td>Repeat Step 2 to Step 3 for other DBs that have to be recovered.</td>
</tr>
<tr>
<td>5</td>
<td>Start Cisco Prime IP Express.</td>
</tr>
</tbody>
</table>

Virus Scanning While Running Cisco Prime IP Express

If you have virus scanning enabled on your system, it is best to configure it to exclude certain Cisco Prime IP Express directories from being scanned. Including these directories might impede Cisco Prime IP Express operation. The ones you can exclude are the .../data, .../logs, and .../temp directories and their subdirectories.

Troubleshooting Databases

The following sections describe troubleshooting the Cisco Prime IP Express databases.

Related Topics

- Using the cnr_exim Data Import and Export Tool, page 8-10
- Using the cnrdb_recover Utility, page 8-11
- Using the cnrdb_verify Utility, page 8-12
- Using the cnrdb_checkpoint Utility, page 8-13
- Restoring DHCP Data from a Failover Server, page 8-13
Using the cnr_exim Data Import and Export Tool

The cnr_exim data import and export tool now supports the following for a user:

- Exporting all the data
- Exporting and importing license related data
- Importing all of the data

The cnr_exim tool also serves to export unprotected resource record information. However, cnr_exim simply overwrites existing data and does not try to resolve conflicts.

**Note**

You cannot use cnr_exim tool for import or export of data from one version of Cisco Prime IP Express to another. It can be used only for import or export of data from or to the same versions of Cisco Prime IP Express.

Before using the cnr_exim tool, exit from the CLI, then find the tool on:

- **Windows** — ...in\cnr_exim.exe
- **Linux** — .../usrbin/cnr_exim

You must reload the server for the imported data to become active.

Note that text exports are for reading purposes only. You cannot reimport them.

The text export prompts for the username and password (the cluster defaults to the local cluster). The syntax is:

```
> cnr_exim –e exportfile [-N username -P password -C cluster]
```

To export (importable) raw data, use the –x option:

```
> cnr_exim –e exportfile –x
```

To export DNS server and zone components as binary data in raw format, use the –x and –c options:

```
> cnr_exim –e exportfile –x –c "dnsserver,zone"
```

The data import syntax is (the import file must be in raw format):

```
> cnr_exim –i importfile [-N username -P password -C cluster]
```

You can also overwrite existing data with the –o option:

```
> cnr_exim –i importfile –o
```

Table 8-2 describes all the qualifying options for the cnr_exim tool.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–a value</td>
<td>Allows exporting and importing of protected or unprotected RRs. Valid values are: protectedRR unprotectedRR On export or import, all RRs are exported by default, so you must use a value to export or import just the protected or unprotected RRs.</td>
</tr>
</tbody>
</table>
Using the `cnrdb_recover` Utility

The `cnrdb_recover` utility is useful in restoring the Cisco Prime IP Express databases to a consistent state after a system failure. You would typically use the `–c` and `–v` options with this command (Table 8-3 describes all of the qualifying options). The utility is located in the installation bin directory.

### Table 8-3 `cnrdb_recover` Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>–c</code></td>
<td>Performs a catastrophic recovery instead of a normal recovery. It not only examines all the log files present, but also recreates the <code>.ndb</code> (or <code>.db</code>) file in the current or specified directory if the file is missing, or updates it if is present.</td>
</tr>
<tr>
<td><code>–e</code></td>
<td>Retains the environment after running recovery, rarely used unless there is a DB_CONFIG file in the home directory.</td>
</tr>
</tbody>
</table>
In the case of a catastrophic failure, restore a snapshot of all database files, along with all log files written since the snapshot. If not catastrophic, all you need are the system files at the time of failure. If any log files are missing, `cnrdb_recover –c` identifies the missing ones and fails, in which case you need to restore them and perform the recovery again.

Use of the catastrophic recovery option is highly recommended. In this way, the recovery utility plays back all the available database log files in sequential order. If, for some reason, there are missing log files, the recovery utility will report errors. For example, the following gap in the log files listed:

```
log.0000000001
log.0000000053
```

results in the following error that might require you to open a TAC case:

```
db_recover: Finding last valid log LSN:file:1 offset 2411756
db_recover: log_get: log.0000000002: No such file or directory
db_recover: DBENV->open: No such for or directory
```

### Using the `cnrdb_verify` Utility

The `cnrdb_verify` utility is useful for verifying the structure of the Cisco Prime IP Express databases. The command requires a file parameter. Use this utility only if you are certain that there are no programs running that are modifying the file. **Table 8-4** describes all its qualifying options. The utility is located in the installation bin directory. The syntax is described in the usage information when you run the command:

```
C:\Program Files\IP Express\Local\bin>cnrdb_verify
```

```
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–h dir</td>
<td>Specifies a home directory for the database environment. By default, the current working directory is used.</td>
</tr>
<tr>
<td>–N</td>
<td>Prevents acquiring shared region locks while running, intended for debugging errors only, and should not be used under any other circumstances.</td>
</tr>
<tr>
<td>–o</td>
<td>Ignores database sort or hash ordering and allows <code>cnrdb_verify</code> to be used on nondefault comparison or hashing configurations.</td>
</tr>
<tr>
<td>–P password</td>
<td>User password, if the file is protected.</td>
</tr>
<tr>
<td>–q</td>
<td>Suppresses printing any error descriptions other than exit success or failure.</td>
</tr>
</tbody>
</table>
Using the \texttt{cnrdb\_checkpoint} Utility

The \texttt{cnrdb\_checkpoint} utility is useful in setting a checkpoint for the database files so as to keep them current. The utility is located in the installation bin directory. The syntax is described in the usage information when you run the command:

\begin{verbatim}
C:\Program Files\IP Express\Local\bin>cnrdb_checkpoint ?
\end{verbatim}

Restoring DHCP Data from a Failover Server

You can restore DHCP data from a failover server that is more current than the result of a shadow backup. Be sure that the failover partner configurations are synchronized, then, on the failover partner:

\textbf{On Windows}

1. Set the default path; for example:
   \begin{verbatim}
   SET PATH=%PATH%;.;C:\PROGRAM~1\NETWORK~1\LOCAL\BIN
   \end{verbatim}
2. Stop the server agent:
   \begin{verbatim}
   net stop "Network Registrar Local Server Agent"
   \end{verbatim}
3. Delete the eventstore, ndb, and logs directories:
   \begin{verbatim}
   del C:\CiscoPrimeIPExpress\Local\data\dhcpeventstore\*.*
   del C:\CiscoPrimeIPExpress\Local\data\dhcp\ndb\dhcp.ndb
   del C:\CiscoPrimeIPExpress\Local\data\dhcp\ndb\logs\*.*
   \end{verbatim}
4. Restart the server agent:
   \begin{verbatim}
   net start "Network Registrar Local Server Agent"
   \end{verbatim}

\textbf{On Linux}

1. Stop the server agent:
   \begin{verbatim}
   /etc/init.d/nwreglocal stop
   \end{verbatim}
2. Determine the processes running:
   \begin{verbatim}
   /opt/nwreg2/local/usrbin/cnr_status
   \end{verbatim}
3. Kill the remaining processes:
   \begin{verbatim}
   kill -9 pid
   \end{verbatim}
4. Delete the eventstore, ndb, and logs directories:
   \begin{verbatim}
   rm /var/nwreg2/data/dhcpeventstore/.*.*
   rm -r /var/nwreg2/data/dhcp/ndb/*
   \end{verbatim}
5. Restart the server agent:
   
   /etc/init.d/nwreglocal start
PART 3

Address Management
Managing Address Space

Address blocks provide an organizational structure for addresses used across the network. Address blocks can consist of static addresses or dynamic addresses allocated to DHCP servers for lease assignment. An address block can have any number of child address blocks and can culminate in one or more child subnets. The address block administrator is responsible for these objects. This administrator can create parent and child address blocks or subnets, which are always the leaf nodes of the address space. Static subnets can be further subdivided into one or more IP address ranges. However, dynamically added subnets create their own subnets that the administrator cannot modify or delete.

Note

For IPv6 address management, see the “Viewing IPv6 Address Space” section on page 27-14.

Related Topics

- Address Block Administrator Role, page 9-1
- Viewing Address Space, page 9-2
- Pulling Replica Address Space from Local Clusters, page 9-2
- Address Blocks and Subnets, page 9-3

Address Block Administrator Role

The address block administrator role manages address space at a higher level than that of specific subnet or static address allocations. This is actually a middle manager role, because there is likely to be a higher authority handing out address blocks to the system.

Related Topics

- Required Permissions, page 9-1
- Role Functions, page 9-2

Required Permissions

To exercise the functions available to the address administrator, you must have at the:

- **Regional cluster**—The regional-addr-admin role assigned. This role should probably be unencumbered by further lease-history, and dhcp-management subrole restrictions.
- **Local cluster**—The addrblock-admin role assigned.
Role Functions

These functions are available to the address block administrator at the:

- **Regional cluster:**
  - Address aggregation. For example, if the 10.0.0.0/16 address block exists at the regional cluster and a local cluster administrator creates the 10.1.1.0/24 address block, the local address block (through replication) is rolled up under its parent at the regional cluster. This allows a unified view of the address space at the regional cluster without affecting the local cluster configuration.
  - Address delegation. Administrators can delegate address space to the local cluster, thereby giving up authority of the delegated object.
  - Lease history reports. This provides a single vantage point on the lease history of multiple DHCP servers. The administrator can query the history data at the local cluster to constrain the scope of the history report. Lease histories can be queried by VPN (if defined), time range and criteria that contain the following choices: IP address, MAC address, IP address range, or all. This is an important feature to meet government and other agency mandates concerning address traceability. For details on querying lease history, see the “Querying Leases” section on page 23-28.
  - Polling configurations. The administrator can control the intervals and periods of local cluster polling for replication, and IP histories. You can also set the lease history and compacting intervals at the CCM server level. (See Chapter 6, “Managing the Central Configuration.”)
  - Check the DHCP and address data consistency.

- **Local cluster:**
  - Manage address blocks, subnets, and address types.
  - Check the DHCP and address data consistency.

Viewing Address Space

The address space is a hierarchical tree of address blocks and subnets in IPv4 and prefixes in IPv6, sorted in IP address order. You can choose the level of depth at which to display the tree. You can also expand and contract nodes, which recursively expands or contracts all child nodes. If you pick a new level, this overrides the previous expansion or contraction.

**Local Advanced and Regional Web UI**

To view the address space as a hierarchical tree:

- From the **Design** menu, choose **Address Tree** under the **DHCPv4** submenu to open the DHCP Address Tree page. Note that you can choose a VPN (if configured).
- From the **Design** menu, choose **Address Tree** under the **DHCPv6** to open the DHCP v6 Address Tree page.

Pulling Replica Address Space from Local Clusters

You may choose to pull address space from the replica data of the local clusters instead of explicitly creating it.
Pulling replica address space from a local cluster where IPv4 subnets were removed does not clear the server name on the subnet. Although the subnet is no longer used, it is still considered allocated to the server. Hence, the delete operation does not appear for the subnet, so that you cannot delete the subnet from the regional cluster. To push or reallocate the subnet to a different cluster, or remove it from the regional cluster, you must first reclaim the subnet (see the “Reclaiming Subnets” section on page 9-9). This clears the reference to the local server.

**Regional Web UI**

**Step 1** In the DHCP Address Tree (or DHCP v6 Address Tree) page, click the **Pull Replica** icon in the Address Tree pane.

**Step 2** In the Select Pull Replica Address Space (or Select Pull Replica IPv6 Address Space) page:
- To omit reservations while pulling replica, check the **Omit Reservations** check box.
- Select the Data Synchronization Mode (**Update**, **Complete** or **Exact**)

**Step 3** Click **Report** at the bottom of the page.

**Step 4** Confirm the summary, and click **Run**.

**Step 5** Click **OK**.

---

### Address Blocks and Subnets

An address block is an aggregate of IP addresses based on a power-of-two address space that can be delegated to an authority. For example, the 192.168.0.0/16 address block (part of the RFC 1918 private address space) includes $2^{16}$ (or 65536) addresses. Address blocks can be further divided into child address blocks and subnets. For example, you might want to delegate the 192.168.0.0/16 address block further into four child address blocks—192.168.0.0/18, 192.168.64.0/18, 192.168.128.0/18, and 192.168.192.0/18.

**Note** The DHCP server also uses address blocks to manage subnet allocation for on-demand address pools (see the “Configuring Virtual Private Networks and Subnet Allocation” section on page 24-17). Address blocks used for dynamic address pools must be created using the `dhcp-address-block` command in the CLI. The unified address view in the web UI also displays these dynamic address blocks, but does not provide an edit link to them, because they have been delegated in their entirety to the DHCP server. They should not be further subdivided for subnet allocation. The DHCP server automatically handles these address blocks as it receives subnet requests. These address pools are indicated by a **D** (for “Delegated”).

A subnet is the leaf node of the address space and cannot be further subdivided. If you create the 192.168.50.0/24 subnet, you can subsequently create an address block by that same name, and the subnet will become a child of the address block. However, you cannot further subdivide or delegate the 192.168.50.0/24 subnet.
Subnets can have one or more defined address ranges. Address blocks cannot have address ranges. When you create an address range for a subnet by using the web UI, it becomes a static range, meaning that it cannot be allocated dynamically using DHCP. However, the web UI shows any dynamic ranges defined by DHCP scopes for the subnet. Displaying the ranges as such indicates where overlaps may occur between assigning static addresses for the address space and dynamic addresses for scopes.

The address space view shows the hierarchy of address block and subnets and their parent-child relationships. The hierarchy does not go down to the level of address ranges for each subnet. These are displayed when you access the subnet.

Related Topics

- Viewing Address Blocks, Subnets, and Address Types, page 9-4
- Knowing When to Add Address Blocks, page 9-5
- Adding Address Blocks, page 9-5
- Delegating Address Blocks, page 9-6
- Pushing Subnets to Local DHCP Servers, page 9-7
- Creating Reverse Zones from Subnets, page 9-8
- Reclaiming Subnets, page 9-9
- Adding Children to Address Blocks, page 9-9
- Adding Address Ranges to Subnets, page 9-10
- Viewing Address Utilization for Address Blocks, Subnets, and Scopes, page 9-10

Viewing Address Blocks, Subnets, and Address Types

You can view the address blocks and subnets created for a network.

Local Advanced and Regional Web UI

From the Design menu, choose Address Tree under the DHCPv4 submenu to open the DHCP Address Tree page.

To choose a level of depth for the address space, click one of the addresses in the Address Tree pane on the left. The details about the address appear in the page. The Address Type column identifies the type of object displayed, an address block or a subnet. The Owner column identifies the owner of the address space, and the Region column identifies the assigned region for the address space.

Address spaces that were assigned dynamically are indicated by a D (for “Delegated”) in the Address Type column. You cannot delete this delegated address space.

To refresh the view, click the Refresh icon.

You can add, modify, and delete address types. From the Design menu, choose Address Types under the DHCP Settings submenu to open the List/Add Address Types page. Click the Add Address Type icon in the Address Types pane on the left, to open the Add Address Type page, and modify settings on the Edit Address Type page. You can also pull replica address types and push address types to the local clusters on the List/Add Address Types page.
Knowing When to Add Address Blocks

This use case describes the set of user actions associated with adding a new address block to the network in a shared management network. These preconditions are assumed:

1. From summary IP address utilization reports, an address block administrator notes that the top level address block of the company is nearing the 90% utilization mark.
2. The address block administrator submits a request for more address space from ARIN (or some other numbering authority) and the request is granted.

Once the address space is made available, the regional address administrator:

1. Adds the new blocks to the central address block map, and based on a review of the utilization reports, creates and delegates address blocks to be used by the local clusters. The action of delegating the address blocks causes them to be pushed to the local clusters.
2. Allocates the new address space to network elements as needed, using router and failover synchronization features to simplify the configuration tasks:
   - Allocates subnets to a failover pair (gets a scope template for the subnet, either from the subnet or the failover pair).
   - Finds a free subnet (finds the address block of the right type).
   - Allocates the free subnet to an address destination (DHCP server or other destination).

Adding Address Blocks

Once you configure your network, you can add DHCPv4 address blocks.

Local Advanced and Regional Web UI

To view the CCM address blocks, from the Design menu choose Address Blocks under the DHCPv4 submenu to open the List/Add DHCP Address Blocks page.

To add an address block, click the Add Address Block icon in the Address Blocks pane on the left. Enter its network address in the Address field, then choose the address mask from the drop-down list. For example, enter 192.168.50.0 in the Address field, then choose 24 in the drop-down list to create the 192.168.50.0/24 address block, which is all the addresses in the range 192.168.50.0 through 192.168.50.255.

For a review of the number of available addresses for each subnet mask, see Table 9-1. These available hosts exclude the two network and broadcast addresses in each range.

<table>
<thead>
<tr>
<th>Table 9-1 Subnet Masking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Mask</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>/8</td>
</tr>
<tr>
<td>/9</td>
</tr>
<tr>
<td>/10</td>
</tr>
<tr>
<td>/11</td>
</tr>
<tr>
<td>/12</td>
</tr>
<tr>
<td>/13</td>
</tr>
</tbody>
</table>
Delegating Address Blocks

Address block delegation is the coordinated actions of marking the delegated address block at the regional cluster as being delegated to a local cluster and creating the delegated address block in the local cluster. To delegate an address block to a local cluster, the address block cannot have child address blocks or subnets. The delegated address block created at the local server must have the same address size as the one at the regional cluster.

You can delegate only one address block to one local cluster at a time; you cannot delegate it to multiple local clusters. You can also delegate an address block to an owner.

To delegate an address block, you must:

1. Have the central configuration administrator create a local cluster to which to delegate the address block (see the “Configuring Server Clusters” section on page 6-2).
2. Have the central configuration administrator synchronize the regional cluster with the local cluster (see the “Synchronizing with Local Clusters” section on page 6-10). The local cluster will have address source references to the regional cluster through the synchronization process.
3. Delegate the address block to the cluster or an owner.
Regional Web UI

For example:

**Step 1**
Have the central configuration administrator create a local cluster, ServProv-One:

a. Log into the regional cluster as the central configuration administrator.

b. From the **Operate** menu, choose **Manage Clusters** under the **Servers** submenu to open the List/Add Remote Clusters page.

c. Click the **Add Cluster** icon in the Manage Clusters pane on the left, to open the Add Cluster dialog box.

d. Enter the cluster name **ServProv-One** and the connection data, then click **Add Cluster**.

e. In the Edit Remote Cluster ServProv-one page, click the **Resync** button.

**Step 2**
As regional address administrator, create an address block:

a. Log into the regional cluster as the regional address administrator.

b. From the **Design** menu, choose **Address Blocks** under the **DHCPv4** submenu to open the List/Add Address Blocks page.

c. Click the **Add Address Block** icon in the Address Blocks pane on the left.

d. Enter **192.168.50.0** in the Address/Mask field, then choose **24** in the mask drop-down list.

e. Click **Add Address Block**.

**Step 3**
Delegate the address block to a cluster or owner:

a. Click the name of the address block to open the Edit Address Block page.

b. Click **Delegate** at the top of the page to open the Delegate Address Block dialog.

c. Choose either a local cluster or an owner to which to delegate the address block.

d. Click **Delegate Address Block**. The Edit Address Block page now indicates that the address block is delegated.

e. If there are further modifications to the address block, click **Save**.

f. The List/Add Address Blocks page now identifies the address block as being delegated (D). To undelegate it, edit the address block again, then click **Reclaim Address Block**.

---

**Pushing Subnets to Local DHCP Servers**

You can push subnets to local DHCP servers.

**Local Advanced and Regional Web UI**

**Step 1**
Have the central configuration administrator create a local cluster and resynchronize it with the local cluster.

**Step 2**
Create a subnet at the regional cluster:

a. From the **Design** menu, choose **Subnets** under the **DHCPv4** submenu. This opens the List/Add Subnets page.
b. Click the **Add Subnets** icon in the Subnets pane on the left.

c. Enter at least the network address and choose the mask of the subnet, then click **Add Subnet**.

### Step 3

Have the central configuration administrator create a scope template so that it can create a scope to contain a subnet:

a. Log into the regional cluster as the central configuration administrator.

b. From the **Design** menu, choose **Scope Templates** under the **DHCPv4** submenu to open the List/Add DHCP Scope Templates page.

c. Click the **Add Scope Templates** icon in the left pane, to open the Add DHCP Scope Template page.

d. Enter the name for the scope template, and click **Add Scope Template**.

e. In the Edit DHCP Scope Template **scopename** page, among other entries on this page, enter the `create-range` expression in the Range Expression field to create a scope with that subnet. (If you choose a policy for the scope template, be sure that the policy exists at the local cluster, or you must push the policy to the local cluster. See the “Pushing Policies to Local Clusters” section on page 6-16.)

### Step 4

As regional address administrator, add the subnet to the local cluster DHCP server:

a. Log into the regional cluster as the regional address administrator.

b. From the **Design** menu, choose **Subnets** under the **DHCPv4** submenu to open the List/Add Subnets page.

c. Select the subnet from the Subnets pane on the left, to open the corresponding Edit Subnet page.

d. Click **Push** at the top of the page. This opens the Push Subnet page.

e. Choose the scope template from the drop-down list.

f. Choose the router and the router interface from the drop-down lists.

g. Choose the cluster from the drop-down list.

h. Click **Push Subnet**.

---

### Creating Reverse Zones from Subnets

You can create reverse zones from subnets directly on the List/Add Subnets page instead of having to do so manually (see the “Adding Reverse Zones from Subnets” section on page 14-12). Click the **Create** icon in the Reverse Zone column of the List/Add Subnets page to open the Create Reverse Zone(s) for Subnet page. On that page, choose a configured zone template from the drop-down list, then click **Report** to return to the List/Add Subnets page.

### Related Topics

- Reclaiming Subnets, page 9-9
- Adding Address Ranges to Subnets, page 9-10
- Viewing Address Utilization for Address Blocks, Subnets, and Scopes, page 9-10
- Pushing Subnets to Local DHCP Servers, page 9-7
Reclaiming Subnets

Once you delegate a subnet to the DHCP server, you can reclaim it if necessary.

**Local Advanced and Regional Web UI**

1. From the **Design** menu, choose **Subnets** under the **DHCPv4** submenu to open the List/Add Subnets page.
2. Select the subnet from the Subnets pane on the left, to open the corresponding Edit Subnet page.
3. Click **Reclaim** at the top of the page. This opens the Reclaim Subnet page.
4. If you want to force deleting the subnet, check the **Force Delete** check box.
5. Click **Reclaim Subnet**.

**Note**
When you push or reclaim subnets for a managed or virtual router, this sets the primary and secondary relationships that are set for the router for all the related subnets and scopes as well.

Adding Children to Address Blocks

You might want to subdivide undelegated address blocks into child address blocks or subnets.

**Local Advanced and Regional Web UI**

1. From the **Design** menu, choose **Address Blocks** under the **DHCPv4** submenu to open the List/Add Address Blocks page.
2. Click the name of an address block that is not marked as delegated (D). This opens the Edit Address Block page.
3. To add a child address block, add an address that is part of the address block network address in the Address/Mask field of the Child Address Blocks section of the Edit Address Block page. Choose a higher mask value than the parent address block, then click **Add**.
   
   An error message appears if you try to set the same network address for a child address block as for a child subnet.

   Omitting a value when you click **Add** automatically adds the subdivisions of the parent address space with the appropriate mask value. For example, if the parent space is 192.168.50.0/24, you omit any child subnet value, and click **Add**, the web UI adds the children in this order:

   - 192.168.50.0/26
   - 192.168.50.64/26
   - 192.168.50.128/26
   - 192.168.50.192/26

4. To add a child subnet, add an address in the Address/Mask field of the Child Subnets section of the page that is part of the address block network address, but choose a higher mask value than the parent address block. Then click **Add**.
An error message appears if you try to set the same network address for a child address block as for a child subnet.

If you omit a value when you click Add, this automatically adds the subdivisions of the parent address space with the appropriate mask value. For example, if the parent space is 192.168.50.0/24, you omit any child subnet value, and click Add, the web UI adds the children in this order:

- 192.168.50.0/26
- 192.168.50.64/26
- 192.168.50.128/26
- 192.168.50.192/26

### Adding Address Ranges to Subnets

You can edit the subnet data and add any number of address ranges to a subnet. These ranges must be in the designated network of the subnet.

#### Local Advanced and Regional Web UI

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>From the Design menu, choose Subnets under the DHCPv4 submenu to open the List/Add Subnets page.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Click the name of the subnet to which you want to add address ranges, in the Subnets pane on the left. This opens the Edit Subnet page.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Enter the starting address of the range in the Start field in the IP Ranges area of the page, then add the ending address in the End field. If you add just the host numbers in these fields, the relative address in the range determined by the address mask is used.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Click Add IP Range.</td>
</tr>
</tbody>
</table>

### Viewing Address Utilization for Address Blocks, Subnets, and Scopes

You can view the current address utilization for address blocks, subnets, and scopes.

#### Tip

For address utilization for IPv6 prefixes, see the “Viewing Address Utilization for Prefixes” section on page 27-30.

#### Local Advanced and Regional Web UI

The function is available on the DHCP Address Tree page, List/Add DHCP Address Blocks page, and List/Add Subnets page. When you click the Current Usage tab, the utilization detail appears.

#### Note

To ensure the proper subnet-to-server mapping on this page, you must update the regional address space view so that it is consistent with the relevant local cluster. Do this by pulling the replica address space, or reclaiming the subnet to push to the DHCP server (see the “Reclaiming Subnets” section on page 9-9). Also ensure that the particular DHCP server is running.
The other columns in the Current Usage tab identify:

- **Type**—Whether the address space is an address block, subnet, or scope.
- **Active Dynamic**—Addresses that are part of a dynamic range managed by DHCP and that are currently leased, but not reserved.
- **Free Dynamic**—Addresses that are not currently leased.
- **Active Reserved**—Addresses that are part of a dynamic range and are reserved.

In the Current Usage tab, the Utilization Detail column items are expandable so that you can view the scope data for an address block or subnet. If you click the address block, subnet, or scope name in this column, this opens the Utilization Detail pane for selected item.

The Utilization Detail pane is read-only, and shows detailed address utilization attributes for the address block, subnet, or scope. The address utilization attributes are described in Table 9-2.

**Table 9-2 Address Utilization Attributes**

<table>
<thead>
<tr>
<th>Utilization Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Addresses</strong></td>
<td></td>
</tr>
<tr>
<td>total-dynamic</td>
<td>Total number of leases, excluding reserved ones.</td>
</tr>
<tr>
<td>total-reserved</td>
<td>Total number of reserved leases.</td>
</tr>
<tr>
<td><strong>Free Dynamic</strong></td>
<td></td>
</tr>
<tr>
<td>avail</td>
<td>Number of dynamic leases that are currently available for issue to clients.</td>
</tr>
<tr>
<td>other_avail</td>
<td>Number of dynamic leases that the DHCP failover partner currently has available for issue to clients.</td>
</tr>
<tr>
<td><strong>Active Dynamic</strong></td>
<td></td>
</tr>
<tr>
<td>offered</td>
<td>Number of dynamic leases that are currently offered to clients, but not yet acknowledged as being leased.</td>
</tr>
<tr>
<td>leased</td>
<td>Number of dynamic leases that are currently acknowledged as leased to clients.</td>
</tr>
<tr>
<td>expired</td>
<td>Number of dynamic leases that are past the lease expiration period, but will not be available for other clients (except after the policy grace-period expires).</td>
</tr>
<tr>
<td>pend_avail</td>
<td>Number of dynamic leases that are waiting acknowledgement from the failover partner that it did not reissue the lease.</td>
</tr>
<tr>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>reserved-active</td>
<td>Number of reserved leases that clients are actively using.</td>
</tr>
<tr>
<td>reserved-inactive</td>
<td>Number of reserved leases that clients are not actively using.</td>
</tr>
<tr>
<td><strong>Unavailable</strong></td>
<td></td>
</tr>
<tr>
<td>unavail</td>
<td>Number of unreserved dynamic leases that a client declines or the server marks with an address conflict (usually indicating configurations that need correcting).</td>
</tr>
<tr>
<td>reserved-unavail</td>
<td>Number of reserved leases that a client declines or the server marks with an address conflict (usually indicating configurations that need correcting).</td>
</tr>
<tr>
<td>Utilization Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Deactivated</strong></td>
<td></td>
</tr>
<tr>
<td><em>leased-deactivated</em></td>
<td>Number of dynamic leases that clients are actively leasing (that are not offered, expired, or released), but that an administrator deactivated.</td>
</tr>
<tr>
<td><em>reserved-leased-deactivated</em></td>
<td>Number of reserved leases that clients are actively leasing (that are not offered, expired, or released), but that an administrator deactivated.</td>
</tr>
</tbody>
</table>
Managing Hosts

This chapter explains how to configure hosts in DNS zones. Before you proceed with the concepts in this chapter, read Chapter 14, “Managing Zones,” which explains how to set up the basic properties of a primary and secondary DNS server and its zones.

Related Topics

Managing Hosts in Zones, page 10-1
Adding Additional RRs for the Host, page 10-2
Editing Hosts, page 10-2
Removing Hosts, page 10-3

Managing Hosts in Zones

You can manage the resource records (RRs) for a host by configuring the host rather than the individual RRs. When you define a host, the DNS server automatically creates an Address (A) RR in IPv4, or an AAAA RR in IPv6, for each address you specify. If you specify one or more aliases for the host, the server also creates a Canonical Name (CNAME) RR for each alias. You can also have the server create a Pointer (PTR) RR for the host in the reverse zone for the host, if the reverse zone exists.

Local Basic or Advanced Web UI

| Step 1 | From the Design menu, choose Hosts under the Auth DNS submenu. This opens the List/Add Hosts for Zone page. |
| Tip | You can sort by hostname, IP address, IPv6 address (if appropriate), or alias by clicking the corresponding column heading on the List/Add Host for Zone page. However, for zones with a large number of hosts (more than 50,000), restrict the sort to the hostname. Sorting based on IP address or alias can take significantly longer, and could fail if you exceed the memory capacity of the CCM server. |
| Step 2 | Enter the name of the host and its IPv4 or IPv6 address or comma-separated addresses. |
| Step 3 | If the host has alias names, enter a comma-separated list. |
| Step 4 | If you want to create a corresponding Pointer (PTR) RR for the host and you know that the reverse zone for the host exists, check the Create PTR Records? check box. |
Chapter 10  Managing Hosts

Adding Additional RRs for the Host

Step 5  Click Add Host.
Step 6  To confirm, from the Design menu, choose Forward Zones under the Auth DNS submenu. This opens the List/Add Forward Zones page.
Step 7  Click the Resource Records tab to view RR’s for the selected zone.

Note  If you want to view the list of hosts for a particular zone, click the Hosts tab.

CLI Commands

To create A RRs, alias RRs, and PTR RRs for existing reverse zones in a single operation, use zone name addHost hostname address alias for each host. To list the created zones, use zone name listHosts.

Adding Additional RRs for the Host

You add additional RRs for the host based on the dns edit mode you chose, either staged or synchronous. For details, see the “Adding Resource Records” section on page 16-2. Reload the DNS server if you want these RRs to become active server RRs.

Local Basic or Advanced Web UI

For example, to add additional CNAME RRs, add the alias hostname in the Name field under the Resource Records tab of the List/Add Forward Zones page, choose CNAME from the Type drop-down list, add the canonical name of the host in the Data field, then click Add Resource Record. Note that the DNS specification does not allow a CNAME RR with the same name as that of another RR.

For an MX RR, add the origin hostname in the Name field; choose MX from the Type drop-down list; add the integer preference value, a space, and the domain name of the mail exchanger for the origin host in the Data field; then click Add Resource Record. These entries should appear in the list at the bottom of the page.

CLI Commands

To create a CNAME record, use zone name addRR alias CNAME canonical for protected RRs or zone name addDNSRR alias CNAME canonical for unprotected RRs. To create an MX record, use zone name addRR hostname MX preference mxname for protected RRs or zone name addDNSRR hostname MX preference mxname for unprotected RRs.

Editing Hosts

Editing a host involves:

• Adding additional addresses or aliases
• Modifying its Resource Records (RR’s).
Local Basic or Advanced Web UI

Step 1  From the **Design** menu, choose **Hosts** under the **Auth DNS** submenu. This opens the List/Add Hosts for Zone page.

If you have multiple zones configured, select the zone from the list of zones in the Hosts pane on the left.

Step 2  Click the hostname to add additional IP addresses or aliases, and click **Save**.

Step 3  To modify the RRs, click the **Edit RRs** button to open the Edit View RR List page.

CLI Commands

To edit the host, you must remove and reenter its RRs by using `zone name removeRR name type data` or `zone name removeDNSRR name type data`, then `zone name addRR name ttl class type data` or `zone name addDNSRR name ttl type data`.

Removing Hosts

Removing a host removes all A, CNAME, and PTR RRs for that host.

Local Basic or Advanced Web UI

On the List/Add Hosts in Zone page (see the “Editing Hosts” section on page 10-2 for the possible ways to get there), click the **Delete** icon next to the host you want to remove, then confirm the deletion.

CLI Commands

Remove the host by using `zone name removeHost`, then re-add it by using `zone name addHost`.
Managing Owners and Regions

This chapter explains how to configure owners and regions that can be applied to DHCP address blocks, subnets, prefixes, links, and zones.

Related Topics
- Managing Owners, page 11-1
- Managing Regions, page 11-2
- Pushing Owners or Regions to Local Clusters, page 11-3
- Pulling Owners and Regions from the Replica Database, page 11-3

Managing Owners

You can create owners to associate with address blocks, subnets, prefixes, links, and zones. You can list and add owners on a single page. Creating an owner involves creating a tag name, full name, and contact name.

Local Advanced and Regional Advanced Web UI

**Step 1** From the Administration menu, choose Owners under the Settings submenu to open the List/Add Owners page. The regional cluster also includes pull and push functions.

**Step 2** Click the Add Owner icon in the Owners pane on the left. This opens the Add Owner page.

**Step 3** Enter a unique owner tag.

**Step 4** Enter an owner name.

**Step 5** Enter an optional contact name.

**Step 6** Click Add Owner.

**Step 7** To edit an owner, click its name in the Owners pane on the left.

CLI Commands

Use `owner tag create name` to create an owner. For example:

```bash
nrcmd> owner owner-1 create "First Owner" contact="Contact at owner-1"
```
Managing Regions

You can create regions to associate with address blocks, subnets, prefixes, links, and zones. You can list and add regions on a single page. Creating a region involves creating a tag name, full name, and contact name.

Local Advanced and Regional Advanced Web UI

Step 1  From the Administration menu, choose Regions under the Settings submenu to open the List/Add Regions page. The regional cluster also includes pull and push functions.

Step 2  Click the Add Regions icon in the Regions pane on the left.

Step 3  Enter a unique region tag.

Step 4  Enter a region name.

Step 5  Enter an optional contact name.

Step 6  Click Add Region.

Step 7  To edit a region, click its name in the Regions pane on the left.

CLI Commands

Use region tag create name. For example:

\[\text{nrcmd> region region-1 create "Boston Region" contact="Contact at region-1"}\]

Centrally Managing Owners and Regions

As a regional or local CCM administrator, you can:

- Push owners and regions to local clusters.
- Pull local cluster owners and regions to the central cluster.

Each of these functions involves having at least one regional CCM administrator subrole defined (see the “Roles, Subroles, and Constraints” section on page 5-2).

Table 11-1 describes the subroles required for these operations.

<table>
<thead>
<tr>
<th>Central Administrator Management Action</th>
<th>Required Regional Subroles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create, modify, pull, push, or delete owners or regions</td>
<td>owner-region</td>
</tr>
</tbody>
</table>

Related Topics

Pushing and Pulling Owners or Regions, page 11-3
Pushing and Pulling Owners or Regions

You can push owners or regions to, and pull them from, local clusters on the List/Add Owners page or List/Add Regions page, respectively, in the regional cluster web UI.

Related Topics

Pushing Owners or Regions to Local Clusters, page 11-3
Pulling Owners and Regions from the Replica Database, page 11-3

Pushing Owners or Regions to Local Clusters

Pushing owners or regions to local clusters involves choosing one or more clusters and a push mode.

Regional Web UI

Step 1 From the Administration menu, choose Owners or Regions under the Settings submenu.

Step 2 On the List/Add Owners or List/Add Regions page, click the Push All icon in the left pane, or click Push at the top of the Edit Owner page or Edit Region page, for a particular owner or region. This opens the Push Owner or Push Region page.

Step 3 Choose a push mode using one of the Data Synchronization Mode radio buttons.

- If you are pushing all the owners or regions, you can choose Ensure, Replace, or Exact.
- If you are pushing a single owner or region, you can choose Ensure or Replace.
  
  In both the above cases, Ensure is the default mode.

  Choose Replace only if you want to replace the existing owner or region data at the local cluster.
  Choose Exact only if you want to create an exact copy of the owner or region data at the local cluster, thereby deleting all owners or regions that are not defined at the regional cluster.

Step 4 Choose one or more local clusters in the Available field of the Destination Clusters and move it or them to the Selected field.

Step 5 Click Push Data to Clusters.

Step 6 On the View Push Owner Data Report or View Push Region Data Report page, view the push details, then click OK to return to the List/Add Owners or List/Add Regions page.

Pulling Owners and Regions from the Replica Database

When you pull an owner or region, you are actually pulling it from the regional cluster replica database. Creating the local cluster initially replicates the data, and periodic polling automatically updates the replication. However, to ensure that the replica data is current with the local cluster, you can force an update before pulling the data.

Regional Web UI

Step 1 From the Administration menu in the regional cluster web UI, choose Owners or Regions under the Settings submenu.
Step 2 On the List/Add Owners or List/Add Regions page, click the **Pull Replica** icon in the left pane. This opens the Select Replica Owner Data to Pull or Select Replica Region Data to Pull page.

Step 3 Click the **Replicate** icon in the Update Replica Data column for the cluster. (For the automatic replication interval, see the “Replicating Local Cluster Data” section on page 6-11.)

Step 4 Choose a replication mode using one of the Mode radio buttons.

Leave the default Replace mode enabled, unless you want to preserve any existing owner or region properties at the local cluster by choosing Ensure.

**Note** We do not recommend that you create an exact copy of the owner or region data at the local cluster by choosing Exact.

Step 5 Click **Pull All Owners** or **Pull All Regions** next to the cluster, or expand the cluster name and click **Pull Owner** or **Pull Region** to pull an individual owner or region in the cluster.

Step 6 On the Report Pull Replica Owners or Report Pull Replica Regions page, click **Run**.

Step 7 On the Run Pull Replica Owners or Run Pull Replica Region page, view the change set data, then click **OK**. You return to the List/Add Owners or List/Add Regions page with the pulled owners or regions added to the list.
Managing Reports

This chapter explains how to manage the Cisco Prime IP Express address space reporting tool, which is available from a regional cluster by using the web UI. Before you proceed with this chapter, become familiar with the concepts in the previous chapters of the Guide.

Related Topics

ARIN Reports and Allocation Reports, page 12-1
Managing ARIN Reports, page 12-1

ARIN Reports and Allocation Reports

Using the Cisco Prime IP Express web UI, you can generate:

- American Registry of Internet Numbers (ARIN) reports, including:
  - Organization and point of contact (POC) reports
  - IPv4 address space utilization reports
  - Shared WHOIS project (SWIP) allocation and assignment reports
- Allocation reports that show how addresses are deployed across the routers and router interfaces of your network, including:
  - Allocation by owner reports
  - Allocation by router interface or by network reports

Managing ARIN Reports

ARIN, which is one of the five Regional Internet Registries (RIRs), manages IP resources in Canada, the United States of America, and many Caribbean and North Atlantic islands.

ARIN allocates blocks of IP addresses to Internet Service Provider (ISP), which, in turn, reassign blocks of address space to their customers. ARIN distinguishes between allocating IP address space and assigning IP address space. It allocates address space to smaller IRs for subsequent distribution to the IRs’ members and customers. It assigns address space to an ISP, or other organization, for use only within the network of that organization and only for the purposes documented in its requests and reports to ARIN.
Note

ARIN manages IP address resources under the auspices of the Internet Corporation for Assigned Names and Numbers (ICANN). In other geographies, ICANN has delegated authority for IP resources to different regional Internet Registries. Cisco Prime IP Express does not currently support the reports that these registries might require, nor does it now support IPv6 reports or autonomous system (AS) numbers.

ARIN maintains detailed documentation about its policies and guidelines on its website.

http://www.arin.net

Be sure that you are familiar with these policies and guidelines before proceeding with ARIN reports.

The three options that you can specify for ARIN reports are:

- **New**—For a newly added POC or organization.
- **Modify**—Includes changed POC or organization data, such as phone numbers and addresses.
- **Remove**—Signals that you want to remove the POC or organization from the ARIN database.

Related Topics

Managing Point of Contact and Organization Reports, page 12-2
Managing IPv4 Address Space Utilization Reports, page 12-5
Managing Shared WHOIS Project Allocation and Assignment Reports, page 12-6

Managing Point of Contact and Organization Reports

Cisco Prime IP Express provides reports that can submit Points of Contact (POC) and organizational information to ARIN. After you fill in these reports, you need to e-mail the information to ARIN. Submit the POC report (also called a template) to ARIN before preparing other reports.

Each POC is uniquely identified by a name called a POC handle and is associated with one or more Organization Identifiers (Org IDs) or resource delegations, such as an IP address space allocation or assignment. A POC handle, which ARIN assigns, can represent either an individual or a role.

The Organization report creates an Org ID and associates POC records with it. Create the Organization report after you create the POC report.

To manage POC and organization reports, log into the Cisco Prime IP Express regional web UI as a member of an administrator group assigned to the regional-addr-admin role.

Related Topics

Creating a Point of Contact Report, page 12-2
Registering a Point of Contact, page 12-3
Editing a Point of Contact Report, page 12-3
Creating an Organization Report, page 12-4
Registering an Organization, page 12-4
Editing an Organization Report, page 12-5

Creating a Point of Contact Report

You create POCs so that managers can interact with ARIN to request and administer IP resources and so that network professionals can manage network operation issues.
Regional Web UI

**Step 1**  From the Administration menu, choose Contacts under the Settings submenu to open the List ARIN Points of Contact page.

**Step 2**  Click the Add Contact icon in the Contacts pane on the left, to open the Add Point of Contact page.

**Step 3**  Enter data in the fields on the page:
- **Name**—A unique identifier for the POC (required).
- **First Name**—The first name of the point of contact (required).
- **Last Name**—The last name of the point of contact (required).
- **Type**—From the drop-down list, choose Person or Role (optional, with preset value Person).

**Step 4**  Click Add Point of Contact.

**Registering a Point of Contact**

You must register the POC with ARIN to receive a POC handle.

Regional Web UI

**Step 1**  From the Administration menu, choose Contacts under the Settings submenu to open the List ARIN Points of Contact page.

**Step 2**  Click the required contact in the Contacts pane on the left.

**Step 3**  Click the Register Report tab to view the ARIN template file.

**Step 4**  Copy and paste the template file into an e-mail and send the file to ARIN.

**Editing a Point of Contact Report**

Edit a POC report after ARIN returns a POC handle to your organization or if your POC has changed.

Regional Web UI

**Step 1**  From the Administration menu, choose Contacts under the Settings submenu to open the List ARIN Points of Contact page.

**Step 2**  Click the required contact in the Contacts pane on the left. The Edit Point of Contact page opens.

**Step 3**  Enter values for Middle Name, Handle, and Description (optional)

**Step 4**  In the Poc Emails field:
- a. Enter the e-mail address for the POC.
- b. Click Add Email Address to add additional e-mail addresses.
Managing ARIN Reports

Chapter 12   Managing Reports

Step 5  In the Poc Phones field:
  a. Enter a phone number and extension, if applicable, then choose a type (Office, Mobile, Fax, or Pager) from the drop-down list,
  b. Click Add Phone to add additional telephones.

Step 6  Miscellaneous Settings. Add these additional attributes as strings or lists of text.

Step 7  After making the changes, click Save.

Creating an Organization Report

Each organization is represented in the ARIN WHOIS database by a unique Org ID, consisting of an organization name, its postal address, and its POCs. While organizations may have more than one Org ID, ARIN recommends consolidating IP address resources under a single Org ID.

If you do not have an Org ID with ARIN, or you are establishing an additional Org ID, you must first create and submit a POC report. When ARIN confirms it has received your POC information, use Cisco Prime IP Express to complete an Organization form and submit that information.

Regional Web UI

Step 1  From the Administration menu, choose Organizations under the Settings submenu to open the List/Add ARIN Organizations page.

Step 2  Click the Add Organization icon in the Organizations pane on the left, to open the Add Organization page.

Step 3  Enter data in the fields on the page:
  • Organization Name—Name of the organization that you want to register with ARIN.
  • Description—A text description of the organization.
  • Organization Admin POC—From the drop-down list, choose the POC who administers IP resources from the drop-down list.
  • Organization Technical Points Of Contact—From the drop-down list, choose one or more POCs who manage network operations, or click Add Point of Contact to add new contact information.

Step 4  Click Add Organization. This opens the Edit Organization page where you can add more details.

Registering an Organization

You must register your Organization with ARIN to receive an Organization ID.

Regional Web UI

Step 1  From the Administration menu, choose Organizations under the Settings submenu to open the List ARIN Organizations page.

Step 2  Click the required organization in the Organizations pane on the left.

Step 3  Click the Register Report tab to view the ARIN template file.
Step 4  Copy and paste the template file into an e-mail and send the file to ARIN.

**Editing an Organization Report**

You might need to change organizational information that you have registered with ARIN.

**Regional Web UI**

Step 1  From the **Administration** menu, choose **Organizations** under the **Settings** submenu to open the List ARIN Organizations page.

Step 2  Click the required organization in the Organizations pane on the left.

Step 3  Enter or change data in the fields.

- **Miscellaneous Settings**—Add these additional attributes as strings or lists of text.
- **Organization Abuse Points of Contact**—From the drop-down list, choose one or more POCs who handle network abuse complaints, or click **Add Point of Contact** to add new contact information.
- **Organization NOC Points of Contact**—From the drop-down list, choose one or more POCs in network operations centers, or click **Add Point of Contact** to add new contact information.

Step 4  Click **Save**.

Step 5  Submit the updated report to ARIN as described in the “Registering an Organization” section on page 12-4.

**Managing IPv4 Address Space Utilization Reports**

Address space utilization reports serve two purposes:

- To make an initial request for IPv4 address space after you receive a POC handle and an Org ID.
- To support a request for an additional allocation of IPv4 addresses when your business projections show that you are running out of IP addresses.

**Note**

The ARIN website contains extensive information about how it initially allocates address space and its threshold criteria for requesting additional address space. In general, for a single-homed organization, the minimum allocation from ARIN is a /20 block of addresses. For a multihomed organization, the minimum allocation is a /22 block of addresses. ARIN recommends that an organization requiring a smaller block of addresses contact an upstream ISP to obtain addresses.

The Cisco Prime IP Express utilization report corresponds to the ARIN ISP Network Request template (ARIN-NET-ISP-3.2.2).

**Regional Web UI**

Step 1  From the **Operate** menu, choose **ARIN Address Space Usage** under the **Reports** submenu to open the Select Address Space Report page.
Managing ARIN Reports

Step 2  In the Select the Report Type field, choose **Utilization** from the drop-down list.
The Select the Filter Type field is updated with the value, **by-owner**. The browser redisplay the Select Address Space Report page with two new fields: Network Name and Network Prefix Length.

Step 3  In the Select Owner field, choose the owner of this address block from the drop-down list.

Step 4  Enter values for the Network Name and Network Prefix Length.

Step 5  Click **Generate Report**. The browser displays an ARIN template file (ARIN-NET-ISP-3.2.2).
Several sections of the report require that you manually enter data because the information is generated and maintained outside the Cisco Prime IP Express application.

Step 6  Click **Save Report**. The browser displays the Address Space Utilization Report as an unformatted text file.

Step 7  Copy the Address Space Utilization Report to a text editor to manually enter the data that Cisco Prime IP Express does not generate.

Step 8  Copy and paste the edited report into an e-mail and send the file to ARIN.

Managing Shared WHOIS Project Allocation and Assignment Reports

The ARIN shared WHOIS project (SWIP) provides a mechanism for finding contact and registration information for resources registered with ARIN. The ARIN database contains IP addresses, autonomous system numbers, organizations or customers that are associated with these resources, and related POCs.

The ARIN WHOIS does not locate any domain- or military-related information. Use whois.internic.net to locate domain information, and whois.nic.mil for military network information.

The regional web UI also provides two allocation and assignment report pages:
- View ARIN SWIP Reallocated Report
- View ARIN SWIP Reassigned Report

Managing BYOD Reports

There are two types of BYOD reports:
- Registered Devices
- Scopes

Registered Devices

Registered Device report displays the list of devices that are registered through BYOD web server. The report can be exported in the csv format. Only an admin user is allowed to delete a device using the Registered Device Report page.

Scopes

Scope report displays the list of scopes that are used for BYOD. The report can be exported in the csv format.
PART 4

Domain and Zone Administration
Introduction to the Domain Name System

The Domain Name System (DNS) handles the growing number of Internet users. DNS translates names, such as www.cisco.com, into IP addresses, such as 192.168.40.0 (or the more extended IPv6 addresses), so that computers can communicate with each other. DNS makes using Internet applications, such as the World Wide Web, easy. The process is as if, when phoning your friends and relatives, you could autodial them based on their names instead of having to remember their phone numbers.

Related Topics

How DNS Works, page 13-1
Domains, page 13-2
Nameservers, page 13-5
Reverse Nameservers, page 13-6
Authoritative and Caching DNS servers, page 13-7
High-Availability DNS, page 13-7
DNS Database, page 13-7
DNS Views, page 13-7
About EDNS, page 13-8

How DNS Works

To understand how DNS works, imagine a typical user, John, logging in to his computer. He launches his web browser so that he can view the website at a company, ExampleCo (see Figure 13-1 on page 13-2). He enters the name of their website—http://www.example.com. Then:

1. John’s workstation sends a request to the DNS server about the IP address of www.example.com.
2. The DNS server checks its database to find that www.example.com corresponds to 192.168.1.4.
3. The server returns this address to John’s browser.
4. The browser uses the address to locate the website.
5. The browser displays the website on John’s monitor.
Domains

John can access the ExampleCo website because his DNS server knows the www.example.com IP address. The server learned the address by searching through the domain namespace. DNS was designed as a tree structure, where each named domain is a node in the tree. The top-most node of the tree is the DNS root domain (.), under which there are subdomains, such as .com, .edu, .gov, and .mil (see Figure 13-2 on page 13-2).

The fully qualified domain name (FQDN) is a dot-separated string of all the network domains leading back to the root. This name is unique for each host on the Internet. The FQDN for the sample domain is example.com., with its domain example, parent domain .com, and root domain “.” (dot).

Related Topics

Learning ExampleCo Address, page 13-3
Establishing a Domain, page 13-3
Difference Between Domains and Zones, page 13-3
Learning ExampleCo Address

When John’s workstation requests the IP address of the website www.example.com (see Figure 13-3 on page 13-3):

1. The local DNS server looks for the www.example.com domain in its database, but cannot find it, indicating that the server is not authoritative for this domain.
2. The server asks the authoritative root nameserver for the top-level (root) domain “.” (dot).
3. The root nameserver directs the query to a nameserver for the .com domain that knows about its subdomains.
4. The .com nameserver determines that example.com is one of its subdomains and responds with its server address.
5. The local server asks the example.com nameserver for the www.example.com location.
6. The example.com nameserver replies that its address is 192.168.1.4.
7. The local server sends this address to John’s Web browser.

Establishing a Domain

ExampleCo has a website that John could reach because it registered its domain with an accredited domain registry. ExampleCo also entered its domain name in the .com server database, and requested a network number, which defines a range of IP addresses.

In this case, the network number is 192.168.1.0, which includes all assignable hosts in the range 192.168.1.1 through 192.168.1.254. You can only have numbers 0 through 255 (2^8) in each of the address fields, known as octets. However, the numbers 0 and 255 are reserved for network and broadcast addresses, respectively, and are not used for hosts.

Difference Between Domains and Zones

The domain namespace is divided into areas called zones that are points of delegation in the DNS tree. A zone contains all domains from a certain point downward, except those for which other zones are authoritative.
A zone usually has an authoritative nameserver, often more than one. In an organization, you can have many nameservers, but Internet clients can query only those that the root nameservers know. The other nameservers answer internal queries only.

The ExampleCo company registered its domain, example.com. It established three zones—example.com, marketing.example.com, and finance.example.com. ExampleCo delegated authority for marketing.example.com and finance.example.com to the DNS servers in the Marketing and Finance groups in the company. If someone queries example.com about hosts in marketing.example.com, example.com directs the query to the marketing.example.com nameserver.

In Figure 13-4, the domain example.com includes three zones, with the example.com zone being authoritative only for itself.

ExampleCo could choose not to delegate authority to its subdomains. In that situation, the example.com domain is a zone that is authoritative for the subdomains for marketing and finance. The example.com server answers all outside queries about marketing and finance.

As you begin to configure zones by using Cisco Prime IP Express, you must configure a nameserver for each zone. Each zone has one primary server, which loads the zone contents from a local configuration database. Each zone can also have any number of secondary servers, which load the zone contents by fetching the data from the primary server. Figure 13-5 shows a configuration with one secondary server.
Nameservers

DNS is based on a client/server model. In this model, nameservers store data about a portion of the DNS database and provide it to clients that query the nameserver across the network. Nameservers are programs that run on a physical host and store zone data. As administrator for a domain, you set up a nameserver with the database of all the resource records (RRs) describing the hosts in your zone or zones (see Figure 13-6 on page 13-5).

Figure 13-6  Client/Server Name Resolution

The DNS servers provide name-to-address translation, or name resolution. They interpret the information in a fully qualified domain name (FQDN) to find its address.

Each zone must have one primary nameserver that loads the zone contents from a local database, and a number of secondary servers, which load a copy of the data from the primary server (see Figure 13-7 on page 13-6). This process of updating the secondary server from the primary server is called a zone transfer.

Even though a secondary nameserver acts as a kind of backup to a primary server, both types of servers are authoritative for the zone. They both learn about hostnames in the zone from the zone authoritative database, not from information learned while answering queries. Clients can query both servers for name resolution.

As you configure the Cisco Prime IP Express DNS nameserver, you specify what role you want the server to perform for a zone—primary, secondary, or caching-only. The type of server is meaningful only in context to its role. A server can be a primary for some zones and a secondary for others. It can be a primary or secondary only, or it can serve no zones and just answer queries by means of its cache.

In Cisco Prime IP Express, the authoritative and caching services are separated and are handled by two separate servers. The authoritative server holds authoritative zone data and responds only to queries for which it is authoritative. The caching server is the recursive/caching server and does not contain any authoritative zone data.
Reverse Nameservers

The DNS servers described so far perform name-to-address resolution. They can do this easily by searching through their database for the correct address, because they index all the data by name. However, there are times when you need address-to-name resolution so that you can interpret certain output, such as computer log files.

Finding a domain name when you only know the address, however, would require searching the entire namespace. DNS solves this problem by supporting a domain namespace that uses addresses as names, known as the in-addr.arpa domain. This reverse zone contains subdomains for each network based on the network number. For consistency and natural grouping, the four octets of a host number are reversed.

The IP address as a domain name appears backward, because the name is in leaf-to-root order. For example, the ExampleCo example domain network number is 192.168.1.0. Its reverse zone is 1.168.192.in-addr.arpa. If you only know the DNS server address (192.168.1.1), the query to the reverse domain would find the host entry 1.1.168.192.in-addr.arpa that maps back to example.com.

Reverse domains are handled through Pointer (PTR) RRs, as indicated in Figure 13-8.
Authoritative and Caching DNS servers

The DNS server functionality is enhanced to provide separate DNS servers for authorization and caching. With this enhancement, Cisco Prime IP Express supports DNS64, DNSSEC, Domain Redirect, full IPv6, and has improved caching performance.

High-Availability DNS

Because there can be only one primary DNS server per zone, you risk the failure of dynamic updates if the primary DNS server goes down. These updates can occur on the primary DNS server only; a secondary DNS server cannot record these changes, but must forward them to the primary. To solve this problem, a second primary server can become a hot standby that shadows the main primary. This is called High-Availability (HA) DNS (see Chapter 19, “Configuring High-Availability DNS Servers”). Both servers in this failover configuration must synchronize so that their primary zones and related attributes are identical. Cisco Prime IP Express provides settings on the main server to identify the main and backup for synchronization, and the timeout period to go over into failover mode.

DNS Database

In Cisco Prime IP Express 8.2, the DNS database is restructured to enhance the functioning of the DNS server. The DNS authoritative database is replaced with the new resource record database and three new databases are added to the DNS operational database. With this enhancement, Cisco Prime IP Express eases the server failure recovery, backup, and restore process. The new database structure also supports DNS views, simplified DNS configuration, and so on.

The following three new DNS databases have been added:

- **Resource Record Database**—Replaces the DNS authoritative database and stores the resource added or modified by RR FQDN and the zone OID. Both the DNS and the CCM servers access the same resource record database through a common API.
- **Resource Record Index Database**—Stores names of protected resource records. This database is updated every time a protected resource record is added or deleted.
- **Zone State Database**—Stores the zone state configuration. This database is also updated along with the resource record database.

DNS Views

DNS zone views allows you to present alternate versions of zone data to different communities of clients using a single name server.

For example, a DNS server for example.com could maintain two views of the zone, where the view of example.com that can be queried internally includes many hosts that do not exist in the external view. Each zone view is treated as an independent copy of the zone. The DNS server, when answering queries on the zone, uses the match criteria defined in each view to determine the matching zone for the client. The query is then answered based on that zone contents. In some cases, the zone contents may only vary slightly between views. For more information, see Chapter 15, “Configuring DNS Views”.

About EDNS

To send a DNS message above 512 bytes over UDP, you need to use an extension of the DNS protocol known as Extended DNS (EDNS). The EDNS protocol expands the number of flags, label types, and return codes available to the DNS protocol. A version of EDNS specified by RFC 2671 is known as EDNS0. EDNS uses a pseudo resource record known as OPT Resource Record (OPT RR). OPT RR differentiates conventional DNS from EDNS. OPT RRs appear only in the route transmission between DNS clients and servers, they do not appear in the zone files or caches. A DNS endpoint that marks a DNS packet as EDNS must insert an OPT RR in the additional data section of the DNS request or response.

The DNS server supports all EDNS0 extensions. You can modify the UDP payload size of the DNS server. The minimum UDP payload size of the DNS server is 512 bytes and the maximum is 4 KB, with the default set to 4 KB.

Note

The DNS Server can handle requests from clients that do not support EDNS0, however, the DNS server is not permitted to use any extended capabilities, when it handles requests from clients that do not support EDNS0. The response to client requests are inserted into a default 512 byte message. To notify clients that the server supports EDNS0, an OPT RR is inserted into the additional section of the DNS message by the server.
Managing Zones

The Domain Name System (DNS) is a distributed database for objects in a computer network. By using a nameserver approach, the network consists of a hierarchy of autonomous domains and zones. The namespace is organized as a tree that often resembles the organizations that are responsible for the administration boundaries. For an introduction to the protocol, see Chapter 13, “Introduction to the Domain Name System.”

The basic function of DNS nameservers is to provide data about network objects by answering queries. You can configure the Cisco Prime IP Express DNS server and zones by accepting the system defaults or changing them.

This chapter describes the basics of configuring the Cisco Prime IP Express DNS servers, and their primary and secondary zones. Chapter 16, “Managing Resource Records,” describes how to manage DNS resource records (RRs) and hosts, and Chapter 17, “Managing Authoritative DNS Server Properties,” describes how to set some of the more advanced zone and DNS server properties.

Each zone will have an explicit view id that defines its view. The list of views for any given zone distribution will be defined by the list of zones. Synchronizing the zone will always sync the associated views and named ACLs for both primary and secondary zones. Chapter 15, “Configuring DNS Views,” describes how to manage DNS views.

Related Topics

Staged and Synchronous Modes, page 14-1
Creating and Applying Zone Templates, page 14-2
Managing Primary DNS Servers, page 14-4
Managing Secondary Servers, page 14-13
Managing DNS ENUM Domain, page 14-15
Adding Subzones, page 14-19
Enabling DNS Updates, page 14-22
Managing Zone Distributions, page 14-22

Staged and Synchronous Modes

You can perform additions or edits to DNS zones, RRs, and hosts in one of two modes in regional cluster—staged or synchronous:

- **Staged**—Changes to zones, their hosts and RRs are written to the CCM regional database. You must synchronize the zones to propagate pending changes to the local DNS servers. When adding or deleting zones, or when updating zone settings, you must also reload the local DNS servers following synchronization.
**Synchronous**—After committing changes to the CCM regional database, they are immediately propagated to the local DNS servers. Changes to hosts and RRs will immediately be active, but similar to staged mode, changes to zones may also require a DNS server reload. If propagation cannot occur because the local servers are unreachable, you must manually synchronize the zones to propagate pending changes once the connection is restored.

The zone distribution determines which local DNS servers will publish the zone. Once assigned, synchronization can occur in synchronous edit mode, on a zone basis, or for all zones in the distribution.

Staged edit mode is the default value for regional servers. To choose the mode, select Session Settings from the **username** menu in the Web UI. The username drop-down list is available at the top right of the window adjacent to the Log Out link.

On local clusters, edits are always performed in synchronous mode. When HA DNS is configured, zone changes are immediately propagated to the HA partner. If propagation cannot occur because the partner unreachable, host and RR changes will be synchronized automatically once the connection is restored, but you must manually synchronize the zones to propagate pending changes.

### Creating and Applying Zone Templates

A zone template is a convenient way to create a boilerplate for primary zones that share many of the same attributes. You can apply a zone template to any zone, and override the zone attributes with those of the template. You can create zone templates in the local and regional cluster web UIs and in the CLI.

**Caution**

Be careful applying a template to an existing zone. The template overwrites all explicitly set attributes for the zone (other than its name), possibly causing severe consequences if the zone is already configured in a network. To make a limited attribute change to multiple zones using a template, be sure to change only that attribute (or attributes), leaving the others unset, before you apply the template to the zones.

### Local Basic or Advanced and Regional Web UI

**Step 1**

From the **Design** menu, choose **Zone Templates** under the **Auth DNS** submenu.

**Step 2**

You can add a zone template at the local and regional clusters, and you can also pull and push zone templates at the regional cluster in the web UI.
Chapter 14  Managing Zones

Creating and Applying Zone Templates

To add a zone template at the local cluster or explicitly add one at the regional cluster, click the **Add Zone Templates** icon in the Zone Templates pane. This opens the Add Zone Template dialog box, enter the name and click **Add Zone Template**.

To make the zone template meaningful, you would enter the suggested serial number, nameserver, contact e-mail address, and list of nameservers, because they are required for the zone itself. You might also want to specify any zone owners or zone distributions. You do not necessarily need to add these values for the zone template, because you can do so for the zone once it is created from the template. However, the template name and zone default TTL are required. (For a description of the minimally required zone attributes, see the “Creating Primary Zones” section on page 14-5.)

After you enter these values, click **Save** at the bottom of the page.

At the regional cluster, to pull a zone template from one or more local clusters, click the **Pull Replica** icon in the Zone Templates pane. This opens the Select Replica Zone Template Data to Pull dialog box which shows a tree view of the regional server replica data for the local clusters’ zone templates. The tree has two levels, one for the local clusters and one for the templates in each cluster. You can pull individual templates from the clusters, or you can pull all of their templates:

- To pull individual zone templates, expand the tree for the cluster, choose a pull criterion next to its name, then click **Pull Zone Template**.
- To pull all the templates from a cluster, choose a pull criterion, then click **Pull All Zone Templates**.
- To update all the replica data for a cluster, click the **Pull Replica** icon.

The pull selection criteria are:

- **Ensure**—Pulls each template, except if an existing template by that name already exists at the regional cluster, in which case it does not overwrite the regional cluster data.
- **Replace**—Pulls each template and overwrites the data for it if it already exists at the regional cluster, without affecting any additional templates at the regional cluster. This is the default and recommended setting.
- **Exact**—Pulls each template, overwrites the data for it if it already exists at the regional cluster, and removes any additional templates at the regional cluster.

At the regional cluster, to push a zone template to one or more local clusters:

- To push all the zone templates on the page List Zone Templates page—Click the **Push All** icon in the Zone Templates pane.
- To push individual zone templates on the page List Zone Templates page—Click **Push**.

Both of these actions open a version of the Push Zone Template Data to Local Clusters page. This page provides a choice of the synchronization mode and the destination clusters. Move the desired cluster or clusters from the Available field to the Selected field, then click one of the data synchronization mode radio buttons:

- **Ensure**—Pushes each template, except if an existing template by that name already exists at the local cluster, in which case it does not overwrite the local cluster data. This is the default and recommended setting.
- **Replace**—Pushes each template and overwrites the data for it if it already exists at the local cluster, without affecting any additional templates at the local cluster.
- **Exact**—Available for “push all” operations only, it pushes each template, overwrites the data for it if it already exists at the local cluster, and removes any additional templates at the local cluster.
Step 3 You can apply the template to a new or existing zone:

a. **New zone**—Select the template from the Template drop-down list when you create the zone, as described in the “Configuring Primary Forward Zones” section on page 14-4.

b. **Existing zone**—After you create a zone (see the “Configuring Primary Forward Zones” section on page 14-4), you can apply the template when you edit the zone on the Edit Zone page. Click the template name in the Template drop-down list, then click **Apply Template**.

**CLI Commands**

Use `zone-template name create` to create the zone template. (See the “Configuring Primary Forward Zones” section on page 14-4 for how to apply the template to a zone.) For example:

```bash
nrcmd> zone-template zone-template-1 create serial=1
```

To apply a template to a zone, use `zone-template name apply-to zone`. Note that the syntax permits one or more comma-separated zones and also the `all` keyword for all zones. You can also clone a template from an existing template by using `zone-template clone-name create clone=template`, and then make adjustments to the clone. For example:

```bash
nrcmd> zone-template zone-template-1 apply-to example.com,boston.example.com
nrcmd> zone-template cloned-template create clone=zone-template-1 owner=owner-1
```

**Managing Primary DNS Servers**

Adding a zone involves creating a domain name. You can also define an owner and use a zone template. If you do not use a template, you must also define the Start of Authority (SOA) and Name Server (NS) properties for the zone.

You do not need to create a loopback zone for the local host, because Cisco Prime IP Express automatically creates one. A loopback zone is a reverse zone that a host uses to resolve its loopback address, 127.0.0.1, to localhost so that it can direct network traffic to itself. The loopback zone is 127.in-addr.arpa, which appears on the list of reverse zones.

**Related Topics**

- Configuring Primary Forward Zones, page 14-4
- Adding Primary Reverse Zones, page 14-11
- Adding Primary Reverse Zones, page 14-11
- Getting Zone Counts on the Server, page 14-13

**Configuring Primary Forward Zones**

This section explains how to configure a primary nameserver with a primary forward zone. When you are done with this procedure, follow the procedure in the “Adding Primary Reverse Zones” section on page 14-11 to configure a reverse zone for each network that you use.
Managing Primary DNS Servers

**Tip**

For an example of adding a forward zone, see the “Create the Zone Infrastructure” section on page 5-28.

### Related Topics

- Creating Primary Zones, page 14-5
- Editing Primary Zones, page 14-7
- Confirming Zone Nameservers, page 14-8
- Synchronizing Zones, page 14-8
- Zone Commands, page 14-8
- Importing and Exporting Zone Data, page 14-9

### Creating Primary Zones

Creating a primary zone requires, at a minimum, adding certain key Start of Authority (SOA) attributes and nameservers for the zone. The advantage of Basic mode in the web UI is that many of these settings are already done for you.

#### Local Basic Web UI

**Step 1** From the **Design** menu, choose **Forward Zones** under the **Auth DNS** submenu to open the List/Add Forward Zones page.

**Step 2** Click the **Add Forward Zone** icon in the Forward Zones pane, enter the zone name (in domain name format).

**Step 3** Enter the name of the nameserver host, such as `ns1`.

**Step 4** Enter the contact e-mail name, such as `hostmaster`.

**Step 5** Click **Add DNS Zone**. Basic mode creates the zone with preset values:

- Zone default TTL—**24h**
- Start of Authority (SOA) serial number—**1**
- SOA secondary refresh time—**3h**
- SOA secondary retry time—**60m**
- SOA secondary expiration time—**1w**
- SOA minimum TTL—**10m**

#### Local Advanced and Regional Web UI

**Step 1** From the **Design** menu, choose **Forward Zones** under the **Auth DNS** submenu to open the List/Add Forward Zones page.

**Step 2** Click the **Add Forward Zone** icon in the Forward Zones pane, enter the zone name (in domain name format).

**Step 3** Enter the name of the nameserver host, such as `ns1`.

**Step 4** Enter the contact e-mail name, such as `hostmaster`.

**Step 5** Enter the serial number.
Step 6 Click **Add Zone**.

Step 7 Choose an owner or region, if necessary, from the drop-down list.

Step 8 Apply an existing zone template, if necessary (see the “Creating and Applying Zone Templates” section on page 14-2). Click the name of the configured template in the drop-down list.

**Caution** Be careful applying a template to a zone that is already live. Explicitly defined attributes on the template replace the existing ones defined for the zone.

Step 9 Modify the top attributes, if necessary:

a. Owner and region

b. Preconfigured zone distribution (see the “Managing Zone Distributions” section on page 14-22)

c. Zone default TTL

Step 10 In the SOA attributes, enter a:

a. Serial number, such as 1.

A primary DNS server uses a serial number to indicate when its database changes and uses any incrementing of this number to trigger a zone transfer to a secondary server. The serial number you can enter here is the suggested one only, and the DNS server does not always accept it. If you edit the serial number to be less than the actual serial number that the server maintains, the server logs a warning message and ignores the suggested serial number. The actual serial number always equals or is higher than the suggested one. You can get the actual serial number by using `zone name get serial` (if the DNS server is running; if the server is not running, or listing or showing the zone attributes, it always returns the suggested serial number), or by refreshing the DNS Server Value for the zone Serial Number attribute. You must explicitly enter this suggested serial number when creating a zone.

b. Nameserver host, such as `ns1`.

Enter either just the hostname or its fully qualified name (such as `ns1.example.com`, but you must end it with a trailing dot). Use the fully qualified name if the primary nameserver is in a different zone. The primary DNS server becomes the ns value in the zone SOA record. You must also specify one or more authoritative nameservers for the zone—these become the Name Server (NS) records for the zone. In the CLI, the primary DNS server automatically becomes the first NS record and also appears as the first entry in the nameservers attribute list.

c. Contact e-mail name, such as `hostmaster`.

The fully qualified contact e-mail name becomes a slightly altered version of the e-mail address in that dots (.) are substituted for the at symbol (@). If using the fully qualified value, end the address with a trailing dot (for example, enter `hostmaster@example.com` as `hostmaster.example.com`). Precede any dot before the @ in the original address with a backslash (\) (for example, enter `hostmaster.marketing@example.com` as `hostmaster\marketing.example.com`).

Step 11 Enter an authoritative nameserver name under Nameservers further down the page, then click **Add Nameserver**.

Authoritative nameservers validate the data in their zones. Both primary and secondary servers can be authoritative. The crucial difference is where they get their zone data. A primary server obtains its data from an administrator, as stored in the server configuration database, and from DNS updates, typically from a DHCP server. A secondary server obtains the zone data from its designated master servers by way of a zone transfer.
You must add at least one nameserver for a zone—Cisco Prime IP Express does not consider the zone data complete unless you do so. The nameservers you list should be those that you want people outside your domain to query when trying to resolve names in your zone. You must add the authoritative nameservers in addition to the primary server for the zone. If the primary DNS server for the zone is in the zone, you must create a host address for it.

For every DNS internal-to-zone nameserver, you must create an Address (A) resource record (RR) to associate the server domain name with an IP address:

- Click **Host** to open the List Zones page.
- Click the zone name to open the List/Add Hosts for Zone page.
- Enter the hostname of the authoritative server.
- Enter its IP address.
- Click **Add Host**. The server hostname and address appear in the list.
- To edit the host, click its name to open the Edit Host page. Click **Modify** to implement the changes.

**Step 12** Configure additional attributes as needed.

**Step 13** Click **Save**.

### CLI Commands

To create a primary zone, use `zone name create primary nameserver contact`. You must specify a primary DNS server; this server becomes the first authoritative DNS nameserver. For example:

```
nrcmd> zone example.com create primary ns1 hostmaster
```

The serial number defaults to 1. You can get the actual serial number by using `zone name get serial` (if the DNS server is running; if the server is not running, or listing or showing the zone attributes, it always returns the suggested serial number).

To add additional authoritative nameservers for the zone, enter a comma-separated list of fully qualified domain names using `zone name set nameservers=list`. Note that only the first server entered is confirmed by the command. Use `zone name show` to show all the server names.

Use `zone name addRR hostname A address` to add the authoritative server hostname and address. To list the host, use `zone name listHosts`. To remove the host, use `zone name removeRR hostname A`.

If you want to apply an existing template while creating a zone, use the `template` attribute. For example:

```
nrcmd> zone example.com create primary ns1 hostmaster template=zone-template-1
```

**Note** In this example, even though you need to specify the nameserver and contact as part of the syntax, the template definition (if any) overwrites them.

To apply a template after creating the zone, use `zone name applyTemplate template`. For example:

```
nrcmd> zone example.com applyTemplate zone-template-1
```

### Editing Primary Zones

You can edit a primary zone to modify its properties, apply a template to it, or use the zone definition to create a template from it.
Local Advanced and Regional Web UI

Step 1  From the Design menu, choose Forward Zones under the Auth DNS submenu to open the List/Add Forward Zones page.
Step 2  Select the zone in the Forward Zones pane to open the Edit Zone page.
Step 3  Make attribute changes as necessary.
Step 4  To apply a template to the zone, choose a template name from the drop-down list at the bottom of the page, then click Apply Template.

Caution  Be careful applying a template to a zone that is already live. Explicitly defined attributes on the template replace the existing ones defined for the zone.

Step 5  To use the zone definitions to create a template from them while modifying the zone, click Modify Zone and Save Template. On the Save New Zone Template page, give the template a name in the Value field, then click Save Zone Template. You return to the List/Add Zones page.

Confirming Zone Nameservers

Confirm your zone NS RR configuration by looking at the RRs that you created.

Local Advanced and Regional Web UI

Select the zone from the Forward Zones pane, and click the Resource Records tab. There should be an A record for each nameserver host in the zone. Edit these records or add more on this page.

See the “Adding Resource Records” section on page 16-2.

CLI Commands

Use zone name listRR to check the RRs you added.

Synchronizing Zones

If a zone needs to be synchronized, in the regional server, click the Zone Sync tab for the Primary Forward/Reverse zone. Click the Sync Zone - Report button to open a Synchronize Zone page. Expert mode includes an additional Sync CCM Hosts from RR Data - Report button.

Manual zone synchronization should only be used when there is an inconsistency between the HA main and HA backup that is not being resolved automatically by the servers.

Zone Commands

The List/Add Zones (Forward/Reverse) page includes a Commands button. When clicked, this opens the Commands dialog box. These commands serve specific purposes:

- Scavenge zone—See the “Scavenging Dynamic Records” section on page 29-19.
- Get scavenge start time—See the “Scavenging Dynamic Records” section on page 29-19.
• **Synchronize HA Zone (Forward Zones)**—See the “Synchronizing HA DNS Zones” section on page 19-5.

**Note** You can see the **Synchronize HA Zone** command only if the server is an HA main server. You cannot see this command if it is an HA backup server.

## Importing and Exporting Zone Data

The easiest and quickest way to create a primary zone is to import an existing BIND format zone file, defined in RFC 1035. You can also export these same kinds of files to another server. BIND 4.x.x uses a boot file, called named.boot, to point the server to its database files. You can import your entire BIND 4.x.x configuration using the `import` command in the CLI. BIND 8 and BIND 9 use a configuration file, called named.conf, with a different syntax.

You can import and export zone data only by using the CLI.

When a BIND file contains an `$INCLUDE` directive, BIND searches for the include file relative to the directory that the directory directive in the named.boot file specifies. In contrast, the `nrcmd` program searches for the include file relative to the directory containing the zone file being processed.

To avoid this problem, ensure that the BIND configuration uses absolute paths whenever specifying an include file in a zone file. If your zone files contain relative paths when specifying include files, and the directory containing the zone file is not the same as the directory that the directory directive in the named.boot file specifies, your configuration cannot load properly. You need to convert the relative paths in your zone files to absolute paths so that you can import your BIND configuration into Cisco Prime IP Express. Here is an example of a configuration and how to fix paths in directory hierarchy, configuration files, and zone files:

- **Directory hierarchy:**
  ```
  /etc/named.conf  
  /etc/named.boot  
  /usr/local/domain/primary/db.example  
  /usr/local/domain/primary/db.include  
  /usr/local/domain/secondary  
  ```

- **Configuration file (/etc/named.conf):**
  ```
  #BIND searches for zone files and include files relative to /usr/local/domain  
  option directory /usr/local/domain  
  #BIND finds zone file in /usr/local/domain/primary  
  zone example.com {  
      type master;  
      file primary/db.example;  
  }  
  #end of /etc/named.conf
  ```

- **Configuration file (/etc/named.boot):**
  ```
  #BIND searches for zone files and include files relative to /usr/local/domain  
  directory /usr/local/domain  
  #BIND finds zone file in /usr/local/domain/primary  
  primary example.com primary/db.example  
  #end of /etc/named.boot
  ```

- **Incorrect zone file (/usr/local/domain/primary/db.example):**
  ```
  #BIND searches for include file relative to /usr/local/domain  
  $INCLUDE primary/db.include  
  #end of /usr/local/domain/primary/db.example
  ```
To make the configuration loadable, change the relative path ($INCLUDE primary/db.include) in the file db.example to an absolute path ($INCLUDE /usr/local/domain/primary/db.include).

Table 14-1 describes the named.boot and named.conf file directives that BIND 4 and BIND 9 support, and the corresponding Cisco Prime IP Express user interface location or syntax, if any.

**Table 14-1  BIND-to-CLI Command Mappings**

<table>
<thead>
<tr>
<th>BIND 4 Command</th>
<th>BIND 9 Command</th>
<th>Mapping to User Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl name {</td>
<td>acl name create</td>
<td>Web UI: List/Add Access</td>
</tr>
<tr>
<td>addr-match-list</td>
<td>match-list=addr-match-list</td>
<td>Control Lists page fields</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>(see the “Access Control Lists” section on page 29-9).</td>
</tr>
<tr>
<td>key id {</td>
<td>key name create</td>
<td>Web UI: List/Add Encryption</td>
</tr>
<tr>
<td>algorithm string; secret string ;</td>
<td>secret algorithm=alg</td>
<td>Keys page fields.</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: acl name create value</td>
</tr>
<tr>
<td>limit transfers-in num</td>
<td>options { transfers-in num ;</td>
<td>Web UI: Edit DNS Server page, set xfer-client-concurrent-limit.</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: session set visibility=3</td>
</tr>
<tr>
<td>options listen-on port</td>
<td>options { listen-on port {addr-match-list} ;</td>
<td>Web UI: Edit DNS Server page, set Listening port.</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: dns set local-port-number=port</td>
</tr>
<tr>
<td>options max-cache-ttl num</td>
<td>options { max-cache-ttl num ;</td>
<td>Web UI: Edit DNS Server, set Max. RR caching TTL.</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: dns set max-cache-ttl=num</td>
</tr>
<tr>
<td>options no-fetch-glue</td>
<td>options { fetch-glue no ;</td>
<td>Web UI: Edit DNS Server page, enable Don’t fetch missing glue records.</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: dns enable no-fetch-glue</td>
</tr>
<tr>
<td>options notify yes</td>
<td>options { notify yes ;</td>
<td>Web UI: Edit DNS Server page, enable Send zone change notification (NOTIFY).</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: dns enable notify</td>
</tr>
<tr>
<td>options rrset-order order order ...</td>
<td>options { rrset-order order ; order ; ... ;</td>
<td>Web UI: Edit DNS Server page, enable Enable round-robin.</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: dns enable round-robin</td>
</tr>
<tr>
<td>options support-ixfr yes</td>
<td>options { request-ixfr yes ;</td>
<td>Web UI: Edit DNS Server page, enable Request incremental transfers (IXFR).</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: dns enable ixfr-enable</td>
</tr>
<tr>
<td>options transfer-format many-answers</td>
<td>options { transfer-format many-answers ;</td>
<td>Web UI: Edit DNS Server page, enable Use multirec format for zone transfers.</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: dns enable axfr-multirec-default</td>
</tr>
<tr>
<td>primary zonename file</td>
<td>zone &quot;name&quot; { type master; ;</td>
<td>Web UI: Add Zone page fields.</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: zone name create primary file=file</td>
</tr>
<tr>
<td>secondary zonename addr list [backupfile]</td>
<td>zone &quot;name&quot; { type slave; ;</td>
<td>Web UI: Add Secondary Zone page fields.</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td>CLI: zone name create secondary ip-addr [ip-addr...</td>
</tr>
</tbody>
</table>
Adding Primary Reverse Zones

For a correct DNS configuration, you must create a reverse zone for each network that you use. A reverse zone is a primary zone that DNS clients use to convert IP addresses back to hostnames, and resides in a special in-addr.arpa domain. You can create a reverse zone manually or import it from BIND. You can also create reverse zones from subnets (see the “Adding Reverse Zones from Subnets” section on page 14-12).

Related Topics

Adding Reverse Zones as Zones, page 14-11
Adding Reverse Zones from Subnets, page 14-12

Adding Reverse Zones as Zones

You can manually add a reverse zone as a zone.

Local Basic or Advanced and Regional Web UI

From the Design menu, choose Reverse Zones under the Auth DNS submenu to open the List/Add Reverse Zones page. This page is almost identical to the List/Add Forward Zones page. Then, add a reverse zone the same way you would add a forward zone, as described in the “Configuring Primary Forward Zones” section on page 14-4, except use the reverse of the forward zone network number added to the special in-addr.arpa domain as the zone name. Use the same template or SOA and nameserver values as you used for the related forward zone.

You can enter a DHCPv4 subnet or DHCPv6 prefix value in the Name field, which converts the subnet or prefix into an appropriate reverse zone name.

To create a reverse zone by using an IPv4 subnet or an IPv6 prefix, do the following:

---

**Step 1** From the Design menu, choose Reverse Zones under the Auth DNS submenu.

**Step 2** In the List/Add Reverse Zones page, click the Add Reverse Zone icon in the Reverse Zones pane, enter values in the Name field, for example:

- 209.165.201.1/24—Creates a reverse zone by using an IPv4 subnet.
- 2001:db8:ff80::/64—Creates a reverse zone by using an IPv6 prefix.

**Step 3** Enter the required fields to create the reverse zone:

- Nameserver—Enter ns1.example.com. (include the trailing dot).
Managing Primary DNS Servers

Step 4 Click **Add Reverse Zone**. The List/Add Reverse Zones page appears.

To create a reverse zone by using the name of an IPv6 prefix, do the following:

**Step 1** From the **Design** menu, choose **Prefixes** under the **DHCPv6** submenu.

**Step 2** Click the **Add Prefixes** icon in the Prefixes pane to open the Add IPv6 Prefix dialog box.

**Step 3** Enter a prefix name (for example, **prefix-1**) and address (for example, **2001:db8:ff80:ff80::**).

**Step 4** Choose a prefix length from the drop-down list (for example, **64**).

**Step 5** Click **Add IPv6 Prefix**. The prefix is added to the List/Add DHCP v6 Prefixes page.

To create a reverse zone from the prefix,

a. Click the **Reverse Zone** tab.

b. Select a zone template

c. Click **Report**, and then click **Run**.

**CLI Commands**

Use `zone name create primary` and `zone name addRR PTR` to add the primary reverse zone and pointer records for the server. You can also apply a zone template.

To create a reverse zone by using:

- An IPv4 subnet
  For example, you can enter:
  ```
  nrcmd> zone 209.165.201.1/24 create primary ns1.example.com. hostmaster.example.com.
  ```

- An IPv6 prefix
  For example, you can enter:
  ```
  nrcmd> zone 2001:db8::/64 create primary ns1.example.com. hostmaster.example.com.
  ```

- The name of an IPv6 prefix
  For example, you can enter:
  ```
  nrcmd> prefix prefix-1 create 2001:db8:ff80:ff80::/64
  nrcmd> zone prefix-1 create primary ns1.example.com. hostmaster.example.com.
  ```

**Adding Reverse Zones from Subnets**

An alternative to creating reverse zones manually is to create them from existing subnets. You can do this in the web UI only.
Managing Secondary Servers

When you configure a zone, choose at least one secondary server. If you have only one nameserver and it becomes unavailable, there is nothing that can look up names. A secondary server splits the load with the primary or handles the whole load if the primary is unavailable. When a secondary server starts up, it contacts the primary and pulls the zone data over. This is known as a zone transfer.

You can configure a secondary DNS server to be responsible for a secondary zone, which makes the server a secondary for that zone. You also need to give the address of the master server from which to perform zone transfers. Cisco Prime IP Express must know about this master server.

Related Topics

Adding Secondary Forward Zones, page 14-13
Enabling Zone Transfers, page 14-14

Adding Secondary Forward Zones

You can add a secondary forward zone at the local cluster.

Local Basic or Advanced Web UI

Step 1 From the Design menu, choose Secondary Zones under the Auth DNS submenu to open the List/Add Secondary Zones page.

Step 2 Click the Add Secondary Zone icon in the Secondary Zones pane to open the Add Secondary Zone dialog box.
Chapter 14  Managing Zones

Managing Secondary Servers

A secondary zone requires a name and a list of one or more master servers. You can also enable restricting zone transfers to a set of hosts, then enter the access control list (ACL) of the restricted hosts in the restrict-xfer-acl field. Enter other attribute values as necessary.

Step 3  Click Add Secondary Zone.

Clicking the name of the secondary zone in the Secondary Zones pane opens the Edit Secondary Zone page where you can edit the secondary zone. Click Save on this page.

You can add the secondary reverse zone the same way you do a secondary forward zone, except that the address must be a reverse zone address.

CLI Commands

To create a secondary zone, use zone name create secondary address. You must specify the primary DNS server IP address to perform the zone transfer.

For example:

```plaintext
nrcmd> zone shark.zone. create secondary 172.18.123.177
```

If you are using HA DNS server pair, the IP addresses must be provided by separating the addresses with comma. The HA DNS backup server will be used when the primary server is unavailable.

For example:

```plaintext
nrcmd> zone shark.zone. create secondary 172.18.123.177,172.18.123.45
```

Enabling Zone Transfers

A secondary server periodically contacts its master server for changes, called a zone transfer. The interval is defined in the server SOA record as the secondary refresh time. You can restrict zone transfers by setting the restrict-xfer attribute to true (the preset value is false) on the master server. You have to set the restrict-xfer-acl setting accordingly.

Note

If you restrict zone transfers, the nslookup utility ls command may fail because it tries to do a full zone transfer, unless you include the IP address that ls runs from in the zone restrict-xfer-acl list.

Local Advanced and Regional Web UI

Step 1  In the Forward Zones pane, click the name of the primary zone to open the Edit Zone page.

Step 2  In the zone attributes area, you can set the restrict-xfer attribute to false (the preset value). If you set the attribute to true, you can also specify a list of servers to which to restrict the zone transfers by using the restrict-xfer-acl attribute, separating the IP addresses with commas.

Secondary zones can also restrict zone transfers from other secondary zones, so that the restrict-xfer and restrict-xfer-acl attributes are also available for secondary zone configurations.

Step 3  Click Save.

Step 4  You can force zone transfers for the DNS server in two ways:

- On the Secondary Zones pane, click the Full Zone Transfer button.
Managing DNS ENUM Domain

Creating separate ENUM domains simplifies the management of Naming Authority Pointer (NAPTR) Electronic Numbering (ENUM). It simplifies to a great extent the setup and management of E.164 numbers and how available services are connected to the E.164 numbers. When you create an ENUM zone and add the corresponding E.164 numbers, Cisco Prime IP Express automatically creates a forward zone and the respective NAPTR resource records.

Related Topics

- Adding DNS ENUM Domains, page 14-16
- Adding DNS ENUM Numbers, page 14-17
- Pulling and Pushing ENUM Domains, page 14-17
- Pulling ENUM Domains from the Replica Database, page 14-18
- Managing DNS ENUM Defaults, page 14-15

Managing DNS ENUM Defaults

To configure the default ENUM settings, do the following:

**Local Basic or Advanced Web UI**

1. From the **Design** menu, choose **Defaults** under the DNS ENUM submenu to open the Manage DNS ENUM Defaults page.
2. Enter the Top-level Domain.
3. Enter the Local Prefix such as +46.
4. Enter the Default Services values: select a service type, enter a URI, and click **Add Service**.
5. Select a Zone Template.
6. Click **Save**.

**CLI Commands**

Using the CLI, you can set the default ENUM domain, default top-level domain and local prefix, service, and zone template by using:

```
dns-enum-config set [number-prefix prefix | zone-template name]
```
To add the default service, use:
```
dns-enum-config addService <type> <subtype> <uri> [{<order> [preference]}]
```
To remove the default service user, use:
```
dns-enum-config removeService <type> <subtype> <uri>
```

### Adding DNS ENUM Domains

Adding an ENUM domain involves creating a domain name. You can also define an owner and use a zone template.

When you create an ENUM zone, Cisco Prime IP Express automatically creates a forward zone. For example, if you create an ENUM domain for E.164 number prefix 100 and the default top-level domain is set to e164enum.net, a forward zone 0.0.1.e164enum.net is automatically created and appears in the list of forward zones.

This section explains how to configure an ENUM domain.

#### Local and Regional Web UI

1. From the **Design** menu, choose **Domains** under the DNS ENUM submenu to open the List/Add DNS ENUM Domains page.
2. Click the **Add Domains** icon in the Domains pane to open the Add ENUM Domain dialog box.
3. Enter the E.164 number prefix for the domain, such as 897.
4. Enter the name of the nameserver host, such as ns1.
5. Enter the contact e-mail name, such as hostmaster.
6. Click **Add ENUM Domain**. The domain will be created with the default local prefix such as +4689. The Basic mode creates the zone with the following preset values:
   - Zone default TTL-24h
   - Start of Authority (SOA) serial number-1
   - SOA secondary refresh time-3h
   - SOA secondary retry time-60m
   - SOA secondary expiration time-1w
   - SOA minimum TTL-10m

#### CLI Commands

The ENUM domain commands are shown in the table below:

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td><code>dns-enum-domain prefix create [zone-template= name] [nameservers [person]]</code></td>
</tr>
<tr>
<td>Delete</td>
<td><code>dns-enum-domain prefix delete</code></td>
</tr>
</tbody>
</table>
Adding DNS ENUM Numbers

Cisco Prime IP Express supports NAPTR RRs. These records help with name resolution in a particular namespace and are processed to get to a resolution service.

In addition to the option of adding NAPTR resource records, you can now directly add the E.164 numbers and associate the corresponding services with the numbers. When you add a DNS ENUM number, you need to specify either the E.164 number prefix of the parent domain or the Zone templates, and a NAPTR resource record is created for the E.164 number. This approach uses a reversed E.164 number and treats every digit as a node on the DNS name hierarchy. For example, the E.164 address +4689761234 creates a NAPTR RR 4.3.2.1.6.7.9.8 for the +46 E.164 prefix domain.

For more information on NAPTR resource records, see the “Using NAPTR Records” section on page 16-8.

Local and Regional Web UI

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>From the Design menu, choose Numbers under the DNS ENUM submenu to open the List/Add DNS ENUM Numbers page.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Click the Add Numbers icon in the Numbers pane to open the Add ENUM Number dialog box.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Enter the E.164 number along with the E.164 number prefix, such as 1234.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Select a service type, enter a URI, and click Add Service.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Enter the E.164 number prefix for the parent domain.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Select the Zone Template if you have not specified the E.164 prefix.</td>
</tr>
<tr>
<td>Step 7</td>
<td>Select a Ported option and enter the Ported Nameserver FQDN.</td>
</tr>
<tr>
<td>Step 8</td>
<td>Click Add ENUM Number. The number will be created and added under the domain +4689.</td>
</tr>
</tbody>
</table>

CLI Commands

Using the CLI, you can add ENUM number by using:

```
dns-enum-number <number> create <type> <subtype> <uri> [zone-template=name] [domain-prefix]
```

Pulling and Pushing ENUM Domains

You can push ENUM Domains to and pull ENUM Domains from local clusters on the List/Add DNS ENUM Domains page in the regional cluster web UI.

Pushing ENUM Domains to Local Clusters

To push ENUM domains to the local cluster, do the following:

Regional Basic and Advanced Web UI

| Step 1 | From the Design menu, choose Domains under the DNS ENUM submenu to view the List/Add DNS ENUM Domains page in the regional web UI. |
Managing Zones

Managing DNS ENUM Domain

Step 2  Click the **Push All** icon in the Domains pane to push all the ENUM domains listed on the page, or select the ENUM domain on the Domains pane and click the **Push** icon.

Step 3  Choose a push mode using one of the Data Synchronization Mode radio buttons.
- If you are pushing all the ENUM Domains, you can choose Ensure, Replace, or Exact.
- If you are pushing an ENUM Domain, you can choose Ensure or Replace.

In both cases, Ensure is the default mode.

Choose Replace only if you want to replace the ENUM domain data at the local cluster. Choose Exact only if you want to create an exact copy of the ENUM domain data at the local cluster, thereby deleting all ENUM domain data that is not defined at the regional cluster.

Step 4  Click **Push Data to Clusters**.

Pulling ENUM Domains from the Replica Database

To pull ENUM domains from the replica database, do the following:

Regional Basic and Advanced Web UI

---

Step 1  From the **Design** menu, choose **Domains** under the DNS ENUM submenu to view the List/Add DNS ENUM Domains page in the regional web UI.

Step 2  Click the **Pull Replica** icon in the Domains pane.

Step 3  Click the **Replica** icon in the Update Replica Data column for the cluster. (For the automatic replication interval, see the “Replicating Local Cluster Data” section on page 6-11)

Step 4  Choose a replication mode using one of the Mode radio buttons.

Step 5  Leave the default Replace mode enabled, unless you want to preserve any existing ENUM domains data at the local cluster by choosing Ensure.

Step 6  Click the **Pull all ENUM Domains** button to view the pull details, and then click **Run**.

Pulling and Pushing ENUM Numbers

You can push ENUM numbers to and pull ENUM numbers from local clusters on the List/Add DNS ENUM Numbers page in the regional cluster web UI.

Pushing ENUM Numbers to Local Clusters

To push ENUM numbers to the local cluster, do the following:

Regional Basic and Advanced Web UI

---

Step 1  From the **Design** menu, choose **Numbers** under the DNS ENUM submenu to view the List/Add DNS ENUM Numbers page in the regional web UI.

Step 2  Click the **Push All** icon in the Numbers pane to push all the ENUM numbers listed on the page, or select the ENUM number on the Numbers pane and click the **Push** icon.
Step 3  Choose a push mode using one of the Data Synchronization Mode radio buttons.

- If you are pushing all the ENUM numbers, you can choose Ensure, Replace, or Exact.
- If you are pushing an ENUM number, you can choose Ensure or Replace.

In both cases, Ensure is the default mode.

Choose Replace only if you want to replace the ENUM number data at the local cluster. Choose Exact only if you want to create an exact copy of the ENUM number data at the local cluster, thereby deleting all ENUM number data that is not defined at the regional cluster.

Step 4  Click **Push Data to Clusters**.

---

**Pulling ENUM Numbers from the Replica Database**

To pull ENUM numbers from the replica database, do the following:

**Regional Basic and Advanced Web UI**

Step 1  From the **Design** menu, choose **Numbers** under the DNS ENUM submenu to view the List/Add DNS ENUM Number page in the regional web UI.

Step 2  Click the **Pull Replica** icon in the Numbers pane.

Step 3  Click the **Replica** icon in the Update Replica Data column for the cluster. (For the automatic replication interval, see the “Replicating Local Cluster Data” section on page 6-11)

Step 4  Choose a replication mode using one of the Mode radio buttons.

Step 5  Leave the default Replace mode enabled, unless you want to preserve any existing ENUM number data at the local cluster by choosing Ensure.

Step 6  Click the **Pull all ENUM Numbers** button to view the pull details, and then click **Run**.

---

**Adding Subzones**

As the zone grows, you might want to divide it into smaller pieces called subzones. You can delegate administrative authority for these subzones, and have them managed there or served by separate servers. This partitioning is called subzone delegation. Establish subzone delegation by performing these tasks:

1. Choose a subzone name.
2. Specify a nameserver name.
3. Specify a nameserver address.

**Related Topics**

- Choosing Subzone Names and Servers, page 14-20
- Creating and Delegating Subzones, page 14-20
- Undelegating Subzones, page 14-21
- Editing Subzone Delegation, page 14-22
Choosing Subzone Names and Servers

After you decide to divide the zone into subzones, you must create names for them. Involve the people responsible for the subzones in deciding their names, and try to maintain a consistent naming scheme.

These suggestions can help you avoid subzone naming problems:

- Consider not naming a subzone by its organizational name. In a changing business environment, organizations merge and are renamed. Naming a subzone after an organization could result in a name that is no longer meaningful over time.
- Consider not using geographical names that indicate the subzone location. Geographical names are meaningless to people outside your organization.
- Do not use cryptic names; make them obvious.
- Do not use existing or reserved top-level domain names as subzones. Using existing names can result in routing problems.

After you choose a subzone name, specify its nameservers, the ones the parent domain nameservers use when queried about the subzone. To ensure that the subzone is always reachable, you should specify two nameservers. They must be authoritative for this zone as either primary or secondary.

Whenever a subzone nameserver changes its name or address, the subzone administrator must inform its parent zone so that the parent zone administrator can change the subzone nameserver and glue records. A glue record is an A record with the address of a subzone authoritative nameserver. If the subzone administrator fails to inform its parent, the glue records are invalid. The common symptom is that a host cannot reach a host in another domain by its name, only by its address.

Note
Cisco Prime IP Express detects lame delegation by reporting missing subzone NS records in the parent zone, if NS record addresses do not match, and if glue A records are required.

Creating and Delegating Subzones

You delegate a subzone by creating it in the parent zone. There should be one NS record for each nameserver to which the subzone is delegated. Each NS record requires a corresponding A record describing the address of the nameserver, unless the nameserver is outside the parent zone or subzone. This A record is called a glue record. Such a zone which creates the NS RRs and corresponding A records (glue records) for point of delegation in the parent zone is called a parented zone. A zone that does not create the NS RRs and corresponding A records (glue records) for point of delegation in the parent zone is called an unparented zone.

Consider a zone example.com with a parent zone .com and a subzone subdomain.example.com. If example.com is a parented zone, NS RRs for the example.com appears in two places; within the example.com and within its parent zone .com. Within example.com are authoritative records for the nameservers for the zone, at the point of delegation for either a subdomain of the zone or in the parent zone. The parent zone .com will contain non-authoritative NS RRs for example.com at its point of delegation and subdomain.example.com will have non-authoritative NS RRs in example.com at its point of delegation.

See the “Choosing Subzone Names and Servers” section on page 14-20.

Local Basic or Advanced Web UI

Step 1
Create a zone as a subdomain of the parent domain on the List/Add Forward Zones page:
• If applying a zone template, go to Step 2.
• If not applying a zone template, on the List/Add Forward Zones page, click the Add Forward Zone icon and add the SOA records and the nameserver with its address.

Step 2 If Cisco Prime IP Express detects a parent zone based on the subzone name, the Create Subzone in Parent Zone page appears. Click Create as Subzone (or Create as Unparented Zone if you do not want it to be a subzone) on this page.

Creating as subzone will create the NS RRs and corresponding A records (glue records) for point of delegation in the parent zone.

Step 3 If you configured a nameserver in the subzone, you need to create a glue Address (A) record for it. In the field provided, enter the IP address of the nameserver, then click Specify Glue Records. (If there are multiple subzone nameservers, there are multiple fields for the glue records.)

Step 4 Click Report to show the intended changesets for the added records.

Step 5 Click Return after viewing the actual changesets implemented.

Step 6 To confirm the added records for the subzone, click the View icon in the RRs column for the subzone. The glue A record or records for the subzone nameserver should appear. Click Return to Zone List.

Step 7 To confirm the added records for the parent zone, click the View icon in the RRs column for the parent zone. The subzone nameserver (NS) record or records plus the glue A record or records for them should appear. Click Return to Zone List.

**CLI Commands**

On the subzone primary nameserver machine, create the subdomain:

```
nrcmd> zone boston.example.com. create primary bostonDNSserv1 hostmaster
```

On the parent zone nameserver machine, add an NS record for the subzone nameserver, then Create a glue A record for the subzone nameserver:

```
nrcmd> zone example.com. addRR boston NS bostonDNSserv1.boston.example.com.
nrcmd> zone example.com. addRR bostonDNSserv1.boston.example.com. A 192.168.40.1
```

**Undelegating Subzones**

If you undelegate a subzone, you need to remove any associated NS and glue A records from the parent zone.

**Note** If you delete the subzone, Cisco Prime IP Express cleans up the delegation records automatically.

**Local Basic or Advanced and Regional Web UI**

On the corresponding Edit Zone page, click the Resource Records tab, delete the NS record for the subzone, then delete the glue A record for the subzone server host.

**CLI Commands**

Use `zone name removeRR NS` and `zone name removeRR A` to remove the subzone NS and glue A records.
Editing Subzone Delegation

You can edit the subzone RRs.

Local Basic or Advanced and Regional Web UI

Step 1 On the corresponding Edit Zone page, click the Resource Records tab, edit the NS RR for the subzone by clicking the Edit icon next to the record to open the Edit RR in Zone page.

Step 2 Edit the NS record data.

Step 3 Click Modify Resource Record.

Step 4 Edit the glue A RR for the subzone server in the same way as in the previous steps.

CLI Commands

Use zone name removeRR to delete the NS and glue A records, then use zone name addRR to replace them.

Enabling DNS Updates

DNS Update (RFC 2136) integrates DNS and DHCP so that they can work together. DNS update automatically records the association between the hosts and their DHCP-assigned addresses. Using DHCP and DNS update, you can configure a host automatically for network access whenever it attaches to the network. You can locate and access the host using its unique DNS hostname.

DNS update is described more fully in Chapter 29, “Configuring DNS Update.” The chapter includes sections on the following:

- **Update policy (the Update Policies tab)**—Determines what kind of RRs you want updated when name-to-address associations change through DHCP.

- **Update map (the Update Maps tab)**—Defines an update relationship between a DNS server or HA DNS pair and a DHCP failover pair, DHCP policies, client-class, or access control list. (See the “Creating DNS Update Maps” section on page 29-7.)

Managing Zone Distributions

Creating a zone distribution simplifies creating multiple zones that share the same secondary zone attributes. It simplifies to a great extent the setup and management of multiple clusters sharing zone relationships such as primary to secondary or main to backup in the case of DNS HA.

The zone distribution requires adding one or more predefined secondary servers. Running a zone distribution synchronization adds secondary zones managed by secondary (slave) servers for each primary zone managed by a primary server. You can also use zone distributions to synchronize zone data from the CCM database to the local DNS server and regional and local cluster zone data. Synchronizing the zone data will always sync the associated views and named ACLs for both primary and secondary zones.
The distribution must be in a star topology, that is, one primary server and multiple secondary servers. The authoritative (master) server can only be the local primary server where the zone distribution default is defined. Starting with Cisco Prime IP Express 6.2, you can manage one zone distribution at the local cluster and multiple distributions at the regional clusters.

Related Topics

Preparing the Zone Distribution Map, page 14-23
Creating a Zone Distribution, page 14-24
Pulling Zone Distributions from Replica Data, page 14-26

Preparing the Zone Distribution Map

To prepare for creating a zone distribution, draw a zone distribution map diagram on paper.

Step 1

Start by identifying the HA DNS pair that is primary (or the primary server if HA is not involved) for all the zones that you include in the map:

a. Create a box for each server in the HA DNS pair. For example, the server pair for the Chicago-cluster consists of the servers at 192.168.50.1 and 192.168.60.1.

b. Write the IP addresses of each server in each box.

c. Write a P (for Primary) inside each box (see Figure 14-1 on page 14-23).

Figure 14-1  Diagramming a Zone Distribution Map

Step 2

Identify the role as master for each server by writing an M below the box. In the example, both primary servers are, by definition, also masters that will send copies of their zones to other servers over zone transfers. Even so, write the M below the box to make later steps easier.
Step 3
Identify all slave servers that will receive zone transfers directly from these masters. Below the master
server boxes on the page, add a box for each slave, and write its IP address inside the box. For example,
the slave servers at 192.168.70.1 and 192.168.80.1 get zone transfers from the Chicago-cluster masters.

Step 4
Write an S above each slave server box.

Step 5
Draw arrows from the M to each S representing the zone transfer flow (see the diagram). In this HA DNS
example, the arrows go from each master to both slaves.

Step 6
As you can see from the diagram, you can extend the boxes further so that the original slaves can become
masters to another set of servers (a.b.c.d and w.x.y.z).

Step 7
Enter the IP address in each box with an M below it in the Master Servers list when creating the zone
distribution.

In the CLI, set the master-servers attribute to the list of IP addresses; for example:
```
nrcmd> zone-dist dist-1 create Chicago-cluster master-servers=192.168.50.1,192.168.60.1
```

Step 8
From the Secondary Servers drop-down list on the Add or Edit Zone Distribution Secondary Server page,
choose the cluster associated with the slave server IP addresses in the boxes that have an S above them.

In the CLI, use `zone-dist name addSecondary cluster`; for example:
```
nrcmd> zone-dist dist-1 addSecondary Boston-cluster
```

### Creating a Zone Distribution

**Note**
If you move a zone from one zone distribution to another, synchronize the first zone distribution, move
the zone, then synchronize the second zone distribution.

### Local Basic or Advanced and Regional Web UI

**Step 1**
From Deploy menu, choose Zone Distributions (for the regional cluster) or Zone Distribution (for the
local cluster). The option is available if the server is configured with authoritative service. This opens
the regional List/Add Zone Distributions page or the local View Zone Distribution page. Note that the
default zone distribution is predefined at both clusters; however, the default cluster is the only one
available at the local cluster.

**Step 2**
To add a new zone distribution, click the Add Zone Distribution icon to open the Add Zone Distribution
dialog box. To edit an existing zone distribution, select its name to open the Edit Zone Distribution page.

**Step 3**
In the Primary Server field, enter the cluster (or configured HA DNS pair) that has the primary server.
This primary server is authoritative for the zones that you will determine further down the page. This
selection is subtractive: the next zone distribution you create will no longer have the cluster that you set
here as one of the choices.

**Step 4**
In the Master Servers list, add the IP address (and optional key) for each master server. The master server
is generally the primary server. However, you might want to set up a hierarchy of primaries and
secondaries where you need to define the master servers for each of the secondary relationships. You
might also want to determine the HA DNS server pairs from the master server list. You can also add an
optional TSIG key (see the “Transaction Security” section on page 29-10) to the master server address
by hyphenating the entry in the format `address-key`. For each entry, click Add IP Key.
**Step 5**  For a zone distribution, you need to add at least one secondary server. Click Add Secondary Server on the Edit Zone Distribution page to open the Add Zone Distribution Secondary Server page. Here, choose the cluster of the secondary server. Optionally, if the master servers are other than the primary servers indicated for the zone distribution, add the master server addresses, separating multiple addresses with commas. After clicking Add Server returns you to the Edit page, you can connect to the secondary server cluster, delete it, or edit it to change the master servers.

To manage the secondary servers in the zone distribution, click the View icon in the Manage Servers column to open the List Secondary Servers page. You can also edit the secondary server on an Edit Zone Distribution Secondary Server page.

**Step 6**  Choose the forward and reverse zones for the zone distribution. The default zone distribution includes all the created forward and reverse zones. For all other created zone distributions, you must move the zone or zones into the Selected column.

**Step 7**  Click Save.

**Step 8**  Synchronize the zone distribution with the local cluster DNS servers. A synchronization:

- Pushes staged zone, RR, or host edits to the primary server cluster or HA DNS pair for the regional cluster in Ensure, Replace, or Exact modes, or from the local cluster in Exact mode.
- Creates secondary zones for secondary servers, in Exact mode.

**Step 9**  Click the Synchronize Zone Distribution tab, and choose a synchronization mode:

- **Update**—Adds new zones, RR sets, and hosts; replaces existing hosts if there are conflicts; and creates new secondary zones.
- **Complete**—Like Ensure mode, except that it always replaces existing RR sets and hosts, and modifies the master server list on existing secondary zones.
- **Exact**—Like Complete mode, except that it deletes extra zones, RR sets, hosts, and secondary zones no longer on the primary.

**Step 10**  Click Report in the Synchronize Zone Distribution tab(or the same icon in the Synchronize All Zone Distributions area of the page at the regional cluster). This opens the Sync Zone Distribution page that shows a preview of the data synchronized.

---

**CLI Commands**

To create the zone distribution, use `zone-dist name create primary-cluster`. (The primary cluster can also be the HA DNS pair.) For example:

```
nrcmd> zone-dist dist-2 create Chicago-cluster
```

To set the master server or servers, use `zone-dist name set master-servers=addresses`, separating the addresses with commas. For example:

```
nrcmd> zone-dist zone-dist-2 set master-servers=192.168.50.1,192.168.60.1
```

To add the secondary server, use `zone-dist name addSecondary secondary-cluster`. For example:

```
nrcmd> zone-dist zone-dist-2 AddSecondary Boston-cluster
```

You must associate the zone distribution directly with the zone or zone template. Use `zone name set dist-map=zone-dist-list` or `zone-template name set dist-map=zone-dist-list`, separating the zone distribution entries with commas. For example:

```
nrcmd> zone example.com set dist-map=zone-dist-2
nnrcmd> zone-template zone-template-1 set dist-map=zone-dist-2
```
To synchronize the zone distributions, use `zone-dist name sync`. You can do a synchronization in update, complete, or exact mode, and you can exclude RRs and secondary zones:

- At the local cluster, this synchronizes staged edits to the DNS server and primary zones to secondaries. Regardless of the synchronization mode, this always synchronizes the exact list of authoritative zones.

- At the regional cluster, this synchronizes primary zones with the local clusters, and primaries to secondaries. This replaces primary zones at the local cluster in Update and Complete modes, and deletes extra primary zones at the local cluster in Exact mode.

- For secondary zones, the same synchronization logic occurs at the local and regional clusters. In Update mode, this ensures that corresponding secondary zones exist on the server. In Complete mode, existing zones are updated to use the master server list specified by the zone distribution map. In Exact mode, any zones not matching the distribution map are deleted.

   For example:

   ```
   nrcmd> zone-dist zone-dist-1 sync exact no-rrs no-secondaries
   ```

### Pulling Zone Distributions from Replica Data

You can pull zone distributions from the local replica data instead of explicitly creating them.

Tip

For an example of pulling local zone data to create a zone distribution, see the "Pull Zone Data and Create a Zone Distribution" section on page 5-36.

### Regional Web UI

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>From Deploy menu, choose Zone Distribution. This opens the regional List/Add Zone Distribution page.</td>
</tr>
<tr>
<td>Step 2</td>
<td>On the List/Add Zone Distribution page, click the Synchronize Zone Distribution tab in the Zone Distributions pane.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Choose the data synchronization mode (Update, Complete, or Exact). These modes are described in the table on that page.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Click Report at the bottom of the dialog box.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Click Run.</td>
</tr>
</tbody>
</table>
Configuring DNS Views

DNS Zone Views let you present alternate versions of zone data to different communities of clients using a single name server. For example, a DNS server for example.com can maintain two views of the zone, where the view of example.com that can be queried internally includes many hosts that do not exist in the external view. Each zone view is treated as an independent copy of the zone. The DNS server, when answering queries on the zone, uses the match criteria defined in each view to determine the matching zone for the client. The query is answered based on that zone contents. In some cases, the zone contents may only vary slightly between views.

Related Topics

- DNS Views Processing, page 15-1
- Key Points to Remember When you Work on DNS Views, page 15-1
- Managing DNS Views, page 15-2
- Synchronizing DNS Views, page 15-3
- Pushing and Pulling DNS Views, page 15-3

DNS Views Processing

DNS Views allow a name server to segregate the data and provide a different view of the data based on the clients accessing it. When DNS receives a DNS request, the request is processed to associate it with a view.

The DNS client servers like Caching DNS, Secondary DNS, Primary for Notifies, DHCP and so on are automatically assigned the correct view and no configuration is required.

Key Points to Remember When you Work on DNS Views

Following are some of the key points or attributes you should know when you work on DNS Views:

- **View ID**—Defines a unique integer identifier for the view that is assigned by the CCM server or the user while creating DNS views.

- **View Priority**—Each Zone View will be assigned a unique priority to determines its order in the view list. The lowest non-zero priority value will have the highest priority and will be processed first. A zero priority is reserved for the default view, which will always be last. The web UI will provide an option to reorder views without explicitly setting the priority.
• **Default View**—The default view is created with view-id=0, priority=0, and client and destination ACLs set to any. Requests that do not match a named view will always fall into the default view. By default, zones will be created with a view-id=0, which will automatically place them in the default view. The default view cannot be modified or deleted.

• **acl-match-clients attribute**—Specifies the ACL that maps clients to this view based on the source address (subnet or prefix) or TSIG key.

## Managing DNS Views

You can create, edit, and delete DNS Views from local or regional cluster. You can also push or pull views and ACLs in Ensure, Replace, and Exact modes from or to the regional CCM server.

### Local Basic or Advanced and Regional Web UI

To create DNS Views:

1. From the **Design** menu, choose **Views** under the **Auth DNS** submenu.
2. On the **Views** pane, click the **Add View** icon.
3. Specify the name for the DNS views.
4. Specify the view id. If you do not specify, the application automatically assigns a view id to the view.
5. You can specify the ACL that maps the client to this view in the **acl-match-clients** field.
6. Click the **Add DNS View** button.
7. To edit a DNS View, click its name in the Views pane on the left.

### Note

You can create a maximum of 100 views.

## Reorder DNS Views

When you create a set of DNS Views, you can specify the priority order. To specify the priority order:

1. From the **Design** menu, choose **View** under the **Auth DNS** submenu to open the List/Add Zone Views page.
2. Click the **Reorder Views** icon in the **Views** pane to open the Reorder dialog box.
3. Set the priority for the DNS Views rules by either of the following methods:
   - Select the view and click the **Move up** or **Move down** icon to reorder the rules.
   - Select the view and click the **Move to** button, and enter the row number to move the view.
4. Click **Save** to save the reordered list.
Synchronizing DNS Views

Zone distribution sync, single zone sync, and HA DNS zone sync will always sync associated views and named ACLs for both primary and secondary zones. The synchronization modes applied while running zone distribution or HA DNS sync vary. When you run:

- **Zone Distribution Sync**—views will be synchronized in Replace mode for all zone distribution sync types (Update, Complete, and Exact), while ACLs will use Ensure mode. If caching DNS servers are included in the zone distribution, the associated views and named ACLs will be synchronized to these servers and the masters list will be configured as exceptions for the unique set of domain names in the distribution. The option to omit secondaries from the sync will also be extended to exclude caching servers.

- **HA DNS Sync**—views will be updated in Replace mode for both Update and Complete sync, while Exact sync will sync views in Exact mode.

Pushing and Pulling DNS Views

You can also push views and ACLs to and pull views and ACLs from the regional cluster in Ensure, Replace, and Exact modes.

Pushing DNS Views to Local Clusters

You can push the views you create from the regional cluster to any of the local clusters.

**Regional Web UI**

**Step 1**
From the Design menu, choose Views under the Auth DNS submenu.

**Step 2**
On the Views pane, click the Push All icon in the left pane, or select a DNS View and click Push at the top of the Edit Zone View page. This opens the Push Data to Local Clusters or Push Zone View page.

**Step 3**
Choose a push mode using one of the Data Synchronization Mode radio buttons.
- If you are pushing all the DNS Views, you can choose Ensure, Replace, or Exact.
- If you are pushing a DNS View, you can choose Ensure or Replace.

In both the above cases, Ensure is the default mode.

Choose Replace only if you want to replace the existing DNS View data at the local cluster. Choose Exact only if you want to create an exact copy of the DNS View at the local cluster, thereby deleting all DNS Views that are not defined at the regional cluster.
Pushing and Pulling DNS Views

Step 4  Choose one or more local clusters in the Available field of the Destination Clusters and move it or them to the Selected field.

Step 5  Click Push Data to Clusters.

Pulling DNS Views from Local Clusters

Instead of explicitly creating views, you can pull them from the local clusters. In the regional web UI, you may first want to update the view replica data by clicking the Replica icon next to the cluster name.

Regional Web UI

Step 1  From the Design menu, choose Views under the Auth DNS submenu.

Step 2  On the List/Add Zone Distribution page, click the Pull Replica icon in the Views pane.

Step 3  Choose the data synchronization mode (Update, Complete, or Exact). These modes are described in the table on that page.

Step 4  Click Report at the bottom of the dialog box.

Step 5  Click Run.
Managing Resource Records

This chapter explains how to configure some of the more advanced DNS zone and server parameters by using the Cisco Prime IP Express web UI and CLI. Before you proceed with the concepts in this chapter, read Chapter 14, “Managing Zones,” which explains how to set up the basic properties of a primary and secondary DNS server and its zones.

Related Topics

Managing Resource Records, page 16-1
Managing Hosts in Zones, page 16-9

Managing Resource Records

Resource records (RRs) comprise the data within a DNS zone. Although there is no fixed limit to the number of RRs a zone may own, in general, a zone may own one or more RRs of a given type (the zone always has a Start of Authority, or SOA, record). There are some exceptions depending on the types involved. All RRs have the entries described in Table 16-1.

<table>
<thead>
<tr>
<th>RR Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Owner of the record, such as a zone or hostname.</td>
</tr>
<tr>
<td>Class (not required for all formats)</td>
<td>Cisco Prime IP Express supports only the IN (Internet) class.</td>
</tr>
<tr>
<td>TTL (time to live)</td>
<td>Amount of time to store the record in a cache, in seconds. If you do not include a TTL, Cisco Prime IP Express uses the zone default TTL, defined as a zone attribute.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of the record, such as A (AAAA for IPv6), NS, SOA, and MX. There are many types that various RFCs define, although fewer than ten are in common use.</td>
</tr>
<tr>
<td>Record data</td>
<td>Data types whose format and meaning varies with record type.</td>
</tr>
</tbody>
</table>

Related Topics

Adding Resource Records, page 16-2
Protecting Resource Record Sets, page 16-3
Editing Resource Records, page 16-4
Adding Resource Records

Before adding or modifying RRs, keep in mind the two distinct dns edit modes that you can set and work in: staged and synchronous (see the “Staged and Synchronous Modes” section on page 14-1).

Administrator roles required for RR management are the dns-admin role at the local cluster and the central-dns-admin role at the regional cluster. The host-admin role at the local cluster and the central-host-admin role at the regional cluster can view host records only.

Local Basic or Advanced and Regional Web UI

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Choose <strong>Forward Zones</strong> from <strong>Design &gt; Auth DNS</strong> to open the List/Add Forward Zones page.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>In the Forward Zone pane, click the zone name to open the Edit Zone page. Note that resource record edits is managed jointly by CCM and DNS and a system lock is used to prevent DNS and CCM from accessing the resource record database at the same time.</td>
</tr>
</tbody>
</table>

Tip Records are listed in the formats that their respective RFCs specify, with only the first record in a set labeled with its name, and in DNSSEC order. To reduce or increase the items in the table, change the page size value at the bottom of the page, then click **Change Page Size**.

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Click the <strong>Resource Records</strong> tab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4</td>
<td>Add the RR name, TTL (if not using the default TTL), type, and data as appropriate.</td>
</tr>
<tr>
<td>Step 5</td>
<td>By default, RRs are protected, which means that DNS Updates cannot overwrite them (see the “Protecting Resource Record Sets” section on page 16-3). To unprotect the RRs, click the <strong>Locked</strong> icon to the left of the record name to change it to the Unlocked icon. Likewise, to protect the record, click the <strong>Unlocked</strong> icon to change it to the Locked icon.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Click <strong>Add Resource Record</strong>.</td>
</tr>
</tbody>
</table>

CLI Commands

Use `zone name addRR` to add a protected RR of a certain type. You can specify the name as a relative name, if the owner is in the same domain, an absolute name (by supplying the FQDN), or the same name as the zone name (by using the at `[@]` symbol).

For example:

```
nrcmd> zone example.com addRR -sync host101 A 192.168.50.101
```

Use `zone name addDNSRR type data` to add an unprotected RR.
Protecting Resource Record Sets

When an RR is protected, DNS Updates cannot modify the record. Most administratively created RRs are protected. However, RRs created by DNS Updates must be unprotected to allow the server to modify them. You can set this protection status for each RR set on the List/Add DNS Server RRs for Zone page.

Note that only the primary DNS server can recognize this protection status; secondary servers do not recognize the protection status of their RRs.

Caution

Zone scavenging can remove RRs that are unprotected. See the “Scavenging Dynamic Records” section on page 29-19 for details.

Local Basic or Advanced and Regional Web UI

To protect an existing RR, do the following:

Step 1 Choose Forward Zones from Design > Auth DNS to open the List/Add Forward Zones page.
Step 2 In the Forward Zone pane, click the zone name to open the List/Add Forward Zones page.
Step 3 Click the Resource Records tab.
Step 4 On the Resource Records tab, click the Resource Record name in the list of Resource Records to edit the resource record.
Step 5 Click Protect Set button to unprotect the selected RR set.
Step 6 Click Save to save the resource record attribute modification.

You can also unprotect an RR. To unprotect an RR while adding, click the Locked icon next to the Resource Record name field. The icon changes to the Unlocked icon.

To unprotect an existing RR, do the following:

Step 1 Choose Forward Zones from Design > Auth DNS to open the List/Add Forward Zones page.
Step 2 In the Forward Zone pane, click the zone name to open the List/Add Forward Zones page.
Step 3 Click the Resource Records tab.
Step 4 On the Resource Records tab, click the Resource Record name in the list of Resource Records to edit the resource record.
Step 5 Click Unprotect Set button to unprotect the selected RR set.
Step 6 Click Save to save the resource record attribute modification.

Note

The icon to the left of the RR set name indicates the status of the Resource Record, whether it is protected or unprotected.
Managing Resource Records

CLI Commands

To protect the RR sets, use `zone name protect-name rrset-name`; to unprotect the zone, use the `unprotect-name rrset-name` action instead. For example:

```
nrcmd> zone example.com protect-name boston
100 Ok
protected boston
nrcmd> zone example.com unprotect-name boston
100 Ok
unprotected boston
```

Editing Resource Records

You can edit RRs as an individual record or as an RR set:

- **Individual RRs**—Click the Edit icon next to the record name to open the Edit RR in Zone page.
- **RR sets**—Click the name of the record to open the Edit RR Set in Zone page.

For a description of the fields to enter data, see the “Adding Resource Records” section on page 16-2.

Removing Resource Records

You can remove RRs from a zone.

Local Basic or Advanced and Regional Web UI

On the local or regional the Resource Records tab for the Zone page:

- To remove an entire record name set, click the Delete icon next to the record set name in the list, then confirm the deletion.
- To remove individual records from the set, click the name of the record set to open the edit page, click the Delete icon next to the individual record in the list, then confirm the deletion.

CLI Commands

The CLI includes two removal commands, depending on the type of RR to remove:

- Use `zone name removeRR` to remove any RR. You must specify the owner. If you omit the data, Cisco Prime IP Express removes all records of the specified type for the specified owner. Similarly, if you omit the type, Cisco Prime IP Express removes all records for the specified owner.
- Use `zone name removeDNSRR` to remove unprotected RRs only.
Listing Records

To view the resource records:

Local Basic or Advanced and Regional Web UI

To list down RRs of a zone, do the following:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Choose <strong>Forward Zones</strong> from <strong>Design &gt; Auth DNS</strong> to open the List/Add Forward Zones page.</td>
</tr>
<tr>
<td>Step 2</td>
<td>In the Forward Zone pane, click the zone name to open the List/Add Forward Zones page.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Click the <strong>Resource Records</strong> tab to view the resource records.</td>
</tr>
</tbody>
</table>

CLI Commands

Use `zone name listRR` to display RRs in the named zone. You can also specify whether you want all records or only staged (CCM) or synchronized (DNS) records (see the “Filtering Records” section on page 16-7 for details). For example:

```
nrcmd> zone example.com listRR protected
```

You can get an exact count of the total RRs for the DNS server by using `dns getRRCount [zone name | forward | reverse | primary | secondary | all]`. Options let you request the RR count for a single zone or all zones of a given type.

Searching Server-Wide for Records and Addresses

With Cisco Prime IP Express, you can search for RRs and IP addresses server-wide. The search is a filter mechanism whereby you can specify a combination of RR and address attributes to target one or more RRs or addresses configured for the network. The search function is available at the local cluster only.

You can search RRs by:

- IP address
- Protection state
- Name prefix
- Type
- Zone

Local Advanced Web UI

To search resource records by IP address, do the following:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>From the <strong>Operate</strong> menu, choose <strong>DNS &gt; RR By IP Address</strong> from the <strong>Reports</strong> submenu to open the IP Address Search page.</td>
</tr>
<tr>
<td>Step 2</td>
<td>To search by IP address, enter an IP address, then click <strong>Search</strong>.</td>
</tr>
</tbody>
</table>
Managing Resource Records

**Note** In an IP address search, the DNS server does not search all forward zones for RRs that have the specified address in the data field. Instead, the server looks up the matching PTR record in the reverse zone and returns all the respective RRs in the forward zone.

---

To search resource records, do the following:

**Step 1** From the Operate menu, choose DNS > Resource Records from Reports submenu to open the DNS Resource Record Search page.

**Step 2** Choose a filter attribute from the drop-down list.

**Step 3** Choose a filter type from the drop-down list depending on the filter attribute you chose:

- **RR Protection State**—RR Protection Status, either locked or unlocked.
- **RR Name Prefix**—RR Name Prefix.
- **RR Type**—RR Type.
- **Zone**—Zone List, Regular expression, or Zone Flags

**Step 4** Enter or select a Value, based on the Type selected. To clear the filter, click **Clear Filter**.

**Step 5** Click **Add Element** to add the search element to the filter elements list. The Filter Elements heading changes to identify the filter attribute and value used for the filter. If you add more than one element, the heading identifies the ANDed values of the elements. For example, if you add an element for a name prefix search for user, then add another element for an RR type search for A records, the filter element heading will identify the search as **RR Name Prefix = user AND RR Type = A**.

**Step 6** You can add as many elements as you like (remembering that the search results are an intersection of the filter elements). View the filter elements list by clicking the plus sign (+).

**Step 7** Click **Search**.

**Step 8** Check the table of resulting RRs from the search, which shows for each RR its zone, hostname, TTL, type, and associated data. If necessary, change the page size to see more entries at one time (you might still need to page forward and back). The RRs are sorted in DNSSEC order.

**Tip** If the search results are less than expected due to the ANDing of the filter elements, look at the filter list for any element that might be compromising the search, delete it by clicking the Delete icon next to it, then redo the search.

---

**CLI Commands**

Use **dns findRR** to find RRs across the zones. The command syntax is of two kinds:

```
# nccmd> dns findRR -name (fqdn | domainaddr)
# nccmd> dns findRR [-namePrefix nameprefix]
#                     [-rrTypes RRtypelist]
#                     [-protected | -unprotected]
#                     [-zoneType (forward | reverse | primary | secondary | ALL)]
```
Managing Resource Records

You can search by domain or its address, or enter the beginning characters of the RR name (the name prefix). If you search by RR name prefix, you can narrow the search by a list of RR types, protection status, or zone type. The output clearly indicates the zone for each found entry. For example:

```
nrcmd> dns findRR -namePrefix user -rrTypes A
userhost101.example.com IN A 192.168.50.101
userhost102.example.com IN A 192.169.50.102
userhost103.boston.example.com IN A 192.168.50.103
```

Filtering Records

You might want to filter records to display only one type of record, such as an A (or IPv6 AAAA) or PTR record. (See also the “Searching Server-Wide for Records and Addresses” section on page 16-5.)

Local Basic or Advanced and Regional Web UI

You can filter RRs right from the Edit Zone page. Look for the Name and Type fields just below the Add Resource Record button.

By default, RRs are sorted alphabetically by name, starting with the top-of-zone records (marked with the @ symbol), and secondarily sorted by type, then data. You can also sort them by:

- **Protected state**—You can click All, Unprotected, or Protected.
- **Name prefix**—Starting characters in the name. Note that the * character is not a wildcard. For example, entering al returns alberta, allen.wrench, and allie, whereas entering al* returns al* and al*ert.
- **RR type**—Click one of the RR types in the drop-down list, such as A (or IPv6 AAAA) or TXT.

When the selection is complete, click Filter List. This returns just the filtered entries in the table below the fields. To return to the full, unfiltered list, click Clear Filter.

CLI Commands

Use `zone zonename findRR` to search on RR name prefixes, RR types, or protection status:

```
nrcmd> zone zonename findRR [-namePrefix nameprefix] [-rrTypes RRtypelist] [-protected | -unprotected]
```

Using Service Location (SRV) Records

The service location (SRV) RR is used to advertise services to the network. This RR is defined in the RFC 2782, “A DNS RR for specifying the location of services (DNS SRV).” The SRV can have an associated A or AAAA record. Windows domain controller is one service that uses the SRV records.

The RFC defines the format of the SRV record (DNS type code 33) as:

```
_service._protocol.name ttl class SRV priority weight port target
```

There should always be an A record associated with the SRV record target so that the client can resolve the service back to a host. In the Microsoft Windows implementation of SRV records, the records might look like this:

```
myserver.example.com A 201.165.201.1
_ldap._tcp.example.com SRV 0 0 389 myserver.example.com
_kdc._tcp.example.com SRV 0 0 88 myserver.example.com
```
Managing Resource Records

An underscore (_) always precedes the service and protocol names. In the example, _kdc is the Key Distribution Center. The priority and weight help a client choose between target servers providing the same service (the weight differentiating those with equal priorities). If the priority and weight are all set to zero, the client orders the servers randomly.

Note

For a description of how Windows clients interoperate with DNS and DHCP servers, including scavenging dynamic RRs, see the “Configuring DNS Update for Windows Clients” section on page 29-22.

Using NAPTR Records

Cisco Prime IP Express supports Naming Authority Pointer (NAPTR) RRs. These records help with name resolution in a particular namespace and are processed to get to a resolution service. Because NAPTR records are a proposed standard, RFC 3403, Cisco Prime IP Express only validates their numeric record fields. However, the proposed standard requires a value for each field, even if it is null (""), and there are no preset values.

When using a NAPTR record to locate a Session Initiation Protocol (SIP) proxy, see the proposed standard, RFC 2916 or RFC 3263. In RFC 2916, the ENUM working group of the Internet Engineering Task Force specifies NAPTR records to map E.164 addresses to Universal Resource Identifiers (URIs). Using the NAPTR record resolves a name in the E.164 international public telecommunication namespace to a URI, instead of providing the name of a service to use as a resolver. The U flag was added to the NAPTR record for this purpose.

For example, to specify a SIP proxy for the phone number +4689761234, add a NAPTR record at the name 4.3.2.1.6.7.9.8.6.4.e164.arpa. with this content:

```
100 10 "u" "sip+E2U" "/^.*$/sip:info@example.com/" .
```

This sets these fields of the NAPTR record:

- `order` = 100
- `preference` = 10
- `flags` = "u"
- `service` = "sip+E2U"
- `regexp` = "/^.*$/sip:info@example.com/"
- `replacement` = .

After you configure these fields, the DNS client dealing with phone number +4689761234 can now find an SIP service URI by replacing the number with sip:info@tele2.se. The E.164 zone mostly uses the NAPTR record for wholesale replacement of the input telephone number. Section 3.2.3 of RFC 2916 includes an example of one transformation to a Lightweight Directory Access Protocol (LDAP) query that preserves some of the digits. The E.164 zone does not map to service location (SRV) records because it wants to obtain a SIP URL that is more humanly readable to the left of the at (@) symbol.

Local Basic or Advanced and Regional Web UI

1. On the Edit Zone page, click the **Resource Records** tab.
2. Enter the owner of the record in the Name field.
3. Enter the TTL (if necessary).
4. Click NAPTR in the Type drop-down list.
Step 5 Enter the data as a string embedded in quotes and separated by spaces:

a. Order
b. Preference
c. Flags
d. Service
e. Regular expression
f. Replacement string

For example:
"100 10 u sip+E2U /^.*$/sip:info@tele2.se/ ."

Step 6 Click Add Resource Record.

CLI Commands

Use `zone name addRR` to add a protected resource record to a zone.

Managing Hosts in Zones

You can manage the RRs for a host by configuring the host record rather than the individual RRs. When you define a host, the DNS server automatically creates an Address (A) RR for IPv4, or an AAAA RR for IPv6, for it. If the reverse zone for the host exists, the server can also create the associated Pointer (PTR) RR for it.

See Chapter 10, “Managing Hosts” for details.
Managing Authoritative DNS Server Properties

This chapter explains how to set the Authoritative DNS server parameters. Before you proceed with the tasks in this chapter, read Chapter 14, “Managing Zones,” which explains how to set up the basic properties of a primary and secondary zone.

Note
Change in majority of DNS server attributes requires reloading of server to become effective.

Related Topics
Managing DNS Authoritative Servers, page 17-1
Setting DNS Server Properties, page 17-2
Setting Advanced Authoritative DNS Server Properties, page 17-6
Tuning DNS Properties, page 17-9
Troubleshooting DNS Servers, page 17-9

Managing DNS Authoritative Servers

You can view its health, statistics, and logs; start, stop, and reload it; run certain commands (see the “Running DNS Authoritative Server Commands” section on page 17-1); and edit the server attributes.

To view the server status and health, or stop, start, and reload the server, in the local cluster web UI, click Deploy, then choose DNS Server under the DNS submenu to open the Manage DNS Authoritative Server page.

Related Topics
Running DNS Authoritative Server Commands, page 17-1
Configuring DNS Server Network Interfaces, page 17-2

Running DNS Authoritative Server Commands

Access the commands by using the Commands button. Clicking the Commands button opens the DNS Commands dialog box in the local web UI. Each command has its own Run icon (click it, then close the dialog box):

- **Force all zone transfers**—A secondary server periodically contacts its master server for changes. See the “Enabling Zone Transfers” section on page 14-14.
Setting DNS Server Properties

• **Scavenge all zones**—Cisco Prime IP Express provides a feature to periodically purge stale records. See the “Scavenging Dynamic Records” section on page 29-19.

• **Synchronize All HA Zones**—Synchronizes all the HA zones. You have the option to choose the type of synchronization. The **Use Server Algorithms** option is checked by default. You can override this by checking either **Push All Zones From Main to Backup** check box or **Pull All Zones From Backup to Main** check box.

**Note**
The **Synchronize All HA Zones** command is an **Expert** mode command which you can see only if the server is an HA main server. You cannot see this command if it is an HA backup server. You can also, synchronize zones separately, which you can do from the Zone Commands for Zone page (see the “Synchronizing HA DNS Zones” section on page 19-5).

**Note**
If you find a server error, investigate the server log file for a configuration error, correct the error, return to this page, and refresh the page.

**Configuring DNS Server Network Interfaces**

You can configure the network interfaces for the DNS server from the Manage Servers page in the local web UI.

**Local Advanced Web UI**

**Step 1** From the **Operate** menu, choose **Manage Servers**.

**Step 2** Click **Local DNS Server** on the Manage Servers pane to open the Local DNS Server page.

**Step 3** Click the **Network Interfaces** tab for the DNS server to view the available network interfaces that you can configure for the server. By default, the server uses all of them.

**Step 4** To configure an interface, click the Configure icon in the Configure column for the interface. This adds the interface to the Configured Interfaces table, where you can edit or delete it.

**Step 5** Clicking the name of the configured interface opens a new page, where you can change the address and port of the interface.

**Step 6** Click **Modify Interface** when you are done editing, then click **Go to Server Interfaces** to return to the Manage Servers page.

**Note**
The IPv6 functionality in DNS requires IPv4 interfaces to be configured except if the DNS server is isolated and standalone (it is its own root and is authoritative for all queries).

**Setting DNS Server Properties**

You can set properties for the DNS server, along with those you already set for its zones. These include:
• **General server properties**—See the “Setting General DNS Server Properties” section on page 17-3 and “Scavenging Dynamic Records” section on page 29-19.

• **Delegation-only zones**—See the “Specifying Delegation-Only Zones” section on page 17-3.

• **Round-robin server processing**—See the “Enabling Round-Robin” section on page 17-4.

• **Subnet sorting**—See the “Enabling Subnet Sorting” section on page 17-4.

• **Enabling incremental zone transfers**—See the “Enabling Incremental Zone Transfers (IXFR)” section on page 17-5.

• **Enabling NOTIFY packets**—See the “Enabling NOTIFY” section on page 17-5.

### Setting General DNS Server Properties

You can display DNS general server properties, such as the name of the server cluster or host machine and the version number of the Cisco Prime IP Express DNS server software. You can change the internal name of the DNS server by deleting the current name and entering a new one. This name is used for notation and does not reflect the official name of the server. Cisco Prime IP Express uses the server IP address for official name lookups and for DNS updates (see Chapter 29, “Configuring DNS Update”).

The following subsections describe some of the more common property settings. They are listed in the “Setting DNS Server Properties” section on page 17-2.

#### Local Basic or Advanced Web UI

<table>
<thead>
<tr>
<th>Step 1</th>
<th>To access the server properties, choose <strong>DNS Server</strong> from the <strong>Deploy</strong> menu to open the Manage DNS Authoritative Server page. The page displays all the DNS server attributes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Modify the attributes as per your requirements.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Click <strong>Save</strong> to save the DNS server attribute modifications.</td>
</tr>
</tbody>
</table>

#### CLI Commands

Use `dns [show]` to display the DNS server properties.

### Specifying Delegation-Only Zones

You can instruct the server to expect only referrals when querying the specified zone. In other words, you want the zone to contain only NS records, such as for subzone delegation, along with the apex SOA record of the zone. This can filter out “wildcard” or “synthesized” data from authoritative nameservers whose undelegated (in-zone) data is of no interest. Enable the DNS server `delegation-only-domains` attribute for this purpose.
Enabling Round-Robin

A query might return multiple A records for a nameserver. To compensate for most DNS clients starting with, and limiting their use to, the first record in the list, you can enable round-robin to share the load. This method ensures that successive clients resolving the same name will connect to different addresses on a revolving basis. The DNS server then rearranges the order of the records each time it is queried. It is a method of load sharing, rather than load balancing, which is based on the actual load on the server.

Tip
Adjust the switchover rate from one round-robin server to another using the TTL property of the server A record.

Local Basic or Advanced Web UI

On the Manage DNS Authoritative Server page, under the Miscellaneous Options and Settings section, find the Enable round-robin (round-robin) attribute. It is set to enabled by default in Basic mode.

CLI Commands

Use `dns get round-robin` to see if round-robin is enabled (it is by default). If not, use `dns enable round-robin`.

Enabling Subnet Sorting

If you enable subnet sorting, as implemented in BIND 4.9.7, the Cisco Prime IP Express DNS server confirms the client network address before responding to a query. If the client, server, and target of the query are on the same subnet, and the target has multiple A records, the server tries to reorder the A records in the response by putting the closest address of the target first in the response packet. DNS servers always return all the addresses of a target, but most clients use the first address and ignore the others.

If the client, DNS server, and target of the query are on the same subnet, Cisco Prime IP Express first applies round-robin sorting and then applies subnet sorting. The result is that if you have a local response, it remains at the top of the list, and if you have multiple local A records, the server cycles through them.

Local Basic or Advanced Web UI

On the Manage DNS Authoritative Server page, in A-Z view, find the Enable subnet sorting (subnet-sorting) attribute, set it to enabled, then click Save.

CLI Commands

Use `dns enable subnet-sorting` or `dns disable subnet-sorting` (the preset value).
Enabling Incremental Zone Transfers (IXFR)

Incremental Zone Transfer (IXFR, described in RFC 1995) allows only changed data to transfer between servers, which is especially useful in dynamic environments. IXFR works together with NOTIFY (see the “Enabling NOTIFY” section on page 17-5) to ensure more efficient zone updates. IXFR is enabled by default.

Primary zone servers always provide IXFR. You should explicitly enable IXFR on the server (you cannot set it for the primary zone) only if the server has secondary zones. The DNS server setting applies to the secondary zone if there is no specific secondary zone setting.

Local Basic or Advanced Web UI

On the Manage DNS Authoritative Server page, under the Zone Default Settings section, you can find the Request incremental transfers (IXFR) attribute. It is set it to enabled by default. For a secondary zone, you can also fine-tune the incremental zone transfers by setting the `ixfr-expire-interval` attribute. This value is the longest interval the server uses to maintain a secondary zone solely from IXFRs before forcing a full zone transfer (AXFR). The preset value of one week is usually appropriate. Then, click Save.

CLI Commands

Use `dns enable ixfr-enable`. By default, the `ixfr-enable` attribute is enabled.

Restricting Zone Queries

You can restrict clients to query only certain zones based on an access control list (ACL). An ACL can contain source IP addresses, network addresses, TSIG keys (see the “Transaction Security” section on page 29-10), or other ACLs. The `restrict-query-acl` on the DNS server serves as a default value for zones that do not have the `restrict-query-acl` specifically set.

Enabling NOTIFY

The NOTIFY protocol, described in RFC 1996, lets the Cisco Prime IP Express DNS primary server inform its secondaries that zone changes occurred. The NOTIFY packets also include the current SOA record for the zone giving the secondaries a hint as to whether or not changes have occurred. In this case, the serial number would be different. Use NOTIFY in environments where the namespace is relatively dynamic.

Because a zone master server cannot know specifically which secondary server transfers from it, Cisco Prime IP Express notifies all nameservers listed in the zone NS records. The only exception is the server named in the SOA primary master field. You can add additional servers to be notified by adding the IPv4 addresses to the notify-set on the zone configuration.

Note

For NS records that point at names that the DNS server is not authoritative for, those IP addresses need to be explicitly set in the notify-set if the user wants those servers to get notifies.

You can use IXFR and NOTIFY together, but this is not necessary. You can disable NOTIFY for a quickly changing zone for which immediate updates on all secondaries does not warrant the constant NOTIFY traffic. Such a zone might benefit from having a short refresh time and a disabled NOTIFY.
Setting Advanced Authoritative DNS Server Properties

Local Basic or Advanced Web UI

Step 1  On the Manage DNS Authoritative Server page, under the Zone Transfer Settings section, find the notify attribute (Expert mode only), then check the Enabled check box to enable it.

Step 2  Set any of the other NOTIFY attributes (notify-defer-cnt, notify-min-interval, notify-rcv-interval, notify-send-stagger, notify-source-address, notify-source-port, and notify-wait).

Step 3  Click Save.

Step 4  To add nameservers in addition to those specified in NS records, from the Design menu, choose Forward Zones under the Auth DNS submenu.

Step 5  Click the zone in the Forward Zones pane to open the Edit Zone page.

Step 6  Add a comma-separated list of IP addresses of the servers using the notify-set attribute on the Edit Zone page.

Step 7  Set the notify attribute to true.

Step 8  Click Save on that page.

CLI Commands

Use dns enable notify. NOTIFY is enabled by default. You can also enable NOTIFY at the zone level, where you can use zone name set notify-set to specify an additional comma-separated list of servers to notify beyond those specified in NS records.

Setting Advanced Authoritative DNS Server Properties

You can set these advanced server properties:

- **SOA time-to-live**—See the “Setting SOA Time to Live” section on page 17-6.
- **Secondary server attributes**—See the “Setting Secondary Refresh Times” section on page 17-7.
- **Port numbers**—See the “Setting Local and External Port Numbers” section on page 17-8.
- **Handle Malicious DNS Clients**—See the “Handling Malicious DNS Clients” section on page 17-8.

Setting SOA Time to Live

The SOA record time to live (TTL) is usually determined by the zone default TTL. However, you can explicitly set the SOA TTL, which sets the maximum number of seconds a server can cache the SOA record data. For example, if the SOA TTL is set for 3600 seconds (one hour), an external server must remove the SOA record from its cache after an hour and then query your nameserver again.

Cisco Prime IP Express responds to authoritative queries with an explicit TTL value. If there is no explicit TTL value, it uses the default TTL for the zone, as set by the value of the defttl zone attribute. Databases originating from versions of Cisco Prime IP Express earlier than Release 3.5 do not have the defttl zone attribute, and use the minimum TTL in the zone SOA record for the default TTL.
Normally, Cisco Prime IP Express assumes the default TTL when responding with a zone transfer with RRNs that do not have explicit TTL values. If the default TTL value for the zone is administratively altered, Cisco Prime IP Express automatically forces a full zone transfer to any secondary DNS server requesting a zone transfer.

### Local Basic or Advanced and Regional Web UI

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>On the List/Add Zone page, set the Zone Default TTL, which defaults to 24 hours.</td>
</tr>
<tr>
<td>Step 2</td>
<td>If you want, set the SOA TTL, which is the TTL for the SOA records only. It defaults to the Zone Default TTL value.</td>
</tr>
<tr>
<td>Step 3</td>
<td>You can also set a TTL value specifically for the NS records of the zone. Set the NS TTL value under Nameservers. This value also defaults to the Zone Default TTL value.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Click <strong>Save</strong>.</td>
</tr>
</tbody>
</table>

### CLI Commands

Use `zone name set defttl`.

### Setting Secondary Refresh Times

The secondary refresh time is how often a secondary server communicates with its primary about the potential need for a zone transfer. A good range is from an hour to a day, depending on how often you expect to change zone data.

If you use NOTIFY, you can set the refresh time to a larger value without causing long delays between transfers, because NOTIFY forces the secondary servers to notice when the primary data changes. For details about NOTIFY, see the “Enabling NOTIFY” section on page 17-5.

### Local Basic or Advanced and Regional Web UI

On the List/Add Zone page, set the Secondary Refresh field to the refresh time, which defaults to three hours. Make any other changes, then click **Save**.

### CLI Commands

Use `zone name set refresh`. The preset value is 10800 seconds (three hours).

### Setting Secondary Retry Times

The DNS server uses the secondary retry time between successive failures of a zone transfer. If the refresh interval expires and an attempt to poll for a zone transfer fails, the server continues to retry until it succeeds. A good value is between one-third and one-tenth of the refresh time. The preset value is one hour.

### Local Basic or Advanced and Regional Web UI

On the List/Add Zone page, set the Secondary Retry field to the retry time, which defaults to one hour. Make any other changes, then click **Save**.
CL Commands

Use `zone name set retry`.

Setting Secondary Expiration Times

The secondary expiration time is the longest time a secondary server can claim authority for zone data when responding to queries after it cannot receive zone updates during a zone transfer. Set this to a large number that provides enough time to survive extended primary server failure. The preset value is seven days.

Local Basic or Advanced and Regional Web UI

On the List/Add Zone page, set the Secondary Expire field to the expiration time, which defaults to seven days. Make any other changes, then click Save.

CLI Commands

Use `zone name set expire`.

Setting Local and External Port Numbers

If you are experimenting with a new group of nameservers, you might want to use nonstandard ports for answering requests and asking for remote data. The local port and external port settings control the TCP and UDP ports on which the server listens for name resolution requests, and to which port it connects when making requests to other nameservers. The standard value for both is port 53. If you change these values during normal operation, the server will appear to be unavailable.

The full list of default ports is included in the “Default Ports for Cisco Prime IP Express Services” section on page 1-9.

Local Basic or Advanced Web UI

On the Manage DNS Authoritative Server page, in A-Z view, find the Listening Port (`local-port-num`) and Remote DNS servers port (`remote-port-num`) attributes, set them to the desired values (they are both preset to 53), then click Save.

Handling Malicious DNS Clients

When trying to resolve query requests, DNS servers may encounter malicious DNS clients. A client may flood the network with suspicious DNS requests. This affects the performance of the local DNS server and remote nameservers.

Using Cisco Prime IP Express, you can resolve this problem by barring malicious clients. You can configure a global ACL of malicious clients that are to be barred, using the blackhole-acl attribute.

Local Basic or Advanced Web UI

On the Manage DNS Authoritative Server page, expand Miscellaneous Options and Settings to view various attributes and their values. For the blackhole-acl attribute value, enter, for example, 10.77.240.73. Then click Save.
## Tuning DNS Properties

Here are some tips to tune some of the DNS server properties:

- **Notify send min. interval DNS server attribute** (*notify-min-interval in the CLI*)—Minimum interval required before sending notification of consecutive changes on the same zone to a server. The preset value is two seconds. For very large zones, you might want to increase this value to exceed the maximum time to send an outbound full zone transfer. This is recommended for secondary servers that receive inbound incremental zone transfers and send out full transfers to other secondaries. These include older BIND servers that do not support incremental zone transfers. Inbound incremental transfers may abort outbound full transfers.

- **Notify delay between servers DNS server attribute** (*notify-send-stagger in the CLI*)—Interval to stagger notification of multiple servers of a change. The preset value is one second, but you may want to raise it to up to five seconds if you need to support a large number of zone transfers distributed to multiple servers.

- **Notify wait for more changes DNS server attribute** (*notify-wait in the CLI*)—Time to delay, after an initial zone change, before sending change notification to other nameservers. The preset value is five seconds, but you may want to raise it to 15, for the same reason as given for the *notify-min-interval* attribute.

- **Max. memory cache size DNS server attribute** (*mem-cache-size in the CLI*)—Size of the in-memory record cache, in kilobytes. The preset value is 50 MB and this is used to make queries for Authoritative DNS server faster. The rule of thumb is to make it as large as the number of authoritative RRs.

- **Maximum UDP payload size DNS server attribute** (*max-udp-payload-size*)—The maximum UDP payload size of the DNS server that responds to the client. You can modify this attribute from a minimum of 512 bytes to a maximum of 4 KB. The default value for this attribute is set to the maximum, that is, 4 KB on the DNS server.

- **IXFR check box in the Foreign Servers section of the Edit DNS Server page, or remote-dns address/mask create ixfr in the CLI**—Adding an entry for a server or group of servers allows controlling whether or not IXFR should occur when doing zone transfers from those servers.

## Troubleshooting DNS Servers

Useful troubleshooting hints and tools to diagnose the DNS server and ways to increase performance include:

- **Restoring a loopback zone**—A loopback zone is a reverse zone that enables a host to resolve the loopback address (127.0.0.1) to the name *localhost*. The loopback address is used by the host to enable it to direct network traffic to itself. You can configure a loopback zone manually or you can import it from an existing BIND zone file.

- **Listing the values of the DNS server attributes**—Click DNS, then DNS Server to open the Edit DNS Server page in the web UI. In the CLI, use *dns show*.

- **Adjusting certain attribute values that could have inherited preset values from previous releases during an upgrade**—The DNS server operating with legacy preset values for critical settings are probably not optimal for current systems and can cause performance issues. We strongly recommend that you update the legacy settings to use the new preset values. Table 17-1 lists the old and new preset values, along with a recommended setting for each attribute.
Troubleshooting DNS Servers

For many of these attributes, you must enter Expert mode in the web UI or use `set session visibility=3` in the CLI. To change the preset value to the current one, unset the attribute. To change to the recommended setting, change the attribute value.

Be sure to reload the DNS server after saving the settings.

- **Choosing from the DNS log settings to give you greater control over existing log messages**—Use the Log settings attribute on the Edit DNS Server page in the web UI, or `dns set log-settings` in the CLI, with one or more of these keyword or numeric values, separated by commas (see Table 17-2). Restart the server if you make any changes to the log settings.

<table>
<thead>
<tr>
<th>DNS Attribute</th>
<th>Recommended Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>axfr-multirec-default</td>
<td>on</td>
</tr>
<tr>
<td>mem-cache-size</td>
<td>50000 (KB)</td>
</tr>
</tbody>
</table>

For many of these attributes, you must enter Expert mode in the web UI or use `set session visibility=3` in the CLI. To change the preset value to the current one, unset the attribute. To change to the recommended setting, change the attribute value.

Be sure to reload the DNS server after saving the settings.

**Table 17-2 DNS Log Settings**

<table>
<thead>
<tr>
<th>Log Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Server configuration and deinitialization.</td>
</tr>
<tr>
<td>ddns</td>
<td>High level dynamic update messages.</td>
</tr>
<tr>
<td>xfr-in</td>
<td>Inbound full and incremental zone transfers.</td>
</tr>
<tr>
<td>xfr-out</td>
<td>Outbound full and incremental zone transfers.</td>
</tr>
<tr>
<td>notify</td>
<td>NOTIFY transactions.</td>
</tr>
<tr>
<td>datastore</td>
<td>Data store processing that provides insight into various events in the server embedded databases.</td>
</tr>
<tr>
<td>scavenger</td>
<td>Scavenging of dynamic RRs (see the “Scavenging Dynamic Records” section on page 29-19).</td>
</tr>
<tr>
<td>scavenger-details</td>
<td>More detailed scavenging output (disabled by default).</td>
</tr>
<tr>
<td>server-operations</td>
<td>General high-level server events, such as those pertaining to sockets and interfaces.</td>
</tr>
<tr>
<td>ddns-refreshes</td>
<td>DNS update refreshes for Windows clients (disabled by default).</td>
</tr>
<tr>
<td>ddns-refreshes-details</td>
<td>RRs refreshed during DNS updates for Windows clients (disabled by default).</td>
</tr>
<tr>
<td>tsig</td>
<td>Logs events associated with Transaction Signature (TSIG) DNS updates (see the “Transaction Security” section on page 29-10).</td>
</tr>
<tr>
<td>tsig-details</td>
<td>More detailed logging of TSIG DNS updates (disabled by default).</td>
</tr>
<tr>
<td>activity-summary</td>
<td>Summary of activities in the server. You can adjust the interval at which these summaries are taken using the <code>activity-summary-interval</code> attribute, which defaults to five-minute intervals (you can adjust this interval using <code>dns set activity-summary-interval</code>).</td>
</tr>
<tr>
<td>query-errors</td>
<td>Logs errors encountered while processing DNS queries.</td>
</tr>
<tr>
<td>config-details</td>
<td>Generates detailed information during server configuration by displaying all configured and assumed server attributes (disabled by default).</td>
</tr>
</tbody>
</table>
Troubleshooting DNS Servers

Using the nslookup utility to test and confirm the DNS configuration—This utility is a simple resolver that sends queries to Internet nameservers. To obtain help for the nslookup utility, enter help at the prompt after you invoke the command. Use only fully qualified names with a trailing dot to ensure that the lookup is the intended one. An nslookup begins with a reverse query for the nameserver itself, which may fail if the server cannot resolve this due to its configuration. Use the server command, or specify the server on the command line, to ensure that you query the proper server. Use the –debug, or better yet, the –d2, flag to dump the responses and (with –d2) the queries being sent.

Using the dig utility to troubleshoot DNS Server—dig (domain information groper) is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried. Most DNS administrators use dig to troubleshoot DNS problems because of its flexibility, ease of use, and clarity of output. To obtain help for the dig utility, enter help at the prompt after you invoke the command. Although dig is normally used with command-line arguments, it also has a batch mode of operation for reading lookup requests from a file. Unlike earlier versions, the BIND9 implementation of dig allows multiple lookups to be issued from the command line. Unless you specifically query a specific name server, dig tries each of the servers listed in /etc/resolv.conf. When no command line arguments or options are given, dig performs an NS query for the root ".". A typical invocation of dig looks like: dig @server name type where server is the name or IP address of the name server to query.

Table 17-2 DNS Log Settings (continued)

<table>
<thead>
<tr>
<th>Log Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>incoming-packets</td>
<td>Incoming data packets.</td>
</tr>
<tr>
<td>outgoing-packets</td>
<td>Outgoing data packets.</td>
</tr>
<tr>
<td>xfer-in-packets</td>
<td>Incoming full zone transfer (XFR) packets.</td>
</tr>
<tr>
<td>query-packets</td>
<td>Incoming query packets.</td>
</tr>
<tr>
<td>notify-packets</td>
<td>NOTIFY packets.</td>
</tr>
<tr>
<td>ddns-packets</td>
<td>DNS Update packets.</td>
</tr>
<tr>
<td>xfer-out-packets</td>
<td>Outgoing XFR packets.</td>
</tr>
<tr>
<td>ha-details</td>
<td>Generates detailed logging of High-Availability (HA) DNS information.</td>
</tr>
<tr>
<td>scp</td>
<td>Allows log messages associated with SCP message handling.</td>
</tr>
<tr>
<td>optRR</td>
<td>Causes logging related to OPT RR processing.</td>
</tr>
<tr>
<td>ha-messages</td>
<td>Enables detailed logging of HA messages.</td>
</tr>
</tbody>
</table>
Managing Caching DNS Server Properties

This chapter explains how to set the Caching DNS server parameters. Before you proceed with the tasks in this chapter, see Chapter 13, “Introduction to the Domain Name System” which explains the basics of DNS.

Related Topics

- Managing DNS Caching Servers, page 18-1
- Defining Forwarders, page 18-2
- Using Exceptions, page 18-3
- Managing DNS64, page 18-4
- Managing DNSSEC, page 18-5
- Setting DNS Caching Server Properties, page 18-5
- Setting Advanced Caching DNS Server Properties, page 18-9
- Caching DNS Domain Redirect, page 18-12

Managing DNS Caching Servers

You can view its health, statistics, and logs; start, stop, and reload it; run certain commands (see the “Running DNS Caching Server Commands” section on page 18-1); and edit the server attributes.

To view the server status and health, or stop, start, and reload the server, in the local cluster web UI, choose CDNS Server from the Deploy > DNS menu to open the Manage DNS Caching Server page.

Related Topics

- Running DNS Caching Server Commands, page 18-1
- Configuring CDNS Server Network Interfaces, page 18-2

Running DNS Caching Server Commands

Access the commands by using the Commands button. Clicking the Commands button opens the CDNS Commands dialog box in the local web UI. Each command has its own Run icon (click it, then close the dialog box):

- **Flush the CDNS cache**— This command allows you to flush either all RRs or RRs for a particular zone from the in-memory cache. See the “Flushing CDNS Cache” section on page 18-10
- **Flush Resource Record**— This command that lets you specify an RR name and optionally a type to remove from the in-memory cache.
Defining Forwarders

You can specify a domain for which forwarding should occur. The forwarder definition is by a list of names of servers or a list of IP addresses with an optional port number, or both.

Note
You can specify IPv4 and/or IPv6 addresses and for the changes to take effect, you must reload the CDNS server.

Tip
To force a caching DNS server to only talk to a forwarder, define a forwarder for the DNS root (.).

Local Basic or Advanced Web UI

To define a forwarder:

Step 1
From the Design menu, choose Forwarders under the Cache DNS submenu. This opens the List/Add Forwarders page.
Chapter 18  Managing Caching DNS Server Properties

Using Exceptions

Step 2 Click the Add Forwarders icon in the Forwarders pane to open the Add DnsForwarder dialog box.

Step 3 Enter the forwarder name and click Add DnsForwarder.

Step 4 In the Edit Forwarders page, enter the hostname, and click Add Host and enter the IP address for the forwarder then click Add Address.

Step 5 Click Save.

CLI Commands

Use the following cdns commands to:

- Specify the address (or space-separated addresses) of nameservers to use as forwarders, use cdns addForwarder.
- List the current forwarders, use cdns listForwarders.
- Edit your forwarder list, you must remove any offending forwarder and reenter it.
- Remove a forwarder or list of forwarders, use cdns removeForwarder.

Note For any change to the forwarders to take effect, you should restart the CDNS server.

Using Exceptions

If you do not want the CDNS server to use the standard resolution method to query the root nameserver for certain domains, use exceptions. This bypasses the root nameservers and targets a specific server (or list of servers) to handle name resolution.

Let us say that example.com has four subsidiaries: Red, Blue, Yellow, and Green. Each has its own domain under the .com domain. When users at Red want to access resources at Blue, their CDNS server follows delegations starting at the root nameservers.

These queries cause unnecessary traffic, and in some cases fail because internal resources are often barred from external queries or sites that use unreachable private networks without unique addresses.

Exceptions solve this problem. The Red administrator can list all the other example.com domains that users might want to reach and at least one corresponding nameserver. When a Red user wants to reach a Blue server, the Red server queries the Blue server instead following delegations from the root servers down.

To enable resolution exceptions, simply create an exception for the domain listing the IP address(es) and/or hostname(s) of the authoritative nameserver(s).

Note Exceptions can contain both IPv4 and/or IPv6 addresses and require a CDNS server reload to take effect.

Local Basic or Advanced Web UI

Step 1 From the Design menu, choose Exceptions under the Cache DNS submenu. This opens the List/Add Exceptions page.

Step 2 Click the Add Exceptions icon in the Exceptions pane to open the Add DnsException dialog box.
Managing DNS64

DNS64 with NAT64 provides access to the IPv4 Internet and servers for hosts that have only IPv6 addresses. DNS64 synthesizes AAAA records from A records, when a IPv6 client queries for AAAA records, but none are found. It also handles reverse queries for the NAT64 prefix(es).

Local Advanced Web UI

Step 1 From the Design menu, choose DNS64 under the Cache DNS submenu. This opens the Manage DNS64 page.

Step 2 Check the true option if you want to enable the DNS64 processing.

Step 3 If needed, add the IPv6 prefix to use for synthesizing AAAA records in the Prefix field. The prefix length must be 32, 40, 48, 56, 64, or 96, and bits 64-71 of the prefix must be zero.
In the **Expert** mode, you have the following extra options:

- to specify the IPv6 Suffix to use for synthesizing AAAA records in the **Suffix** field. The suffix is ignored if the dns64 prefix is 96 bits long.
- to set to `true` the **synthesize-all** attribute which forces DNS64 to always synthesize AAAA records from A records when they are requested.

**Step 4**

Click **Save** to save your settings.

---

**CLI Commands**

To create DNS64 in the DNS Caching server, use `cdns64 create`. To enable DNS64, use `cdns64 enable dns64` (see the cdns64 command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions).

---

**Managing DNSSEC**

DNSSEC enables the server to determine the security status of all Resource Records that are retrieved. You can manage DNSSEC only in the Advanced mode. DNSSEC requires a root trust anchor to establish trust for the DNS root servers. The initial DNSSEC root trust anchor, root.anchor, is stored in the `.../data/cdns` directory and is the default value of the `auto-trust-anchor-file` attribute. Additional trust anchors may be added by adding them to the `.../data/cdns` directory and to the `auto-trust-anchor-file` if the zone supports automated updates according to RFC 5011 or the `trust-anchor-file` attribute if not. The `cdnssec` command controls and configures DNSSEC processing in the Cisco Prime IP Express DNS Caching server.

---

**Local Basic or Advanced Web UI**

**Step 1**

From the **Design** menu, choose **DNSSEC** under the **Security** submenu to open the Manage DNSSEC page.

**Step 2**

Enable DNSSEC validation by selecting the **enabled** option.

**Step 3**

The page displays all the DNSSEC attributes. Modify the attributes as per your requirements.

**Step 4**

Click **Save** to save your settings.

---

**CLI Commands**

To create DNSSEC in the DNS Caching server, use `cdnssec create`. To enable DNS64, use `cdns64 enable dns64` (see the cdnssec command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions).

---

**Setting DNS Caching Server Properties**

You can set properties for the Caching DNS server. These include:

- **General server properties**—See the “Setting General CDNS Server Properties” section on page 18-6.
Setting General CDNS Server Properties

You can view CDNS general server properties, such as log settings, basic cache settings, SNMP traps, and root nameservers.

The following subsections describe some of the most common property settings. They are listed in the “Setting DNS Caching Server Properties” section on page 18-5.

Local Basic or Advanced Web UI

Step 1 To access the server properties, choose CDNS Server from the Deploy > DNS submenu to open the Manage DNS Caching Server page.

Step 2 Select Local CDNS Server from the CDNS Server pane, to open the Edit Local CDNS Server page. The page displays all the CDNS server attributes.

Step 3 Click Save to save the CDNS server attribute modifications.

CLI Commands

Use cdns show to display the CDNS server properties (see the cdns command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions).

Specifying Log Settings

This setting determines which detailed events the Caching DNS server logs, as set using a bit mask. Logging these additional details can help analyze a problem. Leaving detailed logging enabled for a long period, however, can fill the log files and cause the loss of important information.

The possible options are:

- **config**—Controls logging pertaining to server configuration and server de-initialization (unconfiguration).
- **server-ops**—Controls high level logging of server operations.
- **server-detailed-ops**—Controls detailed logging of server operations.
- **scp**—Controls logging pertaining to SCP message processing.
- **activity-summary**—This causes a summary message to appear at an interval specified by activity-summary-interval. The summary provides detailed statistics about the servers operation.
- **query**—Causes logging of all DNS queries to the server.
Specifying Activity Summary Settings

**Note**
To specify the activity summary settings, you have to check `activity-summary` under the Log Settings.

You can specify the interval at which to log activity-summary information using the Statistics Interval (`activity-summary-interval`) attribute.

The Caching DNS server logs sample and/or total statistics based on the option you check for the attribute Statistics Type (`activity-summary-type`).

The option checked for the attribute Statistics Settings (`activity-summary-settings`) determines the category of statistics that is logged as part of activity summary. The possible settings are:

- query—Logs statistics related to incoming queries.
- query-type—Logs statistics on the RR types that are being queried.
- cache—Logs statistics on the RR cache.
- resol-queue—Logs statistics on the resolution queue.
- responses—Logs statistics about query responses.
- memory—Logs statistics on memory usage.
- redirect—Logs statistics on redirect usage.

Specifying Caching Settings

To set the cache TTLs, see the "Setting Cache TTLs" section on page 18-7.

Use the `Prefetch` attribute to set whether message cache elements should be prefetched before they expire to keep the cache up to date. Turning it on gives about 10 percent more traffic and load on the machine, but popular items do not expire from the cache.

When prefetch is enabled, records are assigned a prefetch time that is within 10 percent of the expiration time. As the server processes client queries and looks up the records, it checks the prefetch time. Once the record is within 10 percent of its expiration, the server will issue a query for the record in order to keep it from expiring.

Setting Cache TTLs

TTL is the amount of time that any nameserver is allowed to cache data learned from other nameservers. Each record added to the cache arrives with some TTL value. When the TTL period expires, the server must discard the cached data and get new data from the authoritative nameservers the next time it sends a query. TTL attributes, `cache-min-ttl` and `cache-max-ttl` defines the minimum and the maximum time Cisco Prime IP Express retains the cached information. These parameters limit the lifetime of records in the cache whose TTL values are very large.

Local Basic and Advanced Web UI

**Step 1**
On the Edit Local CDNS Server tab, in A-Z view, you can find:

- the Maximum Cache TTL (`cache-max-ttl`) attribute, set it to the desired value (the preset value is 24 hours)
• the Min Cache TTL (cache-min-ttl) attribute, set it to the desired value (the preset value is 0)

**Step 2**
Click **Save** to save the changes.

**CLI Commands**

Use:

- `cdns set cache-max-ttl` to set the Maximum Cache TTL.
- `cdns set cache-min-ttl` to set the Minimum Cache TTL.

**Defining Root Nameservers**

Root nameservers know the addresses of the authoritative nameservers for all the top-level domains. When you first start a newly installed Cisco Prime IP Express DNS server, it uses a set of preconfigured root servers, called root hints, as authorities to ask for the current root nameservers.

When Cisco Prime IP Express gets a response to a root server query, it caches it and refers to the root hint list. When the cache expires, the server repeats the process. The time to live (TTL) on the official root server records is currently six days, so Cisco Prime IP Express requeries every six days, unless you specify a lower maximum cache TTL value (see the “Setting Cache TTLs” section on page 18-7).

Because the configured servers are only hints, they do not need to be a complete set. You should periodically (every month to six months) look up the root servers to see if the information needs to be altered or augmented.

**Local Basic or Advanced Web UI**

On the Edit Local CDNS Server tab, under the Root Name Servers category, enter the domain name and IP address of each additional root nameserver, clicking **Add Root Nameserver** after each one, then click **Save**.

**CLI Commands**

Use `cdns addRootHint`.

**Dynamic Allocation of UDP Ports**

The Caching DNS server uses a large number of UDP port numbers, by default approximately 60000 port numbers. These numbers are divided among the processing threads. The large number of port numbers reduce the risk of cache poisoning via Birthday Attacks. The Caching DNS server uses the default pool of UDP ports (2048) and the maximum allowable size of the default pool of UDP ports is 4096.

Currently, Cisco Prime IP Express uses the port range from 1024 to 65535. Based on the number of outstanding resolution queries, the Caching DNS server adjusts the pool size by adding or removing ports. The Caching DNS server allocates and releases the UDP ports dynamically when the server is running. If you reload the server, all the UDP ports are released and randomly picked again.

Cisco Prime IP Express uses `outgoing-range-avoid` attribute that allows you to define ports or ranges of ports that will be excluded from use by the DNS server when sending queries.
Note

You need to ensure that UDP ports needed by other applications are in the port exclusion list. Otherwise, these applications may not be able to bind to their port(s) if the DNS server is using the port.

Local Basic or Advanced Web UI

On the Edit Local CDNS Server tab, expand Additional Attributes to view various attributes and their values. For the query-source-port-exclusion-list attribute value, enter a range of ports that need to be excluded. Then click Modify Server.

Setting Advanced Caching DNS Server Properties

You can set these advanced server properties:

- Maximum memory cache sizes—See the “Setting Maximum Memory Cache Sizes” section on page 18-9.
- Network Settings—See the “Specifying Network Settings” section on page 18-10.
- Flush cache—See the “Flushing CDNS Cache” section on page 18-10.
- Prevent DNS cache poisoning—See the “Detecting and Preventing DNS Cache Poisoning” section on page 18-11.
- Handle unresponsive nameservers—See the “Detecting and Preventing DNS Cache Poisoning” section on page 18-11.

Setting Maximum Memory Cache Sizes

The maximum memory cache size property specifies how much memory space you want to reserve for the DNS in-memory cache. The larger the memory cache, the less frequently the Caching DNS server will need to re-resolve unexpired records.

Local Advanced Web UI

On the Edit Local CDNS Server tab, in the Caching category, set it to the desired value for the RRSet Cache Size (rrset-cache-size), then click Save. The default size is 100MB.

To set the size of the message cache, use the Message Cache Size (msg-cache-size) attribute. The default value for Message Cache Size is 200 MB. The message cache stores query responses. It should generally be twice the size of the RRSet Cache Size (rrset-cache-size).

CLI Commands

- Use `cdns set rrset-cache-size` to set RRSet Cache Size.
- Use `cdns set msg-cache-size` to set Message Cache Size.
Specifying Network Settings

The \textit{listen-ip-version} attribute lets you to choose the ip packets to accept and issue. You can check IPv4, IPv6, both, or none. The \textit{listen-protocol} attribute lets you to choose the packet protocol to answer and issue, UDP, TCP, both, or none.

Flushing CDNS Cache

The Cisco Prime IP Express cache flushing function lets you remove all or a portion of cached data in the memory cache of the server.

Local Basic or Advanced Web UI

\begin{itemize}
  \item \textbf{Step 1} From the \textbf{Deploy} menu, choose \textbf{CDNS Server} under the \textbf{DNS} submenu, to open the Manage DNS Caching Server page.
  \item \textbf{Step 2} On the Manage DNS Caching Server page, click the Commands link to open the CDNS Command dialog box. There will be two types of cache flushing commands.
    \begin{itemize}
      \item Flush the CDNS cache—allows you to either flush all cache entries for a particular zone or the entire cache if no zone is provided. To remove all data for a specific zone, enter the zone name in the Zone field. To clear the whole cache, leave the Zone field empty.
      \item The Flush Resource Record—allows you to flush an RR name or an RRSet when the type field is specified.
        \begin{itemize}
          \item Remove common RR types (A, AAAA, NS, SOA, CNAME, DNAME, MX, PTR, SRV, NAPTR, and TXT) from a specific domain—enter the required RR name as the FQDN for the Flush Resource Record command and leave the RR type field empty.
          \item Remove a specified RR type for a domain—specify the domain in the FQDN field, and the RR type in the RR type field.
        \end{itemize}
    \end{itemize}
\end{itemize}

\begin{itemize}
  \item \textbf{Note} When no type is specified, the server flushes types A, AAAA, NS, SOA, CNAME, DNAME, MX, PTR, SRV, TXT, and NAPTR.
\end{itemize}

CLI Commands

\begin{itemize}
  \item To:
    \begin{itemize}
      \item Remove all cached entries at or below a given domain, use \texttt{cdns flushCache domain}. If no domain is given, it flushes all RRs in the cache.
      \item Flush RRs from the cache associated with the given RR name, use \texttt{cdns flushName name type}. When type is provided, it flushes all entries with the given name and type. If no type if provided, it flushes types A, AAAA, NS, SOA, CNAME, DNAME, MX, PTR, SRV, TXT, and NAPTR.
    \end{itemize}
\end{itemize}
Detecting and Preventing DNS Cache Poisoning

Cisco Prime IP Express enhances the CDNS server performance to address the CDNS related issues such as DNS cache poisoning attacks (CSCsq01298), as addressed in a Cisco Product Security Incident Response Team (PSIRT) document number PSIRT-107064 with Advisory ID cisco-sa-20080708-dns, available at:
http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20080708-dns

DNS Cache Poisoning Attacks

A cache poisoning attack can change an existing entry in the DNS cache as well as insert a new invalid record into the DNS cache. This attack causes a hostname to point to the wrong IP address. For example, let us say that www.example.com is mapped to the IP address 192.168.0.1, and this mapping is present in the cache of a DNS server. An attacker can poison the DNS cache and map www.example.com to 10.0.0.1. If this happens, if you try to visit www.example.com, you will end up contacting the wrong web server.

A DNS server that uses a single static port for receiving responses to forwarded queries are susceptible to malicious clients sending forged responses.

The DNS transaction ID and source port number used to validate DNS responses are not sufficiently randomized and can easily be predicted, which allows an attacker to create forged responses to DNS queries. The DNS server will consider such responses as valid.

Handling DNS Cache Poisoning Attacks

To reduce the susceptibility to the DNS cache poisoning attack, the DNS server randomizes the UDP source ports used for forwarded queries. Also, a resolver implementation must match responses to the following attributes of the query:

- Remote address.
- Local address.
- Query port.
- Query ID.
- Question name (not case-sensitive).
- Question class and type, before applying DNS trustworthiness rules (see [RFC2181], section 5.4.1).

The response source IP address must match the query's destination IP address and the response destination IP address must match the query's source IP address. A mismatch must be considered as format error, and the response is invalid.

Resolver implementations must:

- Use an unpredictable source port for outgoing queries from a range (either 53, or > 1024) of available ports that is as large as possible and practicable.
- Use multiple different source ports simultaneously in case of multiple outstanding queries.
- Use an unpredictable query ID for outgoing queries, utilizing the full range available (0 to 65535). By default, CDNS uses about 60000 port numbers.
Chapter 18  Managing Caching DNS Server Properties

Caching DNS Domain Redirect

The Expert mode Caching DNS server setting randomize-query-case, when enabled, specifies that when sending a recursive query, the query name is pseudo-randomly camel-cased and the response is checked to see if this camel-casing is unchanged. If randomize-query-case is enabled and the casing has changed, then the response is discarded. The randomize-query-case is disabled by default, disabling this feature.

Local Basic or Advanced Web UI

The DNS server statistics appears on the Statistics tab of the Manage DNS Caching Server Statistics page. The Statistics displays the answers-unwanted values. You can refresh the DNS Caching Server Statistics.

Handling Unresponsive Nameservers

When trying to resolve query requests, Caching DNS servers may encounter unresponsive nameservers. A nameserver may be unresponsive to queries, respond late. This affects the performance of the local DNS server and remote nameservers.

Using Cisco Prime IP Express, you can resolve these problems by barring unresponsive nameservers. You can configure a global ACL of unresponsive nameservers that are to be barred, using the acl-do-not-query attribute.

When Cisco Prime IP Express receives a list of remote nameservers to transmit a DNS query request to, it checks for the name-servers listed in the acl-do-not-query list and removes them from this list. Conversely, all incoming DNS requests from clients or other nameservers are also filtered against the acl-blacklist. The acl-blacklist attribute

Note

Using the acl-do-not-query does not affect the configuration of communication with certain servers such as forwarders.

Use the acl-query attribute to specify which clients are allowed to query the server. By default any client is allowed to query the server. A client that is not in this list will receive a reply with status REFUSED. Clients on the acl-blacklist do not get any response whatsoever.

Local Advanced Web UI

On the Edit Local CDNS Caching Server tab, expand Query Access Control to view the various attributes and their values. For the Do Not Query (acl-do-not-query) attribute value, enter, for example, 10.77.240.73. Then click Save.

Caching DNS Domain Redirect

DNS domain redirect enables Internet Service Providers (ISP), enterprises, or organizations to redirect the resolution of DNS name away from known bad domains or non-existing domains (NXDOMAIN) for a specified ACL and/or a domain list. Cisco Prime IP Express supports DNS domain and NXDOMAIN redirect to override the caching DNS server response to A or AAAA resource record queries.

To ensure that the caching DNS server redirects queries for non-existing or known bad domains, you can create DNS redirect rules. A domain redirect rule comprises of a priority, an ACL, an action, and a list of domains. The domain redirect rules take precedence over exceptions and forwarders.
Every query to a caching DNS server is first verified against the list of redirect rules in the order of priority. When a resource record query matches the criteria of rule, the specified action is taken. If the resource record query action results for redirect, deny, refuse, the corresponding action is taken. If it is nxdomain, the query is performed in the normal process and if it results in an NXDOMAIN status, then it is redirected to the specified destination.

**Note**
The Deny and Refuse rules are applicable to all the queries for the specified domains, while the redirect rules and NXDOMAIN are applicable only to the queries of A and AAAA records.

### Local Basic or Advanced Web UI

To add, edit, or view the Domain Redirect Rule:

1. From the **Design** menu, choose **Domain Redirect** under the **Cache DNS** submenu to open the List/Add CDN Domain Redirect Rules page.
2. Click the **Add CDN Domain Redirect Rule** icon in the Domain Redirect pane to open the Add CDN Domain Redirect dialog box.
3. Enter a rule name in the **Rule Name** field, and an ACL list in the **Acl List** field.
4. Choose an action from the below list:
   - **deny**—To ignore the resource record query.
   - **refuse**—To block a resource record query.
   - **redirect**—To override the caching DNS response for known bad domains and redirect it to a specified IP address.
   - **nxdomain**—To override the caching DNS response for domains that were not found and redirect to a specified IP address.

**Note**
The rules with the actions Deny and Refuse do not use a destination IP.

5. Click **Add CDN Domain Redirect** to save the redirect rule. The **List/Add CDN Domain Redirect Rules** page appears with the newly added redirect rule.

6. Enter the domains that have to be monitored for the redirection.

**Note**
The NXDOMAIN action do not takes domain list.

7. Enter the IPv4 and IPv6 destinations IP addresses.

8. Click **Save** to save your settings, or **Revert** to cancel the changes.

To delete an domain redirect rule, select the rule on the Domain Redirect pane, click the **Delete** icon, and then confirm or cancel the deletion.

### CLI Commands

Use the following cli commands to:

- Add the domain redirect rule, separated by spaces, use `cdns-redirect rule-name create`. 
Caching DNS Domain Redirect

- List the domains the domain redirect rule, use `cdns-redirect list [-priority]`. The priority is optional. It lists the redirect entries according to priority instead of the default, alphabetically by name.
- Remove domain redirect rule, use `cdns-redirect rule-name delete`.

Reorder DNS Domain Redirect Rules

When you create a set of domain redirect rules, you can specify the priority in which order the rules will apply. To set the priority or reorder the rules:

---

**Step 1**  
From the **Design** menu, choose **Domain Redirect** under the **Cache DNS** submenu to open the **List/Add CDNS Domain Redirect Rules** page.

**Step 2**  
Click the **Reorder Rules** icon in the Domain Redirect pane to open the Reorder dialog box.

**Step 3**  
Set the priority for the DNS domain redirect rules by either of the following methods:

- Select the rule and click the **Move up** or **Move down** icon to reorder the rules.
- Select the rule and click the **Move to** button, and enter the row number to move the rule.

**Step 4**  
Click **Save** to save the reordered list.
Configuring High-Availability DNS Servers

A second primary server can be made available as a hot standby that shadows the main primary server. This configuration is called High-Availability (HA) DNS. The Cisco Prime IP Express web UI and CLI have features with which you can duplicate the primary setup required for HA DNS for the server pair. The server pair is responsible for detecting communication failures and the like. After the HA DNS is configured, the shadowing and error detection is done automatically. In a Cisco Prime IP Express deployment where Cisco Prime IP Express DHCP is updating Cisco Prime IP Express DNS, the failure detection and failover also happens automatically.

Related Topics

HA DNS Processing, page 19-1
Configuring an HA DNS Server Pair from Main Server, page 19-3
DNS Server Configuration for HA DNS, page 19-4
Synchronizing HA DNS Zones, page 19-5
HA DNS Configuration Synchronization, page 19-5
HA DNS Statistics, page 19-8

HA DNS Processing

In normal state, both the main and backup primary servers are up and running. The main server processes all DNS updates from clients and sends all accepted updates to the hot standby backup. The main server will forward RR updates to the backup server and the backup server only accepts updates from the main in normal state. In normal states, updates from DDNS clients are ignored or dropped by a backup server. Both servers can respond to queries and zone transfer requests. The main and backup partners exchange heartbeat messages to detect if the other is not available.

If the main goes down, the backup waits a short time, then begins servicing the DNS updates from clients that the main would normally service and records the updates. When the main returns, the backup sends it the updates, and the main synchronizes with the backup any updates that were not sent and which it had before it went down.

Whenever you add a new zone in the HA main server, it is automatically synchronized to the HA backup. The synchronization is done on a per-zone basis. This allows updates to all other zones while a given zone is in the process of getting synchronized.

If the hot standby backup goes down, the main waits a short time, then records the updates that the partner did not acknowledge. When the backup server comes back up, the main sends the recorded updates to the backup.
Both the main and backup can traverse the following states:

- **Startup**—The servers establish communication and agree on the HA version to use. In this state, the servers do not accept DNS updates or RR edits, and they defer scavenging, if enabled.

- **Negotiating**—Each server is waiting for the other to get ready to synchronize. In this state, DNS Updates and RR edits are not allowed.

- **Normal**—Both servers are up and healthy, exchanging DNS updates and heartbeat messages. The main accepts DNS updates and RR edits, sends RR Update messages to the backup, and performs history trimming and scavenging, if enabled. The backup ignores DNS updates, refuses RR edits, but processes RR Update messages from the main server. The backup also performs history trimming, but defers scavenging, if enabled. In this state, the synchronization takes place.

- **Communication-Interrupted**—The server goes into this state after not getting a response or request from the partner during the communication timeout (ha-dns-comm-timeout) period (preset to 30 seconds). The server continues listening for communication from the partner (they both send heartbeat messages every 12 seconds) and tries to connect, meanwhile accepting DNS updates and RR edits and disabling scavenging.

- **Partner-Down**—The server administrator notifies the partner that it will be down for an extended time. This manual intervention is possible only in Communication-Interrupted state. Either server continues listening for communication from the partner and tries to connect, accepts DNS updates and RR edits, and performs scavenging.

When a DNS server starts up, it:

1. Tries to establish a connection with its partner.
2. Transitions to Negotiating state.
3. Transitions to Normal state, after it receives a Negotiating response.

Once the server is in Normal state, the zone level synchronization begins. Zone synchronization is always managed by the Main HA server. The zones traverse through the following states:

- **Sync-Pending State**—A zone enters this state when the HA DNS server transitions to the normal state or if a manual sync is requested. In this state RR updates for the zone will be accepted on the main server, and forwarded to the backup server.

- **Synchronizing State**—The RR synchronization for the zone takes place in the synchronizing state. RR updates are not accepted, and notifies are disabled.

- **Sync-Complete State**—A zone transitions to this state from the synchronizing state once it has successfully synchronized resource record changes with its corresponding zone on the HA DNS backup. In this state, the zone on the HA DNS main server accepts all dynamic DNS update requests, allow resource record configuration changes, and re-enables notifies. Resource record modifications will be forwarded to the backup server.

- **Sync-Failed State**—A zone transitions to the sync-failed state from the synchronizing state if it fails to sync. The zone will accept resource record updates on the main server, and changes will be forwarded to the backup. The server will retry synchronizing the zone after ha-dns-zonesync-failed-timeout. A manual sync request or server restart will also restart zone synchronization.

HA DNS is fully integrated with DHCP servers, and the partners are updated when hosts get added to the network (see Chapter 29, “Configuring DNS Update”). From the DHCP side of HA DNS, the DHCP server sends DNS updates to a single DNS server at a time.

DHCP autodetects the main being down and start sending updates to the backup. The DHCP server tries to contact the main DNS server, twice. It tries the backup partner if both of the attempts are unsuccessful.
The backup detects the main server down and starts accepting updates from DDNS clients. When the servers come up again, HA communication will be automatically established and the servers will get into Normal state where they carry out zone synchronization and make sure that both have the same RRs, etc.

If both DNS partners are communicating, the backup server drops the update, whereby the DHCP server times out and retries the main DNS server. If both servers are unreachable or unresponsive, the DHCP server continually retries each DNS partner every 4 seconds until it gets a response.

An Expert mode command is added in the local cluster DNS Server Commands page to synchronize all HA zones, if it is configured as the main HA server. The following three options are provided:

- Sync All RRs using the HA server synchronization algorithms
- Sync All RRs from Main to Backup
- Sync All RRs from Backup to Main

For zone level sync, an Advanced mode command is added in the local cluster Zone Commands page, if the local cluster is configured as the main HA server. The sync is run using the HA server algorithms by default. In Expert mode, the following three options are provided:

- Sync All RRs using the HA server synchronization algorithms
- Sync All RRs from Main to Backup
- Sync All RRs from Backup to Main

HA DNS status is modified to include the zone synchronization status. Status includes count and percentage of synchronized zones, zones pending synchronization, and zones that have failed synchronization.

Zone status has been modified to also include the HA synchronization status (ha-server-pending, sync-pending, sync-complete, synchronizing, or sync-failed), if HA is configured.

The ha-dns-comm-timeout attribute managed through the HA pair indicates the time required to determine if a partner is unreachable, after network communication is not acknowledged, which triggers the Communication-Interrupted state (see the description of this state in the “HA DNS Processing” section on page 19-1). The preset value is 30s. The server tries to communicate and then back off at multiples of the ha-dns-comm-timeout interval.

### Configuring an HA DNS Server Pair from Main Server

The attributes needed to set up an HA DNS server pair from the main server are:

- `ha-dns`—Enabled or disabled. The preset value is enabled.
- `main`—cluster for the main primary DNS server.
- `backup`—cluster for the backup primary DNS server.

The specific IP addresses for the main or backup is specified only when the cluster IP is only used for management and DNS works on a different interface.

#### Local Basic or Advanced and Regional Web UI

**Step 1** Create a cluster for the backup server.

**Step 2** From the **Deploy** menu, choose **HA** under the **DNS** submenu to open the View/Add HA DNS Server Pair page.

**Step 3** Click the **Add HA Pair** icon in the HA Pairs pane to open the Add HA DNS Server dialog box.
Step 4 Enter the name of the server pair in the name field. This can be any identifying text string.

Step 5 Click the cluster name of the main DNS server in the Main Server drop-down list.

Note If you change the IP address of your local host machine, you must modify the localhost cluster (on the Edit Cluster page) to change the address in the IP Address field. Do not set the value to 127.0.0.1.

Step 6 Click the cluster name of the backup DNS server in the Backup Server drop-down list. This cannot be the same as the main server cluster. Set the ha-dns-main-server and ha-dns-backup-server attributes only if the server is configured with different interfaces for configuration management and update requests. (Configure the HA DNS protocol only with the interface used to service updates.)

Step 7 Click Add HA DNS Server.

Step 8 Once the server pair appears on the List/Add HA DNS Server Pair page, synchronize the servers:
   a. Select the HA in the HA Pairs pane and click the Sync HA DNS Server Pair tab.
   b. Choose the direction of synchronization (Main to Backup or Backup to Main).
   c. Choose the operation type (Update, Complete, or Exact). See the table on the page for details on the operations for each operation type.
   d. Click the Report button to display the prospective synchronization changes on the View HA DNS Sync Report page.
   e. Click Run Complete to complete the synchronization.
   f. Click Return to return to the List HA DNS Server Pairs page.

Step 9 Reload both DNS servers to begin HA communication.

CLI Commands

Create the HA DNS server pair (ha-dns-pair name create mainaddr backupaddr). Then synchronize the servers using ha-dns-pair name sync, specifying the synchronization operation (update, complete, or exact) and direction (main-to-backup or backup-to-main). Be sure to reload both DNS servers. For example:

```
nrcmd> ha-dns-pair examplehadnspair create localhost test-cluster
nrcmd> ha-dns-pair examplehadnspair sync exact main-to-backup
nrcmd> dns reload
```

See the ha-dns-pair command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions. The CLI provides an additional command for the DNS server to set the HA DNS partner down, if necessary, which is possible only while in Communication-Interrupted state:

```
nrcmd> dns setPartnerDown
```

The partner down is useful because it limits the bookkeeping data a server maintains, thus optimizing its performance. When both servers start communicating again, the sync sends all the zone RRs rather than trying to determine individual changes.

DNS Server Configuration for HA DNS

The log settings, ha-details and ha-messages, enable logging of HA DNS-related information.
Synchronizing HA DNS Zones

Local Advanced Web UI

To manually synchronize an HA DNS zone:

**Step 1**  
From the Design menu, choose Forward Zones or Reverse Zones under the Auth DNS submenu to open the List/Add Forward Zones or List/Add Reverse Zones page.

**Step 2**  
Click the Commands button for the zone which you want to synchronize on the Edit Zone page.

**Step 3**  
Click the command icon next to Synchronize HA Zone to synchronize the HA DNS zone.

Synchronizing the HA DNS zone will always sync the associated views and named ACLs for primary zones.

---

**Note**  
In the Expert mode, you have the option to choose the type of synchronization. The Use Server Algorithms option is checked by default. If you click the command icon next to the Synchronize HA Zone without choosing another option, server algorithms will be used to synchronize the zone. You can override this by checking either Push Full Zone From Main to Backup check box or Pull Full Zone From Backup to Main check box.

---

**CLI Commands**

Use zone name ha-sync-all-rrs to manually schedule HA zone synchronization for the zone, or to raise its priority, if the zone is already in the sync-pending state (see the zone command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions).

---

**HA DNS Configuration Synchronization**

Throughout this procedure the source system is referred as DNS HA main server and destination as DNS HA backup server. When you enable the HA DNS with large DNS configuration, you will notice that the process takes long time to complete. This section provides a workaround, which you can use until the defect is addressed.

---

**Warning**  
To perform this process, you must have HA main server and HA backup server running on the same OS, Cisco Prime IP Express version, and DNS configuration.

---

**Initial Setup Considerations**

If the configuration information resides on a system that will be eventually used as HA DNS backup system, and if you bring in a new system online as the HA DNS main, the backup system functions as source and main system functions as destination.
Warning

The HA DNS backup server must not contain any pre-existing Cisco Prime IP Express configuration that needs to be maintained, as all the DNS configuration data in the HA DNS backup server will be lost on completion of this procedure.

Migration Procedure

This section describes the migration procedure used to migrate Cisco Prime IP Express product databases from the HA DNS main server to the HA DNS backup server.

Related Topics

- Pre-install Cisco Prime IP Express on the HA DNS backup server, page 19-6
- Pre-migration Steps for HA DNS Main Server, page 19-6
- Restart Cisco Prime IP Express on the HA DNS Main Server, page 19-7
- Copy Cisco Prime IP Express Database Files to HA DNS Backup Server, page 19-7
- Reconfigure Cisco Prime IP Express on the HA DNS Backup Server, page 19-7
- Configure Cisco Prime IP Express HA DNS on the HA DNS Main Server, page 19-8
- Reload the DNS Servers, page 19-8

Pre-install Cisco Prime IP Express on the HA DNS backup server

You need to pre-install Cisco Prime IP Express on the HA DNS backup system before migrating the database directory from the HA DNS main system, to reduce the time required during the Cisco Prime IP Express software installation process. During the installation process, the installer will verify whether any previous configuration is up to date with the Cisco Prime IP Express data schema for the version being installed. Even if the versions are identical, the time required to perform this verification can be avoided by pre-installing Cisco Prime IP Express on the HA DNS backup system.

Pre-migration Steps for HA DNS Main Server

You must ensure that the service of DHCP server is available and running on different systems, especially when there is a large DNS configuration. If the servers are found on the same system, the migration from HA DNS main server to backup server may cause DHCP conflict, and DHCP clients may be destabilized.

Follow the pre-migration steps as below:

**Step 1**
Disable the automatic start-on-reboot setting for the DHCP server.
```
ncmd> server dhcp disable start-on-reboot
```

**Step 2**
Stop the Cisco Prime IP Express on the HA DNS main server using the Windows Service Control manager (Windows) or nwreglocal script in /etc/init.d (Linux).

**Step 3**
Once the Cisco Prime IP Express is stopped by using Windows Process Manager (Windows) or ps command line utility (Linux), navigate to the parent directory of the Cisco Prime IP Express data directory, InstallDir\Cisco Prime IP Express\Local\ (Windows) or /var/nwreg2/local/ on (Linux).

**Step 4**
Using tar or an equivalent compression utility, bundle up the contents of the data subdirectory. InstallDir is the directory where you have installed your Cisco Prime IP Express: tar -cvf cnrdatadir.tar data.
Replace all the .bak database backup directories temporarily from HA DNS main server. The HA backup server does not need these backup directories and replacing them reduces the overall archive size. Be sure that you do not replace any other database files other than .bak; otherwise, the HA DNS backup cluster may not function properly.

**Restart Cisco Prime IP Express on the HA DNS Main Server**

**Step 1** Restart the Cisco Prime IP Express servers on the HA DNS main system using the Windows Service Control manager (Windows) or nwreglo cal script in /etc/init.d (Linux).

**Step 2** Restore the DHCP server start-on-reboot attribute values to their pre-migration values:

```
nrcmd> server dhcp enable start-on-reboot  
nrcmd> server dhcp start
```

**Copy Cisco Prime IP Express Database Files to HA DNS Backup Server**

**Step 1** Use FTP or an equivalent network file copy mechanism to transfer the Cisco Prime IP Express database archive that was generated in the previous step to the parent directory of the Cisco Prime IP Express data directory (typically C:\CiscoPrimeIPExpress\Local\ on Windows, and /var/nwreg2/local/ on Linux) on the HA DNS backup server.

**Step 2** Ensure that the mechanism used to transfer the database archive preserves binary file data. If FTP sessions default to ASCII mode, change it to binary mode in order to produce a usable database on the HA DNS backup server.

**Step 3** Stop the Cisco Prime IP Express product on the HA DNS backup server completely using the Windows Service Control manager (Windows) or nwreglo cal script in /etc/init.d (Linux). Ensure that the product is completely stopped, either by using the Windows Process Manager or the ps command line utility on Linux, navigate to the parent directory of the Cisco Prime IP Express data directory (typically C:\CiscoPrimeIPExpress\Local\ on Windows, and /var/nwreg2/local/ on Linux).

**Step 4** Ensure to recursively remove all contents of the existing data directory, to prevent any conflicts with the database archive that is about to be extracted. Using tar or an equivalent utility, extract the contents of the database archive file: tar -xvf cnrdatadir.tar.

**Reconfigure Cisco Prime IP Express on the HA DNS Backup Server**

**Step 1** Start the Cisco Prime IP Express servers on the HA DNS backup system using the Windows Service Control manager (Windows) or nwreglo cal script in /etc/init.d (Linux).

**Step 2** Rectify the conflicts, if any, between HA DNS main system and any DHCP server configuration settings.

**Step 3** The DHCP integrity will be compromised if the DHCP server has a configuration similar to that of HA DNS main system. To know more on increasing the DHCP service availability, refer to the Cisco Prime IP Express product documentation. Cisco recommends that you completely remove any DHCP related
configuration on the HA DNS backup system using either the web UI or nrcmd CLI. You can restore the original DHCP server-start-on-reboot attribute values, only after you confirm that the configuration values do not conflict with that of the HA DNS main system.

```
nrcmd> server dhcp enable start-on-reboot
```

**Step 4** Edit the localhost Cluster object in the HA DNS backup server to reflect the values in use on the local server.

---

### Configure Cisco Prime IP Express HA DNS on the HA DNS Main Server

**Step 1** In HA DNS main server, define appropriate Cluster objects for both the HA DNS main and HA DNS backup servers.

**Step 2** Create an HA Pair object by specifying appropriate Cluster names for the main and backup DNS server roles, and enable HA DNS for the HA Pair.

**Step 3** Generate the report of changesets and exchange them between the two servers using the default report generation settings (Main-to-backup, Complete).

**Step 4** Perform the changeset synchronization while the list of changesets is displayed.

---

### Reload the DNS Servers

**Step 1** Reload the DNS servers on both HA DNS systems to initiate the DNS RR synchronization process. Do it either through the Manage Servers page on the HA DNS main cluster when the HA DNS main server’s DNS server has finished reloading, or to save a little time, initiate through separate connections to both clusters to perform the reloads in parallel instead of series.

**Step 2** When the DNS servers are synchronizing, Cisco Prime IP Express does not allow DNS configuration updates (such as DDNS), but provides DNS queries and zone transfer. You can monitor the DNS server log files on the main and backup clusters to follow the progress of the DNS server synchronization process. The servers are fully operational when HA DNS enters Normal state.

---

### HA DNS Statistics

You can view HA DNS statistics.

**Local Basic or Advanced Web UI**

Click the Statistics tab on the Manage DNS Authoritative Server page to open the DNS Server Statistics page. The statistics appear under the HA Statistics and Max Counter Statistics subcategories of both the Total Statistics and Sample Statistics categories.

**CLI Commands**

Use `dns getStats ha [total]` to view the HA DNS Total counters statistics, and `dns getStats ha sample` to view the Sampled counters statistics.
PART 5

Dynamic Host Administration
Introduction to Dynamic Host Configuration

All hosts seeking Internet access must have an IP address. As Internet administrator, you must perform the following for every new user and for every user whose computer was moved to another subnet:

1. Choose a legal IP address.
2. Assign the address to the individual workstation.
3. Define workstation configuration parameters.
4. Update the DNS database, mapping the workstation name to the IP address.

These activities are time consuming and error prone, hence the Dynamic Host Configuration Protocol (DHCP). DHCP frees you from the burden of individually assigning IP addresses. It was designed by the Internet Engineering Task Force (IETF) to reduce the amount of configuration required when using TCP/IP. DHCP allocates IP addresses to hosts. It also provides all the parameters that hosts require to operate and exchange information on the Internet network to which they are attached.

DHCP localizes TCP/IP configuration information. It also manages allocating TCP/IP configuration data by automatically assigning IP addresses to systems configured to use DHCP. Thus, you can ensure that hosts have Internet access without having to configure each host individually.

Related Topics

How DHCP Works, page 20-1
Cisco Prime IP Express DHCP Implementations, page 20-4
DNS Update, page 20-6
DHCP Failover, page 20-8
Client-Classes, page 20-9

How DHCP Works

DHCP makes dynamic address allocation possible by shifting workstation configuration to global address pools at the server level. DHCP is based on a client/server model. The client software runs on the workstation and the server software runs on the DHCP server.

Related Topics

Sample DHCP User, page 20-2
Typical DHCP Administration, page 20-2
Leases, page 20-3
Scopes and Policies, page 20-3
Sample DHCP User

After Beth’s workstation (bethpc) is configured with DHCP, these actions occur when she first starts up:

1. Her workstation automatically requests an IP address from a DHCP server on the network.
2. The DHCP server offers her a lease that is an IP address with the configuration data necessary to use the Internet. Nobody else uses the leased address, and it is valid only for her workstation.
3. Before the address lease expires, bethpc renewes it, thereby extending the expiration time. It continues to use the lease right up to its expiration or if it cannot reach the server.
4. If Beth relocates to another department and her workstation moves to a different subnet, her current address expires and becomes available for others. When Beth starts her workstation at its new location, it leases an address from an appropriate DHCP server on the subnet (see Figure 20-1 on page 20-2).

As long as the DHCP server has the correct configuration data, none of the workstations or servers using DHCP will ever be configured incorrectly. Therefore, there is less chance of incurring network problems from incorrectly configured workstations and servers that are difficult to trace.

Figure 20-1    Hosts Request an IP Address

The example shows the DHCP protocol with a set of DHCP servers that provide addresses on different subnets. To further simplify the administration of address pools, network routers are often configured as DHCP relay agents to forward client messages to a central DHCP server. This server is configured with address pools for a group of subnets.

Typical DHCP Administration

To use DHCP, you must have at least one DHCP server on the network. After you install the server:

- Define a scope of IP addresses that the DHCP server can offer to DHCP clients. You no longer need to keep track of which addresses are in use and which are available.
- Configure a secondary server to share the distribution or handle leases if the first DHCP server goes down. This is known as DHCP failover, and is described further in the “DHCPv6 Failover” section on page 28-3.
Leases

One of the most significant benefits of DHCP is that it can dynamically configure workstations with IP addresses and associate leases with the assigned addresses. DHCP uses a lease mechanism that offers an automated, reliable, and safe method for distributing and reusing addresses in networks, with little need for administrative intervention. As system administrator, you can tailor the lease policy to meet the specific needs of your network.

Leases are grouped together in an address pool, called a scope, which defines the set of IP addresses available for requesting hosts. A lease can be reserved (the host always receives the same IP address) or dynamic (the host receives the next available, unassigned lease in the scope). The DHCP server of the site is configured to lease addresses 192.168.1.100 through 192.168.1.199 (see Figure 20-2 on page 20-3).

If you plan not to have more network devices than configured addresses for the scope, you can define long lease times, such as one to two weeks, to reduce network traffic and DHCP server load.

Figure 20-2  DHCP Hosts Requesting Leases from a DHCP Server

Scopes and Policies

A scope contains a set of addresses for a subnet, along with the necessary configuration parameters. You must define at least one scope for each subnet for which you want dynamic addressing.

A policy includes lease times and other configuration parameters that a DHCP server communicates to clients. Use policies to configure DHCP options that the DHCP server supplies to a client upon request. Policies ensure that the DHCP server supplies all the correct options for scopes without having to do so separately for each scope (see Figure 20-3 on page 20-4).

The difference between scopes and policies is that scopes contain server information about addresses, such as which address is leasable and whether to ping clients before offering a lease. Policies contain client configuration data, such as the lease duration and address of the local DNS server.

Policies are especially useful if you have multiple scopes on a server. You can create policies that apply to all or selected scopes. The Cisco Prime IP Express policy hierarchy is a way to define policies from least to most specific. For example, you usually specify a router option for each policy, which means that you would need a policy for each scope. Scope-specific policies like this can be defined in a scope-embedded policy. More general policies, such as those referring to lease times, can be applied in
a system-wide policy (see the “Configuring DHCP Policies” section on page 22-1). You can also write extensions to handle policy assignments (see the “Using Extensions to Affect DHCP Server Behavior” section on page 24-10).

**Figure 20-3** Scopes and Policies

<table>
<thead>
<tr>
<th>Scopes</th>
<th>Policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.35-50</td>
<td>Lease time 3 days</td>
</tr>
<tr>
<td>192.168.97-110</td>
<td>DNS server 192.168.1.1</td>
</tr>
<tr>
<td>192.168.10-20</td>
<td>Least time 3 days</td>
</tr>
<tr>
<td>192.168.4-14</td>
<td>DNS server 192.168.4.1</td>
</tr>
</tbody>
</table>

**Cisco Prime IP Express DHCP Implementations**

The Cisco Prime IP Express DHCP server provides a reliable method for automatically assigning IP addresses to hosts on your network. You can define DHCP client configurations, and use the Cisco Prime IP Express database to manage assigning client IP addresses and other optional TCP/IP and system configuration parameters. The TCP/IP assignable parameters include:

- IP addresses for each network adapter card in a host.
- Subnet masks for the part of an IP address that is the physical (subnet) network identifier.
- Default gateway (router) that connects the subnet to other network segments.
- Additional configuration parameters you can assign to DHCP clients, such as a domain name.

Cisco Prime IP Express automatically creates the databases when you install the DHCP server software. You add data through the web UI or CLI as you define DHCP scopes and policies.

The Cisco Prime IP Express DHCP server also supports allocating addresses in virtual private networks (VPNs) and subnets to pool manager devices for on-demand address pools. These features are described in the following sections.

**Related Topics**

- DHCP and IPv6, page 20-4
- Virtual Private Networks, page 20-5
- Subnet Allocation and DHCP Address Blocks, page 20-5

**DHCP and IPv6**

For details on the Cisco Prime IP Express implementation of DHCP v6, see Chapter 27, “Managing DHCPv6 Addresses.”
Virtual Private Networks

Virtual private networks (VPNs) allow the possibility that two pools in separate networks can have the same address space, with these two pools having overlapping private network addresses. This can save address resources without having to use valuable public addresses. These VPN addresses, however, require a special designator to distinguish them from other overlapping IP addresses. Cisco Prime IP Express DHCP servers that are not on the same VPN as their clients can now allocate leases and addresses to these clients, and can distinguish the addresses from one VPN to another.

Through changes made to the Cisco Prime IP Express DHCP server and Cisco IOS DHCP Relay Agent, the DHCP server can service clients on multiple VPNs. A VPN distinguishes a set of DHCP server objects, making them independent of otherwise identical objects in other address spaces. You can define multiple VPNs containing the same addresses. You create a VPN based on the VPN identifier configured in the Cisco IOS Relay Agent.

Figure 20-4 shows a typical VPN-aware DHCP environment. The DHCP Relay Agent services two distinct VPNs, blue and red, with overlapping address spaces. The Relay Agent has the interface address 192.168.1.1 on VPN blue and is known to DHCP Server 1 as 172.27.180.232. The server, which services address requests from DHCP Client 1 in VPN blue, can be on a different network or network segment than the client, and can be in a failover configuration with DHCP Server 2 (see the “DHCPv6 Failover” section on page 28-3). The Relay Agent can identify the special, distinguished route of the client address request to the DHCP server, as coordinated between the Relay Agent and Cisco Prime IP Express administrators. The DHCP servers can now issue leases based on overlapping IP addresses to the clients on both VPNs.

Subnet Allocation and DHCP Address Blocks

Cisco Prime IP Express supports creating on-demand address pools as a network infrastructure for address provisioning and VPNs. Traditionally, the DHCP server is limited to interact with individual host devices. Through subnet allocation, the server can interact with VPN routers and other provisioning devices to provision entire IP subnets. This Cisco Prime IP Express feature enhances the on-demand address pool capability currently supported by the Cisco IOS Relay Agent.
Cisco Prime IP Express supports explicitly provisioned subnets. You must explicitly configure the DHCP server address space and subnet allocation policies before the server can allocate pools or leases. You can thereby configure a server as a pool manager to manage subnets and delegate them to client devices.

You manage DHCP subnet allocation using DHCP server address block objects in Cisco Prime IP Express. A DHCP address block is a range of contiguous IP addresses delegated to the DHCP server for assignment. The server expects to subdivide these addresses into pools so that it or other servers or devices can allocate them. DHCP address blocks are parents to subnets. These DHCP address blocks are distinct from the address blocks you can create using the Cisco Prime IP Express web UI, which are static. DHCP address blocks cannot include static address ranges or lease reservations.

Figure 20-5 shows a sample environment where a DHCP server allocates entire subnets to access concentrators or other provisioning devices, in addition to servicing individual clients. The traditional client/server relationship is shown on the left of the diagram, while the subnet allocation to access concentrators is shown on the right of the diagram. Dialup customers, for example, connect to the service provider network at two ISP gateways (routers), which connect to the management network segment where the DHCP server resides. The gateways provision addresses to their connected clients based on the subnet requested from the DHCP server.

Figure 20-5 Sample DHCP Subnet Allocation Configuration

DNS Update

Although DHCP frees you from the burden of distributing IP addresses, it still requires updating the DNS server with DHCP client names and addresses. DNS update automates the task of keeping the names and addresses current. With the Cisco Prime IP Express DNS update feature, the DHCP server can tell the corresponding DNS server when a name-to-address association occurs or changes. When a client gets a lease, Cisco Prime IP Express tells the DNS server to add the host data. When the lease expires or when the host gives it up, Cisco Prime IP Express tells the DNS server to remove the association.
In normal operation, you do not have to manually reconfigure DNS, no matter how frequently clients’ addresses change through DHCP. Cisco Prime IP Express uses the hostname that the client workstation provides. You also can have Cisco Prime IP Express synthesize names for clients who do not provide them, or use the client lookup feature to use a preconfigured hostname for the client.

**Related Topics**

- Effect on DNS of Obtaining Leases, page 20-7
- Effect on DNS of Releasing Leases, page 20-7
- Effect on DNS of Reacquiring Leases, page 20-8

### Effect on DNS of Obtaining Leases

For ExampleCo, the administrator creates a scope on the DHCP server and allocates 100 leases (192.168.1.100 through 192.168.1.199). Each workstation gets its owner name. The administrator also configures the DHCP server to use DNS update and associates it with the correspondingly configured DNS server. The administrator does not need to enter the names in the DNS server database.

Monday morning, Beth (user of bethpc) tries to log into a website without having an address. When her host starts up, it broadcasts an address request (see Figure 20-6 on page 20-7).

![Figure 20-6 DNS Update at ExampleCo Company](image)

The DHCP server then:

1. Gives bethpc the next available (unassigned) IP address (192.168.1.125).
2. Updates her DNS server with the hostname and address (bethpc 192.168.1.125).

Beth can now access the website. In addition, programs that need to translate the name of Beth’s machine to her IP address, or the other way around, can query the DNS server.

### Effect on DNS of Releasing Leases

Later that day, Beth learns that she needs to travel out of town. She turns off her host, which still has a leased address that is supposed to expire after three days. When the lease is released, the DHCP server:

1. Acknowledges that the IP address is now available for other users (see Figure 20-7 on page 20-8).
2. Updates the DNS server by removing the hostname and address. The DNS server no longer stores data about bethpc or its address.

**Figure 20-7 Relinquishing a Lease**

**Effect on DNS of Reacquiring Leases**

When Beth returns from her trip to start up her host again:
1. Her workstation broadcasts for an IP address.
2. The DHCP server checks if the host is on the correct network. If so, the server issues an address. If not, the server on the correct network issues the address.
3. The DHCP server updates the DNS server again with the host and address data.

**DHCP Failover**

Cisco Prime IP Express failover protocol is designed to allow a backup DHCP server to take over for a main server if the main server is taken offline for any reason. Starting in 8.2, this protocol is TCP based and supports both DHCPv4 and DHCPv6 (prior to 8.2, the protocol was UDP based and only supported DHCPv4). The existing DHCP clients can keep and renew their leases without the need to know which server is responding to their requests.

You can create and synchronize failover pairs at the local and regional clusters in Cisco Prime IP Express. For details, see Chapter 28, “Managing DHCP Failover”.

**Allocating Addresses Through Failover**

In order to keep the failover pair operating in spite of a network partition, in which both can communicate with clients but not with each other, you must allocate more addresses than the addresses needed to run a single server. Configure the main server to allocate a percentage of the currently available (unassigned) addresses in each scope or prefix delegation address pool to its partner. These addresses become unavailable to the main server. The partner uses them when it cannot talk to the main server and does not know if it is down.
The backup server needs enough addresses from each scope or prefix to satisfy the requests of all new DHCP clients that arrive during the period in which the backup does not know if the main server is down. The default backup percentage for a failover pair is 50%, this ensures that during the failover the other partner has equal number of addresses.

Even during PARTNER-DOWN state, the backup server waits for the lease expiration and the maximum client lead time (MCLT), a small additional time buffer, before reallocating any leases. When these times expire, the backup server offers:

- Leases from its private pool of addresses.
- Leases from the main server pool of addresses.
- Expired leases to new clients.

During the working hours, if the administrative staff can respond within two hours to a COMMUNICATIONS INTERRUPTED state to determine if the main server is working, the backup server needs enough addresses to support a reasonable upper bound on the number of new DHCP clients that might arrive during those two hours.

During off-hours, if the administrative staff can respond within 12 hours to the same situation, and considering that the arrival rate of previously unheard from DHCP clients is also less, the backup server then needs enough addresses to support a reasonable upper bound on the number of DHCP clients that might arrive during those 12 hours.

Consequently, the number of addresses over which the backup server requires sole control would be the greater of the numbers of addresses given out during peak and non-peak times, expressed as a percentage of the currently available (unassigned) addresses in each scope or prefix.

**Note**

Starting in 8.2, the default *use-safe-period* is enabled for the DHCP failover pair and the default safe period is 4 hours. This ensures that if the failover partner is in COMMUNICATIONS-INTERRUPTED state for 4 hours, it will enter PARTNER-DOWN state automatically after the safe period elapses.

---

**Client-Classes**

Assigning classes to clients is an important adjunct to DHCP addressing and addresses quality of service issues. You can use the Cisco Prime IP Express client and client-class facility to provide differentiated services to users that are connected to a common network. You can group your user community based on administrative criteria, and then ensure that each user receives the appropriate class of service.

Although you can use the Cisco Prime IP Express client-class facility to control any configuration parameter, the most common uses are for:

- **Lease periods**—How long a set of clients should keep their addresses.
- **IP address ranges**—From which lease pool to assign clients addresses.
- **DNS server addresses**—Where clients should direct their DNS queries.
- **DNS hostnames**—What name to assign clients.
- **Denial of service**—Whether unauthorized clients should be offered leases.

One way to use the client-class facility is to allow visitors access to some, but not all, of your network. For example, when Joe, a visitor to ExampleCo, tries to attach his laptop to the example.com network, Cisco Prime IP Express recognizes the laptop as being foreign. ExampleCo creates one class of clients
known as having access to the entire network, and creates another visitor class with access to a subnet only. If Joe needs more than the standard visitor access, he can register his laptop with the Cisco Prime IP Express system administrator, who adds him to a different class with the appropriate service.

The following sections describe how DHCP normally processes an address assignment, and then how it would handle it with the client-class facility in effect.

**Related Topics**

- **DHCP Processing Without Client-Classes**, page 20-10
- **DHCP Processing with Client-Classes**, page 20-11
- **Defining Scopes for Client-Classes**, page 20-11
- **Choosing Networks and Scopes**, page 20-12

**DHCP Processing Without Client-Classes**

To understand how you can apply client-class processing, it is helpful to know how the DHCP server handles client requests. The server can perform three tasks:

- Assign an IP address.
- Assign the appropriate DHCP options (configuration parameters).
- Optionally assign a fully qualified domain name (FQDN) and update the DNS server with that name.

The DHCP server:

1. Assigns an address to the client from a defined scope—To choose an address for the client, the DHCP server determines the client subnet, based on the request packet contents, and finds an appropriate scope for that subnet.

   If you have multiple scopes on one subnet or several network segments, which is known as multinetting, the DHCP server may choose among these scopes in a round-robin fashion, or you can change the priority of the scope choice by using the DHCP server address allocation priority feature (see the “Configuring Multiple Scopes Using Allocation Priority” section on page 21-12). After the server chooses a scope, it chooses an available (unassigned) address from that scope:

   a. It assigns DHCP option values from a defined policy. Cisco Prime IP Express uses policies to group options. There are two types of policies: scope-specific and system default. For each DHCP option the client requests, the DHCP server searches for its value in a defined sequence.

   b. If the scope-specific policy contains the option, the server returns its value to the client and stops searching.

   c. If not found, the server looks in the system default policy, returns its value, and stops searching.

   d. If neither policy contains the option, the server returns no value to the client and logs an error.

   e. The server repeats this process for each requested option.

2. With DNS update in effect, the server assigns an FQDN to the client. If you enabled DNS update, Cisco Prime IP Express enters the client name and address in the DNS host table. See the “DNS Update” section on page 20-6. The client name can be:

   a. Its name as specified in the client lease request (the default value).

   b. Its MAC address (hardware address; for example, 00:d0:ba:d3:bd:3b).

   c. A unique name using the default prefix `dhcp` or a specified prefix.
DHCP Processing with Client-Classes

When you enable the client-class facility for your DHCP server, the request processing performs the same three tasks of assigning IP addresses, options, and domain names as described in the “DHCP Processing Without Client-Classes” section on page 20-10, but with added capability. The DHCP server:

1. **Considers the client properties and client-class inclusion before assigning an address**—As in regular DHCP processing, the DHCP server determines the client subnet. The server then checks if there is a client-class defined or a MAC address for this client in its database. If there is:
   a. A client-class defined by a client-class lookup ID expression, the client is made a member of this client-class.
   b. No MAC address, it uses the default client. For example, the default client could have its client-class name set to Guest, and that client-class could limit (using options and address selection) what network operations such clients are permitted.
   c. No MAC address and no default client, the server handles the client through regular DHCP processing.
   d. No client-specifier, but a MAC address, the MAC address is converted into a client-specifier. An unknown client is mapped to the default client, if the default client is defined.

The scopes must have addresses on client-accessible subnets. That is, they must have a selection tag that associates them with a client-class. To assign the same clients to different address pools, you must use separate scopes.

For example, a scope would either have a selection tag of Employee or Guest, but not both. In this case, there are two scopes for each subnet; one with the selection tag Employee, and the other with Guest. Each scope has a different associated policy and address range that provides the appropriate access rights for the user group.

2. **Checks for client-class DHCP options**—In regular DHCP processing, the server checks the scope-specific and system default DHCP options. With client-class, it also first checks the client-specific and client-class-specific options.

3. **Provides additional FQDN assignment options**—Beyond the usual name assignment process of using the hostname the client requests, the server can:
   - Provide an explicit hostname that overrides it.
   - Drop the client-requested hostname and not replace it.
   - Synthesize a hostname from the client MAC address.

Defining Scopes for Client-Classes

The motivating factor for using client-classes is often to offer an address from one or another address pool to a client. Another motivating factor might be to provide clients with different option values or lease times. Offering clients addresses from separate pools requires defining more than one scope.

To get more than one scope on a subnet, they must come from the same network segment. Networks are not configured directly in Cisco Prime IP Express, but are inferred from scope configurations. Scopes become related (end up in the same network):

- **Implicitly**—Two scopes have the same network number and subnet mask. These scopes naturally end up on the same network without explicit configuration.
Client-Classes

• **Explicitly**—One scope is marked as a secondary to another. This is required when the scope marked as a secondary has a network and subnet mask unrelated to the primary. An example is putting a set of 10.0.0.0 network addresses on a normal, routable network segment.

When the Cisco Prime IP Express DHCP server reads the scope configuration from its database, it places every scope in a network, and logs this information. Scopes with the same network number and subnet mask end up on the same network, while a secondary scope ends up on the primary scope network.

### Choosing Networks and Scopes

When a DHCP packet arrives, the server determines the address from which it came by:

• Gateway address (`giaddr`), if there was one, for packets sent through a BOOTP relay.

• Interface address of the interface on which the broadcast packet arrived, if the DHCP client is on a network segment to which the DHCP server is also directly connected.

In all cases, the DHCP server determines a network from the gateway or interface address. Then, if the network has multiple scopes, the server determines from which scope to allocate an address to the DHCP client. It always looks for a scope that can allocate addresses to this type of client. For example, a DHCP client needs a scope that supports DHCP, and a BOOTP client needs one that supports BOOTP. If the client is a DHCP client and there are multiple scopes that support DHCP, each with available (unassigned) addresses, the DHCP server allocates an IP address from any of those scopes, in a round-robin manner, or by allocation priority.

Selection tags and client-classes let you configure the DHCP server to allocate IP addresses from:

• One or more scopes on a network to one class of clients.

• A different set of scopes to a different class of clients.

In the latter case, the gateway or interface address determines the network. The client-class capability, through the mechanism of the selection tags, determines the scope on the network to use.
Configuring Scopes and Networks

The Dynamic Host Configuration Protocol (DHCP) is an industry-standard protocol for automatically assigning IP configuration to workstations. DHCP uses a client/server model for address allocation. As administrator, you can configure one or more DHCP servers to provide IP address assignment and other TCP/IP-oriented configuration information to your workstations. DHCP frees you from having to manually assign an IP address to each client. The DHCP protocol is described in RFC 2131. For an introduction to the protocol, see Chapter 20, “Introduction to Dynamic Host Configuration.”

This chapter describes how to set up DHCP policies and options. Before clients can use DHCP for address assignment, you must add at least one scope (dynamic address pool) to the server.

Related Topics

Configuring DHCP Servers, page 21-1
Defining and Configuring Scopes, page 21-2
Managing DHCP Networks, page 21-23

Configuring DHCP Servers

When configuring a DHCP server, you must configure the server properties, policies, and associated DHCP options. Cisco Prime IP Express needs:

- The DHCP server IP address.
- One or more scopes (see the “Defining and Configuring Scopes” section on page 21-2).

Related Topics

General Configuration Guidelines, page 21-1
Configuring DHCP Server Interfaces, page 21-2

General Configuration Guidelines

Here are some guidelines to consider before configuring a DHCP server:

- **Separate the DHCP server from secondary DNS servers used for DNS updating**—To ensure that the DHCP server is not adversely affected during large zone transfers, it should run on a different cluster than your secondary DNS servers.
Defining and Configuring Scopes

- **Configure a separate DHCP server to run in remote segments of the wide area network (WAN)**—Ensure that the DHCP client can consistently send a packet to the server in under a second. The DHCP protocol dictates that the client receive a response to a DHCPDISCOVER or DHCPREQUEST packet within four seconds of transmission. Many clients, notably early releases of the Microsoft DHCP stack, actually implement a two-second timeout.

- **Lease times**—See the “Guidelines for Lease Times” section on page 23-3.

### Configuring DHCP Server Interfaces

To configure the DHCP server, accept the Cisco Prime IP Express defaults or supply the data explicitly:

- **Network interface**—Ethernet card IP address, which must be static and not assigned by DHCP.

- **Subnet mask**—Identifies the interface network membership. The subnet mask is usually based on the network class of the interface address, in most cases 255.255.255.0.

By default, the DHCP server uses the operating system support to automatically enumerate the active interfaces on the machine and listens on all of them. You can also manually configure the server interface. You should statically configure all the IP addresses assigned to NIC cards on the machine where the DHCP server resides. The machine should not be a BOOTP or DHCP client.

#### Local Advanced Web UI

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>From the <strong>Operate</strong> menu, choose <strong>Manage Servers</strong> under the <strong>Servers</strong> submenu to open the Manage Servers page.</td>
</tr>
<tr>
<td>2</td>
<td>Select the Local DHCP server on the Manager Servers pane.</td>
</tr>
<tr>
<td>3</td>
<td>Click the Network Interfaces tab to view the available network interfaces that you can configure for the server. By default, the server uses all of them.</td>
</tr>
<tr>
<td>4</td>
<td>To configure an interface, click the <strong>Edit</strong> icon in the Configure column for the interface. This adds the interface to the Configured Interfaces table, where you can edit or delete it.</td>
</tr>
<tr>
<td>5</td>
<td>Clicking the name of the configured interface opens the Edit DHCP Server Network Interface page, where you can change the address and ports (in Expert mode) of the interface.</td>
</tr>
<tr>
<td>6</td>
<td>Click <strong>Save</strong> when you are done editing.</td>
</tr>
<tr>
<td>7</td>
<td>Click <strong>Revert</strong> to return to the Manage Servers page.</td>
</tr>
</tbody>
</table>

#### CLI Commands

Use **dhcp-interface** to manually control which network interface cards’ IP addresses the DHCP server will listen on for DHCP clients. By default, the DHCP server automatically uses all your server network interfaces, so use this command to be more specific about which ones to use.

### Defining and Configuring Scopes

This section describes how to define and configure scopes for the DHCP server. A scope consists of one or more ranges of dynamic addresses in a subnet that a DHCP server manages. You must define one or more scopes before the DHCP server can provide leases to clients. (For more on listing leases and defining lease reservations for a scope, see Chapter 23, “Managing Leases.”)
Creating and Applying Scope Templates

Scope templates apply certain common attributes to multiple scopes. These common attributes include a scope name based on an expression, policies, address ranges, and an embedded policy option based on an expression (see the “Using Expressions in Scope Templates” section on page 21-4).

Local Advanced and Regional Web UI

Scope templates you add or pull from the local clusters are visible on the List DHCP Scope Templates page. To get there, from the Design menu choose Scope Templates from DHCPv4 submenu. This functionality is available only to administrators assigned the dhcp-management subrole of the regional central-cfg-admin or local ccm-admin role.

To explicitly create a scope template, click Add Scope Templates on the Scope Templates pane. This opens the Add DHCP Scope Template dialog box, which includes the template name. You can also choose an existing policy for the scope template. The other fields require expression values (see the “Create a Scope Template” section on page 5-37 that describes these fields).

Related Topics

Using Expressions in Scope Templates, page 21-4
Additional Scope Template Attributes, page 21-8
Editing Scope Templates, page 21-8
Applying Scope Templates to Scopes, page 21-8
Cloning a Scope Template, page 21-9

CLI Commands

Create a scope template using scope-template name create. For example:

```
nrcmd> scope-template example-scope-template create
```

You can also associate a policy with the scope template:

```
nrcmd> scope-template example-scope-template set policy=examplepolicy
```
Using Expressions in Scope Templates

You can specify expressions in a scope template to dynamically create scope names, IP address ranges, and embedded options when creating a scope. Expressions can include context variables and operations.

Note

Expressions are not the same as DHCP extensions. Expressions are commonly used to create client identities or look up clients. Extensions (see Chapter 30, “Using Extension Points”) are used to modify request or response packets.

If you apply the template to a scope that already has ranges defined, the address range expression of the scope template is not evaluated for that scope.

Table 21-1 lists the scope expression functions. Note that these functions are not case-sensitive.

Table 21-1 Expression Functions

<table>
<thead>
<tr>
<th>Expression Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Context Variables</strong></td>
<td></td>
</tr>
<tr>
<td>bcast-addr</td>
<td>Derived from the broadcast address in the subnet, such as 192.168.50.255. Use in any expression field.</td>
</tr>
<tr>
<td>first-addr</td>
<td>Derived from the first address in the subnet, such as the first address in 192.168.50.64/26 is 192.168.50.65. Use in any expression field.</td>
</tr>
<tr>
<td>last-addr</td>
<td>Derived from the last address in the subnet, such as the last address in 192.168.50.64/26 is 192.168.50.127. Use in any expression field.</td>
</tr>
<tr>
<td>mask-addr</td>
<td>Derived from the network mask address in the subnet, such as 255.255.255.0. Use in any expression field.</td>
</tr>
<tr>
<td>mask-count</td>
<td>Derived from the number of bits in the network address of the subnet, such as 24. Use in the Scope Name Expression or Embedded Policy Option Expression field.</td>
</tr>
<tr>
<td>naddrs</td>
<td>Derived from the number of IP addresses in the subnet, such as 255. Use in the Scope Name Expression field.</td>
</tr>
<tr>
<td>nhosts</td>
<td>Derived number of usable hosts in the subnet, such as 254. Use in any expression field.</td>
</tr>
<tr>
<td>subnet</td>
<td>Derived from the IP address and mask of the subnet, such as 192.168.50.0/24. Use in the Scope Name Expression or Embedded Policy Option Expression field.</td>
</tr>
<tr>
<td>subnet-addr</td>
<td>Derived from the subnet address, such as 192.168.50.0. Use in any expression field.</td>
</tr>
<tr>
<td><strong>template.attribute</strong></td>
<td>Attribute of the scope template, such as template.ping-timeout. Use in the Embedded Policy Option Expression field.</td>
</tr>
<tr>
<td><strong>Arithmetic Operations</strong> (unsigned integer arguments only)</td>
<td></td>
</tr>
<tr>
<td>(+ arg1 arg2)</td>
<td>Adds the two argument values, such as (+ 2 3).</td>
</tr>
<tr>
<td>(– arg1 arg2)</td>
<td>Subtracts the second argument value from the first one, such as with ping-timeout defined as 100, (– template.ping-timeout 10) yields 90.</td>
</tr>
<tr>
<td>(* arg1 arg2)</td>
<td>Multiplies the values of two arguments.</td>
</tr>
</tbody>
</table>
Table 21-1  Expression Functions (continued)

<table>
<thead>
<tr>
<th>Expression Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(( l \ arg1 \ arg2))</td>
<td>Divides the value of the first argument by that of the second one (which</td>
</tr>
<tr>
<td></td>
<td>cannot be zero).</td>
</tr>
<tr>
<td>(concat ( arg1) \ ... ( argn))</td>
<td>Concatenates the arguments into a string, to be used in the Scope Name</td>
</tr>
<tr>
<td></td>
<td>Expression field. Examples: With subnet=192.168.50.0/24 and</td>
</tr>
<tr>
<td></td>
<td>template.ping-timeout=100:</td>
</tr>
<tr>
<td></td>
<td>((concat &quot;ISP-&quot; subnet) --&gt; ISP-192.168.50.0/24)</td>
</tr>
<tr>
<td></td>
<td>((concat subnet &quot;;&quot; (+ template.ping-timeout 10)) --&gt; 192.168.50.0/24-110</td>
</tr>
<tr>
<td></td>
<td>((concat &quot;ISP-&quot; subnet &quot;;&quot; (+ template.ping-timeout 10)) --&gt;</td>
</tr>
<tr>
<td></td>
<td>ISP-192.168.50.0/24-110</td>
</tr>
<tr>
<td></td>
<td>See also the “Scope Name Expression Example” section on page 21-7.</td>
</tr>
<tr>
<td>(create-option \opt \val)</td>
<td>Use create-option in the Embedded Policy Option Expression field to create</td>
</tr>
<tr>
<td></td>
<td>new DHCP options for the scope. The first argument can be an integer or</td>
</tr>
<tr>
<td></td>
<td>string to represent the option number or name. The second argument can be</td>
</tr>
<tr>
<td></td>
<td>a string or blob to give the option a value.</td>
</tr>
<tr>
<td></td>
<td>You can also specify custom defined and unknown options. For undefined</td>
</tr>
<tr>
<td></td>
<td>options, the option number must be specified and the data is used as is as</td>
</tr>
<tr>
<td></td>
<td>(as blob data). If the data is a string, the string is used as is and if</td>
</tr>
<tr>
<td></td>
<td>the data is a number or address, it is used as is.</td>
</tr>
<tr>
<td></td>
<td>Examples:</td>
</tr>
<tr>
<td></td>
<td>((list (create-option &quot;domain-name&quot; &quot;example.com&quot;))</td>
</tr>
<tr>
<td></td>
<td>( (create-option 3 &quot;10.10.10.1&quot;))</td>
</tr>
<tr>
<td></td>
<td>((create-option &quot;routers&quot; &quot;10.10.10.1,10.10.10.2,10.10.10.3&quot;))</td>
</tr>
<tr>
<td></td>
<td>((create-option &quot;routers&quot; (create-ipaddr subnet 10)))</td>
</tr>
<tr>
<td></td>
<td>See also the “Embedded Policy Option Expression Example” section on page 21-7.</td>
</tr>
<tr>
<td>(create-vendor-option \set-name \opt \val)</td>
<td>Use the create-vendor-option in the Embedded Policy Option Expression</td>
</tr>
<tr>
<td></td>
<td>field to creates a DHCP vendor option. The \set-name\ specifies the option</td>
</tr>
<tr>
<td></td>
<td>definition set for the vendor option. The \opt\ can be the literal string</td>
</tr>
<tr>
<td></td>
<td>or integer identifying the vendor option in the set. The \val\ is</td>
</tr>
<tr>
<td></td>
<td>representation of the option value.</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>((list (create-option &quot;routers&quot; (create-ipaddr subnet 1)))</td>
</tr>
<tr>
<td></td>
<td>((create-vendor-option &quot;dhcp-cablelabs-config&quot; 125)</td>
</tr>
<tr>
<td></td>
<td>((concat &quot;;(tftp-servers 2 ( create-ipaddr subnet 2 ))&quot;)))</td>
</tr>
</tbody>
</table>
## Defining and Configuring Scopes

### Chapter 21      Configuring Scopes and Networks

#### Local Advanced and Regional Web UI

There are three fields on the Add DHCP Scope Template page for which you must specify an expression:

- **Scope Name Expression**—Must return a string
- **Range Expression**—Must return IP addresses
- **Embedded Policy Option Expression**—No requirements

#### CLI Commands

Use the following `scope-template` command attributes:

- `scope-name`
- `ranges-exp`
- `options-exp`

### Table 21-1   Expression Functions (continued)

<table>
<thead>
<tr>
<th>Expression Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create Range Operation</strong></td>
<td></td>
</tr>
</tbody>
</table>
| `(create-range start end)` | Use this operation in the Range Expression field. It creates an IP address range for the scope. The first argument is the start of the address range and can be an integer or IP address string. The second argument is the end of the range and can be an integer or IP address string. Do not include the local host or broadcast address determined by the mask (such as 0 and 255 for /24 subnets) in the range. Validation ensures that the range must be in the subnet defined by the template and that the first argument value must be lower than the second. An integer value determines the position of the address in the given subnet. Examples (with subnet=192.168.50.0/26):

  - `(create-range "192.168.50.65" "192.168.50.74")`  
    --> 192.168.50.65 - 192.168.50.74
  - `(create-range 1 10)`  
    --> 192.168.50.65 - 192.168.50.74
  - See also the “Range Expression Example” section on page 21-7. |
| **Create IP Operation** | |
| `(create-ipaddr net host)` | Use this operation in the Embedded Policy Option Expression or Range Expression fields. It creates an IP address string. The net argument is a string or variable. The host argument is an integer. Example:

  - `(create-ipaddr subnet 4)` |
| **List Operation** | Arguments must all be create-option or create-range operations. Nesting is not supported. Examples:

  - `(list (create-option "routers" "10.10.10.1")
    (create-option "domain-name" "example.com"))
  - `(list (create-range 1 5) (create-range 10 20))` |
Scope Name Expression Example

You might want to set an expression so that the template constructs scope names starting with "ISP-" and followed by the subnet of the scope and a derivative of its ping timeout value. You would use the following expression in the Scope Name Expression field:

\[(\text{concat } "ISP-" \text{ subnet } "-" (+ \text{template.ping-timeout } 10))\]

The elements of the example expression are:

- \(\text{concat} \ldots\)—Concatenation operation, which concatenates all the following values into one value.
- "ISP-"—String with which to start the scope name.
- \text{subnet}—Keyword variable that indicates to use the existing subnet defined for the scope.
- "-"—Indicates to include this hyphen to construct the value.
- \((+ \text{template.ping-timeout } 10)\)—Indicates to add the ping-timeout property value for the scope to the number 10.

If the scope subnet happens to be 192.168.50.0/24 and its ping-timeout value 100, the resulting constructed scope name would be:

ISP-192.168.50.0/24-110

Range Expression Example

You might want to set an expression so that the template constructs only certain address ranges for scopes. You can either be explicit about the actual starting and ending addresses, or you can make them relative to the subnet. Here are two ways of requesting relative ranges in the Range Expression field:

\[(\text{create-range first-addr last-addr})\]
\[(\text{create-range 1 10})\]

The first \text{create-range} operation creates the address range based on the first through last usable address in the subnet. For the 192.168.50.0/24 subnet, for example, the address range would be 192.168.50.1 through 192.168.50.254. Because the second operation specifies integers instead of full IP addresses, it makes the range relative to the subnet based on its mask. If the template discovers the subnet to be 192.168.50.0/26, it takes the first through tenth address in this subnet, which would be 192.168.50.65 through 192.168.50.74.

To set the range expressions in the CLI, you should place the expression into a file and use a command such as:

\[\text{nrcmd> scope-template example-template set ranges-expr=@file}\]

where file is the name of the file that you created with the expressions.

Embedded Policy Option Expression Example

An embedded policy is important because the DHCP server looks at it before it looks at the assigned, named policy of the scope. This is usually where you would set the DHCP options on a scope. You might want to set an expression so that the template constructs DHCP options for the scope embedded policy. Here are some examples:

\[(\text{create-option } "domain-name" "example.com")\]
\[(\text{create-option } 3 "10.10.10.1")\]
\[(\text{create-option } "routers" (\text{create-ipaddr subnet 10}))\]
The first `create-option` operation associates the value `example.com` with the `domain-name` option for the scope. The second operation associates the address `10.10.10.1` with the `routers` option (number 3). The third operation creates an IP address for the `routers` option based on the tenth address in a subnet.

To set the policy options expressions in the CLI, you should place the expression into a file and use a command such as:

```
nrcmd> scope-template example-template set options-expr=@file
```

where `file` is the name of the file that you created with the expressions.

---

**Note**

Trying to specify the expression directly on the CLI command line will likely fail because of embedded spaces and special characters such as the quotes. Use the `@file` syntax as it avoids any potential issues with the CLI command parser. But the Web UI does not support the `@file` syntax. You can enter complex expressions directly in the Web UI.

### Additional Scope Template Attributes

The optional additional attributes appear in functional categories. For a description of each attribute, click the attribute name to open a help window. For example, you might want to enable dynamic DNS updates for the scope, or set the main and backup DHCP failover servers.

After you complete these fields, click **Save**.

### Editing Scope Templates

To edit a scope template, click its name on the List/Add DHCP Scope Templates page. The Edit DHCP Scope Template page is essentially the same as the Add DHCP Scope Template page (see the “Creating and Applying Scope Templates” section on page 21-3) except for an additional attribute unset function. Make your changes, then click **Save**.

In the CLI, edit a scope template attribute by using `scope-template name set attribute`. For example:

```
nrcmd> scope-template example-scope-template set policy=default
```

### Applying Scope Templates to Scopes

You can apply a scope template to a scope in a few ways.

---

**Caution**

Be careful applying a scope template to an existing scope. The template overwrites all the scope attributes with its own, which can have a detrimental effect if the scope is active.

---

### Local Advanced Web UI

- **While creating a named scope**—On the List/Add DHCP Scopes page, click the **Add Scopes** icon in the Scopes pane, include the name of the scope, add its subnet and choose the policy from the drop-down list. Clicking **Add DHCP Scope** creates a scope with the name specified and with the attributes set for the scope template, including the expressions you might have set (see the “Using Expressions in Scope Templates” section on page 21-4). (Note that Basic mode lets you specify a Class of Service, but not apply a scope template.)
• **While creating a scope, derive its name from the template**—If you set a Scope Name Expression for the scope template (see the “Using Expressions in Scope Templates” section on page 21-4) on the List/Add DHCP Scope Template page, when you add a scope on the List/Add DHCP Scopes page, omit the name of the scope, but add its subnet and mask, then choose the scope template from the Template drop-down list. Clicking Add DHCP Scope creates a scope with a name synthesized from the scope name expression. If you do not set a scope name expression in the template and apply it to the scope without specifying a name for the scope, you get an error. (Note that Basic mode does not provide this functionality.)

• **After creating a named scope**—On the Edit DHCP Scopes page, scroll to the bottom to find the Apply Template button. Choose a preconfigured template from the drop-down list, then click the button. Then click Save. (Be aware of the previous warning that the template attributes overwrite the existing ones of the scope.)

**CLI Commands**

To apply a template to the scope while creating the scope, use `scope name create address mask template=template-name`. For example:

```
nrcmd> scope example-scope create 192.168.50.0 24 template=example-scope-template
```

To derive the scope name from the template during scope creation, use `scope-template name apply-to {all | scope1,scope2,...}`). For example:

```
nrcmd> scope-template example-scope-template apply-to examplescope-1,examplescope-2
```

**Cloning a Scope Template**

In the CLI, you can also clone a scope template from an existing one by using `scope-template clone-name create clone=template`, and then make adjustments to the clone. For example:

```
nrcmd> scope-template cloned-template create clone=example-scope-template-1 ping-timeout=200
```

**Creating Scopes**

Creating scopes is a local cluster function. Each scope needs to have the following:

- **Name**
- **Policy** that defines the lease times, grace period, and options
- **Network address and subnet mask**
- **Range or ranges of addresses**

You can configure scopes at the local cluster only. The web UI pages are different for local basic and advanced modes.

**Local Basic Web UI**

**Step 1**
From the **Design** menu, choose **Scopes** from the **DHCPv4** submenu to open the Manage Scopes page.

**Step 2**
Choose a VPN for the scope from the **username** drop-down list on the top right of the window, if necessary.
Step 3 Click the Add Scopes icon in the Scopes pane, enter a scope name, enter the subnet IP address and choose a mask value from the drop-down list.

Step 4 If desired, choose a preconfigured class of service (client-class) for the scope from the drop-down list.

Step 5 Click Add DHCP Scope.

Step 6 Reload the DHCP server.

---

**Note**

When a scope is created in Basic mode, the range and the router address will be added automatically. If you want to change them, you have to change the mode to Advanced since it cannot be configured on the Basic mode.

---

### Local Advanced Web UI

Step 1 From the Design menu, choose Scopes from the DHCPv4 submenu to open the List/Add DHCP Scopes page.

Step 2 Choose a VPN for the scope from the username drop-down list on the top right of the window, if necessary.

Step 3 Click the Add Scopes icon in the Scopes pane, enter a scope name, or leave it blank to use the one defined in the scope name expression of a scope template, if any (see the “Using Expressions in Scope Templates” section on page 21-4). In the latter case, choose the scope template. You must always enter a subnet and mask for the scope.

Step 4 Choose a policy for the scope from the drop-down list. The policy defaults to the default policy.

Step 5 Click Add DHCP Scope.

Step 6 Add ranges for addresses in the scope. The ranges can be any subset of the defined scope, but cannot overlap. If you enter just the host number, the range is relative to the netmask. Do not enter ranges that include the local host or broadcast addresses (usually 0 and 255). Add the range, then click Add Range.

Step 7 Reload the DHCP server.

---

**Tip**

To view any leases and reservations associated with the scope, see Chapter 23, “Managing Leases.” To search for leases, see the “Searching Server-Wide for Leases” section on page 23-8.

---

### Related Topics

- Getting Scope Counts on the Server, page 21-10
- Configuring Multiple Scopes, page 21-11
- Editing Scopes, page 21-17
- Staged and Synchronous Mode, page 21-18

### Getting Scope Counts on the Server

You can view the created scopes associated with the DHCP server, hence obtain a count, in the web UI.
### CLI Commands

Using the CLI, you can get an exact count of the total scopes for the DHCP server by using `dhcp getScopeCount [name | vpn name | all]`. You can specify a VPN or all VPNs. Omitting the `vpn name` returns a count for the current VPN. Specifying a failover pair name returns the total scopes and networks for the failover pair. Because a failover pair definition includes explicit VPN settings in its matchlist, these counts are not limited to the current VPN only.

To create a scope, use `scope name create`. Each scope must identify its network address and mask. When you create the scope, Cisco Prime IP Express places it in its current virtual private network (VPN), as defined by `session set current-vpn`. You cannot change the VPN once you set it at the time of creation of the scope.

To set a policy for the scope, use `scope name set policy`.

To add a range of IP addresses to the scope, use `scope name addRange`.

### Configuring Multiple Scopes

You can configure multiple scopes (with disjointed address ranges) with the same network number and subnet mask. By default, the DHCP server pools the available leases from all scopes on the same subnet and offers them, in a round-robin fashion, to any client that requests a lease. However, you can also bypass this round-robin allocation by setting an allocation priority for each scope (see the “Configuring Multiple Scopes Using Allocation Priority” section on page 21-12).

Configuring the addresses of a a single subnet into multiple scopes helps to organize the addresses in a more natural way for administration. Even though you can configure a virtually unlimited number of leases per scope, if you have a scope with several thousand leases, it can take a while to sort them. This can be a motivation to divide the leases among multiple scopes.

You can divide the leases among the scopes according to the types of leases. Because each scope can have a separate reservations list, you can put the dynamic leases in one scope that has a policy with one set of options and lease times, and all the reservations in another scope with different options and times.

Note that in cases where some of the multiple scopes are not connected locally, you should configure the router (having BOOTP relay support) with the appropriate helper address.

### Related Topics

- Configuring Multiple Scopes for Round-Robin Address Allocation, page 21-11
- Configuring Multiple Scopes Using Allocation Priority, page 21-12

### Configuring Multiple Scopes for Round-Robin Address Allocation

By default, the DHCP server searches through the multiple scopes in a round-robin fashion. Because of this, you would want to segment the scopes by the kind of DHCP client requests made. When multiple scopes are available on a subnet through the use of secondary scopes, the DHCP server searches through all of them for one that satisfies an incoming DHCP client request. For example, if a subnet has three scopes, only one of which supports dynamic BOOTP, a BOOTP request for which there is no reservation is automatically served by the one supporting dynamic BOOTP.

You can also configure a scope to disallow DHCP requests (the default is to allow them). By using these capabilities together, you can easily configure the addresses on a subnet so that all the DHCP requests are satisfied from one scope (and address range), all reserved BOOTP requests come from a second one, and all dynamic BOOTP requests come from a third. In this way, you can support dynamic BOOTP while minimizing the impact on the address pools that support DHCP clients.
Configuring Multiple Scopes Using Allocation Priority

Allocation priority can be set among the scopes instead of the default round-robin behavior described in the previous section. In this way, you can have more control over the allocation process. You can also configure the DHCP server to allocate addresses contiguously from within a subnet and control the blocks of addresses allocated to the backup server when using DHCP server failover (see Chapter 28, “Managing DHCP Failover”).

A typical installation would set the allocation priority of every scope by using the allocation-priority attribute on the scope. Some installations might also want to enable the allocate-first-available attribute on their scopes, although many would not. There is a small performance loss when using allocate-first-available, so you should only use it when absolutely required.

You can control:

- A hierarchy among scopes of which should allocate addresses first.
- Whether to have a scope allocate the first available address rather than the default behavior of the least recently accessed one.
- Allocating contiguous and targeted addresses in a failover configuration for a scope.
- Priority address allocation server-wide.
- In cases where the scopes have equal allocation priorities set, whether the server should allocate addresses from those with the most or the least number of available addresses.

When there is more than one scope in a network, then the DHCP must decide which scope to allocate an IP address from when it processes a DHCPDISCOVER request from a DHCP client that is not already associated with an existing address. The algorithm that the DHCP server uses to perform this allocation is described in the following section.

Allocation Priority Algorithm

The DHCP server examines the scopes in a network one at a time to determine if they are acceptable. When it finds an acceptable scope, it tries to allocate an IP address from it to fulfill the DHCPDISCOVER request. The allocation-priority scope attribute is used to direct the DHCP server to examine the scopes in a network in a particular order, because in the absence of any allocation priority, the DHCP server examines the scopes in a round-robin order.

Figure 21-1 shows an example of a network with nine scopes (which is unusual, but serves to illustrate several possibilities of using allocation priority).
Six of these scopes were configured with an allocation priority, and three of them were not. The server examines the six that were configured with an allocation priority first, in lowest to highest priority order. As the server finds an acceptable scope, it tries to allocate an IP address from it. If the server succeeds, it then finishes processing the DHCPDISCOVER request using this address. If it cannot allocate an address from that scope, it continues examining scopes looking for another acceptable one, and tries to allocate an address from it.

This process is straightforward if no scopes have the same allocation priority configured, but in the case where (as in the example in) more than one scope has the same nonzero allocation priority, then the server has to have a way to choose between the scopes of equal priority. The default behavior is to examine the scopes with equal priority starting with the one with the fewest available addresses. This uses up all of the addresses in one scope before using any others from another scope. This is the situation shown in Figure 21-1. If you enable the equal-priority-most-available DHCP server attribute, then the situation is reversed and the scope with the most available addresses is examined first when two scopes have equal priority. This spreads out the utilization of the scopes, and more or less evenly distributes the use of addresses across all of the scopes with equal allocation priority set.

You can use this equal-priority-most-available approach because of another feature in the processing of equal priority scopes. In the situation where there are two scopes of equal priority, if the DHCPDISCOVER request, for which the server is trying to allocate an address, also has a limitation-id (that is, it is using the option 82 limitation capability; see the “Subscriber Limitation Using Option 82” section on page 25-14), then the DHCP server tries to allocate its IP address from the same scope as that used by some existing client with the same limitation-id (if any). Thus, all clients with the same limitation-id tend to get their addresses allocated from the same scope, regardless of the number of available addresses in the scopes of equal priority or the setting of the equal-priority-most-available server attribute.
To bring this back to the equal-priority-most-available situation, you might configure equal-priority-most-available (and have several equal priority scopes), and then the first DHCP client with a particular limitation-id would get an address from the scope with the most available addresses (since there are no other clients with that same limitation-id). Then all of the subsequent clients with the same limitation-id would go into that same scope. The result of this configuration is that the first clients are spread out evenly among the acceptable, equal priority scopes, and the subsequent clients would cluster with the existing ones with the same limitation-id.

If there are scopes with and without allocation priority configured in the same network, all of the scopes with a nonzero allocation priority are examined for acceptability first. Then, if none of the scopes were found to be acceptable and also had an available IP address, the remaining scopes without any allocation priority are processed in a round-robin manner. This round-robin examination is started at the next scope beyond the one last examined in this network, except when there is an existing DHCP client with the same limitation-id as the current one sending the DHCPDISCOVER. In this case, the round-robin scan starts with the scope from which the existing client IP address was drawn. This causes subsequent clients with the same limitation-id to draw their addresses from the same scope as the first client with that limitation-id, if that scope is acceptable and has available IP addresses to allocate.

**Address Allocation Attributes**

The attributes that correspond to address allocation are described in Table 21-2.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocation-priority</td>
<td>Scope (set or unset)</td>
<td>If defined, assigns an ordering to scopes such that address allocation takes place from acceptable scopes with a higher priority until the addresses in all those scopes are exhausted. An allocation priority of 0 (the preset value) means that the scope has no allocation priority. A priority of 1 is the highest priority, with each increasing number having a lower priority. You can mix scopes with an allocation priority along with those without one. In this case, the scopes with a priority are examined for acceptability before those without a priority. If set, this attribute overrides the DHCP server priority-address-allocation attribute setting. However, if allocation-priority is unset and priority-address-allocation is enabled, then the allocation priority for the scope is its subnet address. With allocation-priority unset and priority-address-allocation disabled, the scope is examined in the default round-robin fashion.</td>
</tr>
<tr>
<td>allocate-first-available</td>
<td>Scope (enable or disable)</td>
<td>If enabled, forces all allocations for new addresses from this scope to be from the first available address. If disabled (the preset value), uses the least recently accessed address. If set, this attribute overrides the DHCP server priority-address-allocation attribute setting. However, if unset and priority-address-allocation is enabled, then the server still allocates the first available address. With allocate-first-available unset and priority-address-allocation disabled, the scope is examined in the default round-robin fashion.</td>
</tr>
</tbody>
</table>
When trying to allocate an IP address from within a scope, the default action of the DHCP server is to try to allocate the least recently accessed address first, from the list of available leases. But all the operations that require accessing the lease like listing all the leases or all leases in a scope, asking for a specific lease (nrcmd> lease addr), searching leases, or modifying leases (activate, deactivate, or force available) affect the ordering of the leases in the list of available leases with the server.

Operating on a single lease places that lease at the end of the list. Listing leases causes the leases to be arranged in numerical order, making the lowest numbered lease to end up first on the available list. Other operations that require the server to access the lease, like leasequery requests also impacts the order of leases.
Thus, in general there is no way to predict which IP address within a scope is allocated at a given time. Usually this poses no difficulty, but there are times when a more deterministic allocation strategy is desired. To configure a completely deterministic address allocation strategy, you can enable the allocate-first-available attribute on a scope. This causes the available address with the lowest numeric value to be allocated for a DHCP client. Thus, the first client gets the first address in the lowest range, and the second client the second one in that range, and so on. This is shown in Figure 21-2.

Figure 21-2  Address Allocation with allocate-first-available Set

![Diagram showing address allocation with allocate-first-available set](image)

Note that there is some minor performance cost to this deterministic allocation strategy, not so much that you should not use it, but possibly enough so that you should not use it if you do not need it. When using this deterministic allocation strategy approach in a situation where the scope is in a failover relationship, the question of how to allocate the available IP addresses for the backup server comes up on the main server. By default, the address halfway between the lowest and highest ones in the scope becomes the failover-backup-allocation-boundary. The available addresses for the backup server are allocated working down from this boundary (if any addresses are available in that direction). If no address is available below this boundary, then the first available one above the boundary is used for the backup server. You can configure the failover-backup-allocation-boundary for the scope if you want to have a different address boundary than the halfway point.

You would use a deterministic allocation strategy and configure allocate-first-available in situations where you might allocate a scope with a larger number of IP addresses than you were sure you needed. You can later shrink back the ranges in the scope so as to allow moving address space to another network or server. In the nondeterministic approach, the allocated addresses are scattered all over the ranges, and it can be very hard to reconfigure the DHCP clients to free up, say, half of the scope addresses. However, if you configure allocate-first-available, then the allocated addresses tend to cluster low in the scope ranges. It is then probably simpler to remove ranges from a scope that does not need them, so that those addresses can be used elsewhere.
Editing Scopes

Note
You can only make changes to a scope's subnet, if there are no reservations or ranges that conflicts with the change, either in the current scope or any other scope with the same old subnet as those scopes' subnet will also be changed.

Local Advanced Web UI

Step 1
Create a scope, as described in the “Creating Scopes” section on page 21-9.

Step 2
Reload the DHCP server.

Step 3
Click the name of the scope on the Scopes pane on the List/Add DHCP Scopes page to open the Edit DHCP Scope page. (If a server reload is required, a status message indicates it and you must reload first before proceeding.)

Step 4
Modify the fields or attributes as necessary. You can also modify the name of the scope.

Step 5
To edit the scope embedded policy, see the “Configuring Embedded Policies for Scopes” section on page 21-18. To list leases for the scope, see the “Viewing Leases” section on page 23-2.

Step 6
Click Save.

Step 7
Reload the DHCP server.

CLI Commands

After you create a scope, look at the properties for all the scopes on the server, use scope list (or scope listnames, scope name show, or scope name get attribute). Then:

- To reset an attribute, use scope name set. For example, you can reset the name of the scope by using scope name set name=new name
- To enable or disable an attribute, use scope name enable or scope name disable.

See the scope command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions.

Related Topics

- Staged and Synchronous Mode, page 21-18
- Configuring Embedded Policies for Scopes, page 21-18
- Configuring Multiple Subnets on a Network, page 21-19
- Enabling and Disabling BOOTP for Scopes, page 21-20
- Disabling DHCP for Scopes, page 21-20
- Deactivating Scopes, page 21-21
- Setting Scopes to Renew-Only, page 21-21
- Setting Free Address SNMP Traps on Scopes, page 21-21
- Removing Scopes, page 21-22
Staged and Synchronous Mode

New scopes or modifications to scopes can be in one of two modes—staged or synchronous:

- **Staged**—New scopes or modifications to existing scopes are written to the database, but not propagated to the DHCP server until the DHCP server is reloaded.

- **Synchronous**—Most new scopes and scope modifications (including deletions) are immediately propagated to the DHCP server (without the need for a reload). Not all scope changes are possible. For example, changing the primary subnet on a scope is not allowed (a reload is required to effect the change). Furthermore, only scope attribute changes can be propagated without a reload. For example, changes to named policies require a DHCP server reload.

If you add or modify a scope while in staged mode and then change the dhcp edit mode to synchronous, the first change in synchronous mode applies all pending changes for that scope (not just the ones made while in synchronous mode).

Local Basic or Advanced Web UI

To view the current dhcp edit mode or change the dhcp edit mode, click the *username* drop-down list on the top right of the window and choose **Session Settings**. If the scope is up to date in the DHCP server, the *Total synchronized scopes* message appears on the List/Add DHCP Scopes page (in Advanced mode) and the *Scope status: synchronized* message appears on the Edit DHCP Scope page (in both modes). If the scope is not up to date, the *Scope name status: reload required* message is displayed.

CLI Commands

View the dhcp edit mode by using `session get dhcp-edit-mode`, or set the dhcp edit mode using `session set dhcp-edit-mode={sync | staged}`. To view the scopes that are not synchronized with the DHCP server, use `scope report-staged-edits`. For example:

```
nrcmd> scope report-staged-edits
100 Ok
example-scope: [reload-required]
```

Configuring Embedded Policies for Scopes

When you create a scope, Cisco Prime IP Express automatically creates an embedded policy for it. However, the embedded policy has no associated properties or DHCP options until you enable or add them. An embedded policy can be useful, for example, in defining the router for the scope. As the “Types of Policies” section on page 22-1 describes, the DHCP server looks at the embedded policy of a scope before it looks at its assigned, named policy.

**Note**

If you delete a scope policy, you remove all of its properties and attributes.

Local Advanced Web UI

**Step 1**
Create a scope, as described in the “Creating Scopes” section on page 21-9.

**Step 2**
Click the name of the scope on the Scopes pane on the List/Add DHCP Scopes page to open the Edit DHCP Scope page.
Step 3 Click **Create New Embedded Policy** to create a new embedded policy, or **Edit Existing Embedded Policy** if there is already an existing one, to open the Edit DHCP Embedded Policy for Scope page.

Step 4 Modify the fields, options, and attributes on this page. If necessary, unset attributes.

Step 5 Click **Save**.

### CLI Commands

Create a scope first. In the CLI, `scope-policy` uses the same syntax as `policy`, except that it takes the scope name as an argument. Then, to:

- Determine if there are any embedded property values already set for a scope, use `scope-policy scope-name show`.
- Enable or disable an attribute, use `scope-policy name enable` or `scope-policy name disable`.
- Set and unset attributes, use `scope-policy name set` and `unset`.
- List, set, and unset vendor options (see the “Using Standard Option Definition Sets” section on page 22-8).

### Configuring Multiple Subnets on a Network

Cisco Prime IP Express supports multiple logical subnets on the same network segment, which are also called secondary subnets. With several logical subnets on the same physical network, for example, 192.168.1.0/24 and 192.168.2.0/24, you might want to configure DHCP so that it offers addresses from both pools. By pooling addresses this way, you can increase the available number of leases.

To join two logical subnets, create two scopes, and elect one to be primary and the other to be a secondary. After you configure the secondary subnet, a new client on this physical network gets a lease from one or the other scope on a round-robin basis.

### Local Advanced Web UI

Step 1 Create a scope (see the “Creating Scopes” section on page 21-9) that you will make a secondary scope.

Step 2 Click the name of the scope on the Scopes pane on the List/Add DHCP Scopes page to open the Edit DHCP Scope page.

Step 3 To make this a secondary scope, enter the network address of the subnet of the primary scope in the **Primary Subnet** attribute field in the Edit DHCP Scope page.

It is common practice for the **primary-subnet** to correspond directly to the network address of the primary scope or scopes. For example, with examplescope1 created in the 192.168.1.0/24 network, associate examplescope2 with it using **primary-subnet**=192.168.1.0/24. (Note that if Cisco Prime IP Express finds that the defined subnet has an associated scope, it ignores the mask bit definition and uses the one from the primary scope, just in case they do not match.) However, the **primary-subnet** can be a subnet address that does not have a scope associated with it.

Step 4 Click **Save**.

Step 5 Restart or reload the server.
Defining and Configuring Scopes

**CLI Commands**

To assign the secondary scope to a primary one, use `scope name set primary-subnet`, then reload the server.

To remove the secondary scope, use `scope name unset primary-subnet`. When setting the `primary-subnet` attribute, include the number bits for the network mask, using slash notation. For example, represent the network 192.168.1.0 with mask 255.255.255.0 as 192.168.1.0/24. The mask bits are important. If you omit them, a /32 mask (single IP address) is assumed.

**Enabling and Disabling BOOTP for Scopes**

The BOOTstrap Protocol (BOOTP) was originally created for loading diskless computers. It was later used to allow a host to obtain all the required TCP/IP information so that it could use the Internet. Using BOOTP, a host can broadcast a request on the network and get the data required from a BOOTP server. The BOOTP server listens for incoming requests and generates responses from a configuration database for the BOOTP clients on that network. BOOTP differs from DHCP in that it has no concept of lease or lease expiration. All addresses that a BOOTP server allocates are permanent.

You can configure the Cisco Prime IP Express DHCP server to act like a BOOTP server. In addition, although BOOTP normally requires static address assignments, you can choose either to reserve addresses (and use static assignments) or have addresses dynamically allocated (known as *dynamic BOOTP*).

When you need to move or decommission a BOOTP client, you can reuse its lease simply by forcing lease availability. See the “Forcing Lease Availability” section on page 23-18.

**Local Advanced Web UI**

On the Edit DHCP Scope page, under BootP Settings, enable the `bootp` attribute for BOOTP, or the `dynamic-bootp` attribute for dynamic BOOTP. They are disabled by default. Then click **Save**.

**CLI Commands**

Use `scope name enable bootp` to enable BOOTP, and `scope name enable dynamic-bootp` to enable dynamic BOOTP. Reload the DHCP server (if in staged dhcp edit mode).

**Disabling DHCP for Scopes**

You can disable DHCP for a scope if you want to use it solely for BOOTP. See the “Enabling and Disabling BOOTP for Scopes” section on page 21-20. You can also temporarily deactivate a scope by disabling DHCP, but deactivation is more often used if you are enabling BOOTP. See the “Deactivating Scopes” section on page 21-21.

**Local Advanced Web UI**

On the Edit DHCP Scope page, under BootP Settings, disable the `dhcp` attribute and enable the `bootp` attribute. Then click **Save**.

**CLI Commands**

Use `scope name disable dhcp` to disable DHCP. You should also enable BOOTP and reload the server (if in staged dhcp edit mode).
**Deactivating Scopes**

You might want to temporarily deactivate all the leases in a scope. To do this, you must disable both BOOTP and DHCP for the scope.

**Local Advanced Web UI**

On the Edit DHCP Scope page, under Miscellaneous Settings, explicitly enable the `deactivated` attribute. Then click **Save**.

**CLI Commands**

Use `scope name enable deactivated` to disable BOOTP and DHCP for the scope. Reload the DHCP server (if in staged dhcp edit mode).

**Setting Scopes to Renew-Only**

You can control whether to allow existing clients to re-acquire their leases, but not to offer any leases to new clients. A renew-only scope does not change the client associated with any of its leases, other than to allow a client currently using an available IP address to continue to use it.

**Local Advanced Web UI**

On the Edit DHCP Scope page, under Miscellaneous Settings, explicitly enable the `renew-only` attribute. Then click **Save**.

**CLI Commands**

Use `scope name enable renew-only` to set a scope to renew-only.

**Setting Free Address SNMP Traps on Scopes**

You can set SNMP traps to capture unexpected free address events by enabling the traps and setting the low and high thresholds for a scope. You can also set traps based on networks and selection tags instead of scopes.

When setting the threshold values, it is advisable to maintain a small offset between the low and high values, as described in the “Simple Network Management” section on page 1-2). The offset can be as little as 5%, for example, a low value of 20% and a high value of 25%, which are the preset values.

Here are some variations on how you can set the server and scope values for these attributes:

- Get each scope to trap and reset the free address values based on the server settings, as long as at least one recipient is configured.
- Disable the traps at the scope level or specify different percentages for each scope.
- Disable the traps globally on the server, but turn them on for different scopes.
- Set the traps at the network level or selection tags level.
Defining and Configuring Scopes

Local Advanced Web UI

Step 1 To create a trap configuration, click the Deploy menu and choose Traps under the DHCP submenu to open the List/Add Trap Configurations page.

Step 2 Click the Add Traps icon, enter a name for the trap configuration, choose scope from the mode drop-down list, and enter the low and high threshold values (they are 20% and 25%, respectively, by default). Click Add AddrTrapConfig. (You can go back to edit these values if you need to.)

Step 3 Edit the created scope to which you want to apply the threshold settings. Under SNMP Trap Settings, enter the name of the trap in the free-address-config attribute field. Click Save.

CLI Commands

Use addr-trap name create to add a trap configuration. To set the thresholds, use the addr-trap name set method (or include the threshold settings while creating the trap). For example:

```
nrcmd> addr-trap trap-1 create
nrcmd> addr-trap trap-1 set low-threshold
nrcmd> addr-trap trap-1 set high-threshold
```

To set the free-address trap, use scope name set free-address-config=trap-name. For example:

```
nrcmd> scope scope-1 set free-address-config=trap-1
```

Removing Scopes

Caution Although removing a scope from a DHCP server is easy to do, be careful. Doing so compromises the integrity of your network. There are several ways to remove a scope from a server, either by re-using or not re-using addresses, as described in the following sections.

DHCP, as defined by the IETF, provides an address lease to a client for a specific time (defined by the server administrator). Until that time elapses, the client is free to use its leased address. A server cannot revoke a lease and stop a client from using an address. Thus, while you can easily remove a scope from a DHCP server, the clients that obtained leases from it can continue to do so until it expires. This is true even if the server does not respond to their renewal attempts, which happens if the scope was removed.

This does not present a problem if the addresses you remove are not reused in some way. However, if the addresses are configured for another server before the last lease expires, the same address might be used by two clients, which can destabilize the network.

Cisco Prime IP Express moves the leases on the removed scope to an orphaned leases pool. When creating a scope, orphaned leases are associated with appropriate scopes.

Related Topics

Removing Scopes if Not Reusing Addresses, page 21-22
Removing Scopes if Reusing Addresses, page 21-23

Removing Scopes if Not Reusing Addresses

You can remove scopes if you are not reusing addresses.
Local Basic or Advanced Web UI

If you are sure you do not plan to reuse the scope, on the Manage Scopes or List/Add DHCP Scopes page, click the Delete Scopes icon in the Scopes pane after selecting the name, and confirm or cancel the deletion.

CLI Commands

Be sure that you are not immediately planning to reuse the addresses in the scope, then use scope name delete to delete it.

Removing Scopes if Reusing Addresses

If you want to reuse the addresses for a scope you want to remove, you have two alternatives:

- **If you can afford to wait until all the leases in the scope expire**—Remove the scope from the server, then wait for the longest lease time set in the policy for the scope to expire. This ensures that no clients are using any addresses from that scope. You can then safely reuse the addresses.

- **If you cannot afford to wait until all the leases in the scope expire**—Do not remove the scope. Instead, deactivate it. See the “Deactivating Scopes” section on page 21-21. Unlike a removed scope, the server refuses all clients’ renewal requests, which forces many of them to request a new lease. This moves these clients more quickly off the deactivated lease than for a removed scope.

You can use the ipconfig utility in Windows to cause a client to release (/release) and re-acquire (/renew) its leases, thereby moving it off a deactivated lease immediately. You can only issue this utility from the client machine, which makes it impractical for a scope with thousands of leases in use. However, it can be useful in moving the last few clients in a Windows environment off deactivated leases in a scope.

Managing DHCP Networks

When you create a scope, you also create a network based on its subnet and mask. Scopes can share the same subnet, so that it is often convenient to show their associated networks and the scopes. Managing these networks is a local cluster function only. You can also edit the name of any created network.

Related Topics

Listing Networks, page 21-23
Editing Networks, page 21-24

Listing Networks

The List Networks page lets you list the networks created by scopes and determine to which scopes the networks relate. The networks are listed by name, which the web UI creates from the subnet and mask. On this page, you can expand and collapse the networks to show or hide their associated scopes.

In Basic mode, from the Design menu, choose Networks from the DHCPv4 to open the DHCP Network Tree page (for DHCPv6 networks, this opens the DHCP v6 Networks Tree page). On this page, you can:

- **List the networks**—The networks appear alphabetically by name. You can identify their subnet and any assigned selection tags. Click the plus (+) sign next to a network to view the associated scopes.
To expand all network views, click **Expand All**. To collapse all network views to show just the network names, click **Collapse All**.

- **Edit a network name**—Click the network name. See “Editing Networks” section on page 21-24.

### Editing Networks

You can edit a network name. The original name is based on the subnet and mask as specified in the scope. You can change this name to an arbitrary but descriptive string.

#### Local Basic or Advanced Web UI

| Step 1 | From the **Design** menu, choose **Networks** from the **DHCPv4** submenu or **Networks** from the **DHCPv6** submenu to open the DHCP Network Tree page (DHCP v4) or the DHCP v6 Network Tree page (DHCP v6).

For DHCPv6, the DHCP v6 Networks page is for creating networks. Enter a name for the network, choose a template, if desired, and enter the template root prefix name and click **Add Link** (see the “Viewing DHCPv6 Networks” section on page 27-32.).

If you want to edit a network, click the name of the network you want to edit. This opens the Edit DHCP v6 Link page.

| Step 2 | Click **Save**. |
Configuring Policies and Options

This chapter describes how to set up DHCP policies and options. Before clients can use DHCP for address assignment, you must add at least one DHCPv4 scope (dynamic address pool) or DHCPv6 prefix to the server. The policy attributes and options are assigned to the scope or prefix.

Related Topics
Configuring DHCP Policies, page 22-1
Creating DHCP Option Definition Sets and Option Definitions, page 22-8

Configuring DHCP Policies

Every DHCPv4 scope or DHCPv6 prefix must have one or more policies defined for it. Policies define lease duration, gateway routers, and other configuration parameters, in what are called DHCP options. Policies are especially useful if you have multiple scopes or prefixes, because you need only define a policy once.

This section describes how you can define named policies with specific attributes and option definitions, or use system default or embedded policies.

Related Topics
Types of Policies, page 22-1
Policy Hierarchy, page 22-3
Creating and Applying DHCP Policies, page 22-3
Cloning a Policy, page 22-5
Setting DHCP Options and Attributes for Policies, page 22-5
Creating and Editing Embedded Policies, page 22-7

Types of Policies

There are three types of policies—system default, named, and embedded:

- **System default (system_default_policy)**—Provides a single location for setting default values on certain options for all scopes or prefixes. Use the system default policy to define attributes and standard DHCP options that have common values for all clients on all the networks that the DHCP server supports. You can modify the system default options and their values. If you delete a system default policy, it reappears using its original list of DHCP options and their system-defined values (see Table 22-1 on page 22-2).
Chapter 22      Configuring Policies and Options

Configuring DHCP Policies

- **Named**—Policies you explicitly define by name. Named policies are usually named after their associated scope, prefix, or client grouping. For example, the policy might be assigned attributes and options that are unique to a subnet, such as for its routers, and then be assigned to the appropriate scope or prefix.

  Cisco Prime IP Express includes a policy named **default** when you install the DHCP server. The server assigns this policy to newly created scopes and prefixes. You cannot delete this default policy.

- **Embedded**—A policy embedded in (and limited to) a named scope, scope template, prefix, prefix template, client, or client-class. An embedded policy is implicitly created (or removed) when you add (or remove) the corresponding object. Embedded policy options have no default values and are initially undefined.

  Be sure to save the object (scope, prefix, client, or client-class) for which you are creating or modifying an embedded policy. Not doing so is a common error when using the web UI. Click **Modify** for both the embedded policy and the parent object.

### Table 22-1 System Default Policy Option Values

<table>
<thead>
<tr>
<th>System Default Option</th>
<th>Predefined Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>all-subnets-local</td>
<td>False</td>
</tr>
<tr>
<td>arp-cache-timeout</td>
<td>60 seconds</td>
</tr>
<tr>
<td>broadcast-address</td>
<td>255.255.255.255</td>
</tr>
<tr>
<td>default-ip-ttl</td>
<td>64</td>
</tr>
<tr>
<td>default-tcp-ttl</td>
<td>64</td>
</tr>
<tr>
<td>dhcp-lease-time</td>
<td>604800 seconds (7d)</td>
</tr>
<tr>
<td>802.3-encapsulation</td>
<td>False</td>
</tr>
<tr>
<td>interface-mtu</td>
<td>576 bytes</td>
</tr>
<tr>
<td>mask-supplier</td>
<td>False</td>
</tr>
<tr>
<td>max-dgram-reassembly</td>
<td>576 bytes</td>
</tr>
<tr>
<td>non-local-source-routing</td>
<td>False</td>
</tr>
<tr>
<td>path-mtu-aging-timeout</td>
<td>6000 seconds</td>
</tr>
<tr>
<td>path-mtu-plateau-tables</td>
<td>68, 296, 508, 1006, 1492, 2002, 4352, 8166, 17914, 32000</td>
</tr>
<tr>
<td>perform-mask-discovery</td>
<td>False</td>
</tr>
<tr>
<td>router-discovery</td>
<td>True</td>
</tr>
<tr>
<td>router-solicitation-address</td>
<td>224.0.0.2</td>
</tr>
<tr>
<td>tcp-keepalive-garbage</td>
<td>False</td>
</tr>
<tr>
<td>tcp-keepalive-interval</td>
<td>0 seconds</td>
</tr>
<tr>
<td>trailer-encapsulation</td>
<td>False</td>
</tr>
</tbody>
</table>
Policy Hierarchy

To eliminate any conflicting attribute and option values that are set at various levels, the Cisco Prime IP Express DHCP server uses a local priority method. It adopts the more locally defined attribute and option values first while ignoring the ones defined on a more global level, and includes any default ones not otherwise defined. When the DHCP server makes processing decisions for a DHCPv4 client, it prioritizes the attributes and options in this order:

1. Client embedded policy.
2. Client named policy.
3. Client-class embedded policy.
4. Client-class named policy.
5. Scope embedded policy for clients, or address block embedded policy for subnets.
6. Scope named policy for clients (or default policy if a named policy is not applied to the scope), or address block named policy for subnets.
7. Any remaining unfulfilled attributes and options in the system_default_policy. For attributes, the default value for the most local policy applies.

Note

For DHCPv6 policy prioritization, see the “DHCPv6 Policy Hierarchy” section on page 27-12.

Creating and Applying DHCP Policies

This section describes how to create a policy at the DHCP server level and then allow specific scopes or prefixes to reference it. A policy can consist of a:

- **Name**—Not case-sensitive and must be unique.
- **permanent-leases attribute**—A permanent lease never expires.
- **Lease time**—How long a client can use an assigned lease before having to renew the lease with the DHCP server (the lease time attributes are not available for an embedded policy, only the option). The default lease time for both system default and default policies is seven days (604800 seconds). A policy contains two lease times—the client lease time and the server lease time:
  - **Client lease time**—Determines how long the client believes its lease is valid. (Set the client lease time using a DHCP option, not a policy attribute.)
  - **Server lease time**—Determines how long the server considers the lease valid. Note that the server lease time is independent of the lease grace period. The server does not allocate the lease to another client until after the lease time and grace period expire.

Caution

Although Cisco Prime IP Express supports the use of two lease times for special situations, Cisco Systems generally recommends that you not use the server-lease-time attribute.

You can establish these two different lease times if you want to retain information about client DNS names and yet have them renew their leases frequently. When you use a single lease time and it expires, the server no longer keeps that client DNS name. However, if you use a short client lease time and a longer server lease time, the server retains the client information even after the client lease expires. For details on leases, see Chapter 23, “Managing Leases”.
Chapter 22 Configuring Policies and Options

Configuring DHCP Policies

- **Lease grace period**—Time period after the lease expires that it is unavailable for reassignment (not available for an embedded policy).

- **DNS update configuration**—A DNS update configuration specifies the type of DNS updates to perform, the zones involved, the DNS server to be updated, and the related security. The policy determines the forward and reverse DNS update configuration objects, and can also specify the forward zone to use if a DNS server hosts multiple zones. (For details on DNS update configurations, see the “Creating DNS Update Configurations” section on page 29-5.)

- **DHCP options**—To add option values, see the “Setting DHCP Options and Attributes for Policies” section on page 22-5.

## Local Basic or Advanced and Regional Web UI

**Step 1** From the Design menu, choose **Policies** under the **DHCP Settings** submenu to open the List/Add DHCP Policies page.

**Step 2** The default policy and system_default_policy are already provided for you.

**Step 3** Click the **Add Policies** icon in the Policies pane, give the policy a unique name (required).

**Step 4** Set the offer timeout and grace period values or leave them blank.

**Step 5** Enter the DHCP Lease Time, if required and click **Add DHCP Policy** to add the named policy.

**Step 6** In the Edit DHCP Policy page, you can:

- Add the necessary DHCP options (see the “Setting DHCP Options and Attributes for Policies” section on page 22-5) like:
  - **Lease time**—Set the **dhcp-lease-time** (51) option.
  - **Limitation count**—See the “Using Expressions” section on page 26-1.
  - **Use client IDs for reservations**—See the “Overriding Client IDs” section on page 23-17.

  To set vendor-specific options, see the “Using Standard Option Definition Sets” section on page 22-8.

- In Advanced mode, set the policy attributes, which include:
  - **Unavailable timeout**—See the “Setting Timeouts for Unavailable Leases” section on page 23-20.
  - **Inhibit all renew**—See the “Inhibiting Lease Renewals” section on page 23-19.
  - **Inhibit all renew at reboot**
  - **Permanent leases** (not recommended).
  - **Lease retention limit**

- Set the DNS update configuration that determines which forward or reverse zones you want to include in a DNS update (DNS Update Settings). You can set:
  - **forward-dnsupdate**—Name of the update configuration for the forward zone. Note that you can thereby set different update configurations for forward and reverse zones.
  - **forward-zone-name**—If necessary, overrides the forward zone in the update configuration. Use this in case a DNS server is hosting multiple zones.
  - **reverse-dnsupdate**—Name of the update configuration for the reverse zone. If not set on any policy in the policy hierarchy applicable to the client request (see the “Policy Hierarchy” section on page 22-3), the DHCP server uses the **forward-dnsupdate** configuration.

**Step 7** Click **Save**.
Step 8

Reload the DHCP server.

In the regional web UI, you can also pull replica policies and push policies to local clusters. (See the “Managing DHCP Policies” section on page 6-16 for regional policy management.)

CLI Commands

Use `policy name create` to create the policy. Then use `policy name set offer-timeout=value` and `policy name set grace-period=value` to set these two values.

To set policy options, use `policy name setOption`:

- Lease time—Use `policy name set LeaseTime`.
- Subnet mask—Use a combination of `policy name setOption subnet-mask value` and `dhcp enable get-subnet-mask-from-policy`.

To confirm the option settings, use `policy name listOptions` or `policy name getOption`.

To enable permanent leases (not recommended), use `policy name enable permanent-leases`. Note that enabling permanent leases forces the `dhcp-lease-time` option (51) to be set to infinite.

Related Topics

Types of Policies, page 22-1
Policy Hierarchy, page 22-3
Cloning a Policy, page 22-5
Setting DHCP Options and Attributes for Policies, page 22-5
Creating and Editing Embedded Policies, page 22-7
Creating DHCP Option Definition Sets and Option Definitions, page 22-8

Cloning a Policy

In the CLI, you can clone a policy from an existing one by using `policy clone-name create clone=policy`, and then make adjustments to the clone. For example:

```
nrcmd> policy cloned-policy create clone=example-policy-1 offer-timeout=4m
```

Setting DHCP Options and Attributes for Policies

DHCP options automatically supply DHCP clients with configuration parameters, such as domain, nameserver, and subnet router addresses (see the “Creating DHCP Option Definition Sets and Option Definitions” section on page 22-8). Note that the Cisco Prime IP Express user interfaces allow you to set some option values on a policy that actually have no effect on the packet returned to the client (such as hostname and dhcp-server-identifier).

The server searches the policies, in order, for these BOOTP and DHCP attribute values and returns the first occurrence of these values in its reply packet:

- `packet-siaddr` returned in the `siaddr` packet field
- `packet-file-name` returned in the `file` field
- `packet-server-name` returned in the `sname` field
Chapter 22  Configuring Policies and Options

Configuring DHCP Policies

Related Topics

Adding Option Values, page 22-6
Adding Complex Values for Suboptions, page 22-7

Adding Option Values

You can view, set, unset, and edit DHCP option values. When you set an option value, the DHCP server replaces any existing value or creates a new one, as needed for the given option name. Cisco Prime IP Express DHCP options are grouped into categories to aid you in identifying options that you must set in various usage contexts. You can create custom option definitions to simplify entering custom option values (see the “Creating Custom Option Definitions” section on page 22-10).

Local Basic or Advanced and Regional Web UI

Step 1  Create a policy, as described in the “Creating and Applying DHCP Policies” section on page 22-3.

Step 2  On the Edit DHCP Policy page, add each DHCP option to the policy by choosing its number and name in the drop-down list. The choices indicate the data type of the option value (see the “Option Definition Data Types and Repeat Counts” section on page 22-15).

Tip  You can sort the options by Name, Number, or (in the case of DHCPv4) Legacy (grouping).

Step 3  Add the appropriate option value in the Value field. The web UI does error checking based on the value entered. For example, to add the lease time for the policy, click the \( \text{dhcp-lease-time (unsigned time)} \) option in the Number drop-down list, then add a lease time value in the Value field. (Options do not have preset values.)

Tip  If you are configuring an option on a policy while another user is editing the option definition, log out of the session and log back in to get the new option definition.

Step 4  Click Add Option for each option. You must supply a value or you cannot add the option.

Step 5  Click Save.

Tip  If you add new option values or edit existing ones, be sure to save the policy object by clicking Save.

CLI Commands

To view option values, use policy name getOption and policy name listOptions. To set option values, use policy name setOption option. When you set an option value, the DHCP server replaces any existing value or creates a new one, as needed, for the given option name. To unset option values, use policy name unsetOption.
Adding Complex Values for Suboptions

If you are adding more complex option values such as for suboptions, use a parenthesized string format. The format requires that you:

- Enclose each option level (option, suboption, subsuboption) in parentheses.
- Separate multiple values with commas.
- Separate data fields for packed data (missing the suboption code or length) with semicolons.

For example, the `cablelabs-client-configuration` option (122) normally has 10 suboptions as well as some subsuboptions. This example shows the syntax to set the suboption 1, 2, 3, and 4 data values, and includes the two subsuboptions for suboption 3 and the three subsuboptions for suboption 4 (which are packed data and have no code numbers):

```
(1 10.1.1.10) (2 10.2.2.10) (3 (flag 0; provisioning-server server.example.com.)) (4 (as-backoff-retry-initial-time-ms 10; as-backoff-retry-max-time 10s; as-backoff-retry-count 100))
```

The suboption name (such as primary-dhcp-server) is optional. Hence, it is often safer to use just the code number and data value (or just the data value for packed data) to minimize typographical errors and parsing failures. The compacted (and preferred) version of the previous example that strips out the suboption names is:

```
(1 10.1.1.10) (2 10.2.2.10) (3 (0;server.example.com.)) (4 (10;10s;100))
```

Even if you use numerical code values, Cisco Prime IP Express always includes the equivalent names when it displays the suboptions (see the “Creating DHCP Option Definition Sets and Option Definitions” section on page 22-8).

To include suboptions that include enterprise IDs (such as for option 125), use the following format, for example, when entering in the policy option value:

```
(enterprise-id 1((1 10.1.1.1) (2 10.2.2.2) (3 www.cisco.com)))
```

The parentheses surround the enterprise ID itself, the suboptions as a group, and each suboption.

Creating and Editing Embedded Policies

An embedded policy is embedded for a DHCPv4 scope or scope template, DHCPv6 prefix or prefix template, client, or client-class (see Chapter 27, “Managing DHCPv6 Addresses,” for embedded policies in DHCPv6). You can create or edit an embedded policy.

Local Advanced Web and Regional UI

**Step 1**
From the Design menu, choose one of the following that appear for DHCPv4 or DHCPv6 in the local web UI: Scopes, Scope Templates, Clients, Client-Classes, Prefixes, or Links. (The regional web UI can have the selections Scope Templates, Client-Classes, Prefixes, and Links.)

**Step 2**
Click the name of the object on the left pane to open its Edit page.

**Step 3**
Click Create New Embedded Policy or Edit Existing Embedded Policy under the Embedded Policy section of the page. This opens the Edit DHCP Embedded Policy page for the object.

**Step 4**
Make changes to the values as needed, then click Modify Embedded Policy.
Chapter 22  Configuring Policies and Options

Creating DHCP Option Definition Sets and Option Definitions

In Cisco Prime IP Express, you configure option values on policies for such things as lease times and router addresses. Numerous RFCs describe the formatting of DHCP option values, beginning with RFC 2132. Option definitions are used in the web UI and CLI to control formatting of option values in policies. You can define option definitions separately for the DHCPv4 and DHCPv6 address spaces, as:

- **Standard (built-in) options**—Defined by the RFCs. In the web UI, these are in the `dhcp-config` and `dhcp6-config` definition sets. The CLI includes additional `dhcp-default` and `dhcp6-default` definition sets that are hidden, but accessible if you call for them specifically. (See the “Using Standard Option Definition Sets” section on page 22-8.)

- **Custom options**—New or modified definitions in the supplied `dhcp-config` or `dhcp6-config` definition sets. Once you add or modify definitions in the web UI, they are added to the `dhcp-custom` or `dhcp6-custom` definition sets in the CLI. (See the “Creating Custom Option Definitions” section on page 22-10.)

- **Vendor-specific options**—Defined in their own definition sets. The CableLabs definition sets (`dhcp-cablelabs-config` and `dhcp6-cablelabs-config`) are preconfigured in Cisco Prime IP Express. The CLI also includes `dhcp-cablelabs-default`, `dhcp6-cablelabs-default`, `dhcp-cablelabs-custom`, and `dhcp6-cablelabs-custom` definition sets. (See the “Using Standard Option Definition Sets” section on page 22-8.)

### Related Topics

- Using Standard Option Definition Sets, page 22-8
- Creating Custom Option Definitions, page 22-10
- Creating Vendor-Specific Option Definitions, page 22-10
- Option Definition Data Types and Repeat Counts, page 22-15
- Adding Suboption Definitions, page 22-16
- Importing and Exporting Option Definition Sets, page 22-17
- Pushing Option Definition Sets to Local Clusters, page 22-17
- Pulling Option Definition Sets from Replica Data, page 22-18
- Setting Option Values for Policies, page 22-18

### Using Standard Option Definition Sets

Cisco Prime IP Express provides two standard, built-in option definition sets, `dhcp-config` and `dhcp6-config`, for DHCPv4 and DHCPv6 option definitions, respectively. You can create new options definitions in these sets or you can overwrite existing ones. New option definitions or ones that were overwritten are identified by an asterisk (*). You can delete these definitions and there is no deletion confirmation given. However, saving the set after deleting an overwritten definition causes the original definition to reappear in the set.
Caution
Arbitrarily modifying the standard definitions (or adding suboption definitions) can adversely affect configurations.

Local Advanced and Regional Web UI

Step 1
From the Design menu, choose Options under the DHCPv4 or DHCPv6 submenu to open the List/Add DHCP Option Definition Sets page. (DHCP option definition is not available in Basic mode.)

Step 2
Click the dhcp-config (DHCPv4) or dhcp6-config (DHCPv6) link to open the Edit DHCP Option Definition Set page, then click Add/Edit Option Definition icon in the Option Definitions tab. View the predefined definitions on the List DHCP Option Definitions page. These are the definitions that control the formatting of the option values you add to policies. If there are suboption definitions, you can expand to show them.

Step 3
To add a definition, click the Add Option Definition icon. On the Edit DHCP Option Definition page, give the option an number, name, description, type, and repeat count (whether more than one instance of the option is allowed or required). (For details on the data types and repeat count values, see the “Option Definition Data Types and Repeat Counts” section on page 22-15.)

Note
You cannot add an option definition for an option number or name that already exists. However, you can modify any option definition that appears as a hyperlink on the page.

Step 4
Click Add Option Definition. Then, on the List/Add DHCP Option Definition Sets page, click Save.

Step 5
Click the Revert button if you want to revert to the original definitions in that standard set.

Step 6
In the regional web UI, you can also pull replica definition sets and push definition sets to local clusters. (See the “Pulling Option Definition Sets from Replica Data” section on page 22-18 and the “Pushing Option Definition Sets to Local Clusters” section on page 22-17.)

CLI Commands

To view the entire list of standard DHCP option definitions, use option-set dhcp-config [show] or option-set dhcp6-config [show], or option {id | name} option-set show to view a specific definition. For example:

```
nrcmd> option-set dhcp-config
nrcmd> option subnet-mask dhcp-config show
```

To add a definition to a set, use option id option-set create name type. You cannot add a definition for an option ID (number) or name that already exists. For example, to add option number 222 with the name example-option in the dhcp-config option set, with a string type, use:

```
nrcmd> option 222 dhcp-config create example-option AT_STRING
```

To get a particular option attribute value, use option (id | name) optionset get attribute. To modify an option attribute, use option (id | name) optionset set. You can also unset an option attribute.
Creating Custom Option Definitions

You can create custom option definitions in the standard sets. Click the `dhcp-config` or `dhcp6-config` set on the List/Add DHCP Option Definition Sets page. Then proceed with Step 3 in the “Using Standard Option Definition Sets” section on page 22-8.

Creating Vendor-Specific Option Definitions

You can send vendor-specific option data to DHCP clients that request them.

Note

There are several option codes set aside for vendor-specific options, so that you must explicitly specify the option code number for which you are creating a vendor-specific option definition.

In Cisco Prime IP Express, you can create vendor-specific option definitions in the web UI, or in the CLI by using `option id option-set-name create`. (For details on the option data types, see the “Option Definition Data Types and Repeat Counts” section on page 22-15.)

Vendor-specific options are sent in the following DHCP options:

- **vendor-encapsulated-options (43)**—Set this to a binary data type, then add the vendor-specific suboption definitions. (The data type of the parent option definition is a placeholder only. The suboption definitions define the valid option value formatting.)
- **v-i-vendor-info (125) or vendor-options (17) for DHCPv6**—Set this to a vendor-opts data type, then add the vendor-specific suboption definitions.

You can create vendor-specific option definitions for DHCPv4 options 43 and 125, and DHCPv6 option 17. You add the vendor-specific option definitions into a vendor option definition set that you create.

Caution

Changing option definition properties, or deleting the option definition altogether, can have unexpected side effects on policies. If you delete a custom option definition, also check for the policies that include an option value. Changing an option definition changes the way that they are displayed, not what is stored, so that you do not need to modify the policy value unless you want the policy to return a differently formatted option value. Some option types are very similar, and changing between them can have side effects. For example, strings and DNS names are both entered as string values in the user interfaces, but the formatted option values are quite different.

Note

Cisco Prime IP Express 7.0 preconfigures separate CableLabs (enterprise ID 4491) option definitions in the `dhcp-cablelabs-config` and `dhcp6-cablelabs-config` vendor-specific option definition sets.

Local Advanced and Regional Web UI

Step 1  From the Design menu, choose Options under the DHCPv4 or DHCPv6 submenu to open the List/Add DHCP Option Definition Sets page. View the existing DHCPv4 or DHCPv6 options.

Step 2  Click the Add Options icon in the Options pane to open the Add OptionDefinitionSet dialog box.

Step 3  Enter a name for the option definition set, then choose DHCPv4 or DHCPv6 from the DHCP Type drop-down list.

If you are creating vendor-specific option definitions using:
• Option 43, enter a value in the Vendor Option String field. (See the subsequent section for a sample procedure on creating a vendor option set and vendor option values for option 43.)

• Option 125 for DHCPv4 or option 17 for DHCPv6, enter a valid Enterprise Option Enterprise ID value.

Step 4  Click **Add OptionDefinitionSet**.

Step 5  Click the added option definition set name on the left pane.

Step 6  On the Edit DHCP Option Definition Set page, click the **Option Definitions** tab. Any existing option definitions will appear on this page (new or modified standard definitions are marked with an asterisk).

Step 7  Click **Add Option Definition** icon. Enter the ID number of the option definition, along with its name and a description. The ID must be 43, 125, or 17 (for DHCPv6) for the client to recognize a vendor-specific option definition. The option name does not need to match the one specified in the RFC and can be of your own creation.

Step 8  Choose a data type and repeat count (or enter an absolute repeat count in the next field). The data type must be:

- Binary (AT_BLOB) for option 43.
- Vendor-opts (AT_VENDOR_OPTS) for option 125 (for DHCPv4) and option 17 (for DHCPv6).

(For details on the data type and repeat count values, see the “Option Definition Data Types and Repeat Counts” section on page 22-15.)

Step 9  Click **Add Option Definition**. Then, on the List DHCP Option Definitions page, click **Save**.

Using the Local Advanced web UI to create vendor option set and vendor option values for option 43:

Step 1  From the **Design** menu, choose **Options** under the **DHCPv4** or **DHCPv6** submenu to open the List/Add DHCP Option Definition Sets page.

Step 2  Click the **Add Options** icon in the **Options** pane to open the Add OptionDefinitionSet dialog box.

Step 3  Enter values for the following attributes:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the option definition set; for example, AP1130.</td>
</tr>
<tr>
<td>DHCP Type</td>
<td>Byte size of the type identifiers for all children in this set. You must choose DHCP v4 from the drop-down list.</td>
</tr>
<tr>
<td>Vendor Option String</td>
<td>Exact vendor class identifier string from option-60 that the DHCP client device vendor provides. For example, Cisco AP c1130.</td>
</tr>
</tbody>
</table>

Step 4  Click **Add OptionDefinitionSet**. The List/Add DHCP Option Definition Sets page appears.

Step 5  Click AP1130, the name of the option definition set that appears. The Edit DHCP Option Definition Set AP1130 page appears.

Step 6  Click the **Option Definitions** tab, then **Add Option Definition**.
Chapter 22  Configuring Policies and Options

Creating DHCP Option Definition Sets and Option Definitions

Step 7  Enter the values for the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Number of the option code. You must enter 43.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of this attribute. For example, ap1130-option-43.</td>
</tr>
<tr>
<td>Type</td>
<td>Datatype for the option value. You must choose binary from the drop-down list.</td>
</tr>
</tbody>
</table>

Step 8  Click Add Option Definition.

Note that clicking this button does not save the changes that you make to the option definition set. It only lists the option definition set on the List DHCP Option Definitions page.

Step 9  In the Option Definitions tab, click the name of the new option definition (ap1130-option-43), then Add Sub-Option Definition.

Step 10  In the Add DHCP Option Definition page, enter values for the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>The option code for this suboption. For this example, you must enter 241.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of this attribute. For example, &quot;ap1130-suboption-241&quot;.</td>
</tr>
<tr>
<td>Type</td>
<td>Datatype for the suboption value. For this example, you must choose IP Address from the drop-down list.</td>
</tr>
<tr>
<td>Repeat</td>
<td>The repeat count for this type. For this example, you must choose 1+ from the drop-down list.</td>
</tr>
</tbody>
</table>

Step 11  Click Add Option Definition, then Save.

Step 12  Click Design, then Policies under the DHCP Settings submenu to open the List/Add DHCP Policies page.

Step 13  Choose the policy for which to set this option; or, add a new policy in the Advanced mode. Depending on your selection, the Edit DHCP Policy policy_name or the Add DHCP Policy page appears.

Step 14  From the DHCP v4 Vendor Options drop-down list, choose the name of the option definition set (AP1130), and click Select.

Step 15  Choose the option definition from the Name drop-down list (“ap1130-option-43”) and, in the Value field, enter, for example:

(241 3.3.3.3,4.4.4.4)

Step 16  Click Add Option, then, click Save.

Step 17  Reload the DHCP server.

Examples

You can create a vendor option set and vendor option values from the CLI for Cisco Access Point (AP) devices, SunRay devices, and Cisco 79xx IPPhones using the sample procedures described in this section.

Example 22-1  Creating Vendor Option Set for Cisco AP Devices

Using option 43 for Lightweight Access Point Protocol (LWAPP) APs requires vendor option 43 if you are using Cisco Prime IP Express as the DHCP server. This example is specific to the Cisco Aironet 1130 series. You can modify the example to configure option 43 for other vendor options, such as Cisco Aironet 1200 series and Cisco Aironet 1240 series.
Chapter 22  Configuring Policies and Options

Creating DHCP Option Definition Sets and Option Definitions

Step 1  Create a .txt file with the following content:

```
# Version: 1
#  6.2+ Option-set example for Option 43 with suboptions for Cisco APs
#  NOTE: Need to edit vendor option string to Exact match AP Model string in Option-60.
#        For compatibility with pre-6.2 vendor options ensure that
#        name=vendor-option-string. (Not True in this test example.)
#  ======================================================================
{
  ( id-range = 1 )
  ( vendor-option-string = Cisco AP c1130 )
  ( name = APtest )
  ( children = [
    {
      ( id = 43 )
      ( name = pxe-sample )
      ( desc = )
      ( base-type = AT_BLOB )
      ( children = [
        {
          ( id = 241 )
          ( name = controller )
          ( desc = ap controller )
          ( base-type = AT_IPADDR )
          ( repeat = ONE_OR_MORE )
        ]
      ]
    ]
  ]
}
```

Step 2  Save the file as OptionSetCiscoAP.txt at the following location:

- Windows—\Program Files\IP Express\Local\bin
- Linux—/opt/nwreg2/local/usrbin

Step 3  Import the OptionSetCiscoAP.txt file from the CLI using the `import option-set` command. For example:

```
nrcmd> import option-set OptionSetCiscoAP.txt
```

(For information on importing option definition sets, see the “Importing and Exporting Option Definition Sets” section on page 22-17.)

Step 4  Set the vendor-specific option data on a policy using the `policy name setVendorOption opt-name-or-id opt-set-name value` command.

For example, to set vendor option 43 data for the optionset APtest with values (241 3.3.3.3,4.4.4.4), on an existing policy with the name test, use:

```
nrcmd> policy test setVendorOption 43 APtest "(241 3.3.3.3,4.4.4.4)"
nrcmd> save
```

Step 5  Reload the DHCP server.

```
nrcmd> dhcp reload
```
Example 22-2  Creating Vendor Option Set for SunRay Devices

Use this sample procedure to create vendor option set with multiple suboptions for SunRay Devices:

Step 1  
Create a .txt file with the following content:

```
# Option Definition Set Export/Import Utility
# Version: 1
# 6.2 Option-set example for Option 43 with suboptions for Sun SunRay.
# NOTE: Need to edit vendor option string to match Option-60
# For compatibility with pre-6.2 vendor options ensure that
# name=vendor-option-string.
# ======================================================================
{( id-range = 1 )
  ( vendor-option-string = sunray )
  ( name = sunray )
  ( children = [
    {
      ( id = 43 )
      ( name = option43 )
      ( desc = )
      ( base-type = AT_BLOB )
      ( children = [
        {
          ( id = 21 )
          ( name = AuthSrvr )
          ( desc = AuthSrvr )
          ( base-type = AT_IPADDR )
          ( repeat = ONE_OR_MORE )
        }
        {
          ( id = 35 )
          ( name = AltAuth )
          ( desc = AltAuth )
          ( base-type = AT_IPADDR )
          ( repeat = ONE_OR_MORE )
        }
        {
          ( id = 36 )
          ( name = BarrierLevel )
          ( desc = BarrierLevel )
          ( base-type = AT_SHORT )
        }
      ]
    ]
  ]
)
```

Step 2  
Save the file as OptionSetSunRay.txt at the following location:
- Windows—\Program Files\IP Express\Local\bin
- Linux—/opt/nwreg2/local/usrbin

Step 3  
Import the OptionSetSunRay.txt file from the CLI using the `import option-set file` command. For example:
```
nrcmd> import option-set OptionSetSunRay.txt
```

(For information on importing option definition sets, see the “Importing and Exporting Option Definition Sets” section on page 22-17.)
Step 4  Set the vendor-specific option data on a policy using the `policy name setVendorOption opt-name-or-id opt-set-name value` command.

For example, to set vendor option 43 data for the optionset APtest with multiple suboption values (21 3.3.3.3,4.4.4.4) (35 1.1.1.1) (36 0), on an existing policy with the name test, use:

```
nrcmd> policy test setVendorOption 43 APtest "(21 3.3.3.3,4.4.4.4) (35 1.1.1.1) (36 0)"
nrcmd> save
```

Step 5  Reload the DHCP server.

```
nrcmd> dhcp reload
```

---

**Example 22-3  Creating Option Set for Cisco 79xx IPPhones**

Use this sample procedure to create option set for Cisco 79xx IPPhones:

**Step 1**  Define the option.

```
nrcmd> option 150 dhcp-custom create voip-tftp-server AT_IPADDR desc="VOIP Option-150 Server" repeat=ONE_OR_MORE
```

**Step 2**  Display the configured option.

```
nrcmd> option dhcp-config list
```

**Step 3**  Set policy, by using `policy default setoption voip-tftp-server ip-address`. For example:

```
nrcmd> policy default setoption voip-tftp-server 192.168.1.254
```

**Step 4**  Confirm the policy setting.

```
nrcmd> policy default getoption voip-tftp-server
```

**Step 5**  Reload the DHCP server.

```
nrcmd> dhcp reload
```

---

**Option Definition Data Types and Repeat Counts**

The data type values that you can use appear in Table 22-2.

<table>
<thead>
<tr>
<th>AT_INT8</th>
<th>AT_SHORT</th>
<th>AT_INT</th>
<th>AT_STRING</th>
</tr>
</thead>
<tbody>
<tr>
<td>unsigned 8-bit</td>
<td>unsigned 16-bit</td>
<td>unsigned 32-bit</td>
<td>string</td>
</tr>
<tr>
<td>AT_SINT8</td>
<td>AT_SSHORT</td>
<td>AT_SINT</td>
<td>AT_NSTRING</td>
</tr>
<tr>
<td>signed 8-bit</td>
<td>signed 16-bit</td>
<td>signed 32-bit</td>
<td>string (no termination)</td>
</tr>
<tr>
<td>AT_SHRTI</td>
<td>AT_INTI</td>
<td>AT_BLOB</td>
<td></td>
</tr>
<tr>
<td>unsigned 16-bit (Intel)</td>
<td>unsigned 32-bit (Intel)</td>
<td>binary</td>
<td></td>
</tr>
<tr>
<td>AT_SSHTI</td>
<td>AT_SINTI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>signed 16-bit (Intel)</td>
<td>signed 16-bit (Intel)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Creating DHCP Option Definition Sets and Option Definitions

Chapter 22 Configuring Policies and Options

Creating DHCP Option Definition Sets and Option Definitions

You can view these types in the CLI by using **option listtypes**.

To set the repeat count, set the *repeat-count* attribute to one of the following, or enter an absolute number:

- **ZERO_OR_MORE**—0+ in the web UI
- **ONE_OR_MORE**—1+ in the web UI
- **EVEN_NUMBER**—2n in the web UI

In the CLI, for example, use:

```
nrcmd> option 200 ex-opt-def-set set repeat-count=ZERO_OR_MORE
nrcmd> save
```

Adding Suboption Definitions

You can set a suboption definition for the option definition by clicking **Add Suboption Definition** on the Edit DHCP Option Definition page. This opens the Add DHCP Option Definition page, where you can add the same values as for an option definition. The suboption definition you create is associated with its parent option (or parent suboption) definition. You can define up to six option and suboption levels.

**Note**

You can add suboption definitions by using the web UI only. You currently cannot do so by using the CLI.

Suboption definition formats can be packed or type/length/value (TLV):

- **Packed**—A suboption with a zero ID value and an implicit data type. The option value is the only data in the packet. DHCPv6 options are virtually all defined with packed data. There are no markers for type or length and the layout of the data is inherent in the option definition. You cannot have further suboption definitions for packed suboptions.
- **TLV**—A suboption with a value of 1 through 255 (or 65535) that includes a type, length, and value. The data in the packet has the type and length preceding the value.

In most cases, you will not be mixing packed with TLV suboptions for the same option.

**Note**

DHCP server does not support suboption 0 defined by vendor under V-I Vendor-Specific Information (125). Suboption with a zero ID value is used by DHCP server to specify packed data as mentioned above.
Creating DHCP Option Definition Sets and Option Definitions

Chapter 22      Configuring Policies and Options

Creating DHCP Option Definition Sets and Option Definitions

To enter suboption values when editing policies, see the “Adding Complex Values for Suboptions” section on page 22-7.

Importing and Exporting Option Definition Sets

Importing and exporting option definition sets is a way to copy them between servers. In the CLI, you can import and export option sets by using `import option-set file` and `export option-set name file`.

For example, to import an option set for Preboot Execution Environment (PXE) clients, modify and import a sample file located in the /examples/dhcp directory:

```
nrcmd> import option-set /examples/dhcp/OptionSetPXE.txt
```

**Caution**

Do not export the built-in option definition sets (such as dhcp-config and dhcp-cablelabs-config) and then reimport them. Reimporting an edited option definition set without TAC assistance can cause the server to fail.

Some of the guidelines for the file format include:

- The version string in the file must match the version for the import utility.
- The utility imports just the first option definition set found in the file.
- Delimit objects using curly brackets (`{ }`), attributes using parentheses (`()`), and lists of objects in attributes using square brackets (`[ ]`). Delimit string value attributes using quotes (`" "`).

Using some care, you can also edit the text file to make minor modifications to an option definition set. Cisco Prime IP Express provides two sample option definition set text files in the examples/dhcp directory, OptionSetJumpStart.txt and OptionSetPXE.txt:

- **OptionSetJumpStart.txt**—Edit the vendor-option-string to match the dhcp-class-identifier (option 60) that your JumpStart clients are sending.
- **OptionSetPXE.txt**—Edit the vendor-option-string to match the dhcp-class-identifier (option 60) that your Pre-boot Execution Environment (PXE) clients are sending.

Pushing Option Definition Sets to Local Clusters

You can push option definition sets you create from the regional cluster to any of the local clusters. If you want to push a specific option definition set to a cluster, click **Push Option Definition** sets on the List/Add DHCP Option Definition Sets page, which opens the Push DHCP Option Definition Set to Local Clusters page.

This page identifies the data to push, how to synchronize it with the local cluster, and the cluster or clusters to which to push it. The data synchronization modes are:

- **Ensure** (preset value)—Ensures that the local cluster has new data without affecting any existing data.
- **Replace**—Replaces data without affecting other objects unique to the local cluster.
- **Exact**—Available for “push all” operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the local cluster.

Choose the destination cluster or clusters in the Available field and move it or them to the Selected field.
Creating DHCP Option Definition Sets and Option Definitions

Chapter 22 Configuring Policies and Options

Creating DHCP Option Definition Sets and Option Definitions

Tip

The synchronization mode and cluster choice settings are persistent for the duration of the current login session, so that they are in effect each time you access this page, unless you change them.

After making these choices, click **Push Data to Clusters**. This opens the View Push DHCP Option Definition Set Data Report page.

Pulling Option Definition Sets from Replica Data

You may choose to pull option definition sets from the replica data of the local clusters instead of explicitly creating them. (You may first want to update the option definition set replica data by clicking the **Replicate** icon next to the cluster name.) To pull the option definition sets in the web UI, click **Pull Replica Option Definition Sets** to open the Select Replica DHCP Option Definition Set Data to Pull page.

This page shows a tree view of the regional server replica data for the local clusters’ option definition sets. The tree has two levels, one for the local clusters and one for the scope templates in each cluster. You can pull individual option definition sets from the clusters, or you can pull all of their option definition sets. To pull individual ones, expand the tree for the cluster, then click **Pull Option Definition Set** next to its name. To pull all the ones from a cluster, click **Pull All Option Definition Sets from Cluster**. To pull the option definition sets, you must also choose a synchronization mode:

- **Ensure**—Ensures that the regional cluster has new data without affecting any existing data.
- **Replace** (preset value)—Replaces data without affecting other objects unique to the regional cluster.
- **Exact**—Available for “pull all” operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the regional cluster.

Setting Option Values for Policies

You enter option values on a policy. The option definitions in your server configuration control the format and values that you enter.

Local Advanced and Regional Web UI

On the List/Add DHCP Policies page, click a policy to edit it. (Note that you cannot set options for policies in Basic mode.) On the Edit DHCP Policy page:

- To enter a standard DHCPv4 or DHCPv6 option value for a policy, choose it from the DHCPv4 Options or DHCPv6 Options drop-down list, then set a value for the option. Click **Add Option**.
- To enter a vendor-specific DHCPv4 or DHCPv6 option value for a policy, choose an option definition set in the DHCPv4 Vendor Options or DHCPv6 Vendor Options drop-down list, then click **Select**. The page changes to show the drop-down list that includes the option; choose it, then click **Add Option**.

Note that you can also edit policy attributes on this page. Be sure to click **Modify Policy**.

To edit a configured policy option, click the name of the configured option on the Edit DHCP Policy page to open the Edit DHCP Policy Option page. Enter a new value, then click **Modify Option**.
CLI Commands

Use one of these commands:

```
nrcmd> policy name setOption {name | id} value
nrcmd> policy name setV6Option {name | id} value
nrcmd> policy name setVendorOption {name | id} option-set-name value
nrcmd> policy name setV6VendorOption {name | id} option-set-name value
```

To list the options in the policy, use one of these commands:

```
nrcmd> policy name listOptions
nrcmd> policy name listV6Options
nrcmd> policy name listVendorOptions
nrcmd> policy name listV6VendorOptions
```

To add suboption values, see the “Adding Complex Values for Suboptions” section on page 22-7.
Managing Leases

Leases are at the center of the Dynamic Host Configuration Protocol (DHCP). They are the IP addresses allocated to individual clients for a certain time period. The DHCP server automatically allocates these leases with properly configured scopes that include valid IP address ranges. No two clients can have the same leased address. Reservations are leases that always get the same IP address.

This chapter describes how to manage leases and reservations in a network.

Related Topics

- Configuring Leases in Scopes, page 23-1
- Searching Server-Wide for Leases, page 23-8
- Using Client Reservations, page 23-10
- Creating Lease Reservations, page 23-13
- Setting Advanced Lease and Reservation Properties, page 23-14
- Running Address and Lease Reports, page 23-21
- Querying Leases, page 23-29
- DHCP Listener Configuration, page 23-36
- Lease History Database Compression Utility, page 23-37
- Moving Leases Between Servers, page 23-43

Configuring Leases in Scopes

After setting the IP address ranges for a scope, you can monitor and adjust the leases that result from DHCP assignments.

Related Topics

- Viewing Leases, page 23-2
- Lease States, page 23-2
- Guidelines for Lease Times, page 23-3
- Importing and Exporting Lease Data, page 23-4
- Pinging Hosts Before Offering Addresses, page 23-6
- Deactivating Leases, page 23-7
- Excluding Leases from Ranges, page 23-7
Configuring Leases in Scopes

Chapter 23  Managing Leases

Viewing Leases

To view leases, you must first create a range of IP addresses for them in a scope, as described in the “Set Up DHCP” chapter of the Quick Start Guide for Cisco Prime IP Express or the “Defining and Configuring Scopes” section on page 21-2, then wait for the DHCP server to generate leases based on these addresses.

Local Basic Web UI

To view leases, choose Scopes from the Design menu to open the Manage Scopes page, then click the Leases tab for the scope. This opens the page, where you can click each lease to manage it.

See the “Lease States” section on page 23-2 for a description of the values in the State column. For guidelines as to the lease expiration time, see the “Guidelines for Lease Times” section on page 23-3. To open the Manage DHCP Lease page, click the lease IP address.

Local Advanced Web UI

From the Design menu, choose Scopes under the DHCPv4 submenu to open the List/Add DHCP Scopes page. You can then click the Leases tab for the scope; or you can click the name of the scope to open the Edit DHCP Scope page, then click Leases tab in the page.

CLI Commands

Use lease ipaddr show to show the properties of a particular lease based on its IP address. Use scope name listLeases to show all the leases for a named scope. The output is nearly identical for both commands. Note that you cannot list leases in a particular virtual private network (VPN); all the leases in all the VPNs appear in the list.

You can show the most recent MAC address associated with a lease or what lease is associated with a MAC address. The lease addr macaddr command shows the MAC address of the lease, whether or not that lease is reserved or active. The lease addr list –macaddr command lists the lease data only if the IP address for that MAC address was actively leased (and not reserved). You can also list leases by LAN segment and subnet by using lease addr list –subnet network netaddr netmask.

Lease States

A lease can be in one of the states described in Table 23-1.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available</td>
<td>IP address available to be leased.</td>
</tr>
<tr>
<td>Unavailable</td>
<td>Not leaseable. See the “Handling Leases Marked as Unavailable” section on page 23-20 for ways the DHCP server might set a lease to unavailable.</td>
</tr>
<tr>
<td>Leased</td>
<td>Held by a client.</td>
</tr>
<tr>
<td>Offered</td>
<td>Offered to the client.</td>
</tr>
<tr>
<td>Expired</td>
<td>Available when the lease grace period expires.</td>
</tr>
</tbody>
</table>
Table 23-1 Lease States (continued)

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deactivated</td>
<td>Not renewable or leasable after the lease expires. See the “Deactivating Leases” section on page 23-7.</td>
</tr>
<tr>
<td>Pending available</td>
<td>Failover-related. A lease in the pending-available state is available as soon as the server synchronizes its state with the failover partner.</td>
</tr>
<tr>
<td></td>
<td>See Chapter 28, “Managing DHCP Failover”.</td>
</tr>
</tbody>
</table>

Guidelines for Lease Times

To define appropriate values for lease times, consider these events on your network:

- Frequency of changes to DHCP options and default values.
- Number of available IP addresses compared to clients requesting them.
- Number of network interface failures.
- Frequency at which computers are added to and removed from the network.
- Frequency of subnet changes by users.

All these events can cause clients to release IP addresses or the leases to expire at the DHCP server. Consequently, the addresses may return to the free-address pool for reuse. If many changes occur on your network, Cisco recommends a lease time between one and three days for active networks, and between four and ten days for inactive networks. Assigning such a lease time reassigns IP addresses more quickly as clients leave the subnet.

Another important factor is the ratio of available addresses to connected computers. For example, the demand for reusing addresses is low in a class C network having 254 available addresses, of which only 40 are used. A long lease time, such as two months, might be appropriate in such a situation. The demand would be much higher if there were 240 to 260 clients trying to connect at one time. In this situation, you should try to configure more address space. Until you do, keep the DHCP lease time to under a hour.

Tip

Short lease periods increase the demand that the DHCP server be continuously available, because clients will be renewing their leases more frequently. The DHCP failover functionality can help guarantee such levels of availability.

Be careful when creating policies that have permanent leases. A certain amount of turnover among clients occurs, even in a stable environment. Portable hosts might be added and removed, desktop hosts moved, and network adapter cards replaced. If you remove a client with a permanent lease, it requires manual intervention in the server configuration to reclaim the IP address. It would be better to create a long lease, such as six months, to ensure that addresses are ultimately recovered without administrator intervention.

Recommendations for lease durations include:

- Set cable modem lease times to seven days (604800 seconds). The leases should come from private address space, and the cable modems should seldom move around.
- Leases for customer premises equipment (CPE) or laptops should come from public address space and should match the habits of the user population, with as long a lease as possible to reduce load on the server.
Shorter lease times require more DHCP request and response buffers. Set the request and response buffers for optimal throughput (see the “Setting DHCP Request and Response Packet Buffers” section on page 28-13).

Allow the server to determine the lease period, by ensuring that the allow-lease-time-override policy attribute is disabled, which is its normal default. Even if enabled, clients can only request lease times that are shorter than you configure for the server. Some clients always request a fixed lease time (such as an hour) or the same one they had previously. These kinds of requests can cause problems in that the client never gets the full lease time, thereby generating more traffic for the server.

Defer any lease extensions for clients trying to renew leases before the halfway mark in the lease. For details, see the “Deferring Lease Extensions” section on page 24-8.

### Importing and Exporting Lease Data

You can use the CLI to import lease data to, and export from, text files.

#### Import Prerequisites

Before you can import leases, you must perform several configuration steps:

1. Configure a scope or scopes in the DHCP server for the leases that you plan to import.
2. If you want the hostnames for the leases dynamically entered into DNS as part of the import, configure zones in the DNS server to allow dynamic updates from the DHCP server.
3. Set the DHCP server to import mode so that it does not respond to other lease requests during the lease importing.
4. For all the time fields, use either the number of seconds since midnight GMT January 1, 1970, or a day, month, date, time, year format (Mon Apr 15 16:35:48 2002).
5. After you import the leases, take the DHCP server out of import mode so that it can respond to other lease requests.

**Note** Importing permanent leases will fail if you disable the permanent leases option. Enable this option using policy name enable permanent-leases, as necessary.

#### Import and Export Commands

The `import leases` and `export leases` commands use a special file format. Each record, or line, in the file represents one DHCP client:

```
field-1|field-2|field-3|...|field-13
```

Do not use spaces between the vertical line (|) delimiter and the field values. You must include at least the first four required fields. If you include more, you must delimit all the remaining null fields with the vertical line (|) so that there are 13 fields. The fields are, in order:

1. MAC address in `aa:bb:cc:dd:ee:ff` format (required)
2. MAC address type (required)
3. MAC address length (required)
4. IP address in dotted decimal format, `a.b.c.d` (required)
5. Start of lease time (Greenwich Mean Time, GMT) (optional)
6. Lease expiration time (GMT) (optional)
7. Allowable extension time (GMT) (optional)
8. Last transaction time (GMT) (optional)
9. IP address of the DHCP server (optional)
10. Hostname (without domain) (optional)
11. Domain name (optional)
12. Client ID (optional)
13. VPN name (optional; if omitted, the global VPN is used)

For all the time fields, use either the number of seconds since 1970, or the day-month-date-time-year format (such as Mon Apr 9 16:35:48 2007).

When importing leases, the DHCP server might not accept a lease, or a communication failure might drop the lease packet. In the latter case, the server retries the import several times, and after about a minute, reports a failure. If the import fails, check the DHCP server log file to find the lease that caused the error. Then go back to the import file, delete all lease entries up to and including the offending one, and repeat the lease import.

When you use export leases, you can choose between writing the state of all current and expired leases, or just the current leases, to the output file. Example 23-1 shows part of a lease data export from a Cisco Prime IP Express DHCP server. The blank lines between records appear in the example for clarity; they are not in the actual output.

**Example 23-1 Lease Data Export**

```
Wed Aug 30 08:36:57 2000|Fri Sep 01 09:34:05 2000|204.253.96.57|nomad|cisco.com|
00:d0:ba:d3:bd:3b|blue-vpn

00:d0:ba:d3:bd:3b|1|6|204.253.96.77|Thu Aug 17 13:10:11 2000|Fri Sep 01 14:24:46 2000|
204.253.96.57|NPI9F6AF8|cisco.com|blue-vpn

00:d0:ba:d3:bd:3b|1|6|204.253.96.78|Fri Jun 23 15:02:18 2000|Fri Sep 01 14:11:40 2000|
Fri Jun 23 15:02:18 2000|Fri Sep 01 09:56:40 2000|
204.253.96.57|JTB-LOCAL|cisco.com|blue-vpn
```

**Lease Times in Import Files**

For a lease import request, if the DHCP server is:

- Enabled for import-mode and the lease is not already leased to the client, the server accepts any lease time the client specifies.
- Enabled for import-mode, the lease is already leased to the client, defer-lease-extensions is enabled for the server (the default), and the request arrives before the renewal time (T1), the server uses the existing lease time.
- If the request arrives after T1, the server gives the client whatever it asks for. Within about two minutes of the expiration time, defer-lease-extensions is inoperative.
- Not enabled for import-mode, it never accepts a lease time longer than the server-configured one.
- If allow-lease-time-override is enabled for a policy applicable to the request, the server accepts a shorter lease time from the client. The shorter lease time is acceptable to the server, even though you can set a server expert mode client-requested-min-lease-time attribute that creates a floor for the lease time.

- If allow-lease-time-override is not enabled for any applicable policy, the server ignores the dhcp-lease-time request in the incoming packet and uses the server setting.

If your import file specifies a DNS zone name, the server does not use the zone name when it updates the DNS. If the file specifies a hostname, then the server uses the hostname when updating the DNS, unless hostname specification in a client or client-class entry overrides the hostname.

The client hostname should be in a zone other than the zone associated with the DNS update configuration object used for the DNS update. This can be indicated to the DHCP server, only by specifying that zone in a client or client-class entry.

### Pinging Hosts Before Offering Addresses

You can have the DHCP server use the Internet Control Message Protocol (ICMP) echo message capability (also known as ping) to see if anyone responds to an IP address, before assigning it (using the ping-clients attribute). The ping-clients attribute controls whether the server should attempt to ping an address before offering a lease. If enabled, then the ping-timeout attribute may also need to be set. This test allows the DHCP server to check whether an address is not in use before assigning it.

Using ping can help prevent two clients from using the same address. If a client responds to ping, the DHCP server marks that address as unavailable and offers a different address. This test works only for powered-up clients; it is possible for clients to have a lease and be powered down.

You can also configure the ping-clients attribute at the DHCP server. This attribute controls the default value of the ping-clients attribute of a scope, if not explicitly configured on a scope.

#### Note

If you have configured scopes, the scope-specific configuration takes precedence; scopes without explicit configurations assume the global setting.

The ping timeout period is important. Because pinging helps to ensure that no client is using a particular IP address, each ping must wait the entire timeout period. This ping timeout period comes before an offer, so the time specified has a considerable effect on server performance.

- If you set this time too long, it slows down the lease offering process.
- If you set this time too short, it reduces the effectiveness of the ping packet to detect another client using the IP address.

To implement pinging hosts before offering IP addresses, modify the scope by:

- Enabling the ping-clients attribute. It is disabled by default.
- Setting the ping-timeout attribute. It is 300 milliseconds by default.

The server makes unavailable any IP address for which it receives a successful ECHO reply. You can control this action by enabling the DHCP server attribute ignore-icmp-errors (the preset value). If disabled, the DHCP server also uses ICMP DEST_UNREACHABLE and TTL_EXPIRED error messages that it receives after sending ICMP ECHO requests as reasons for making an IP address unavailable.
Deactivating Leases

Deactivating a lease moves a client off of it. If the lease is available, deactivating it prevents the DHCP server from giving it to a client. If the lease is active (held by a client), deactivating it prevents the client from renewing it and the server from giving the lease to another client. You can deactivate a lease only if the server is running. The DHCP server deactivates the lease immediately.

Tip

To force a Windows client to release its lease, run `ipconfig /release` on the client machine.

Local Basic or Advance Web UI

To deactivate a lease, click the address of the lease on the Leases tab for the Scope (see the “Viewing Leases” section on page 23-2) and click Deactivate. The lease now shows as deactivated. To reactivate the lease, click Activate.

CLI Commands

To deactivate a lease, use `lease ipaddr deactivate`. To reactivate a lease, use `lease ipaddr activate`.

Excluding Leases from Ranges

IP address ranges, by definition, must be contiguous. To exclude a lease from an existing range, you must divide the range into two smaller ones. The new ranges consist of the addresses between the original starting and ending range addresses and the address that you want to exclude.

Caution

If the excluded address currently has an active lease, you should first follow the steps in the “Deactivating Leases” section on page 23-7, otherwise you will get a warning message. Deleting an active lease can result in a duplicate IP address if the deleted address is subsequently reconfigured and then reassigned. Information about that lease will no longer exist after you reload the server.

Local Basic Web UI

To exclude a lease from a scope address range, do the following:

Step 1 From the Design menu, choose Scopes under the DHCPv4 submenu to open the Manage Scopes (Address Pools) page.

Step 2 Click the name of the scope to open the Edit DHCP Scope (Address Pool) page.

Step 3 In the Ranges area, click the Delete icon next to the IP address range you want to remove.

Step 4 Add a range that ends just before the excluded IP address.

Step 5 Add another range that begins just after the excluded IP address.

Step 6 Save the scope.

Step 7 Reload the DHCP server.
Local Advanced Web UI

To exclude a lease from a scope address range, the same operations exist as in Basic mode, except that you click the name of the scope on the List/Add DHCP Scopes page, which opens the Edit DHCP Scope page.

CLI Commands

To exclude a lease from a scope address range, discover the lease range (scope name listRanges), deactivate the lease (lease ipaddr deactivate), then remove the range of just that IP address (scope name removeRange). The resulting ranges are then split appropriately.

The following example removes the 192.168.1.55 address from the range. Note that if the lease is in a scope with a defined VPN, you must explicitly define that VPN for the session, or you can include the VPN prefix in the lease command:

nrcmd> session set current-vpn=red
nrcmd> scope examplescope1 listRanges
nrcmd> lease red/192.168.1.55 deactivate
nrcmd> scope examplescope1 removeRange 192.168.1.55 192.168.1.55
nrcmd> scope examplescope1 listRanges

Searching Server-Wide for Leases

Using Cisco Prime IP Express, you can search for leases, server-wide. The search is a filter mechanism whereby you can specify a combination of lease attributes to target one or more leases configured for the network. The lease history search function is available at both local and regional cluster whereas the active lease search function is available only at the local cluster. The search function is provided separately for DHCPv4 and DHCPv6 leases.

You can also search for the active leases using Cisco Prime IP Express.

Local Advanced Web UI

To search for DHCPv4 leases, do the following:

---

Step 1

From the Operate menu, choose Current Leases from the DHCPv4 drop-down list under the Reports submenu to open the DHCP Lease Search page.

You can also go to the DHCP Lease Search page if you choose Lease History from the DHCPv4 drop-down list under the Reports submenu. If you choose Lease History from the DHCPv4 drop-down list under the Reports submenu, the DHCP Lease History Search page is displayed. You have to click the Search button to go to the DHCP Lease Search page.

Note: You can open the DHCP Lease Search page by clicking the Search button in the DHCP Lease History Search page (choose Lease History from the DHCPv4 drop-down list under the Reports submenu to open the DHCP v4 Lease History Search page). This button helps you to toggle between lease history search page and active leases search page.

Step 2

Choose a Filter Attribute from the drop-down list, such as address. DHCPv4 and DHCPv6 have separate lists of filter attributes. Also, the set of filter attributes are different for active and historical leases. Attributes are greyed out after you select them as elements.
Step 3  Choose a filter Type from the drop-down list. You can choose at least Binary or Regular Expression, but the list can contain one or more of the following, depending on the Filter Attribute selected:

- Binary—Value is in binary notation.
- Date Range—Range of date values, From a date and time To a date and time.
- Integer—Value is an integer.
- Integer Range—Integer From value to an integer To value.
- IP Address—Value is an IP address.
- IP Range—IP address From value to an IP address To value.
- IP Subnet—Value is an IP subnet.
- Regular Expression—Value is a Regular Expression in regex syntax. (For common regex usage, see Table 5-4 on page 5-30).

Step 4  Enter a Value, based on the Type selected. To clear the filter, click Clear Filter.

Step 5  Click Add Element to add the search element to the Filter Elements list. You can delete the element by expanding the filter display, then clicking the Delete icon next to the element.

Step 6  Once you assemble a list of elements, you can search on them, so that the elements are ANDed together for the result. Click Search.

Step 7  Check the table of resulting leases from the search, which shows for each an address, state, MAC address, hostname, flags, and expiration date. If necessary, change the page size to see more entries. The leases are ordered by IP address.

Tip  The filter elements are ANDed together for the search. If you find that the search results do not yield what you expect, look at the Filter Elements list again and delete elements that can obstruct the results.

To search for DHCPv6 leases, do the following:

Step 1  From the Operate menu, choose Current Leases from the DHCPv6 drop-down list under the Reports submenu to open the DHCP v6 Lease Search page.

You can also go to the DHCP v6 Lease Search page if you choose Lease History from the DHCPv6 drop-down list under the Reports submenu. If you choose Lease History from the from the DHCPv6 drop-down list under the Reports submenu, the DHCP v6 Lease History Search page is displayed. You have to click the Search button to go to the DHCP v6 Lease Search page.

Step 2  Choose a Filter Attribute from the drop-down list, such as address.

Step 3  Choose a filter Type from the drop-down list. You can choose at least Binary or Regular Expression, but the list can contain one or more of the following, depending on the Filter Attribute selected:

- Binary—Value is in binary notation.
- Date Range—Range of date values, From a date and time To a date and time.
- Integer—Value is an integer.
- Integer Range—Integer From value to an integer To value.
- IPv6 Address—Value is an IPv6 address.
- IPv6 Prefix—Value is an IPv6 prefix.
Using Client Reservations

For the clients to get the lease they want was to create a lease reservation (see the “Creating Lease Reservations” section on page 23-13). It may not always be easy to create reservations for each client, which may come up to millions of reservations. Also, the process to update and synchronize the Cisco Prime IP Express reservations with databases is very complex. The client reservation feature helps in reducing this complexity.

The current functionality supported by Cisco Prime IP Express DHCP server in assigning an IP address to a DHCPv4 client is as follows:

- If a lease based reservation for the client exists and the lease is available, it is used.
- Otherwise, if the client requested an address and it is available, it is used.

CLI Commands

Use `lease list -macaddr mac-addr [-vpn=vpn-name]` to find leases in the DHCPv4 space. Specify the MAC address of the lease. If you omit the VPN designation, you base the search on the current VPN.

For leases in the DHCPv6 space, use the following `lease6 list` syntax:

```
nrcmd> lease6 list
    [-duid=client-id]
    [-lookup-key=key] [-blob | -string]
    [-macaddr=mac-addr]
    [-cm-macaddr=cm-mac-addr]
    [-vpn=vpn-name]
    [-count-only]
```

The `-macaddr` and `-cm-macaddr` options are to search for leases identified by the CableLabs DOCSIS `vendor-opts` option (DHCPv6 option 17). For example, for these two commands:

```
nrcmd> lease6 -macaddr=01:02:03:04:05:06
nrcmd> lease6 -cm-macaddr=01:02:03:04:05:06
```

The `-macaddr` line lists leases where the option 17 device-id suboption (36) contains the requested MAC address. The `-cm-macaddr` line lists leases where the option 17 cm-mac-address suboption (1026) matches the requested MAC address. (See Table C-4 on page C-7 for details on these suboptions.)
Client reservations feature enables you to supply addresses and delegate prefixes through client entries (either stored directly in Cisco Prime IP Express or in LDAP) or through extensions. Also, a client can be located on more than a single scope or prefix and the server will select the address appropriate to the location of the client.

Client-reserved leases are essentially reserved leases. The major difference is that the client for which the lease is reserved is not known to the server in case of client reservations. Client reservations are used when you want to configure leases for many clients or configure many leases for a single client.

Client reservations can be provided to Cisco Prime IP Express using one of the following three primary mechanisms:

- Using internal client database—This has some of the same issues as with lease reservations, but may be a better option if Cisco Prime IP Express internal client database is already being used for other purposes. The fact that the internal client database has to maintain the client alone and not the reservations makes it more advantageous when compared to lease reservations.
- Using LDAP—Cisco Prime IP Express can look up clients in an LDAP repository (external to Cisco Prime IP Express) and these clients may specify client reservations.
- Using extensions—Cisco Prime IP Express can be set up to communicate with external servers or databases using extensions.

The client entries, maintained either within the Cisco Prime IP Express client database or LDAP, can include the addresses and prefixes a client is supposed to use. The attributes to specify the client reservations are:

1. `reserved-addresses`—Specifies the list of addresses reserved for the client. The first available address to match a usable Scope (which must have restrict-to-reservations enabled) are assigned to the client.
2. `reserved-ip6addresses`—Specifies the list of addresses reserved for the client. All available addresses to match a usable Prefix (which must have restrict-to-reservations enabled) are assigned to the client.
3. `reserved-prefixes`—Specifies the list of prefixes reserved for the client. All available prefixes to match a usable Prefix (which must have restrict-to-reservations enabled) are assigned to the client.

The attribute restrict-to-reservations is added to Scope, Scope template, Prefix, and Prefix template objects to specify the client reservations.

For a client in LDAP, you must set up a mapping between the LDAP attribute name and the corresponding client attribute name.

If the LDAP addresses attribute contained a list of the IPv4 addresses for the client, use `ldap_servername.setEntry query-dictionary ldap-attribute=cnr-client-attribute` to map it to the reserved-addresses attribute. For example:

```
nrcmd> ldap ldap-1 setEntry query-dictionary addresses=reserved-addresses
```

**Local Advanced Web UI**

To restrict a scope to client reservations, do the following:

**Step 1** From the Design menu, choose Scopes under the DHCPv4 submenu to open the List/Add DHCP Scopes page. See the “Creating Scopes” section on page 21-9 to create a scope.

**Step 2** Click enabled for restrict-to-reservations attribute in Miscellaneous Settings group in the List/Add DHCP Scopes page.
To modify an existing scope to specify client reservations, click the required scope name to open the Edit DHCP Scope page. Click enabled for restrict-to-reservations attribute in Miscellaneous Settings group. The flag client-reserved shows that a scope is restricted to client reservations.

**Note**

All the IP addresses configured in a scope which has the restrict-to-reservations attribute enabled will be counted in the active lease count, and will count against the licensed IP addresses for this DHCP server. If you are a heavy user of client reservations, you might want to configure only the addresses that you expect to need in the near term, and increase the size of the scope later if you need more addresses.

To restrict a scope template to client reservations, do the following:

**Step 1** Choose Scope Templates from the Design menu to open the List/Add DHCP Scope Templates page. See the “Creating and Applying Scope Templates” section on page 21-3 to create a scope template.

**Step 2** Click enabled for restrict-to-reservations attribute in Miscellaneous Settings group in the List/Add DHCP Scope Template page.

To modify an existing scope template to specify client reservations, click the required scope template name to open the Edit DHCP Scope Template page. Click enabled for restrict-to-reservations attribute in Miscellaneous Settings group.

To restrict a prefix to client reservations, do the following:

**Step 1** From the Design menu, choose Prefixes under the DHCPv6 submenu to open the List/Add DHCP v6 Prefixes page.

**Step 2** Click the Add Prefixes icon in the Prefixes pane, enter the prefix name and address and click the Add IPv6 Prefix.

**Step 3** Click the prefix name on the Prefixes pane to open the Edit DHCPv6 Prefix page. Click enabled for restrict-to-reservations attribute in Non-Parent Settings group.

**Note**

Prefixes which have the restrict-to-reservations attribute enabled are not counted in the total of active leases which must be licensed. Any client which receives a client reservation will have that active lease counted, but that will happen only when the lease is actually held by a client.

To restrict a prefix template to client reservations, do the following:

**Step 1** To restrict a prefix to client reservations, from the Design menu, choose Prefix Templates under the DHCPv6 submenu to open the List/Add DHCP v6 Prefix Templates page.

**Step 2** Click the Add Prefix Templates icon in the Prefix Templates pane, to open the Add Prefix Template dialog box.

**Step 3** Enter the prefix template name and click the Add Prefix Template button.

**Step 4** Click enabled for restrict-to-reservations attribute.
To modify an existing prefix template to specify client reservations, click the prefix template name that you want to restrict to client reservations. Click enabled for restrict-to-reservations attribute.

**Differences Between Client Reservations And Lease Reservations**

Client reservations have the following significant differences over lease reservations:

- There is no validation to assure that there is only a single client reservation for any address. If the external database assigns the same address to two different clients, whichever client request arrives first is granted that lease.

- A client reservation really exists only after the client completes DHCP configuration. Lease reservations are known even if a client transaction never occurs and thus can also be used for clients that do not provide DHCP services at all.

Cisco Prime IP Express 8.2 supports:

- Creating a lease reservation for a particular IP address.
- Configuring the correct cable modem MAC address for the IP address such that Cable Source Verify will work correctly with a Cable Modem Termination System (CMTS).

This works because the Cisco Prime IP Express DHCP server knows about the lease reservation before any DHCP client transaction and will respond correctly to a leasequery request from a CMTS for those addresses. Client reservations are, in contrast, not known to the DHCP server before the arrival of a DHCP client packet at the DHCP server. A leasequery for an IP address which is configured as client-reserved due to some client registration will not (in general) know that the IP address is client reserved.

Thus, any leasequery to which the DHCP server is supposed to respond with a positive result that includes the proper cable modem MAC address, even when no client has actively requested the lease, will not work with client reservations.

**Creating Lease Reservations**

To ensure that a client always gets the same lease, you can create a lease reservation. Managing lease reservations is available only to administrators having the dhcp-admin role at the local cluster, or the central-cfg-admin role with the dhcp-management subrole at the regional cluster.

You can query DHCPv4 and DHCPv6 reservations from the server.

**Note**

All lease reservations are counted in the total of active leases that is compared to the number of IP addresses licensed.

**DHCPv4 Reservations**

When the DHCP edit mode is synchronous, reservation changes are automatically forwarded to the DHCP server, and take immediate effect.
When the edit mode is staged, any change you make to the reservation list on a local cluster modifies the parent scope to indicate that a server reload is required. Any change to the regional reservation list modifies the parent subnet.

**Local Basic Web UI**

To view lease reservations, from the Design menu, choose Scopes under the DHCPv4 submenu to open the Manage Scopes (Address Pools) page, then click the Reservations tab.

To create a reservation on this page, enter the IP address you want to reserve for lease, and enter a lookup key in the Lookup Key field. Click the MAC address (the default) or string or binary radio button, as appropriate for the lookup key entry. Click Add Reservation. The lease IP address, Lookup Key and Scope details are displayed in the List/Add DHCP Reservations page.

**Local Advanced Web UI**

To view the lease reservations for DHCPv4 scopes, from the Design menu, choose Scopes under the DHCPv4 submenu to open the List/Add DHCP Scopes page. Proceed as for the Basic web UI.

Advanced mode also provides a mechanism to create reservations independent of scopes. To configure reservations directly for DHCPv4 scopes, do the following:

1. From the Design menu, choose Reservations to open the List/Add DHCP Reservations page.
2. Click the Add DHCP Reservations icon in the Reservations pane, enter the IP address you want to reserve for lease, and enter a lookup key in the Lookup Key field, then click Add Reservation.
3. Click the MAC address (the default) or string or binary radio button, as appropriate for the lookup key entry. Click Save.

**Tip**

You can use a filter to reduce the size of the list that is displayed. To do this, choose a filter type from the Filter Type drop-down list. The Filter Value is set as for the selection of the Filter Type. Click Set Filter. To set Filter Type as None, click Clear Filter. The lease IP address, Lookup Key and Scope details are displayed in the List/Add DHCP Reservations page.

**Note**

Multiple DHCP servers should not distribute IP addresses on the same subnet, unless they are DHCP Failover partners. When using Failover, the client reservations must be identical on each server. If not, a client for whom a lease reservation exists can receive offers of different IP addresses from different servers. The Failover synchronization function helps you assure that the partner configuration is consistent.

### Setting Advanced Lease and Reservation Properties

Setting advanced lease and reservation properties can include:

- **Reserving currently leased IP addresses**—See the “Reserving Currently Leased Addresses” section on page 23-15.

- **Unreserving leases**—See the “Unreserving Leases” section on page 23-16.
• **Extending leases to non-MAC addresses**—See the “Extending Reservations to Non-MAC Addresses” section on page 23-17.

• **Forcing lease availability**—See the “Forcing Lease Availability” section on page 23-18.

• **Inhibiting lease renewals**—See the “Inhibiting Lease Renewals” section on page 23-19.

• **Handling leases marked as unavailable**—See the “Handling Leases Marked as Unavailable” section on page 23-20.

• **Setting timeouts for unavailable leases**—See the “Setting Timeouts for Unavailable Leases” section on page 23-21.

### Reserving Currently Leased Addresses

You can delete a reservation for one client while reusing it for another one, even though the first client still has the lease.

**Local Advanced Web UI**

To reserve an existing lease, do the following:

**Step 1**
From the **Design** menu, choose **Scopes** under the DHCPv4 submenu, then select the name of the scope to open the Edit DHCP Scope page.

**Step 2**
Click the Leases tab.

**Step 3**
Click the IP address of the lease on the List DHCP Leases for Scope page.

**Step 4**
On the Manage DHCP Lease page, if the IP address is not leased (in available state), enter the lookup key or MAC address for the reservation.

**Step 5**
Click **Make Reservation**. On the List DHCP Leases for Scope page, the lease will appear as reserved.

**Step 6**
Save the scope.

**Step 7**
To remove the reservation, click **Remove Reservation** on the Manage DHCP Lease page, then modify the scope. The lease no longer appears as reserved.

### Example of Reserving an Existing Lease

This CLI command example creates a reservation from an existing lease. It assumes that the dhcp-edit-mode has been set to synchronous to allow the reservations to be added to the server dynamically:

```
nrcmd> reservation 192.168.1.110 create 1,6,00:d0:ba:d3:bd:3b
nrcmd> lease 192.168.1.110 activate
```

Client 1,6,00:d0:ba:d3:bd:3b does a DHCPDISCOVER and gets an offer for 192.168.96.110. The client then does a DHCPREQUEST and gets an ACK message for the same IP address.

As time passes, client 1,6,00:d0:ba:d3:bd:3b does several DHCPREQUESTs that are renewals, which the server acknowledges. Then, at some time before the client lease expiration time, you terminate the reservation:

```
nrcmd> lease 192.168.1.110 deactivate
nrcmd> reservation 192.168.1.110 delete
```
You then add a reservation for a different client for that IP address, even though the address is still leased to the first client:

```
nrcmd> reservation 192.168.1.110 create 1,6,02:01:02:01:02:01
nrcmd> lease 192.168.1.110 activate
```

This action results in an IP address that is leased to one client, but reserved for another. If the new client (1,6,02:01:02:01:02:01) does a DHCPDISCOVER before the original client (1,6,00:d0:ba:d3:bd:3b) does, the new client does not get 192.168.96.110, but gets a random IP address from the dynamic pool.

When the original client (1,6,00:d0:ba:d3:bd:3b) sends its next DHCPREQUEST/RENEW for the lease on 192.168.96.110, it gets a NAK message. Generally, upon receipt of the not-acknowledged message, the client immediately sends a DHCPDISCOVER. On receiving that DHCPDISCOVER, the server cancels the remaining lease time for 192.168.96.110.

The server then gives client 1,6,00:d0:ba:d3:bd:3b whatever lease is appropriate for it—some reservation other than 192.168.96.110, some dynamic lease (if one is available), or nothing (if no dynamic leases are available). When the new client (1,6,02:01:02:01:02:01) tries to renew the random IP address it received, the server sends it a NAK, because it wants to give it the reserved address. When the new client then does a DHCPDISCOVER, it gets the 192.168.96.110 reserved address.

You could also force availability of a lease (see the “Forcing Lease Availability” section on page 23-18). However, doing so does not stop the original client (1,6,00:d0:ba:d3:bd:3b) from using 192.168.96.110. Also, it does not prevent the new client (1,6,02:01:02:01:02:01) from getting 192.168.96.110. In other words, this means that making a reservation for a client is independent of the lease state (and actual lease client) of the IP address for which the reservation is made.

Thus, making a reservation for one client does not cause another client to lose that lease right away, although that client receives a NAK response the next time it contacts the DHCP server (which could be seconds or days). Additionally, the client that reserved the IP address does not get that address if some other client already has it. Instead, it gets another IP address until the:

- IP address it is supposed to receive is free.
- Client sends a DHCPREQUEST as a renewal and receives a NAK response.
- Client sends a DHCPDISCOVER.

### Unreserving Leases

You can remove lease reservations at any time. However, if the lease is still active, the client continues to use the lease until it expires. If you try to reserve the lease for a different client, you will get a warning.

Once you delete the last reservation from regional, it is not possible to select the reservation and push the change to the local cluster. You must push the parent subnet, which will then synchronize the reservation list and thus delete the local copy of the reservation.

There is no push function for DHCPv6 reservations in regional. You always need to push the parent prefix to resynchronize the reservations. This is the preferred method when synchronizing regional delete actions.

### Local Advanced Web UI

To unreserve a lease, choose **Reservations** from the **Design** menu to open the List/Add DHCP Reservations page, then click the **Delete** icon after selecting the reservation you want to remove. This removes the reservation immediately, with no confirmation.
CLI Commands

To unreserve a lease, use `reservation [vpn]/ipaddr delete` or `scope name removeReservation [ipaddr | macaddr | lookupkey] [–mac | –blob | –string]`. However:

- Ensure that the reservation is gone from the `nrcmd` internal database.
- If you use failover on the scope containing the reservation:
  1. Use `reservation [vpn]/ipaddr delete`, or `scope name removeReservation`, on both servers.
  2. On the backup server, if you are in staged dhcp edit mode, use `lease [vpn]/ipaddr delete-reservation`.
  3. Use the same command on the main server.

Save the result of this operation to preserve it across server reloads, because issuing `lease ipaddr delete-reservation` alone affects only the server internal memory.

Extending Reservations to Non-MAC Addresses

You might need to create lease reservations based on something other than the MAC address from the incoming client packet. Often, DHCP client devices attached to a switch port need to get the same IP address, regardless of the MAC address. This approach helps when you replace factory floor devices with identical devices (with different MAC addresses), but want to maintain the same IP address.

Overriding Client IDs

You can set an expression in a client-class `override-client-id` attribute that extracts the MAC address and port of a switch from the relay-agent-info option (82) and creates a client identity from it. Regardless of the client-id in the incoming packet, the identity that allocates an IP address is the same for any device coming in through the same switch port. The expression you use for the attribute depends on the option 82 format. The DHCP server calculates the expression when it assigns the packet to the client-class. The `override-client-id` value becomes the identity of the client from that point onward.

---

**Note**

When using `/v6-override-client-id` expressions, leasequery by client-id requests may need to specify the `override-client-id` attribute to correctly retrieve the information on the lease(s) for the client.

However, when you enable the `use-client-id-for-reservations` attribute in a policy, the server turns the client-id of that request into a string of the form `nn:nn:nn ... nn:nn`, and uses that string to look up the reservation.

The `add-to-environment-dictionary` attribute for a client or client-class also serves to send attribute values to the DHCP extension environment dictionary (see Chapter 30, “Using Extension Points”), specified as name-value pairs. You can configure an `add-to-environment-dictionary` attribute on either a client or a client-class. If you choose to configure this attribute on both a client and client-class, you should ensure that the name-value pairs that you configure on the client have different names than the name-value pairs that you configure on the client-class, because they are all going to be put into the same environment dictionary (which can have only one value for a particular name). Generally, it is best to configure this attribute on a client or a client-class only, but not on both.
Local Advanced Web UI

You can find the override-client-id attribute on the Add DHCP Client-Class page (from the Design menu, choose Client-Class, then Add Client-Class) or Edit DHCP Client-Class page (from the Design menu, choose Client-Class, then the name of the client-class).

You also need to configure a client-class lookup ID for the DHCP server, to put every packet into a particular client-class where you configure the override-client-id expression. From the DHCP menu, choose DHCP Server, then the Local DHCP Server link to open the Edit DHCP Server page. In the Client Class attributes, enter a client-class-lookup-id expression.

To use the client ID for the reservation, configure the policy to enable the use-client-id-for-reservations attribute on the Add DHCP Policy page (from the Design menu, choose Policies, then Add Policy) or Edit DHCP Policy page (from the Design menu, choose Policies, then the name of the policy).

CLI Commands

The syntax for setting the override-client-id attribute is client-class name set override-client-id="expression". The syntax for setting the client-class-lookup-id attribute is dhcp set client-class-lookup-id="expression". The syntax for setting the use-client-id-for-reservations attribute is policy name enable use-client-id-for-reservations.

Reservation Override Example

The following example shows how to override a client ID for a reservation:

Step 1 Create a scope for the reservation:
  a. Enter a subnet address.
  b. If you want dynamic reservations, add an IP address range.

Step 2 Add the reservation for the scope:
  a. Include a value for the lookup key.
  b. Specify the lookup key type as binary.

Step 3 Create a policy for the purpose, enabling the use-client-id-reservations attribute.

Step 4 Create a client-class for the purpose:
  a. Specify the policy created in the previous step.
  b. Include an expression for the override-client-id attribute that returns a blob value with the client ID you want, based on the contents of the packet.

Step 5 Get a lease for a client with the MAC address. This client will then get the override ID.

Forcing Lease Availability

You can force a current lease to become available. You should request that the user release the lease, or do so yourself, before forcing its availability. Forcing lease availability does not require a server reload.

Note

After a lease is forced to be available, the client continues to use it until the client contacts the DHCP server.
Local Advanced Web UI

To force lease availability, do the following:

Step 1  From the Design menu, choose Scopes to open the List/Add DHCP Scopes page.
Step 2  Click the Lease tab for the scope that has leases.
Step 3  Click the IP address of the lease on the List DHCP Leases for Scope page.
Step 4  On the Manage DHCP Lease page, click Force Available. On the List DHCP Leases for Scope page, the lease will now show an empty value in the Flags column.

CLI Commands

To force lease availability, use lease [vpn/]ipaddr force-available. Use scope name clearUnavailable to force all leases in the scope to become available.

Inhibiting Lease Renewals

Normally, the Cisco Prime IP Express DHCP server retains the association between a client and its leased IP address. The DHCP protocol explicitly recommends this association and it is a usually desirable feature. However, for some customers, such as ISPs, clients with long-lived lease associations may be undesirable, because these clients should change their IP addresses periodically. Cisco Prime IP Express includes a feature that allows customers to force lease associations to change when DHCP clients attempt to renew their leases or reboot.

A server can never force a client to change its lease, but can compel the client to do so based on a DHCPRENEW or DHCPDISCOVER request. Cisco Prime IP Express offers configuration options to allow customers to choose which interactions to use to force a client to change its IP address:

- **Inhibiting all lease renewals**—While a client is using a leased address, it periodically tries to extend its lease. At each renewal attempt, the server can reject the lease, forcing the client to stop using the IP address. The client might have active connections that are terminated when the lease terminates, so that renewal inhibition at this point in the DHCP interaction is likely to be user-visible.

- **Inhibiting renewals at reboot**—When a DHCP client reboots, it might have recorded a valid lease binding that did not expire, or it might not have a valid lease. If it does not have a lease, you can prevent the server from granting the last held lease. If the client has a valid lease, the server rejects it, forcing the client to obtain a new one. In either case, no active connections can use the leased address, so that the inhibition does not have a visible impact.

- **Effect on reservations**—Reservations take precedence over renewal inhibition. If a client has a reservation, it can continue to use the reserved IP address, whether or not renewal inhibition is configured.

- **Effect on client-classes**—Client-class testing takes place after renewal inhibition testing. If a client is forced to change IP addresses by renewal inhibition, then client-class processing might influence which address the server offers to the client.
You can enable or disable lease renewal inhibition for a policy, which you can set system wide, for a scope or on a client-by-client basis. The `inhibit-all-renews` attribute causes the server to reject all renewal requests, forcing the client to obtain a new IP address any time it contacts the DHCP server. The `inhibit-renews-at-reboot` attribute permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

The DHCP server needs to distinguish between a client message that it should reject (such as a renewal request) and one that represents a retransmission. When the server processes a message, it records the time the packet arrived. It also records the time at which it made a lease binding to a client, and the last time it processed a message from the client about that binding. It then compares the packet arrival time with the lease binding time (the start-time-of-state) and processes packets from the client within a certain time interval from the start time of the binding. By default, this time interval is one minute.

**Local Advanced Web UI**

To inhibit lease renewals, create a policy on the Edit DHCP Policy page (click **Design**, then **Policies**, then the name of the policy), then enable the `inhibit-all-renews` or `inhibit-renews-at-reboot` attribute. (Both attributes are preset to disabled). Then, modify the policy.

**Handling Leases Marked as Unavailable**

One of the aspects of effective lease maintenance is determining the number of unavailable leases in a scope. This number is sometimes higher than expected. Each unavailable lease is probably an indication of a serious problem. Possible causes for an unavailable lease are:

- **The DHCP server is configured for a ping before an offer, and the ICMP echo message is returned successfully**—A currently active client is using that IP address, causing the DHCP server to mark it as **unavailable**. To prevent the server from doing so, disable pinging an address before offering it to a client. See the “Pinging Hosts Before Offering Addresses” section on page 23-6.

- **The server receives a DHCPDECLINE message from a client to which it leased what it considered to be a good IP address**—The client does an address resolution (ARP) request for the IP address on its local LAN segment, and another client responds to it. The client then returns the address to the server with a DHCPDECLINE packet and sends another DHCPDISCOVER packet to get a new address. The server marks as **unavailable** the address that the client returns. To prevent the server from reacting to DHCPDECLINE messages, you can set a scope attribute, `ignore-declines`.

- **The server receives “other server” requests from the client**—Because all DHCPREQUEST messages that follow DHCPOFFER messages are broadcast, the server can see messages directed to other DHCP servers. A server knows that a message is directed to it by the value of the `server-id` option in the packet. If the Cisco Prime IP Express server recognizes a message directed at another server, in that its own IP address does not appear in the `server-id` option, but the address leased in the message is one that the server controls, it believes that two servers must be trying to manage the address simultaneously. It then marks the local address as **unavailable**. This behavior does not apply in a DHCP failover configuration. Either the two servers are configured with some or all of the same IP addresses, or (in rare cases) the DHCP client placed a wrong `server-id` option value in the packet.

  If you have reason to believe that the client is sending bad `server-id` options (rather than packets actually directed to other servers), Cisco Prime IP Express has a server attribute you can enable that turns this behavior off, the `ignore-requests-for-other-servers` attribute.

- **Inconsistent lease data**—Extremely rare and occurring only during server startup when, while configuring a lease, the server reads the lease data from disk during a refresh of the internal cache. The lease state appears as **leased**, but there is incomplete data to construct a client for that lease,
such as that the lease might not yet have a client-id option value. The server considers the data to be inconsistent and marks the IP address as unavailable. Forcing the lease to be available (such as by using the lease ipaddr force-available command in the CLI) should clear up this problem.

**Setting Timeouts for Unavailable Leases**

During the times when leases become unavailable, as described in the “Handling Leases Marked as Unavailable” section on page 23-20, all unavailable leases remain in that state for a configured time only, after which time they again become available. A policy attribute, unavailable-timeout, controls this time. The system_default_policy policy sets this value to one day by default.

**Running Address and Lease Reports**

You can run these reports on IP addresses and leases:

- **Address Usage**—See the “Running Address Usage Reports” section on page 23-21.
- **Lease History**—See the “Running IP Lease Histories” section on page 23-21.
- **Current Utilization**—See the “Running Lease Utilization Reports” section on page 24-28.
- **Lease Notification**—See the “Receiving Lease Notification” section on page 23-27.

**Running Address Usage Reports**

The address usage reports show the IP addresses that are assigned leases.

**Local Advanced Web UI**

To view the leases for IP addresses, on the Edit DHCP Scope page, click the Leases tab, to open the List DHCP Leases for the scope. To manage a specific lease, click its IP address on the page. This opens the Manage DHCP Lease page.

**CLI Commands**

To view the IP address usage for specified servers, use report.

<table>
<thead>
<tr>
<th>Tip</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you are not already using lease-notification in an automated way, try lease-notification available=100% for a concise scope-by-scope summary of the state of the servers.</td>
</tr>
</tbody>
</table>

**Running IP Lease Histories**

You can extract IP lease history data from a special database so that you can determine past allocation information for a given IP address. You can get a historical view of when a client was issued a lease, for how long, when the client or server released the lease before it expired, and if and when the server renewed the lease and for how long.
Chapter 23      Managing Leases

Running Address and Lease Reports

Cisco Prime IP Express provides a client to control querying IP history data. Through this client, you can:

- Get the MAC addresses associated with a given IP address over a given time.
- See the entire IP history database as a comma-separated file.
- View the attributes of the lease history (the lease history detail report)—See the “Querying IP Lease History” section on page 23-23.

You must use additional administrative functions to trim the IP history database of records, to keep the size of the database from growing without bounds.

Note When the state of an existing lease changes (for example, when it is configured as a reserved IP address or it is deactivated), the change does not appear as a lease history change at the regional. With detail collection disabled, a lease history change appears only when the lease transitions from leased to not leased or is assigned to another client.

Related Topics

Enabling Lease History Recording at the Local Cluster, page 23-22
Querying IP Lease History, page 23-23
Trimming Lease History Data, page 23-26

Enabling Lease History Recording at the Local Cluster

You must explicitly enable lease history recording for the local cluster DHCP server. The DHCP server logs IP history recording errors in the usual DHCP log files.

When the lease history is enabled on a local cluster it impacts the performance of the server and the size of the lease state database. A history record is created for the lease whenever a lease ends (expires or is released); a lease that a client renews over a long period does not cause a history record to be created. The size of each lease history record depends on many factors, but a good estimate is about 1 KB per record. Depending on the rate at which the lease ends and the duration over which lease history is kept, this could result in a sizeable number of lease history records being created and thus requires a considerable disk space. This could be many orders larger than the space needed for the active leases.

Local Advanced Web UI

To enable lease history recording, do the following:

**Step 1** From the **Deploy** menu, choose **DHCP Server** under the DHCP submenu to open the Manage DHCP Server page.

**Step 2** Click the **Local DHCP Server** on the DHCP Server pane.

**Step 3** On the Edit DHCP Server page, look for the **Lease History** attributes:

- **Lease History (ip-history)**—Enable or disable the lease history database for v4-only (DHCPv4), v6-only (DHCPv6), or both.
- **ip-history-max-age**—Maximum age of the lease history to collect. With lease history set to v4 only, v6 only, or both the DHCP server periodically examines the lease history records and deletes any records with lease history bindings older than this age threshold.

**Step 4** Click **Save**.
Step 5  Reload the server.

**CLI Commands**

To enable lease history recording, you must explicitly enable recording IP (lease) history for IP addresses by using `dhcp set ip-history=<value>` (v4-only, v6-only, both, or disable).

**Querying IP Lease History**

Once you have leases, you can query for their history. You can query IP lease history either from a local or a regional cluster. Set up the local cluster containing the DHCP server as part of the regional cluster, and enable polling for the lease history data from the regional cluster (see the “Enabling Lease History Collection” section on page 6-14).

You can adjust the polling criteria for the cluster in the regional cluster web UI by using the attributes described in the “Polling Lease History Data” section on page 6-13.

You must also set the selection criteria for querying the lease history data, as described in the following sections.

**Local and Regional Advanced Web UI**

To query the IPv4 lease history, do the following:

- **Step 1**  From the Operate menu, choose Lease History from the DHCPv4 drop-down list under the Reports submenu to open the DHCP Lease History Search page.

  **Note**  You can use the Search button in the Local Advanced Web UI to move to DHCP Lease Search page. This button helps you to toggle between lease history search page and active leases search page.

- **Step 2**  Choose the Filter attribute and the Type from the drop down lists and enter the value of the filter type selected in the Value field.

- **Step 3**  Click Search to display the list of leases.

To query the IPv6 lease history, do the following:

- **Step 1**  From the Operate menu, choose Lease History from the DHCPv6 drop-down list to open the DHCP v6 Lease History Search page.

  **Note**  You can use the Search button in the Local Advanced Web UI to move to DHCP v6 Lease Search page. This button helps you to toggle between lease history search page and active leases search page.

- **Step 2**  Choose the Filter attribute and the Type from the drop down lists and enter the value of the filter type selected in the Value field.

- **Step 3**  Click Search to display the list of leases.
The regional server only searches its version of the lease history which is as recent as the latest poll. For the most up-to-date data, this might require performing an explicit lease history poll for the regional to retrieve the latest lease history data.

**Using the iphist Utility**

You can query the IP history database at the local as well as regional clusters and direct the results to standard output or a file by using the `iphist` utility. You must run this utility on the same machine as the DHCP server, and you must have superuser/root privileges to read and modify the database file. The default location is:

- **Windows**—\Program Files\IP Express\bin
- **Linux**—/opt/nwreg2/local/usrbin

From the command prompt, change to the above location and run the utility using the syntax:

```
iphist [options] [ipaddr | all] [start-date | start [end-date | end]]
```

The IP address is a single address or the keyword `all`, the start date is in local time or the keyword `start` for the earliest date in the database, and the end date is in local time or the keyword `end` for the last date in the database. However, the output is in Greenwich Mean Time (GMT) by default, unless you use the `-l` option to specify local time.

The full list of command options appears in **Table 23-2**.

**Table 23-2  iphist Command Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-N username</code></td>
<td>Administrator username. If omitted, you are prompted for the username.</td>
</tr>
<tr>
<td><code>-P password</code></td>
<td>Administrator password. If omitted, you are prompted for the password.</td>
</tr>
<tr>
<td><code>-C cluster[:port]</code></td>
<td>Destination server and optional SCP port.</td>
</tr>
<tr>
<td><code>-6</code></td>
<td>Output DHCPv6 leases</td>
</tr>
<tr>
<td><code>-a</code></td>
<td>Shows the lease attributes, visibility 3.</td>
</tr>
<tr>
<td><code>-f &quot;format&quot;</code></td>
<td>Format of the output lines. The default format is: &quot;address,client-mac-addr,binding-start-time,binding-end-time&quot;</td>
</tr>
<tr>
<td><code>-t</code></td>
<td>Print format as title line.</td>
</tr>
<tr>
<td><code>-n &quot;namespace&quot;</code></td>
<td>Specify the namespace for the address.</td>
</tr>
<tr>
<td><code>-o file</code></td>
<td>Sends output to a file.</td>
</tr>
<tr>
<td><code>-l</code></td>
<td>Displays output in local time rather than the default GMT.</td>
</tr>
<tr>
<td><code>-i</code></td>
<td>Displays output for delegated prefix that includes specified IPv6 address (only with <code>-6</code>).</td>
</tr>
<tr>
<td><code>-s</code></td>
<td>Restricts the leases to the self or partner.</td>
</tr>
<tr>
<td><code>-v</code></td>
<td>Displays the database version.</td>
</tr>
<tr>
<td><code>-z debug-args</code></td>
<td>Sets the debug output levels.</td>
</tr>
</tbody>
</table>

Dates can use this syntax (quotation marks are required if space characters are included):

- `month/day/year@hour:min:sec` (for example, 8/28/2007@10:01:15), with the time optional
Running Address and Lease Reports

Chapter 23  Managing Leases

- **month/day/year hour:min:sec** (for example, “8/28/2007 10:01:15”), with the time optional
- **month day hour:min:sec year** (for example, “Aug 28 10:01:15 2007”), with the seconds optional
- Keywords **start**, **end**, or **now** (for the current time)

The date filtering is intended to limit the output to leases that were active during that time. This means that they can begin before the specified start date, as long as they do not end before the start date. They can also not begin after the specified end date. For example, invoking the command:

```
# ./iphist -N user -P password all "Aug 28 00:00 2008" "Dec 31 23:59:59 2008"
```

for the following leases:

- Lease 1  Begin  Jan 01 2008  End  Jun 30 2008
- Lease 2  Begin  Mar 10 2008  End  Sep 01 2008
- Lease 3  Begin  Jun 01 2008  End  Sep 30 2008
- Lease 4  Begin  Jan 01 2009  End  Mar 10 2009

would return just Lease 2 and Lease 3, because they both end after the specified start date of the query, even though they both begin before that date. The other two are out of range, because they either end before the specified start date or begin after the specified end date of the query.

The values on each line depend on the specific lease object that the DHCP server stores. You can specify the values to include using the **iphist –f** format command.

The **format** argument is a list of lease attribute names, enclosed in quotation marks with the names separated by commas, that provides the template for the output lines. The default output is **ipaddress,client-mac-addr,binding-start-time,binding-end-time**.

For example:

```
# ./iphist -f "address,client-mac-addr,binding-start-time,binding-end-time" all
```

The output is a sequence of lines terminated with a newline sequence appropriate to the operating system (\n on UNIX or \r\n on Windows). Each line contains data on a single lease record. The format of the lines is generally comma-separated values enclosed in quotation marks. To use a literal backslash (\) or quotation mark (") inside quotation marks, precede each with a single backslash (\). New lines in attributes are printed as \n.

**Table 23-3** lists some of the common lease object attributes you can include in the output. Also, see the help for the lease command. To get a full list, use **iphist -a**.

**Table 23-3  IP History Query Output Attributes**

<table>
<thead>
<tr>
<th>Lease Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>IP address of the lease.</td>
</tr>
<tr>
<td>binding-start-time</td>
<td>Start time of the lease binding.</td>
</tr>
<tr>
<td>binding-end-time</td>
<td>End time of the lease binding.</td>
</tr>
<tr>
<td>client-binary-client-id</td>
<td>Binary form of the client MAC address.</td>
</tr>
<tr>
<td>client-dns-name</td>
<td>Latest DNS name of the client known by the DHCP server.</td>
</tr>
<tr>
<td>client-domain-name</td>
<td>Domain where the client resides.</td>
</tr>
<tr>
<td>client-flags</td>
<td>A number of client flags.</td>
</tr>
<tr>
<td>client-host-name</td>
<td>Hostname that the client requested.</td>
</tr>
<tr>
<td>client-id</td>
<td>Client ID requested by or synthesized for the client.</td>
</tr>
</tbody>
</table>
Running Address and Lease Reports

Chapter 23      Managing Leases

Trimming Lease History Data

If you enabled IP history trimming at the regional cluster, the IP history database is automatically trimmed so that you can reclaim disk space. Each history record has an expiration time. Trimming is necessary for the DHCP server itself, as well as for the CCM regional server that polls the DHCP server for history data.

The CCM server performs background trimming at the regional cluster, which trims off the lease history data older than a certain age at regular intervals. The trimming interval is set by default to 24 hours, and the age (how far back to go in time before trimming) to 24 weeks. The DHCP server at the local cluster performs daily automatic trimming (at 3:00 A.M. local time), and stores four weeks of data by default.

Regional Web UI

To trim lease history data, you must be a central configuration administrator:

Step 1  From the Operate menu, choose Manage Servers under the Servers submenu to open the Manage Servers page.

Step 2  Click the Local CCM Server on the Manage Servers pane.

Step 3  On the Edit Local CCM Server page, under Lease History Settings, set the following attributes (you can use the s, m, h, d, w, or y suffix with values you enter):

- **trim-lease-hist-interval**—How often to trim the old lease history data automatically, the default being daily. If set to 0, no automatic lease trimming occurs, which is not recommended due to the increasing disk space used. The bounded values are 0 to one year.
• **trim-lease-hist-age**—Provided that the *trim-lease-hist-interval* is not set to 0, how far back in time to trim the old lease history data automatically, the default being 24 weeks. The bounded values are one day to one year.

**Step 4** To force immediate trimming, at the bottom of the page find the Trim/Compact Inputs section (compacting is available only for subnet utilization data). Set the Trim/Compact age to a desired value. This age is how far in time to go back to trim the lease history data. There are no bounds to this value. However, if you set a very small value (such as 1m), it trims or compacts very recent data, which can be undesirable. In fact, if you set it to zero, you lose all of the collected data. Setting the value too high (such as 10y) may end up not trimming or compacting any data.

**Step 5** If you are trimming immediately, click **Trim All Lease History**.

You can adjust the trimming that the DHCP server itself performs by setting the *ip-history-max-age* attribute. If *ip-history* is set, the DHCP server accumulates database records over time as lease bindings change. This parameter establishes a limit on the age of the history records kept in the database. The server periodically examines the lease history records, establishes an age threshold based on this parameter, and deletes any records that represent bindings that ended before the threshold. The preset value is four weeks.

---

**Receiving Lease Notification**

The CLI provides the feature of sending notifications if the number of available IP addresses equals or falls below a certain threshold. The *lease-notification* command specifies, through an *available* attribute, when the notification should occur if the number of available leases reaches or falls below a certain threshold. You can e-mail the report to a user. Although you can use the command interactively, its primary use is in an automated procedure such as a UNIX *cron* task or Windows Scheduled Task.

The following example sets up lease notification for examplescope for when its free addresses fall to 10%. It sends the report to recipients billy, joe, and jane, on a specific Windows mail host:

```
nrcmd> lease-notification available=10% scopes=examplescope recipients=billy,joe,jane mail-host=mailhost
```

The output consists of an explanatory header, a table containing a row for each scope in which the number of free addresses is equal to or less than the threshold, and possible warnings related to the scopes and clusters requested.

Cisco Prime IP Express uses the default cluster and the .nrconfig file by default, unless you specify otherwise. For the command syntax, see the help for the *lease-notification* command.

---

**Related Topics**

- Running Lease Notification Automatically in Linux, page 23-28
- Running Lease Notification Automatically in Windows, page 23-28
- Specifying Configuration Files for Lease Notification, page 23-29
Running Lease Notification Automatically in Linux

You can run `lease-notification` periodically by means of the `cron(1)` command by supplying `crontab(1)` with the command to run.

This example, specified to `crontab`, runs `lease-notification` at 00:15 and 12:15 (15 minutes after midnight and noon), Monday through Friday (note that this encompasses a single command line):

```
15 0,12 * * 1-5 . .profile; /opt/nwreg2/local/usrbin/nrcmd lease-notification available=10\% config=/home/jsmith/.nrconfig addresses=192.32.1.0-192.32.128.0 recipients=jsmith,jdoe@example.com >/dev/null 2>&1
```

You can perform `crontab` editing by running the UNIX `crontab -e` command. Set your EDITOR environment variable before running the command, unless you want to use `ed(1)`. See the `crontab(1)` man page for additional details.

Note that you must supply the full path of the CLI command on the `crontab` command line. You can determine the full path in your environment with the UNIX `which nrcmd` command.

Also, when you run the `lease-notification` command by means of `crontab`, the `nrcmd` command ignores the user environment variables CNR_CLUSTER, CNR_NAME, and CNR_PASSWORD. Because other viewers can view the command being run, do not provide the password through the `--P` option on the command line, for security reasons.

Supply the cluster name, user, and password information for the cluster you want the `nrcmd` command to run from in a `.profile` or other file in the home directory of the user running `crontab -e`. For example:

```
CNR_CLUSTER=host1
export CNR_CLUSTER
CNR_NAME=admin1
export CNR_NAME
CNR_PASSWORD=passwd1
export CNR_PASSWORD
```

The `.profile` specification in the `crontab` entry explicitly reads the file. The first dot (.) is the shell command that reads the file and you must follow it with at least one space character. For notification on a different cluster (or clusters) than where `nrcmd` is running, specify this information:

- Clusters to check in a config file (see the “Specifying Configuration Files for Lease Notification” section on page 23-29).
- Fully specified path as in the sample `crontab` entry at the beginning of this section.

You can prevent others from examining or changing the contents of the `.profile` and the configuration file that you create by changing its permissions with the `chmod go-rwx config-file` UNIX command.

Running Lease Notification Automatically in Windows

Use the Scheduled Tasks service available in Windows Explorer under My Computer to schedule the `lease-notification` command. If you do not find a Scheduled Tasks folder under My Computer, you need to add this optional component from Microsoft Internet Explorer 4.0 or later, or use some third-party task scheduler. You can also use the `at` command to schedule the `nrcmd lease-notification` command.

Put multiple entries in the `at` queue, one for each time of day at which you want to run the job.
Specifying Configuration Files for Lease Notification

If you omit a configuration file, `lease-notification` looks for a default .nrconfig file in your current directory, then in your home directory, and finally in the CNR_INSTALL_PATH/conf directory. Cisco Prime IP Express uses the first file it encounters. Each line of the file must either begin with the character # (comment), a section header enclosed in square brackets, or a parameter/value pair or its continuation. Cisco Prime IP Express strips leading space characters from each line and ignores blank lines.

Querying Leases

Cisco Prime IP Express provides enhanced provisioning capabilities. This function is described in the DHCP Leasequery specification (RFC 4388), with which Cisco Prime IP Express conforms. Part of the implementation of the Cisco uBR access concentrator relay agent is to capture and glean information from DHCP lease requests and responses. It uses this information to:

- Associate subscriber cable modem and client MAC addresses with server-assigned IP addresses.
- Verify source IP addresses in upstream datagrams.
- Encrypt unicast downstream traffic through the DOCSIS Baseline Privacy protocol.
- Avoid broadcasting downstream Address Resolution Protocol (ARP) requests, which can burden the uBR as well as the subscriber hosts, and which malicious clients can compromise.

The uBR device does not capture all DHCP state information through gleaning. The uBR device cannot glean from unicast messages (particularly renewals and releases) because capturing them requires special processing that would degrade its forwarding performance. Also, this data does not persist across uBR reboots or replacements. Therefore, the only reliable source of DHCP state information for the uBR device is the DHCP server itself.

For this reason the DHCP server supports the DHCPLEASEQUERY message, which is similar to a DHCPINFORM message. Access concentrators and relay agents can thereby obtain client location data directly from the DHCP server, for DHCPv4 and DHCPv6 addresses.

Related Topics

- Leasequery Implementations, page 23-29
- Pre-RFC Leasequery for DHCPv4, page 23-30
- RFC 4388 Leasequery for DHCPv4, page 23-31
- Leasequery for DHCPv6, page 23-31
- Leasequery Statistics, page 23-32
- Leasequery Example, page 23-34

Leasequery Implementations

Cisco Prime IP Express provides three Leasequery implementations:

- **DHCPv4 Cisco-proprietary for pre-RFC 4388**—See the “Pre-RFC Leasequery for DHCPv4” section on page 23-30.
- **DHCPv4 compliant with RFC 4388**—See the “RFC 4388 Leasequery for DHCPv4” section on page 23-31.
- **DHCPv6**—See the “Leasequery for DHCPv6” section on page 23-31.
The Cisco-proprietary and the more recent RFC-compliant implementations for DHCPv4 differ in only minor ways and will coexist. The DHCP server accepts Leasequery requests at the same port and returns the specified data for both implementations. The DHCPv6 implementation conforms with RFC 5007 and RFC 5460.

The DHCP server can include lease reservation data in Leasequery responses for DHCPv4 and DHCPv6. Cisco Prime IP Express returns a default lease time of one year (31536000 seconds) for reserved DHCPv4 and lifetime of the leases for DHCPv6 leases in a response. If the IP address is actually leased, Cisco Prime IP Express returns its remaining lease time.

Leasequery is preset to be enabled for all the implementations. To disable it, disable an Expert mode attribute, leasequery.

**Pre-RFC Leasequery for DHCPv4**

Leasequery messages usually contain request fields and options. To illustrate, suppose that after a relay agent reboot or replacement, the relay agent receives a request to forward a datagram downstream to the public broadband access network. Because the relay agent no longer has the downstream location data, it sends a LEASEQUERY message to the DHCP server that includes the gateway IP address (giaddr) of the relay agent and the MAC address or dhcp-client-identifier (option 61) of the target client. If the DHCP server finds the client, it returns the client IP address in the client address (ciaddr) field in the response to the leasequery. If the server cannot find the client address, it returns a DHCPNACK.

In the pre-RFC implementation for DHCPv4, the requestor can query by IP address, client ID option (61), or MAC address, and receives from the server a DHCPACK (with the returned data) or a DHCPNACK message, or the server drops the packet. If the request includes multiple query types, the DHCP server responds to the first one it can find. The giaddr value from the requestor is independent of the ciaddr searched and is simply the return IP address for any responses from the server. The three possible query types are:

- **IP address (ciaddr)**—The request packet includes an IP address in the ciaddr field. The DHCP server returns data for the most recent client to use that address. A packet that includes a ciaddr value must be a request by IP address, despite the values in the MAC address fields (htype, hlen, and chaddr) or dhcp-client-identifier option. Querying by IP address is the most efficient method and the one most widely used, in that the other two methods can put more load on the DHCP server.

- **dhcp-client-identifier option (61)**—The request packet includes a dhcp-client-identifier option value. The DHCP server returns a DHCPACK packet containing the IP address data for the most recently accessed client. If the request omits a MAC address, the server returns all IP addresses and their data for the requested client ID in the cisco-leased-ip (also called the associated-ip) option. If the request includes the MAC address, the server matches the dhcp-client-identifier and MAC address with the client data for the IP address and returns that data in the ciaddr field or cisco-leased-ip (also called the associated-ip) option.

- **MAC address**—The request packet includes a MAC address in the hardware type (htype), address length (hlen), and client hardware address (chaddr) fields, and a blank ciaddr field. The server returns all the IP addresses and most recent lease data for the MAC address in the cisco-leased-ip (also called the associated-ip) option of the reply packet.

The DHCPLEASEQUERY message number in the dhcp-message-type option (53) for the pre-RFC implementation is 13. A server that does not support this type of message is likely to drop the packet. The DHCPACK message reply always contains the physical address of the lease owner in the htype, hlen, and chaddr fields. If the request contains the ciaddr, the data returned is always based on the ciaddr and never the client ID or MAC address.
The requestor can include the parameter-request-list option (55) to request specific options about an address. The reply often contains the dhcp-lease-time option (51) and the original content of the relay-agent-info option (82) that the client sent. If the server does not detect a valid lease for a client, it does not return option 51, and the requestor needs to determine if there is a valid lease.

A DHCPACK from the server can also contain the following Leasequery options:

- **cisco-leased-ip (161)**—Data on all the IP addresses associated with the client; also known as (and later renamed) the associated-ip option.
- **cisco-client-requested-host-name (162)**—Hostname that the client requested in the host-name option (12) or client-fqdn option (81). The requested hostname was dropped in the RFC 4388 implementation.
- **cisco-client-last-transaction-time (163)**—Most recent time duration that a DHCP server contacted the client.

### RFC 4388 Leasequery for DHCPv4

Leasequery became an official RFC 4388 for DHCPv4 in February 2006. Cisco Prime IP Express provides the RFC 4388 implementation alongside the pre-RFC one (see the “Pre-RFC Leasequery for DHCPv4” section on page 23-30) and there are no conflicts between them. However, the RFC 4388 implementation includes a few notable changes:

- The DHCPLEASEQUERY message type contained in the dhcp-message-type option (53) changed its message ID to 10 (the ID 13 was given to the DHCPLEASEACTIVE message), and the reply messages were changed from just DHCPACK and DHCPNACK to be more specific:
  - DHCPLEASEQUERY (10) for queries
  - DHCPLEASEUNASSIGNED (11) for replies of unassigned addresses
  - DHCPLEASEUNKNOWN (12) for replies of unknown addresses
  - DHCPLEASEACTIVE (13) for replies of active addresses
- The reply option names and IDs changed, and the cisco-client-requested-host-name option was dropped so that there are only two reply options:
  - client-last-transaction-time (91)—Most recent time duration that a DHCP server contacted the client.
  - associated-ip (92)—Data on all the IP addresses associated with the client.
- If querying by client ID or MAC address, the request can contain only the dhcp-client-identifier option (61) or MAC address; if the packet contains both, the server drops it.

### Leasequery for DHCPv6

Cisco Prime IP Express supports both the RFC 5007 (UDP) and RFC 5460 (TCP, Bulk) DHCPv6 leasequery capabilities.

**Note**

To use the RFC 5460 (TCP, Bulk) leasequery support, you must create a DHCP Listener for IPv6 (see the “DHCP Listener Configuration” section on page 23-36).

The message types for DHCPv6 Leasequery are:

- LEASEQUERY (14)
Querying Leases

A query can be by:

- QUERY_BY_ADDRESS (1)
- QUERY_BY_CLIENTID (2)
- QUERY_BY_RELAY_ID (3)
- QUERY_BY_LINK_ADDRESS (4)
- QUERY_BY_REMOTE_ID (5)

A DHCPv6 LEASEQUERY_REPLY message can contain one or more of the following options:

- lq-query (44)—Query being performed. The option, used in a request only, includes the query type, link-address (or 0::0), and options to provide data needed for the query.
- client-data (45)—Encapsulates the data for a single client on a single link. The client data can include any number of these or other requested options.
- clt-time (46)—Client last transaction time encapsulated in a client-data option (45); identifies how long ago (in seconds) the server last communicated with the client.
- lq-relay-data (47)—Relay agent data used when the client last communicated with the server. Fields are the peer-address and the relay-message. This option can include further options.
- lq-client-link (48)—Links on which the client has any bindings. Used in reply to a client query when the link-address is omitted and the client is found to be on more than one link.

A DHCPv6 LEASEQUERY_REQUEST message can contain one or more of the following options:

- option_lq_start_time—Bindings updated since the specified time. This option, used for the list of binding updates happened during the offline period.
- option_lq_end_time—Bindings updated during the specified time period.

DHCPv6 uses the Option Request option (oro) to request a list of options in the Leasequery reply.

**Note**

Leasequery by client-id requests may need to specify the override-client-id attribute when using [v6-]override-client-id expressions to correctly retrieve the information on the lease(s) for the client.

Leasequery Statistics

Lease queries provide statistics attributes, in the web UI, on the DHCP Server Statistics page (see the “Displaying Statistics” section on page 7-11), and in the CLI by using dhcp getStats. The Leasequery statistics are:

- **lease-queries**—Number of RFC 4388 message ID 10 (or pre-RFC message ID 13) DHCPv4 Leasequery packets received in the given time interval.
- **lease-queries-active**—Number of RFC 4388 DHCPLEASEACTIVE packets.
- **lease-queries-unassigned**—Number of RFC 4388 DHCPLEASEUNASSIGNED packets.
• **lease-queries-unknown**—Number of RFC 4388 DHCPLEASEUNKNOWN packets.

• **leasequeries**—Number of DHCPv6 Leasequery packets received.

• **leasequery-replies**—Number of responses to DHCPv6 Leasequery packets that might or might not be successful.

• **tcp-current-connections**—Number of currently open TCP connections to the DHCP server for DHCPv6 Active and Bulk Leasequery.

• **tcp-total-connections**—Number of TCP connections that were opened to the DHCP server for DHCPv6 Active and Bulk Leasequery in this time interval.

• **bulk-leasequeries**—Number of LEASEQUERY packets received over all TCP connections in this time interval.

• **bulk-leasequery-replies**—Number of LEASEQUERY-REPLY packets sent over all TCP connections in this time interval.

• **bulk-leasequery-data**—Number of LEASEQUERY-DATA packets sent over all TCP connections in this time interval.

• **bulk-leasequery-done**—Number of LEASEQUERY-DONE packets sent over all TCP connections in this time interval.

• **tcp-lq-status-unspec-fail**—Number of LEASEQUERY-REPLY packets with a status code of UnspecFail(1) sent over TCP in this time interval.

• **tcp-lq-status-unknown-query**—Number of LEASEQUERY-REPLY packets with a status code of UnknownQueryType(7) sent over TCP in this time interval.

• **tcp-lq-status-malformed-query**—Number of LEASEQUERY-REPLY packets with a status code of MalformedQuery(8) sent over TCP in this time interval.

• **tcp-lq-status-not-configured**—Number of LEASEQUERY-REPLY packets with a status code of NotConfigured(9) sent over TCP in this time interval.

• **tcp-lq-status-not-allowed**—Number of LEASEQUERY-REPLY packets with a status code of NotAllowed(10) sent over TCP in this time interval.

• **tcp-lq-status-query-terminated**—Number of LEASEQUERY-REPLY/LEASEQUERY-DONE packets with a status code of QueryTerminated(11) sent over TCP in this time interval.

• **tcp-connections-dropped**—Number of TCP requests that were terminated in this time interval because the TCP connection was closed (or reset) by the DHCPv6 requester. This excludes normal connection closes or server reboots.

• **active-leasequeries**—Number of ACTIVELEASEQUERY packets received over all TCP connections in this time interval.

• **active-leasequery-replies**—Number of LEASEQUERY-REPLY packets sent over all TCP connections in this time interval for active leasequery.

• **active-leasequery-data**—Number of LEASEQUERY-DATA packets sent over all TCP connections in this time interval for active leasequery.

• **active-leasequery-done**—Number of LEASEQUERY-DONE packets sent over all TCP connections in this time interval for active leasequery.

• **tcp-lq-status-data-missing**—Number of LEASEQUERY-REPLY packets with a status code of DataMissing(240) sent over TCP in this time interval.

• **tcp-lq-status-catch-up-complete**—Number of LEASEQUERY-DATA packets with a status code of CatchUpComplete(241) sent over TCP in this time interval.
Leasequery Example

Example 23-2 on page 23-34 shows a packet trace of a DHCPv6 UDP query by client ID without a link-address, but with addresses on more than one link. The first part of the output shows the query message and the second part shows the reply data. The \texttt{lq-query} option identifies the type of query. Note the list of requested options via the Option Request option (\texttt{oro}) in the request, and the two addresses returned in the \texttt{lq-client-links} option in the reply.

\textbf{Example 23-2  Packet Trace of Sample UDP Lease Query}

\begin{verbatim}
+- Start of LEASEQUERY (14) message (113 bytes)
  | transaction-id 22
  | lq-query (44) option (37 bytes)
  | (query-type 2, link-address ::)
  | client-identifier (1) option (10 bytes)
  | 00:03:00:01:02:03:04:05:06
  | oro (6) option (2 bytes)
  | 47
  | server-identifier (2) option (14 bytes)
  | 00:01:00:01:13:06:6a:67:00:23:7e:65:e3
  | client-identifier (1) option (10 bytes)
  | 00:03:00:01:03:05:07:09:11
  | vendor-class (16) option (14 bytes)
  | (enterprise-id 1760,
  | vendor-class (16) option (14 bytes)
  | (enterprise-id 1760,
+- End of LEASEQUERY message

+- Start of LEASEQUERY-REPLY (15) message (72 bytes)
  | transaction-id 22
  | server-identifier (2) option (14 bytes)
  | 00:01:00:01:13:06:6a:67:00:23:7e:65:e3
  | client-identifier (1) option (10 bytes)
  | 00:03:00:01:03:05:07:09:11
  | lq-client-links (48) option (32 bytes)
  | 2001:4f8:ffff:0:8125:ef1b:bdcb:4b4e,2001:4f8:ff00:0:e400:f92:1bfd:60fa
+- End of LEASEQUERY-REPLY message
\end{verbatim}

Example 23-3 on page 23-34 shows a packet trace of a DHCPv6 TCP query by client ID. The first part of the output shows the request message, the second part shows the response message with the binding data of the first client, and the last part will show that the query has ended successfully. The third part will follow the second part if there are more than a single client to be returned.

\textbf{Example 23-3  Packet Trace of Sample TCP Lease Query}

\begin{verbatim}
+- Start of LEASEQUERY (14) message (59 bytes)
  | transaction-id 2
  | lq-query (44) option (37 bytes)
  | (query-type 2, link-address ::)
  | client-identifier (1) option (10 bytes)
  | 00:03:00:01:02:03:04:05:06
  | oro (6) option (2 bytes)
  | 47
\end{verbatim}

\textbf{Note}

The LEASEQUERY-DONE message will not be present in a packet if the LEASEQUERY-REPLY message does not have any binding data.
Querying Leases

The following are the differences between TCP bulk leasequery and UDP leasequery:

**Difference between TCP bulk leasequery and UDP leasequery**

The following are the differences between TCP bulk leasequery and UDP leasequery:
- UDP leasequery supports Query by IPv6 Address and Query by Client Identifier. However, TCP Bulk Leasequery supports all the five query types; that is, Query by IPv6 Address, Query by Client Identifier, Query by Relay Identifier, Query by Link Address, and Query by Remote ID.
- In UDP Leasequery, if the server finds bindings for the relay agent on multiple links, then DHCP server will send an option OPTION_CLIENT_LINK in the reply message. The relay agent will need to resend LEASEQUERY messages using each of the returned link-addresses to obtain the all client's bindings. Whereas in TCP Bulk Leasequery, the server returns multiple bindings of a client on different links; however OPTION_CLIENT_LINK is not supported in Bulk Leasequery reply.

**DHCP Listener Configuration**

Using DHCP Listener Configuration, you can configure objects to enable active and bulk leasequery to the DHCP server over TCP connections. A single object is sufficient, unless you want the DHCP server to support listening for connections on multiple TCP ports or need to restrict the addresses on which the server will accept incoming connections.

**Local Advanced Web UI**

**Step 1**
From the **Deploy** menu, choose **Listener** under the **DHCP** submenu, to open the List/Add DHCP TCP Listener page.

**Step 2**
Click the **Add Listeners** icon in the Listeners pane, enter a name in the **Name** field, then click **Add TCP Listener**.

**Step 3**
Enter an IP address in the **address/ip6address** field, to restrict the interface over which the server will accept connections. This is usually unspecified. If you want to configure a IPv6 listener, then enter **ip6address**. If both address and ip6address are not specified, then the IPv4 address 0.0.0.0 is used.

To restrict the address on which TCP connections are accepted, enter the address in the **address (for IPv4)** or **ip6address (for IPv6)** attribute. If no value is entered in either attribute, IPv4 connections to any IPv4 address of the host are accepted. To specify connections over IPv6, you must enter a value in the **ip6address attribute** (0::0 can be used to accept connections to any IPv6 address of the host). You can only enter a value in either, not both, attributes.

**Note**
You cannot specify both IPv4 and IPv6 listeners for a DHCP server.

**Step 4**
Enter a value for the port in the **Port** field, if the default value is not appropriate. The default port is the server-port for DHCPv4 and DHCPv6-server-port for DHCPv6.

**Step 5**
For Enable attribute, click **true** or **false** radio button. The default value is true.

**Step 6**
Enter a value for Max-connections, if the default value 10 is not appropriate.

**Step 7**
Enter a value for Leasequery-backlog-time, if the default value 120 is not appropriate.

**Step 8**
For leasequery-send-all attribute, click **true** or **false** radio button. The default value is false.

**Step 9**
Click **Save**.

**CLI Commands**

The DHCP Listener commands are shown in Table 23-4.
Lease History Database Compression Utility

The cnr_leasehist_compress utility was added to Cisco Prime IP Express to compress regional cluster (DHCPv4) lease history databases. This utility does not compress the data directly in the databases, but copies the existing data into new databases that are optimized to be as compact as possible. You can download this utility from the Cisco Prime IP Express download section on the Cisco website.

Caution

Use the cnr_leasehist_compress utility only with the regional cluster lease history database, and when you suspect that the database grew significantly, particularly because of DHCPRELEASE packets.

During the copy operation, you can use this utility to:

- Trim records that are older than a certain interval of time—You would typically use the –t option. The interval that you specify with this option uses the Cisco Prime IP Express time interval format; for example, 30d for 30 days or 1y for 1 year.

- Merge records that belong to the same lease and client—You use the cnr_leasehist_compress utility to merge records that belong to clients who have reclaimed the lease on an IP address after releasing it. You would typically use the –m option. The interval that you specify with this option uses the Cisco Prime IP Express time interval format; for example, 120s for 120 seconds or 2m for 2 minutes.

While merging records, the utility also corrects lease history records that were terminated abruptly or have an incorrect binding end time (that may have resulted from a subsequent lease operation). This option of merging records also addresses the vast number of records that are created by certain router configurations that introduce an additional load on the servers.

Before you run the cnr_leasehist_compress utility:

- Stop the Cisco Prime IP Express regional cluster; it does not operate on an active regional cluster database.

- Note that you can use it to compress existing lease history data alone; it does not alter how the regional cluster collects future lease history records. If you suspect chatty clients, ensure that the DHCP server does not process DHCPRELEASE messages, because this results in rapid growth of lease history data. In such instances, you may need to run the utility periodically.

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>dhcp-listener &lt;name&gt; create [&lt;attribute&gt;=&lt;value&gt;]</td>
</tr>
<tr>
<td>Delete</td>
<td>dhcp-listener &lt;name&gt; delete</td>
</tr>
<tr>
<td>List</td>
<td>dhcp-listener list</td>
</tr>
<tr>
<td>List the names</td>
<td>dhcp-listener listnames</td>
</tr>
<tr>
<td>Show</td>
<td>dhcp-listener show</td>
</tr>
<tr>
<td>Set</td>
<td>dhcp-listener &lt;name&gt; set &lt;attribute&gt;=&lt;value&gt; [&lt;attribute&gt;=&lt;value&gt; ...]</td>
</tr>
<tr>
<td>Get</td>
<td>dhcp-listener &lt;name&gt; get &lt;attribute&gt;</td>
</tr>
<tr>
<td>Unset</td>
<td>dhcp-listener &lt;name&gt; unset &lt;attribute&gt;</td>
</tr>
<tr>
<td>Enable</td>
<td>dhcp-listener &lt;name&gt; enable &lt;attribute&gt;</td>
</tr>
<tr>
<td>Disable</td>
<td>dhcp-listener &lt;name&gt; disable &lt;attribute&gt;</td>
</tr>
</tbody>
</table>

---

The cnr_leasehist_compress utility was added to Cisco Prime IP Express to compress regional cluster (DHCPv4) lease history databases. This utility does not compress the data directly in the databases, but copies the existing data into new databases that are optimized to be as compact as possible. You can download this utility from the Cisco Prime IP Express download section on the Cisco website.

Caution

Use the cnr_leasehist_compress utility only with the regional cluster lease history database, and when you suspect that the database grew significantly, particularly because of DHCPRELEASE packets.

During the copy operation, you can use this utility to:

- Trim records that are older than a certain interval of time—You would typically use the –t option. The interval that you specify with this option uses the Cisco Prime IP Express time interval format; for example, 30d for 30 days or 1y for 1 year.

- Merge records that belong to the same lease and client—You use the cnr_leasehist_compress utility to merge records that belong to clients who have reclaimed the lease on an IP address after releasing it. You would typically use the –m option. The interval that you specify with this option uses the Cisco Prime IP Express time interval format; for example, 120s for 120 seconds or 2m for 2 minutes.

While merging records, the utility also corrects lease history records that were terminated abruptly or have an incorrect binding end time (that may have resulted from a subsequent lease operation). This option of merging records also addresses the vast number of records that are created by certain router configurations that introduce an additional load on the servers.

Before you run the cnr_leasehist_compress utility:

- Stop the Cisco Prime IP Express regional cluster; it does not operate on an active regional cluster database.

- Note that you can use it to compress existing lease history data alone; it does not alter how the regional cluster collects future lease history records. If you suspect chatty clients, ensure that the DHCP server does not process DHCPRELEASE messages, because this results in rapid growth of lease history data. In such instances, you may need to run the utility periodically.
• Note that you can use it if you are a service provider and suspect that the regional lease history in your network grew because some devices have known issues, such as repeatedly generating a sequence of DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, DHCPACK, and after 30 seconds, DHCPRELEASE messages. You can choose to drop all DHCPRELEASE messages or those that belong to clients that exceed a configured threshold.

• Note that it writes the new database in the most optimal manner. The new database can initially grow at a considerable rate, but it tapers back to normal after the additional lease history records are collected.

General Comments on Running cnr_leasehist_compress

⚠️ Caution

Follow every step in this procedure carefully. If you skip any step, you might lose lease history data. Note the lease history database that each task involves. Depending on the number of lease history records and the time taken to trim or merge the records, this utility may take several hours or days to run. You can interrupt the utility while it is running if the server reboots before the run is completed. You can resume it later; however, you must specify the same options that you have used in the previous run.

The install-path is the path in which you install Cisco Prime IP Express.

Table 5 describes the qualifying options for the cnr_leasehist_compress utility.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–a</td>
<td>Appends all the lease history records in the temporary active database to those in the new database.</td>
</tr>
<tr>
<td>–c</td>
<td>Generates a report when more than a specified number of records are merged for a client. When used with the –f option, these records are transferred to a log file.</td>
</tr>
<tr>
<td>–d</td>
<td>Specifies a new destination database that contains the compressed lease history records.</td>
</tr>
<tr>
<td>–e attrlist</td>
<td>Overwrites the excluded merge attribute list.</td>
</tr>
<tr>
<td>–f file</td>
<td>Redirects most lease history record warnings to a log file.</td>
</tr>
<tr>
<td>–g</td>
<td>Uses the dbtxn-seq and dbtxn-generation attributes to generate a new sequence in the numbers that are assigned to all lease history records, which are written into the destination database.</td>
</tr>
<tr>
<td>–i ipaddr</td>
<td>Transfers the records of a particular IP address to a log file.</td>
</tr>
<tr>
<td>–l limit</td>
<td>Purges log files after the database reaches the preset limit of 20 files.</td>
</tr>
<tr>
<td>–m time-int</td>
<td>Merges lease records where the binding-start-time of a particular lease falls in the duration of the binding-end-time of the previous lease. The recommended value for this option is 120s.</td>
</tr>
<tr>
<td>–n</td>
<td>Compares adjacent records without merging them.</td>
</tr>
<tr>
<td>–p</td>
<td>Drops detailed lease history records. You can use this option only if you have enabled detailed lease history.</td>
</tr>
</tbody>
</table>
Running Compression on Linux

To run the `cnr_leasehist_compress` utility on Linux, do the following:

**Step 1**  Add `install-path/lib` to the LD_LIBRARY_PATH to provide the utility with access to the Cisco Prime IP Express libraries:

```
$ bash
  # export LD_LIBRARY_PATH=install-path/lib:$LD_LIBRARY_PATH
```

**Step 2**  Stop the Cisco Prime IP Express regional cluster:

```
# /etc/init.d/nwregregion stop
```

**Step 3**  Rename the original `install-path/data/leasehist` directory as `install-path/data/oldleasehist`. The `/leasehist` directory becomes the original database:

```
# mv install-path/data/leasehist \\n# install-path/data/oldleasehist
```

**Step 4**  Create a new `leasehist` directory, which becomes the temporary active database:

```
# mkdir install-path/data/leasehist
```

**Step 5**  Run the `cnr_leasehist_compress` utility to allow the regional cluster to resume activity:

```
# install-path/bin/cnr_leasehist_compress \
> -r 0 \ 
> -s install-path/data/oldleasehist \ 
> -d install-path/data/leasehist \ 
> -p
```

---

**Table 5**  `cnr_leasehist_compress` Options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| `-q records` | Sets an interval for a periodic progress report that is generated while the utility runs. The default value is **100000**. For example: 
  
  +00:00:18 Read 100000 records (0 bad); trimmed 6717; merged 73370; 19912 written (19.91%) |
| `-r records` | Limits the number of records that are read from the source database. |
| `-s path` | Specifies the source database from where the data is copied to a new database. |
| `-t age` | Specifies a value for trimming records that are older than a certain interval of time. Use the standard Cisco Prime IP Express time interval for this option, such as **1y** for 1 year or **30d** for 30 days. |
| `-v` | Emits the version and exits. |
| `-w` | Limits the number of records that are written into the destination database. |
| `-y` | Alters the width of the dump of lease history records. This option is not recommended; however, you can use the value **132 30** for a 132-column output. |
| `-z` | Debrids the database, specified by using the standard Cisco Prime IP Express debug tracing syntax. |
Lease History Database Compression Utility

Chapter 23  Managing Leases

Caution  Running these commands does not compress the original database. The –r 0 option is critical as it instructs the utility to create the temporary active database. The regional cluster remains active while the utility compresses the original database.

Step 6  Restart the Cisco Prime IP Express regional cluster.

# /etc/init.d/nwregregion start

You cannot, however, obtain lease history data from the original database at this time. The regional cluster collects new lease history data and transfers it to the temporary active database. The utility then merges the new lease history data into the new database.

Step 7  Create a new directory called install-path/data/newleasehist. This /newleasehist directory becomes the new lease history database:

# mkdir install-path/data/newleasehist

Tip  After the regional cluster populates the new database, you can optionally create this new directory on a different partition and copy it to the final location.

Step 8  Run the cnr_leasehist_compress utility to trim, merge, and compress the original database into the new database:

# install-path/bin/cnr_leasehist_compress \  > -s install-path/data/oldleasehist \  > -d install-path/data/newleasehist \  > -t trim-time-interval \  > -m merge-time-interval \  > -f /tmp/cnr-compress.log

If the original database contains any detailed lease history records, you must use the –p option to acknowledge that it is acceptable for the utility to not transfer these records into the new database. Otherwise, the utility does not run.

Step 9  Perform the following tasks to append any fresh lease history records to the new database after the utility processes the entire original database.

Note  Do not restart the regional cluster until you have completed the following procedure. If the system reboots during the following procedure, repeat these steps.

a.  Stop the Cisco Prime IP Express regional cluster:

# /etc/init.d/nwregregion stop

b.  Run the cnr_leasehist_compress utility to append new lease history records to the new database:

# install-path/bin/cnr_leasehist_compress \  > -a \  > -s install-path/data/leasehist \  > -d install-path/data/newleasehist \  > -m merge-time-interval \  > -f /tmp/cnr-append.log
Chapter 23      Managing Leases

Lease History Database Compression Utility

Caution

The –a option is critical as it indicates that the utility should append the lease history records in the temporary active database to those in the new database. We recommend that you use the same merge-time-interval value that you used for the original database.

c. After the utility completes the task of appending the newly collected lease history records, rename the temporary active database directory, install-path/data/leasehist, as install-path/data/tmpleasehist:

```
# mv install-path/data/leasehist \
# install-path/data/tmpleasehist
```

d. Rename the new database directory, install-path/data/newleasehist, as install-path/data/leasehist:

```
# mv install-path/data/newleasehist \
# install-path/data/leasehist
```

**Step 10** Start the Cisco Prime IP Express regional cluster:

```
#/etc/init.d/nwregregion start
```

**Step 11** Verify the regional lease history data by using the Cisco Prime IP Express web UI.

**Step 12** Archive the original database, in install-path/data/oldleasehist, and the temporary active database, in install-path/data/tmpleasehist. Ensure that you include all subdirectories and files when you archive the database.

**Step 13** Delete the original database and temporary active database:

```
# rm -rf install-path/data/oldleasehist
# rm -rf install-path/data/tmpleasehist
```

---

**Running Compression on Windows**

To run the `cnr_leasehist_compress` utility on Windows, do the following:

**Step 1** Stop the Cisco Prime IP Express regional cluster:

```
> net stop nwregregion
```

**Step 2** Rename the original `install-path\data\leasehist` directory as `install-path\data\oldleasehist`. This leasehist directory becomes the original database:

```
> rename install-path\data\leasehist install-path\data\oldleasehist
```

**Tip** You can move the original database to a different partition. Ensure that you copy the entire original `/leasehist` directory (including all its subdirectories and files) before you remove it.

**Step 3** Create a new leasehist directory, this new leasehist directory becomes the temporary active database:

```
> mkdir install-path\data\leasehist
```

**Step 4** Run the `cnr_leasehist_compress` utility to allow the regional cluster to resume activity:

```
> install-path\cnr_leasehist_compress -r 0 -s install-path\data\oldleasehist \
-d install-path\data\leasehist -p
```
Chapter 23      Managing Leases

Lease History Database Compression Utility

---

**Caution**
Running these commands, however, does not compress the original database. The `–r 0` option is critical as it instructs the utility to create the temporary active database. The regional cluster remains active while the utility compresses the original database.

---

**Step 5**
Restart the Cisco Prime IP Express regional cluster. However, you cannot obtain lease history data from the original database at this time. The regional cluster collects any new lease history data and transfers it to the temporary active database. The utility then merges the new lease history data into the new database:

```
> net start nwregregion
```

**Step 6**
Create a new directory called `install-path\data\newleasehist`. This newleasehist directory becomes the new lease history database:

```
> mkdir install-path\data\newleasehist
```

---

**Tip**
After the regional cluster populates the new database, you can optionally create this new directory on a different partition and copy it to the final location.

---

**Step 7**
Run the `cnr_leasehist_compress` utility to trim, merge, and compress the original database into the new database:

```
> install-path\cnr_leasehist_compress -s install-path\data\oldleasehist
    -d install-path\data\newleasehist -t trim-time-interval -m merge-time-interval
    -f c:\temp\cnr-compress.log
```

If the original database contains any detailed lease history records, you must use the `–p` option to acknowledge that it is acceptable for the utility to not transfer these records into the new database. Otherwise, the utility does not run.

**Step 8**
Perform the following tasks to append any fresh lease history records to the new database after the utility processes the entire original database.

---

**Note**
Do not restart the regional cluster until you have completed the following procedure. If the system reboots during the following procedure, repeat these steps:

1. **a.** Stop the Cisco Prime IP Express regional cluster:

   ```
   > net stop nwregregion
   ```

2. **b.** Run the `cnr_leasehist_compress` utility to append new lease history records to the new database:

   ```
   > install-path\cnr_leasehist_compress -a -s install-path\data\leasehist
      -d install-path\data\newleasehist -m merge-time-interval
      -f c:\temp\cnr-append.log
   ```

   **Caution**
The `–a` option is critical as it indicates that the utility should append the lease history records in the temporary active database to those in the new database. We recommend that you use the same `merge-time-interval` value that you used for the original database.

3. **c.** After the utility completes the task of appending the newly collected lease history records, rename the temporary active database directory, `install-path\data\leasehist`, as `install-path\data\tpleasehist:`

---

Cisco Prime IP Express 8.2 User Guide

OL-31070-01
d. Rename the new database directory, \install-path\data\newleasehist, as \install-path\data\leasehist:

\rename \install-path\data\newleasehist \install-path\data\leasehist

Step 9 Start the Cisco Prime IP Express regional cluster:

\net start nwregregion

Step 10 Verify the regional lease history data by using the Cisco Prime IP Express web UI.

Step 11 Archive the original database, in \install-path\data\oldleasehist, and the temporary active database, in \install-path\data\tmpleasehist. Ensure that you include all subdirectories and files when you archive the database.

Step 12 Delete the original database and temporary active database:

\del/s \install-path\data\oldleasehist
\del/s \install-path\data\tmpleasehist

Moving Leases Between Servers

There may be a need to move leases to a new DHCP server such as, the configuration of the server grows sufficiently large to exceed the recommended limits. There are different ways to accomplish this task depending on whether the leases are being moved to a new server or an existing server. Either of these techniques requires special considerations and careful execution. A new server is often the simplest way to accomplish by moving the entire configuration and the state database. To move the leases to another server, the \leaseadmin utility is used. This utility allows you to export all or a selected set of leases and also to import this exported lease set.

\Caution

The Leaseadmin utility must only be used on a local cluster (exporting or importing) and the DHCP server must be stopped before running the leaseadmin utility.

The leaseadmin utility was added to Cisco Prime IP Express to allow leases to be moved from one server to another. You must run this utility on the same machine as the DHCP server and you must have superuser/root privileges to read and modify the database file. This utility requires direct access to the lease state database; however, stopping the DHCP server is not sufficient as the stopped server still holds the lease state database open. If the utility is run when the database is still in use, the leaseadmin utility will report the error "Failed to obtain exclusive access to lease state database". The default location is:

- Windows—\Program Files\IP Express\bin
- Linux—/opt/nwreg2/local/bin

From the command prompt, change to the above location and run the utility using the syntax:

\# leaseadmin <options>
Table 23-6 describes the qualifying options for the `leaseadmin` utility.

**Table 23-6 leaseadmin Command Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-e filename</td>
<td>Sends output to a file.</td>
</tr>
<tr>
<td>-x</td>
<td>Sends raw output format (required to import).</td>
</tr>
<tr>
<td>-t</td>
<td>Specifies the record types to be exported. Valid values are: current, history, detail, all, v6leases, v6history</td>
</tr>
<tr>
<td>-s</td>
<td>Restrict the lease records to be exported to a subnet or a prefix.</td>
</tr>
<tr>
<td>-i filename</td>
<td>Import from a file.</td>
</tr>
<tr>
<td>-o</td>
<td>When used with the -i (import) option, overwrites the existing data.</td>
</tr>
<tr>
<td>-d</td>
<td>Specifies the address, subnet, or prefix to be deleted.</td>
</tr>
<tr>
<td>-n vpn</td>
<td>When used with -e (export) or -d (delete) option, specified the VPN.</td>
</tr>
<tr>
<td>-h path</td>
<td>Overrides the default path to the database.</td>
</tr>
<tr>
<td>-v</td>
<td>Displays the database version.</td>
</tr>
<tr>
<td>-z debug-args</td>
<td>Sets the debug output levels.</td>
</tr>
</tbody>
</table>
Advanced DHCP Server Properties

This chapter describes how to set up some of the more advanced DHCP server properties. Before clients can use DHCP for address assignment, you must add at least one scope to the server. This is described in Chapter 21, “Configuring Scopes and Networks.” The additional properties are:

- Configuring BOOTP
- Defining Advanced Server Attributes, page 24-4
- Integrating Windows System Management Servers, page 24-9
- Using Extensions to Affect DHCP Server Behavior, page 24-10
- Tuning the DHCP Server, page 24-15
- Configuring Virtual Private Networks and Subnet Allocation, page 24-17
- Setting DHCP Forwarding, page 24-24

Configuring BOOTP

BOOTP (the BOOTstrap Protocol) was originally created for loading diskless computers. It was later used to allow a host to obtain all the required TCP/IP information to use the Internet. Using BOOTP, a host can broadcast a request on the network and get information required from a BOOTP server. The BOOTP server is a computer that listens for incoming BOOTP requests and generates responses from a configuration database for the BOOTP clients on that network. BOOTP differs from DHCP in that it has no concept of lease or lease expiration. All IP addresses that a BOOTP server allocates are permanent.

You can configure Cisco Prime IP Express to act like a BOOTP server. In addition, although BOOTP normally requires static address assignments, you can choose to either reserve IP addresses (and, therefore, use static assignments) or have IP addresses dynamically allocated for BOOTP clients.

Related Topics

- About BOOTP, page 24-2
- Enabling BOOTP for Scopes, page 24-2
- Moving or Decommissioning BOOTP Clients, page 24-3
- Using Dynamic BOOTP, page 24-3
- BOOTP Relay, page 24-3
About BOOTP

When you configure the DHCP server to return a BOOTP packet, be aware that BOOTP requires information in the DHCP packet in fields other than the option space. BOOTP devices often need information in the boot file (file), server IP address (siaddr), and server hostname (sname) fields of the DHCP packet (see RFC 2131).

Every Cisco Prime IP Express DHCP policy has attributes with which you can configure the information you want returned directly in the file, siaddr, or sname fields. The Cisco Prime IP Express DHCP server also supports a configuration parameter with which you can configure the policy options and determine which of the file, sname, or siaddr values you want returned to the BOOTP device.

Cisco Prime IP Express supports an analogous configuration parameter with which you can configure the options and file, sname, or siaddr values you want returned to the DHCP client. This is in addition to any options requested by the DHCP clients in the dhcp-parameter-request option in the DHCP request. Thus, you can configure both the BOOTP and DHCP response packets appropriately for your devices.

Step 1
Decide which values you want for the BOOTP attributes:
- file—Name of the boot file
- siaddr—Server IP address
- sname—Optional server hostname

Step 2
Decide the list of options and their values that you want returned to the BOOTP client.

Step 3
Set these values in the policy you want associated with the BOOTP request:
- Attributes (packet-siaddr, packet-file-name, packet-server-name) to send to the BOOTP client.
- Option values, such as the server addresses and domain name to return to the BOOTP client.
- List of fields and options you want returned to the BOOTP client.

Step 4
Enable the associated scope or scopes for BOOTP processing.

Step 5
Enable dynamic BOOTP processing if you want to have this scope provide an address for any BOOTP client that requests one. If you do not enable dynamic BOOTP, you must make reservations for each BOOTP client for which you want this scope to provide an address.

Enabling BOOTP for Scopes

You can enable BOOTP processing for a scope. Set certain attributes and BOOTP reply options for a created policy in the local cluster web UI, or use policy name create and policy name set in the CLI, to configure BOOTP. Set the policy attributes and options as a comma-separated list. The attributes are entities to use in a client boot process:
- packet-siaddr—IP address of the next server
- packet-file-name—Name of the boot file
- packet-server-name—Hostname of the server

The server looks through the policy hierarchy for the first instances of these attribute values.
In the CLI, policy name setOption requires spaces (not equal signs) before values.
Also, enable BOOTP and dynamic BOOTP, if desired, and ensure that the DHCP server updates the DNS server with BOOTP requests. The options are:

- Set the option dhcp-lease-time.
- Enable the dynamic-bootp attribute.
- Enable the update-dns-for-bootp attribute.

### Moving or Decommissioning BOOTP Clients

When you move or decommission a BOOTP client, you can reuse its lease. To decommission a BOOTP client, you must remove its lease reservation from the scope and force its lease to be available.

Force the lease to be available in the local cluster web UI, or set scope name removeReservation and lease ipaddr force-available in the CLI.

### Using Dynamic BOOTP

When you use dynamic BOOTP, there are additional restrictions placed on the address usage in scopes, because BOOTP clients are allocated IP addresses permanently and receive leases that never expire.

If you are using DHCP failover, when a server whose scope does not have the dynamic-bootp option enabled goes into PARTNER-DOWN state, it can allocate any available IP address from that scope, no matter whether it was initially available to the main or backup server. However, when the dynamic-bootp option is enabled, the main server and backup servers can only allocate their own addresses. Consequently scopes that enable the dynamic-bootp option require more addresses to support failover.

When using dynamic BOOTP:

1. Segregate dynamic BOOTP clients to a single scope. Disable DHCP clients from using that scope. In the local cluster web UI, under the BOOTP attributes for the scope, disable the dhcp attribute. In the CLI, use scope name disable dhcp.
2. If you are using DHCP failover, set the failover-dynamic-bootp-backup-percentage attribute for the DHCP server to allocate a greater percentage of addresses to the backup server for this scope. This percentage can be as much as 50 percent higher than a regular backup percentage.

### BOOTP Relay

Any router that supports BOOTP relay usually has an address that points to the DHCP server. For example, if you are using a Cisco router, it uses the term IP helper-address, which contains an address for a specific machine. In this case, use this address to forward all BOOTP (and therefore DHCP) broadcast packets. Be sure that you configure this address on the router closest to your host.

If your DHCP clients are not receiving addresses from the DHCP server, check the network configuration, particularly the router or relay agent configuration, to verify that your network devices are set up to point to your Cisco Prime IP Express DHCP server address.
Defining Advanced Server Attributes

You can set advanced DHCP server attributes, including custom DHCP options.

Related Topics

- Setting Advanced DHCP Server Attributes, page 24-4
- Enabling BOOTP for Scopes, page 24-2
- Moving or Decommissioning BOOTP Clients, page 24-3
- Using Dynamic BOOTP, page 24-3
- BOOTP Relay, page 24-3

Setting Advanced DHCP Server Attributes

Table 24-1 describes the advanced DHCP server attributes that you can set in the local cluster web UI and CLI.
### Table 24-1  DHCP Advanced Attributes

<table>
<thead>
<tr>
<th>Advanced Parameter</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>max-dhcp-requests</td>
<td>set/unset</td>
<td>Controls the number of buffers that the DHCP server allocates for receiving packets from DHCP clients and failover partners. If this setting is too large, a burst of DHCP activity can clog the server with requests that become stale before being processed. This results in an increasing processing load that can severely degrade performance as clients try to obtain a new lease, and affects the ability to handle bursts. A low buffer setting throttles requests and could affect server throughput. If the server runs out of buffers, packets are dropped. A good rule or thumb is to increase the buffers if you expect a high load (in a steady state or when experiencing frequent stress times) or you have a fast multiprocessor system. In a nonfailover deployment, the default setting (500) is sufficient. In a failover deployment, you can increase it to 1000 if the DHCP logs indicate a consistently high number of request buffers. You should then also modify the number of DHCP responses (see the max-dhcp-responses parameter) to four times the request buffers. When using LDAP client lookups, buffers should not exceed the LDAP lookup queue size defined by the total number of LDAP connections and the maximum number of requests allowed for each connection. Set the LDAP queue size to match the capacity of the LDAP server to service client lookups. If the following logs messages occur frequently and are not related to short term traffic spikes (such as after a power recovery), you may want to consider increasing the value of the attribute: 4493 DHCP ERROR &quot;DHCP has used xx of its yy request buffers: the server is dropping a request.&quot; 4494 DHCP WARNING &quot;DHCP has used xx of yy request packets. Requests will be ignored if no packet buffers are available.&quot; 5270 DHCP WARNING &quot;DHCP has used xx of its yy request buffers: the server is congested -- will not keep the client last-transaction-time to within value but will keep it to within value seconds.&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Required. The default is 500.</td>
</tr>
</tbody>
</table>
Defining Advanced Server Attributes

**max-dhcp-responses**

- **set/unset**

  Controls the number of response buffers that the DHCP server allocates for responding to DHCP clients and performing failover communication between DHCP partners.

  In a non-failover deployment, the default setting of twice the number of request buffers is sufficient. In a failover deployment, you can increase this so that it is four times the number of request buffers. In general, increasing the number of response buffers is not harmful, while reducing it to below the previously recommended ratios might be harmful to server responsiveness.

  If the following logs messages occur frequently and are not related to short term traffic spikes (such as after a power recovery), you may want to consider increasing the value of the attribute:

  4721 DHCP ERROR "DHCP has used all xx response packets. A request was dropped and they will continue to be dropped if no responses are available."

  5289 DHCP WARNING "DHCP has used xx of yy response packets. Requests will be dropped if no responses are available."

  Required. The default is 1000.

**max-ping-packets**

- **set/unset**

  Controls the number of buffers that the server has available to initiate Ping requests to clients. If you enable the *Ping address before offering it* option at the scope level, packet buffers are used to send and receive ICMP messages. If you enable pinging, you should have enough ping packets allocated to handle the peak load of possible ping requests. The default is 500 ping packets.

**hardware-unicast**

- **enable/disable**

  Controls whether the DHCP server sends unicast rather than broadcast responses when a client indicates that it can accept a unicast. This feature is only available on Windows platform; other operating systems broadcast instead. The default is enabled.

**defer-lease-extensions**

- **enable/disable**

  Controls whether the DHCP server extends leases that are less than half expired. This is a performance tuning attribute that helps minimize the number of disk writes to the lease state database. The default is checked or true. This means that a client renewing a lease less than halfway through can get the remaining part of it only and not be extended. See the “Deferring Lease Extensions” section on page 24-8.
Local Basic or Advanced Web UI

Step 1  From the Deploy menu, choose DHCP Server under the DHCP submenu to open the Manage DHCP Server page.

Step 2  Select the server from the DHCP Server pane.

Step 3  Add or modify attributes on the Edit Local DHCP Server page.

---

### Table 24-1  DHCP Advanced Attributes (continued)

<table>
<thead>
<tr>
<th>Advanced Parameter</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| last-transaction-time-granularity | set/unset | The default value of the last-transaction-time-granularity attribute has changed from 60 seconds to one week. This new default means that the client-last-transaction-time may not accurately reflect the last time the client communicated with the server.

If your deployment depended on this attribute being updated whenever the client communicated with the server, you need to explicitly set the last-transaction-time-granularity attribute to a value appropriate for the deployment.

The last-transaction-time-granularity attribute is effectively not used when you have disabled defer-lease-extensions. Therefore, if you have disabled defer-lease-extensions, this change in the default value does not impact you.

When the server is heavily loaded and has run low on request or response buffers, the server temporarily sets the last-transaction-time-granularity value to one year to reduce its load. |
| discover-queue-limit | set/unset | Specifies the percentage limit of the request buffers that may be used for DHCPDISCOVER and SOLICIT client requests at any time. Once the configured percentage of the request buffers is exceeded, additional DHCPDISCOVER and SOLICIT client requests are discarded. By restricting the requests buffers that can be used by DHCPDISCOVER/SOLICIT requests, the server assures it has request buffers available to process DHCPREQUEST/REQUEST requests and this can greatly reduce the time needed to get clients online during spikes in activity, such as after a power recovery or CMTS reboot.

The DRL (Discriminating Rate Limiter) attribute controls the discriminating rate limiter capability. The Discriminating Rate Limiter is enabled by default and assures that the DHCP server prefers completing DHCP transaction over starting too many new ones. In many situations, this should expedite bringing all clients online. If activity summary logging is enabled, the number of DHCPDISCOVER (DHCPv4) and SOLICIT (DHCPv6) packets dropped because of rate limiting is reported as DRL:number.

The DHCPv4 statistics includes a new queue-limited-discovers-dropped counter and the DHCPv6 statistics includes a new queue-limited-solicits-dropped counter. These counters are used to monitor the packets that are dropped. |
Defining Advanced Server Attributes

Step 4  Click Save after making the changes.

CLI Commands

Use dhcp show and dhcp get to show the current server parameters, then use dhcp set, dhcp unset, dhcp enable, and dhcp disable to change them (see Table 24-1 on page 24-5).

Deferring Lease Extensions

Enabling the defer-lease-extensions attribute (which is its preset value) allows the DHCP server to optimize response to a sudden flood of DHCP traffic. An example of a network event that could result in such a traffic spike is a power failure at a cable internet service provider (ISP) data center that results in all of its cable modem termination systems (CMTS) rebooting at once. If this happens, the devices attached to the CMTSs produce a flood of DHCP traffic as they quickly come back online.

With the defer-lease-extensions attribute enabled, the DHCP server might defer extending the lease expiration time for a client’s renewal request, which typically occurs before T1 (usually before halfway through the lease). Instead of giving the client the full configured lease time, the server grants the remaining time on the existing lease. Because the absolute lease expiration time does not change, the server can avoid database updates that result in a significantly higher server throughput. Another benefit is avoiding having to update the failover partner with an extended lease expiration time.

If a client is at or beyond T1 (typically halfway to its expiration), enabling or disabling this attribute has no effect, and the server always tries to extend the lease expiration time. However, failover and other protocol restrictions can prevent the server from extending the lease for the full configured time.

Note

Deferring lease extensions significantly increases the server performance while remaining in compliance with the DHCP RFC, which stipulates that client binding information is committed to persistent storage when the lease changes.

When deferring lease extensions, it is advisable to leave the policy attribute allow-lease-time-override to its default of disabled, or to change it to disabled if it is enabled.

These three specific situations are described from the server point of view:

- **Client retries**—When the server gets behind, it is possible for a client to retransmit requests. The DHCP server does not maintain enough information to recognize these as retransmissions, and processes each to completion, granting a full lease duration again and updating the database. When the server is already behind, doing extra work worsens the situation. To prevent this, the DHCP server does not extend leases that are less than 30 seconds old, regardless of the state of the defer-lease-extensions attribute.

- **Client reboots**—The effective renew time for a client lease is really the minimum of the configured renew time and the time between client reboots. In many installations this may mean that clients get fresh leases one (in a typical enterprise) or two (in a typical cable network) times per day, even if the renew time is set for many days. Setting the defer-lease-extensions attribute can prevent these early renews from causing database traffic.
Integrating Windows System Management Servers

You can have the DHCP server interact with the Microsoft System Management Server (SMS) so that SMS is current with DHCP changes. Normally, SMS pulls updated data through a DHCPDISCOVER request from the server about any new clients that joined the network. Cisco Prime IP Express, however, pushes these updates to SMS when you use `dhcp updatesms`. Before you do, verify that:

- SMS client installation and initialization step is complete.
- Cisco Prime IP Express Server Agent is set to run under a login account with sufficient privileges.
- SMS site ID is correct and matches that of the SMS server.

These steps describe how to integrate Windows SMS into Cisco Prime IP Express.

**Step 1** Install the Microsoft BackOffice 4.5 Resource Kit on the same machine as the Cisco Prime IP Express DHCP server. Follow the installation instructions and choose the default settings.

**Step 2** After the installation, modify the User Variable search path on the Environment tab of the System control panel to:

```
\program files\ResourceKit\SMS\diagnose
```

**Step 3** If the DHCP and SMS servers are on different machines, install the SMS client on the same machine as the DHCP server. The SMS library has the necessary API calls to communicate with the SMS server. You must assign the correct site code from the DHCP server machine. In your Network Neighborhood, go to the path `\SMS-servername\SMSLOGON\x86.bin\00000409\smsman.exe`.

Run the program and follow the instructions, using the default settings. The program creates two icons that you can use later from the control panel, marked SMS and Remote Control.

**Step 4** Stop and then restart the Cisco Prime IP Express server agent under a trusted domain account with sufficient privileges. Both the DHCP and SMS servers must be aware of this account. Use this short procedure:

a. Stop the local cluster server agent process.

b. Configure the account under which the Cisco Prime IP Express services run. Create an account name that is a member of both the trusted SMS site server group and a member of the DHCP server administrator group, with the corresponding password.

c. Restart the local cluster server agent process.
Step 5 Use `dhcp set sms-library-path` (or the `sms-library-path` attribute under the Microsoft Systems Management Server category on the Edit DHCP Server page) to configure the DHCP server to push lease information to SMS. Include the full path to the SMSRsGen.dll. If you omit a value, the path defaults to the internal server default location of this file. For example:

```
nrcmd> dhcp set sms-library-path /conf/dll
```

When you install the Microsoft BackOffice Resource Kit, the system path is not updated to reflect the location of the SMS data link library (DLL). Use one of these methods to configure this attribute:

a. Set the `sms-library-path` attribute to a relative path:

```
First, modify the system PATH variable to append the path of the directory where the DLL is installed:

  sms-install-directory\diagnose

Then, set `sms-library-path` to the name of the DLL, such as smsrsgen.dll. You can also accept the system default by unsetting the attribute.
```

b. Set `sms-library-path` to an absolute path. If you do not want to change the system path, set this attribute to the absolute path of the DLL location:

```
"\\Program Files\\Resource Kit\\sms\\diagnose\\smsrsgen.dll"
```

Step 6 Set the `sms-network-discovery` DNS attribute to 1 to turn SMS network discovery on. If you use the default of 0, you disable SMS network discovery.

Step 7 Set the `sms-site-code` DHCP server attribute by entering the SMS site code from Step 3. The default string is empty, but for data discovery to be successful, you must provide the site code.

Step 8 Set the `sms-lease-interval` attribute to the SMS lease interval.

The lease interval is the time between sending addresses to SMS, or how long, in milliseconds, the DHCP server should wait before pushing the next lease to the SMS server when you run `server dhcp updateSms`. Early versions of the SMSRsGen.dll file (SMS Version 2.0) did not allow SMS to reliably receive multiple updates within a one-second window (1000 ms); the default value, therefore, was set to 1.1 second (1100 ms). If you install a future version of the Microsoft BackOffice Resource Kit, which might contain an enhanced version of the SMSRsGen.dll file, then reduce this interval or set it to 0 to increase performance.

Step 9 Reload the DHCP server and check the dhcp_startup_log and/or name_dhcp_1_log file.

Step 10 In the CLI, use `server dhcp updateSms` to initiate SMS processing. (This command can take an optional `all` keyword to send all leased addresses from the DHCP server to SMS. If you omit this keyword, the DHCP server sends only new leases activated since the last time the command ran.) Then, verify that both the DHCP and SMS logs indicate successful completion. Note that a server reload during SMS updating interrupts the process, but the process resumes (or restarts) after the server is back up.

Using Extensions to Affect DHCP Server Behavior

Cisco Prime IP Express provides the ability to alter and customize the operation of the DHCP server through *extensions*, programs that you can write in TCL or C/C++. Extensions interact with the server in two ways: by modifying request or response packets, and through environment variables stored in the environment dictionary (see Chapter 30, “Using Extension Points” for details).
For example, you might have an unusual routing hub that uses BOOTP configuration. This device issues a BOOTP request with an Ethernet hardware type (1) and MAC address in the chaddr field. It then sends out another BOOTP request with the same MAC address, but with a hardware type of Token Ring (6). The DHCP server normally distinguishes between a MAC address with hardware type 1 and one with type 6, and considers them to be different devices. In this case, you might want to write an extension that prevents the DHCP server from handing out two different addresses to the same device.

You can solve the problem of the two IP addresses by writing either of these extensions:

- One that causes the DHCP server to drop the Token Ring (6) hardware type packet.
- One that changes the Token Ring packet to an Internet packet and then switches it back again on exit. Although this extension would be more complex, the DHCP client could thereby use either return from the DHCP server.

**Related Topics**

Writing Extensions, page 24-11  
Preventing Chatty Clients by Using an Extension, page 24-13

**Writing Extensions**

You can write extensions in TCL or C/C++:

- **TCL**—Makes it a bit easier and quicker to write an extension. If the extension is short, the interpreted nature of TCL does not have a serious effect on performance. When you write an extension in TCL, you are less likely to introduce a bug that can crash the server.

- **C/C++**—Provides the maximum possible performance and flexibility, including communicating with external processes. However, the complexity of the C/C++ API is greater and the possibility of a bug in the extension crashing the server is more likely than with TCL.

You create extensions at specific extension points. Extension points include three types of dictionaries—request, response, and environment. One or more of these dictionaries are available for each of the following extension points:

1. **init-entry**—Extension point that the DHCP server calls when it configures or unconfigures the extension. This occurs when starting, stopping, or reloading the server. This entry point has the same signature as the others for the extension. It is required for DHCPv6 processing. Dictionary: environment only.

2. **pre-packet-decode**—First extension point that the DHCP server encounters when a request arrives, and calls it before decoding the packet. Dictionaries: request and environment.


4. **post-class-lookup**—Evaluates the result of a client-class-lookup-id operation on the client-class. Dictionaries: request and environment.

5. **pre-client-lookup**—Affects the client being looked up, possibly by preventing the lookup or supplying data that overrides the existing data. Dictionaries: request and environment.

6. **post-client-lookup**—Reviews the operation of the client-class lookup process, such as examining the internal server data structures filled in from the client-class processing. You can also use it to change any data before the DHCP server does additional processing. Dictionaries: request and environment.

7. **generate-lease**—Generates and controls a DHCPv6 address or prefix. Dictionaries: request, response, and environment.
8. **check-lease-acceptable**—Changes the results of the lease acceptability test. Do this only with extreme care. Dictionaries: request, response, and environment.

9. **lease-state-change**—Determines when the lease state changes this only with extreme care. Dictionaries: response and environment.

10. **pre-packet-encode**—Changes the data sent back to the DHCP client in the response, or change the address to which to send the DHCP response. Dictionaries: request, response, and environment.

11. **post-packet-encode**—Allows the server to examine and alter the packet before it sends the packet to the client, or drops the packet. Dictionaries: request, response, and environment.

12. **pre-dns-add-forward**—Alters the name used for the DNS forward (A record) request. Dictionary: environment only.

13. **post-send-packet**—Used after sending a packet for processing that you want to perform outside of the serious time constraints of the DHCP request-response cycle. Dictionaries: request, response, and environment.

14. **environment-destructor**—Allows an extension to clean up any context that it might be holding. Dictionary: environment.

To extend the DHCP server, do the following:

---

**Step 1** Write the extension in Tcl, C or C++ and install it in the server extensions directory, on:

- **UNIX**:
  - Tcl—/opt/nwreg2/extensions/DHCP/tcl
  - C or C++—/opt/nwreg2/extensions/DHCP/dex

- **Windows**:
  - Tcl—\program files\Cisco Prime IP Express\extensions\dhcp\tcl
  - C or C++—\program files\Cisco Prime IP Express\extensions\dhcp\dex

It is best to place these extensions in the appropriate directory for TCL or C/C++ extensions. Then, when configuring the filename, just enter the filename itself, without slash (/) or backslash (\).

If you want to place extensions in subdirectories, enter the filename with a path separator. These are different depending on the operating system on which your DHCP server is running.

**Note** When entering a filename that contains a backslash (\) character in Windows, you must enter it with a double-backslash (\\), because backslash (\) is an escape character in the CLI. For example, enter the filename `debug\myextension.tcl` as `debug\\myextension.tcl`.

---

**Step 2** Use the List/Add DHCP Extensions page in the web UI (In the Advanced mode, from the Deploy menu, choose **Extensions** from the DHCP submenu to open the List/Add DHCP Extensions page) or the **extension** command in the CLI to configure the DHCP server to recognize this extension.

**Step 3** Attach the configured extension to one or more DHCP extension points by using **dhcp attachExtension**.

**Step 4** Reload the server.
Preventing Chatty Clients by Using an Extension

One example of an effective use of an extension is to protect against clients flooding the server with unnecessary traffic. You can use the ChattyClientFilter extension to keep the server from having to do much of the work of processing these chatty client packets. If you have large numbers of clients in your network, you might want to consider implementing this extension.

The ChattyClientFilter extension is available in the /examples/dhcp/dex directory of the Cisco Prime IP Express installation, and compiled and ready to use in /extensions/dhcp/dex/dexextension.so or /extensions/dhcp/dex/dexextension.dll. The extension monitors client requests, based on the MAC address, and disables the client if it generates more than a certain number of packets in a time interval. Disabling a client means that the server discards packets from it. However, the server does not ignore the client entirely, because it continues to monitor traffic from it. If the server detects that the client starts to generate fewer than a certain number of packets in a time interval, it reenables the client and begins to allow packets from it again.

![Chatty Client Filter Flow Diagram](image)

The criteria for disabling and reenabling are set through arguments to the ChattyClientFilter extension. By default, the server disables a client when it receives more than 15 packets within 30 seconds; the server reenables the client when it sends fewer than 5 packets within 10 seconds. Note that these defaults are conservative and do not protect against all situations. For example, the server does not disable a client that sends packets every three seconds. Even allowing for a few retransmissions, a client should never need to send more than a half dozen packets in a short interval.
If you suspect chatty clients, review the DHCP server logs to determine incoming rates, then set the arguments described in Table 24-2 in the ChattyClientFilter code appropriately.

<table>
<thead>
<tr>
<th>ChattyClientFilter Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–c</td>
<td>Ignores the packets when the “drop” attribute of the environment dictionary is set to “true”; default is not to ignore.</td>
</tr>
<tr>
<td>–d packet-count seconds</td>
<td>Drops DHCPRELEASE packets if more than the specified count are received in the specified time interval; default disabled. The server keeps dropping DHCPRELEASE packets until the client suspends sending them for the specified interval. (DHCPv4 clients only.) The basic formula is that the time interval should be at least ((\text{packet-count} + 2) \times 30) seconds.</td>
</tr>
<tr>
<td>–h packet-count</td>
<td>SampleHitsToDisable; default 15 packets.</td>
</tr>
<tr>
<td>–i seconds</td>
<td>SampleTimeInterval; default 30 seconds.</td>
</tr>
<tr>
<td>–l packet-count</td>
<td>QuietHitsToLeaveDisabled; default 5 packets.</td>
</tr>
<tr>
<td>–m seconds</td>
<td>Sets the maximum time a client is disabled, in seconds; default 0 - unlimited.</td>
</tr>
<tr>
<td>–n</td>
<td>NAKs the client if renewing or rebinding; default off. If the client exceeding the SampleHitsToDisable rate does a DHCPREQUEST, the server sends it a DHCPNAK instead of discarding the packet. This can resolve problems with clients (such as cable modems) that cannot renew leases for some reason. Sending the DHCPNAK causes the client to restart its DHCP state machine and send a DHCPDISCOVER. If you use this argument, you must attach the ChattyClientFilter to the <code>check-lease-acceptable</code> extension point. (DHCPv4 clients only.)</td>
</tr>
<tr>
<td>–q seconds</td>
<td>QuietTimeInterval; default 10 seconds.</td>
</tr>
<tr>
<td>–r seconds</td>
<td>StatisticsInterval; default 300 seconds (5 minutes). This argument controls the frequency of periodic logging of the number of clients disabled and reenabled.</td>
</tr>
<tr>
<td>–s</td>
<td>Silently discards dropped packets; default off.</td>
</tr>
<tr>
<td>–w port</td>
<td>Enables web access on specified port (only enabled over IPv4; specify negative port to not bind to 127.0.0.1).</td>
</tr>
</tbody>
</table>

**Note**

The `-h`, `-i`, `-l`, and `-q` defaults are unlikely to be appropriate to most situations as these were designed to address a single type of misbehaving client. Using a longer interval and packet hit count for normal conditions will produce reasonable results. Values such as `-i 120 -h 8 -q 120 -l 8` would allow a client 8 packets over a 120 second period. A normal DHCPDISCOVER/OFFER/REQUEST/ACK is only 2 packets from a client. That is, the proper use of the ChattyClientFilter requires tuning these values for your particular network conditions. Use of the logscan tool which is available from the Cisco Prime IP Express download section on the Cisco website can help in analyzing client activity.
Tuning the DHCP Server

Review the comments in the ChattyClientFilter.cpp file for details on setting the arguments and enabling the extension. In most cases, you would attach it to the post-packet-decode extension point (along with check-lease-acceptable if you use the –n argument).

A sample use for the ChattyClientFilter is to drop DHCPRELEASE packets sent from a DHCPv4 client to prevent the lease history database from growing out of bounds, which can be the case with certain router configurations.

This scenario uses the –d argument. The setup on a Linux system might be:

```
nrcmd> extension dexChattyClientFilter create dex libdexextension.so
       dexChattyClientFilter
       init-entry=dexChattyClientFilterInitEntry
       init-args="-d 2 120"
nrcmd> dhcp attachextension post-packet-decode dexChattyClientFilter
```

For Windows, replace libdexextension.so with dexextension.dll.

This setup results in the server dropping DHCPRELEASE packets if it receives more than two of these packets from the same client in a 120-second interval, and resuming DHCPRELEASEs processing when the client does not send a DHCPRELEASE for at least 120 seconds.

Cisco Prime IP Express supports the mini-web server that can be used to obtain information about the clients being monitored or disabled (traffic being dropped) by the Chatty Client Filter. A typical request might be http://127.0.0.1:<port>/report entered in a web browser.

The web server supports the following requests:

- **status**—Returns a statistics report.
- **report**—Returns a statistics report and a full client report. The client report includes all clients currently being monitored and those that are disabled.
- **disabled-report**—Same as report except only the disabled clients are returned.
- **flush**—Same as report but all clients are REMOVED from the internal monitored and disabled list.
- **csv-client-list**—Returns the client list using CSV format (includes monitored and disabled clients).
- **csv-disabled-client-list**—Same as csv-client-list but only includes clients currently disabled.
- **xml-client-list**—Returns the client list using XML (includes monitored and disabled clients).
- **xml-disabled-client-list**—Returns the disabled client list using XML.

This web server is a very basic server implementation. It only supports the requests mentioned above.

Other helpful hints in tuning your DHCP performance include:

- Set the request (max-dhcp-requests) and response (max-dhcp-responses) buffers for optimal throughput. See Table 24-1 on page 24-5 for details.
- Keep the defer-lease-extensions attribute enabled. This reduces writes to the database.
- Set the last-transaction-time-granularity attribute to at least 60 seconds, optimally a value greater than half your lease interval.
- Disable the allow-lease-time-override attribute for policies offering production leases.
- Minimize your logging and debugging settings. If you require logging, use the log-settings attribute for the DHCP server with a controlled number of attributes, as described in Table 24-3.
### Table 24-3  DHCP Log Settings

<table>
<thead>
<tr>
<th>Log Setting (Numeric Equivalent)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default (1)</td>
<td>Displays basic DHCP activity logging (the default setting).</td>
</tr>
<tr>
<td>incoming-packets (2)</td>
<td>Logs a separate line for each incoming DHCP packet (the default).</td>
</tr>
<tr>
<td>missing-options (3)</td>
<td>Displays missing policy options expected by a client (the default).</td>
</tr>
<tr>
<td>incoming-packet-detail (4)</td>
<td>The same as incoming-packets, but in human-readable form.</td>
</tr>
<tr>
<td>outgoing-packet-detail (5)</td>
<td>Logs each incoming DHCP packet in a human-readable form.</td>
</tr>
<tr>
<td>unknown-criteria (6)</td>
<td>Logs whenever a client entry has a selection-criteria or selection-criteria-excluded that is not found in any scope appropriate for the current network location of that client.</td>
</tr>
<tr>
<td>dns-update-detail (7)</td>
<td>Logs each sent and replied DNS update.</td>
</tr>
<tr>
<td>client-detail (8)</td>
<td>After every client-class client lookup operation, logs the composite of the data found for the client and its client-class. Useful when setting a client-class configuration and debugging problems in client-class processing.</td>
</tr>
<tr>
<td>client-criteria-processing (9)</td>
<td>Logs whenever a scope is examined to find an available lease or to determine if a lease is still acceptable for a client who already has one. Can be very useful when configuring or debugging client-class scope criteria processing. (Causes moderate amount of information to be logged and should not be left enabled as a matter of course.)</td>
</tr>
<tr>
<td>failover-detail (10)</td>
<td>Logs detailed failover activity.</td>
</tr>
<tr>
<td>ldap-query-detail (11)</td>
<td>Logs whenever the DHCP server initiates a query to an LDAP server, receives a response, and retrieves a result or error messages.</td>
</tr>
<tr>
<td>ldap-update-detail (12)</td>
<td>Logs whenever the DHCP server initiates an update lease state to the LDAP server, receives a response, and retrieves a result or error messages.</td>
</tr>
<tr>
<td>ldap-create-detail (13)</td>
<td>Logs whenever the DHCP server initiates a lease state entry create to the LDAP server, receives a response, and retrieves a result or error messages.</td>
</tr>
<tr>
<td>leasequery (14)</td>
<td>Logs a message for every ACK- or NAK-responded lease query packet.</td>
</tr>
<tr>
<td>dropped-waiting-packets (15)</td>
<td>If the value of max-waiting-packets is non-zero, packets can be dropped if the queue length for any IP address exceeds the value. If dropped-waiting-packets is set, the server logs whenever it drops a waiting packet from the queue for an IP address.</td>
</tr>
<tr>
<td>no-success-messages (16)</td>
<td>Inhibits logging successful outgoing response packets.</td>
</tr>
<tr>
<td>no-dropped-dhcp-packets (17)</td>
<td>Inhibits logging dropped DHCP packets.</td>
</tr>
<tr>
<td>no-dropped-bootp-packets (18)</td>
<td>Inhibits logging dropped BOOTP packets.</td>
</tr>
<tr>
<td>no-failover-activity (19)</td>
<td>Inhibits logging normal activity and some warning messages logged for failover. However, serious error log messages continue to appear.</td>
</tr>
<tr>
<td>activity-summary (20)</td>
<td>Enables logging a summary message every five minutes (useful if the following no- type flags are set), showing the activity in the previous interval (you can adjust this interval using dhcp set activity-summary-interval).</td>
</tr>
</tbody>
</table>
Consider setting client caching (see the “Setting Client Caching Parameters” section on page 25-13).

Check the server statistics to aid in monitoring server performance (see the “Displaying Statistics” section on page 7-11).

Consider setting the scope allocation priority (see the “Configuring Multiple Scopes Using Allocation Priority” section on page 21-12).

If pinging hosts before offering addresses, consider adjusting the ping timeout period (see the “Pinging Hosts Before Offering Addresses” section on page 23-6).

To boost performance, consider limiting the number of selection tags.

If using Lightweight Directory Access Protocol (LDAP) servers, consider the performance issues described in the “Configuring Cisco Prime IP Express to Use LDAP” section on page 25-19.

If using DHCP failover, consider using the load balancing feature (see the “Setting Load Balancing” section on page 28-13).

Tip
Be sure to follow any DHCP server attribute changes with a server reload.

### Configuring Virtual Private Networks and Subnet Allocation

This section describes how to configure the Cisco Prime IP Express DHCP server to support virtual private networks (VPNs) and subnet allocation for on-demand address pools.

Configuring VPNs involves an adjustment to the usual DHCP host IP address designation. VPNs use private address spaces that might not be unique across the Internet. Because of this, Cisco Prime IP Express supports IP addresses that are distinguished by a VPN identifier. Relay agents on routers must support this capability as well. The VPN identifier selects the VPN to which the client belongs. VPN for DHCP is currently only supported by Cisco IOS software, the newest versions of which can include VPN IDs in the relayed DHCP messages.

Subnet allocation is a way of leasing subnets to clients (usually routers or edge devices) so that they can, in turn, provide DHCP services. This can occur along with or instead of managing individual client addresses. Subnet allocation can vastly improve IP address provisioning, aggregation, characterization,
and distribution by relying on the DHCP infrastructure to dynamically manage subnets. Subnet allocation through DHCP is currently only supported by Cisco IOS software, the newest versions of which incorporate the on-demand address pools feature.

Related Topics

- Configuring Virtual Private Networks Using DHCP, page 24-18
- Configuring DHCP Subnet Allocation, page 24-22
- VPN and Subnet Allocation Tuning Parameters, page 24-23

### Configuring Virtual Private Networks Using DHCP

VPNs that you create provide a filtering mechanism for:

- Viewing the unified address space (see the "Viewing Address Space" section on page 9-2).
- Listing address blocks (see the "Adding Address Blocks" section on page 9-5).
- Listing subnets (see the "Address Blocks and Subnets" section on page 9-3).
- Querying lease history (see the "Running IP Lease Histories" section on page 23-21).

If you do not configure a VPN, Cisco Prime IP Express uses the global VPN of 0 on each scope.

To configure a VPN whereby a client can request IP addresses from a DHCP server using a relay agent, you must define the VPN and associate a scope with it. Specifically:

1. Ensure that the relay agents that handle DHCP VPN traffic are configured with a version of Cisco IOS software that supports the `vpn-id` suboption of the `relay-agent-info` option (82) in DHCP.
2. Coordinate with the Cisco IOS relay agent administrator that the VPN is identified either by a VPN ID or a VPN Routing and Forwarding instance (VRF) name.
3. Create a scope for the VPN.

### Typical Virtual Private Networks

Figure 20-4 shows a typical VPN scenario with DHCP client 1 as part of VPN blue and DHCP client 2 in VPN red. For example, both DHCP client 1 in VPN blue and client 2 in VPN red have the same private network address: 192.168.1.0/24. The DHCP relay agent has gateway addresses that are in the two VPNs as well as a global one (172.27.180.232). There are two failover DHCP servers, both of which know the relay agent through its external gateway address.

Here is the processing that takes place for the server to issue a VPN-supported address to a client:

1. DHCP client 1 broadcasts a DHCPDISCOVER packet, including its MAC address, hostname, and any requested DHCP options.
2. DHCP relay agent at address 192.168.1.1 picks up the broadcast packet. It adds a `relay-agent-info` option (82) to the packet and includes the `subnet-selection` suboption that identifies 192.168.1.0 as the subnet. The packet also includes the `vpn-id` suboption that identifies the VPN as blue. Because
the DHCP server cannot communicate directly with the requesting client, the server-id-override suboption contains the address of the relay agent as known by the client (192.168.1.1). The relay agent also includes in the packet its external gateway address (giaddr), 172.27.180.232.

3. The relay agent unicasts the DHCPDISCOVER packet to the configured DHCP server on its subnet.

4. DHCP server 1 receives the packet and uses the vpn-id and subnet-selection suboptions to allocate an IP address from the proper VPN address space. It finds the available address 192.168.1.37 in the subnet and VPN, and places it in the yiaddr field of the packet (the address offered to the client).

5. The server unicasts a DHCPOFFER packet to the relay agent that is identified by the giaddr value.

6. The relay agent removes the relay-agent-info option and sends the packet to DHCP client 1.

7. DHCP client 1 broadcasts a DHCPREQUEST message requesting the same IP address that it was offered. The relay agent receives this broadcast message.

8. The relay agent forwards the DHCPREQUEST packet to DHCP server 1, which replies with a unicast DHCPACK packet to the client.

9. For a lease renewal, the client unicasts a DHCPRENEW packet to the IP address found in the dhcp-server-identifier option of the DHCPACK message. This is 192.168.1.1, the address of the relay agent. The relay agent unicasts the packet to the DHCP server. The server does its normal renewal processing, without necessarily knowing whether it was the server that gave out the original address in the first place. The server replies in a unicast DHCPACK packet. The relay agent then forwards the DHCPACK packet to the client IP address identified by the ciaddr field value.

If the server-id-override suboption of the relay-agent-info option (82) exists, the DHCP server uses its value to compare to that of the dhcp-server-identifier option in the reply packet. Any packet that the DHCP client unicasts then goes directly to the relay agent and not to the server (which may, in fact, be inaccessible from the client). Both partners in a failover environment can renew a lease if the packet includes the server-id-override suboption.

### Creating and Editing Virtual Private Networks

To set up the VPN and its index:

**Step 1** Coordinate with the Cisco IOS relay agent administrator that the VPNs are configured either by VPN ID or VRF name on the relay agent. This will determine how to identify the VPN in Cisco Prime IP Express.

**Step 2** Create a VPN to allow provisioning DHCP clients onto the VPN that is configured in the IOS switch or router.

Enter a VPN index, which can be any unique text string except the reserved words all or global. Its associated ID must also be unique. To add an index at the:

- **Local cluster** (Advanced)—From the Design menu, choose VPNs under the DHCP Settings submenu to open the List/Add VPNs page. Give the VPN a numerical key identifier and a unique name in the cluster.

- **Regional cluster**—Add the local cluster containing the VPN (from the Operate menu, choose Manage Clusters under the Servers submenu). Then choose VPNs from the Design menu. This opens the List/Add VPNs page. You can create the VPN on this page or pull the VPN from the local clusters:
  - If creating the VPN, give it a numerical key identifier and a unique name.
  - If pulling the VPN from the local clusters, click the Pull Replica icon in the VPNs pane on the List/Add VPNs page, then pull a specific VPN or all the VPNs from the selected cluster.
You can also push VPNs to the clusters by clicking the Push or Push All icon in the List/Add VPNs page. Then choose the synchronization mode and the clusters to which to push the VPNs on the Push VPN Data to Local Clusters page.

- **In the CLI**—Use `vpn name create key`. For example:
  ```bash
  nrcmd> vpn blue create 99
  ```

**Step 3** Specify the appropriate VPN identifier, either by VPN ID or VRF name. It is rarely both.

- If you use a VPN ID, set the `vpn-id` attribute value for the VPN. The value is usually in hexadecimal, in the form `oui:index`, per IETF RFC 2685. It consists of a three-octet VPN Organizationally Unique Identifier (OUI) that corresponds to the VPN owner or ISP, followed by a colon. It is then followed by a four-octet index number of the VPN itself. Add the VPN ID value to the List/Add VPNs page.
  In the CLI, set the `vpn-id` attribute. For example:
  ```bash
  nrcmd> vpn blue set vpn-id=a1:3f6c
  ```

- If you use a VPN Routing and Forwarding (VRF) instance name, set the `vrf-name` attribute value for the VPN. Cisco routers frequently use VRF names. Add the VRF Name value to the List/Add VPNs page.
  In the CLI, set the `vrf-name` attribute. For example:
  ```bash
  nrcmd> vpn blue set vrf-name=framus
  ```

**Step 4** Add a description for the VPN (optional).

**Step 5** Click Add VPN. You can edit the VPN to change the values on the Edit VPN page.

**Step 6** Create a scope for the VPN.

You must keep the VPN name and scope name as similar as possible for identification purposes.

a. In the web UI, from the Design menu, choose Scopes under the DHCPv4 submenu to open the List/Add DHCP Scopes page.

b. Choose the VPN from the `username` drop-down list on the top right of the window. You cannot change the VPN once you set it at the time of creation of the scope

In the CLI, identify to which VPN the scope belongs in one of three ways:

- Its VPN name, through the `vpn` attribute (which applies the VPN ID to the scope).
- The VPN ID itself, through the `vpn-id` attribute.
- The current session VPN name, by omitting the VPN or its ID on the command line.

You set the default VPN for the current session using `session set current-vpn`. You can then set the usual address range and necessary option properties for the scope. For example:

```bash
nrcmd> scope blue-1921681 create 192.168.1.0 255.255.255.0 vpn=blue
```

Or:

```bash
nrcmd> scope blue-1921681 create 192.168.1.0 255.255.255.0 vpn-id=99
```

Or:

```bash
nrcmd> session set current-vpn=blue
nrcmd> scope blue-1921681 create 192.168.1.0 255.255.255.0
```

Then:

```bash
nrcmd> scope blue-1921681 addRange 192.168.1.101 192.168.1.200
nrcmd> scope-policy blue-1921681 setOption routers 192.168.1.1
```
If you are in the staged dhcp edit mode, reload the DHCP server after you create all the VPNs and scopes.

VPN Usage

The VPN name is used to qualify many DHCP objects in Cisco Prime IP Express, such as IP addresses (leases), scopes, and subnets. For example, lease names can have this syntax:

vpn/ipaddress

For example, red/192.168.40.0

A VPN can be any unique text string except the reserved words global and all. You can use global and all when you lease data. The global VPN maps to the [none] VPN; the all VPN maps to both the specific VPN and the [none] VPN.

In the CLI, if you omit the VPN or its ID in defining an object, the VPN defaults to the value set by session set current-vpn. In the web UI, if the current VPN is not defined, it defaults to the [none] VPN, which includes all addresses outside of any defined VPNs.

These objects have associated VPN properties:

- **Address blocks**—Define the VPN for an address block. Choose Address Blocks from the Design > DHCPv4 menu to open the List/Add DHCP Address Blocks page. Choose the VPN from the username > VPN menu at the top right of the window. In the CLI, use the dhcp-address-block creation and attribute setting commands. For example:

  nrcmd> dhcp-address-block red create 192.168.50.0/24
  nrcmd> dhcp-address-block red set vpn=blue
  nrcmd> dhcp-address-block red set vpn-id=99

- **Clients and client-classes**—In some cases it is best to provision a VPN inside of Cisco Prime IP Express instead of externally, where it might have to be configured for every Cisco IOS device. To support this capability, you can specify a VPN for a client or client-class. Two attributes are provided:
  - default-vpn—VPN that the packet gets if it does not already have a vpn-id or vrf-name value in the incoming packet. You can use the attribute with clients and client-classes.
  - override-vpn—VPN the packet gets no matter what is provided for a vpn-id or vrf-name value in the incoming packet. You can use the attribute with clients and client-classes. Note that if you specify an override VPN on the client-class, and a default VPN for the client, the override VPN on the client-class takes precedence over the default VPN on the client.

At the local cluster—Choose Client-Classes or Clients from the Design > DHCP Settings menu. Create or edit a client-class or client and enter the default-vpn and override-vpn attribute values.

At the regional cluster—Choose Client-Classes from the Design > DHCP Settings menu. Create or pull, and then edit a client-class to enter the default-vpn and override-vpn attribute values.

In the CLI—Use the client-class creation and attribute setting commands. For example:

  nrcmd> client 1,6,00:d0:ba:d3:bd:3b set default-vpn=blue
  nrcmd> client-class CableModem set override-vpn=blue

In a cable modem deployment, for example, you can use the override-vpn attribute to provision the cable modems. The client-class would determine the scope for the cable modem, and the scope would determine the VPN for the uBR. User traffic through the cable modem would then have the vpn-id suboption set and use the specific VPN. The override-vpn value also overrides any default-vpn set for the client.
• **Leases**—List leases, show a lease, or get lease attributes.

In the CLI—To import leases, use `import leases filename`. Each lease entry in the file can include the VPN at the end of the line. If it is missing, Cisco Prime IP Express assigns the [none] VPN. (See also the “Importing and Exporting Lease Data” section on page 23-4.)

```
nrcmd> import leases leaseimport.txt
```

To export the address or lease data to include the VPN, use `export addresses` with the `vpn` attribute, or `export leases` with the `-vpn` option. The VPN value can be the reserved word `global` or `all`:

- **Global**—Any addresses outside the defined VPNs (the [none] VPN).
- **All**—All VPNs, including the [none] VPN.

If you omit the VPN, the export uses the current VPN as set by `session set current-vpn`. If the current VPN is not set, the server uses the [none] VPN.

```
nrcmd> export addresses file=addrexport.txt vpn=red
nrcmd> export leases -server -vpn red leaseexport.txt
```

• **Scopes**—Scopes can include the VPN name or its ID, as described in the “Creating and Editing Virtual Private Networks” section on page 24-19.

**Note**

You cannot change the VPN once you set it at the time of creation of the scope.

• **Subnets**—Listing subnets, showing a subnet, or getting the `vpn` or `vpn-id` attribute for a subnet shows the VPN. See the “Configuring DHCP Subnet Allocation” section on page 24-22.

• **DHCP server**—If the `vpn-communication` attribute is enabled (which it is by default), the DHCP server can communicate with DHCP clients that are on a different VPN from that of the DHCP server by using an enhanced DHCP relay agent capability. This capability is indicated by the `server-id-override` suboption in the relay agent information option (82).

---

**Configuring DHCP Subnet Allocation**

The following section provides an example of setting up subnet allocation using the DHCP server. Figure 20-5 shows a sample subnet allocation configuration with subnets assigned to provisioning devices, along with the conventional DHCP client/server configuration.

Before allocating subnets, the DHCP server first determines what VPN the client is on, in the following order:

1. The server looks for incoming VPN options and uses the value for the VPN.
2. If no VPN options are found, the server uses the relay agent suboption value, then combines the VPN with the subnet address to form the unique identifier.
3. If no relay agent suboption is found, the server looks for client-class information (selection tags).

To configure DHCP subnet allocation:

---

**Step 1**

Create a DHCP address block for a subnet, set the initial subnet mask and its increment, and set other subnet allocation request attributes. Also, associate a policy or define an embedded policy.

- If you use VPNs, you can specify a `vpn` or `vpn-id` attribute (see the “Configuring Virtual Private Networks Using DHCP” section on page 24-18).
The server uses the presence of the **subnet-alloc** DHCP option (220) in the request packet to determine that the packet is a subnet allocation request. You can configure the server to use the **subnet-name** suboption (3) as a selection tag if you set the **addr-blocks-use-selection-tags** attribute for the server or VPN.

You can optionally set a default selection tag by setting the **addr-blocks-default-selection-tags** attribute for the DHCP server or VPN object. This identifies one or more subnets from which to allocate the addresses. If the relay agent sends a VPN string (via a VPN option or relay agent suboption), associated with a subnet, any address block with that string as one of its **addr-blocks-default-selection-tags** values uses that subnet.

The default behavior on the server and for VPNs is that the DHCP server tries to allocate subnets to clients using address blocks that the clients already used. Disabling the **addr-blocks-use-client-affinity** attribute causes the server to supply subnets from any suitable address block, based on other selection data in the clients’ messages.

If you want to support configurations of multiple address blocks on a single LAN segment (analogous to using primary and secondary scopes), add a **segment-name** attribute string value to the DHCP address block. When the relay agent sends a single subnet selection address, it selects address blocks tagged with that **segment-name** string value. However, you must also explicitly enable the LAN segment capability (**addr-blocks-use-lan-segments**) at the server or VPN level.

Instead of associating a policy, you can set properties for the address block embedded policy. As in embedded policies for clients, client-classes, and scopes, you can enable, disable, set, unset, get, and show attributes for an address block policy. You can also set, unset, get, and list any DHCP options for it, as well as set, unset, and list vendor options. Note that deleting an address block embedded policy unsets all the embedded policy properties.

**Step 2** Note that the server allocates subnets based on the relay agent request. If not requested, the default subnet size is a 28-bit address mask. You can change this default, if necessary, by setting the **default-subnet-size** attribute for the DHCP address block.

For example:

```
nrcmd> dhcp-address-block red set default-subnet-size=25
```

**Step 3** You can control any of the subnets the DHCP server creates from the address blocks. Identify the subnet in the form **vpn-name/netipaddress/mask**, with the **vpn-name** optional. Subnet control includes activating and deactivating the subnet as you would a lease. Likewise, you can force a subnet to be available, with the condition that before you do so, that you check that the clients assigned the subnet are no longer using it. First, show any subnets created.

**Step 4** Reload the DHCP server.

---

**VPN and Subnet Allocation Tuning Parameters**

Consider these tuning parameters for VPNs and on-demand address pools.

- **Keep orphaned leases that have nonexistent VPNs**—Cisco Prime IP Express usually maintains leases that do not have an associated VPN in the Cisco Prime IP Express state database. You can change this by enabling the DHCP attribute **delete-orphaned-leases**. The server maintains a lease state database that associates clients with leases. If a scope modification renders the existing leases...
invalid, the lease database then has orphaned lease entries. These are typically not removed even after the lease expires, because the server tries to use this data in the future to reassociate a client with a lease. One downside to this is that the lease database may consume excessive disk space. When you enable the delete-orphaned-leases attribute, such lease database entries are removed during the next server reload. However, be cautious when enabling this attribute, because rendering leases invalid can result in clients using leases that the server believes to be free. This can compromise network stability.

- **Keep orphaned subnets that have nonexistent VPNs or address blocks**—This is the default behavior, although you can change it by enabling the DHCP attribute dhcp enable delete-orphaned-subnets. As the DHCP server starts up, it reads its database of subnets and tries to locate the parent VPN and address block of each subnet. With the attribute enabled, if a subnet refers to a VPN that is no longer configured in the server, or if the server cannot locate a parent address block that contains the subnet, the server permanently deletes the subnet from the state database.

- **Keep the VPN communication open**—This is the default behavior, although you can change it by disabling the DHCP attribute vpn-communication. The server can communicate with clients that reside on a different VPN from that of the server by using an enhanced DHCP relay agent capability. This is signaled by the appearance of the vpn-id suboption of the relay-agent-info option (82). You can disable the vpn-communication attribute if the server is not expected to communicate with clients on a different VPN than the server. The motivation is typically to enhance network security by preventing unauthorized DHCP client access.

### Setting DHCP Forwarding

The Cisco Prime IP Express DHCP server supports forwarding DHCP packets to another DHCP server on a per-client basis. For example, you might want to redirect address requests from certain clients, with specific MAC address prefixes, to another DHCP server. This can be useful and important in situations where the server being forwarded to is not one that you manage. This occurs in environments where multiple service providers supply DHCP services for clients on the same virtual LAN.

Enabling DHCP forwarding requires implementing an extension script. The DHCP server intercepts the specified clients and calls its forwarding code, which checks the specified list of forwarded server addresses. It then forwards the requests rather than processing them itself. You attach and detach extensions to and from the DHCP server by using dhcp attachExtension and dhcp detachExtension.

The DHCP forwarding feature works like this:

1. When DHCP is initialized, the server opens a UDP socket, which it uses to send forwarded packets. To support servers with multiple IP addresses, the socket address pair consists of INADDR_ANY and any port number. This enables clients to use any one of the server IP addresses.

2. When the DHCP server receives a request from a client, it processes these extension point scripts:
   - post-packet-decode
   - pre-client-lookup
   - post-client-lookup

   As the DHCP server processes these scripts, it checks the environment dictionary for this string:

   cnr-forward-dhcp-request

3. When it finds that string and it has the value true (enabled), the server calls its forwarding code.

4. The forwarding code checks the environment dictionary for a string with this key:

   cnr-request-forward-address-list
It expects a list of comma-separated IP addresses with an optional colon-delimited port number, as in this example:

192.168.168.15:1025,192.168.169.20:1027

By default, the server forwards to server-port for DHCPv4 and v6-server-port for DHCPv6. It sends a copy of the entire client request to each IP address and port in turn. If any element in the list is invalid, the server stops trying to parse the list.

5. After the forwarding code returns, the server stops processing the request. In the post-client-lookup extension point script, however, this action might create an optional log message with client-entry details.

The following example of a portion of a TCL extension script tells the DHCP server to forward a request to another server based on the information in the request. You can use such a script if there are multiple device provisioning systems in the same environment. In this case, you would run the extension script on the DHCP server to which routers forward broadcast requests. The script would determine which (if any) other server or servers should handle the request, and tell the original server to forward the request.

The sample script uses a static mapping of MAC address prefix to send modems from a specific vendor to a specific system:

```
proc postPktDecode {req resp env} {
    set mac [\$req get chaddr]
    set addr ""
    ;# Very simple, static classifier that forwards all requests from devices
    ;# with a vendor-id of 01:0c:10 to the DHCP servers at 10.1.2.3 and 10.2.2.3:
    switch -glob -- $mac {
        01:0c:10* {
            set addr "10.1.2.3,10.2.2.3"
        }
    }
    ;# If we decide to forward the packet, the $addr var will have the IP addresses
    ;# where to forward the packet:
    if {$addr != ""} {
        ;# Tell the DHCP server to forward the packet...
        $env put cnr-forward-dhcp-request true
        ;# ...and where to forward it:
        $env put cnr-request-forward-address-list $addr
        ;# No more processing is required.
        return
    }
}
```

A more flexible script could use a per-client configuration object, such as the Cisco Prime IP Express client entry, to indicate which DHCP server should get the request.
Configuring Client-Classes and Clients

Use the Cisco Prime IP Express client and client-class concepts to provide differentiated services to users across a common network. You can group clients based on administrative criteria, and then ensure that each group receives its appropriate class of service (COS). Without client-class processing, the DHCP server provides client leases based solely on their network location.

Related Topics

- Configuring Client-Classes, page 25-1
- Configuring Clients, page 25-9
- Subscriber Limitation Using Option 82, page 25-14
- Configuring Cisco Prime IP Express to Use LDAP, page 25-19

Configuring Client-Classes

You can differentiate client services in the following ways:

- Register clients using the Cisco Prime IP Express database (this section) or the Lightweight Directory Access Protocol (see the “Configuring Cisco Prime IP Express to Use LDAP” section on page 25-19).
- Register intermediary devices (such as cable modems) so that you can differentiate their upstream clients by class of service.
- Use the contents of client packets without the foreknowledge of client data:
  - Known DHCP options that can be in the packet, such as the dhcp-user-class-id DHCP option (77), or the radius-attribute suboption of the relay-agent-info DHCP option (82, see the “Processing Client Data Including External Sources” section on page 25-6).
  - Other data in the packet to extract using an expression in the client-class-lookup-id DHCP server attribute (see the “Calculating Client-Classes and Creating Keys” section on page 25-15).
- Use a two-stage process of first creating a client-class to assign clients, then set a client-lookup-id for certain clients (see the “Expression Processing for Subscriber Limitation” section on page 25-16).

Related Topics

- Client-Class Process, page 25-2
- Defining Client-Classes, page 25-2
- Setting Selection Tags on Scopes and Prefixes, page 25-4
Chapter 25      Configuring Client-Classes and Clients

Configuring Client-Classes

Defining Client-Classes and Their Embedded Policies, page 25-5
Processing Client Data Including External Sources, page 25-6
Troubleshooting Client-Classes, page 25-8

Client-Class Process

Enable or disable client-class processing for the DHCP server and apply a set of properties to groups of clients. With client-class enabled, the server assigns the client to an address from a matching DHCPv4 scope or DHCPv6 prefix. The server acts according to the data in the packet. To configure client-classes:

1. Enable client-class processing for the DHCP server.
2. Define client-classes that include or exclude selection tags (criteria).
3. Apply the selection tags to specific scopes or prefixes (or their templates).
4. Assign clients to these classes.

This process is for clients configured through Cisco Prime IP Express. For processing affected by data from external sources, see the “Processing Client Data Including External Sources” section on page 25-6.

Defining Client-Classes

You enable and define client-classes at the server level.

Local Basic Web UI

**Step 1** Enable client-classes. In the Basic or Advanced mode:

a. From the Deploy menu, choose DHCP Server under the DHCP submenu to open the Manage DHCP Server page.

b. Select the server on the DHCP Server pane.

c. On the Edit DHCP Server tab, enable the client-class attribute.

d. Click Save.

**Step 2** From the Design menu, choose Client Classes under the DHCP Settings submenu to open the List/Add DHCP Client Classes page.

**Step 3** Click the Add Client Classes icon in the Client Classes pane to open the Add DHCP Client Class dialog box.

**Step 4** Enter a name for the client-class.

**Step 5** Set other client-class properties. The hostname and domain name attributes are mainly used for DNS updates if not using a DNS update configuration (see the “Creating DNS Update Configurations” section on page 29-5). The hostname properties are described in the “Defining Client-Class Hostname Properties” section on page 25-5. You can also choose the appropriate policy for the client-class.

**Step 6** Click Add Client Class.

**Step 7** Define the selection criteria.
The critical step in creating a client-class is defining its selection criteria so that you can associate the client-class with a DHCPv4 scope or DHCPv6 prefix. Use the selection-criteria attribute (see also Table 25-1 on page 25-4).

You can enter multiple selection tags by separating them with commas. The values have to match the selection tags set for the desired scope or prefix (see the “Setting Selection Tags on Scopes and Prefixes” section on page 25-4).

**Step 8** To add an embedded policy to the client-class, see the “Editing Client-Classes and Their Embedded Policies” section on page 25-5.

**Step 9** Click *Save*.

**Step 10** Debug as needed. To debug client-class errors, set the DHCP log settings to *client-criteria-processing*.

**Step 11** To delete a client-class, select the client and click the *Delete* icon in the Client Classes pane, and confirm the deletion.

---

**Local Advanced Web UI**

**Step 1** Enable client-classes. In the Basic or Advanced mode:

a. From the *Deploy* menu, choose DHCP Server under the DHCP submenu to open the Manage DHCP Server page.

b. Select the server on the DHCP Server pane.

c. On the Edit DHCP Server tab, enable the *client-class* attribute.

d. Click *Save*.

**Step 2** From the *Design* menu, choose Client Classes under the DHCP Settings submenu to open the List/Add DHCP Client Classes page.

**Step 3** Click the *Add Client Classes* icon in the Client Classes pane to open the Add DHCP Client Class dialog box.

**Step 4** Enter a name for the client-class.

**Step 5** Set other client-class properties. The hostname and domain name attributes are mainly used for DNS updates if not using a DNS update configuration (see the “Creating DNS Update Configurations” section on page 29-5). The hostname properties are described in the “Defining Client-Class Hostname Properties” section on page 25-5. You can also choose the appropriate policy for the client-class.

**Step 6** Click *Add DHCP Client Class*.

**Step 7** Define the selection criteria.

The critical step in creating a client-class is defining its selection criteria so that you can associate the client-class with a DHCPv4 scope or DHCPv6 prefix. Use the selection-criteria attribute (see also Table 25-1 on page 25-4).

You can enter multiple selection tags by separating them with commas. The values have to match the selection tags set for the desired scope or prefix (see the “Setting Selection Tags on Scopes and Prefixes” section on page 25-4).

**Step 8** To add an embedded policy to the client-class, see the “Editing Client-Classes and Their Embedded Policies” section on page 25-5.

**Step 9** Click *Save*.

**Step 10** Debug as needed. To debug client-class errors, set the DHCP log settings to *client-criteria-processing*.
Step 11 To delete a client-class, select the client and click the **Delete** icon in the Client Classes pane, and confirm the deletion.

---

**CLI Commands**

Enable client-classes by using `dhcp enable client-class`. To create the client-class, use `client-class name create`. The name should clearly identify its intent. It is not case-sensitive; classPC is the same as Classpc.

Set properties of the clients in the client-class by using `client-class name set attribute=value`. For example, set the desired policy to associate with the client-class by using `client-class name set policy-name=value`. Associate a scope with the client-class by using `client-class name set selection-criteria`. (See the “Setting Selection Tags on Scopes and Prefixes” section on page 25-4).

Show the properties of a created client-class by using `client-class name [show]`. You can also list the properties for all the client-classes created, or list just their names. To debug the client-class processing, use `dhcp set log-settings=client-criteria-processing`. To delete the client-class, use `client-class name delete`.

---

**Setting Selection Tags on Scopes and Prefixes**

To assign clients to different address pools, you must define the DHCPv4 scope (or template) or DHCPv6 prefix (or template) with the selection tags that you specified in the selection-criteria for the client-class. All the selection-criteria tags that the client-class has must match the tags the scope or prefix has, even though the scope or prefix might have additional tags. If the client-class omits all selection-criteria, no limitations apply to the scope or prefix selection.

For example:

Scope A has tag1, tag2
Scope B has tag3, tag4

Assuming both scopes are on the same network, a client in a client-class with:

- Tag1, tag2, or both, gets leases from scope A.
- Tag3, tag4, or both, gets leases from scope B.
- One or more tags from both scopes (such as tag1 and tag3) does not get leases from either scope.
- No tags gets leases from either scope.

Table 25-1 describes the attributes Cisco Prime IP Express uses to refer to selection tags or selection criteria for network objects.

<table>
<thead>
<tr>
<th>Object</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>selection-criteria</td>
</tr>
<tr>
<td>Client-class</td>
<td>selection-criteria</td>
</tr>
<tr>
<td>Scope</td>
<td>selection-tag-list</td>
</tr>
<tr>
<td>Scope template</td>
<td>selection-tag-list</td>
</tr>
<tr>
<td>Prefix</td>
<td>selection-tags</td>
</tr>
<tr>
<td>Prefix template</td>
<td>selection-tags</td>
</tr>
</tbody>
</table>
Configuring Client-Classes and Clients

Chapter 25

Configuring Client-Classes

<table>
<thead>
<tr>
<th>Table 25-1</th>
<th>Selection Tag and Criteria Attributes Used (continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>Attribute</td>
</tr>
<tr>
<td>Address block</td>
<td>selection-tags</td>
</tr>
<tr>
<td>Subnets</td>
<td>selection-tags</td>
</tr>
</tbody>
</table>

Local Basic or Advanced Web UI

Create or edit a scope or prefix or its template; on the Add or Edit page for the scope or prefix (or its template), find the Selection Tags attribute and enter a list of comma-separated selection tags created in the selection-criteria attribute for the client-class that you want to associate with this scope or prefix (or its template). Then save the changes and reload the DHCP server, if necessary.

CLI Commands

Use `scope name set selection-tag-list`. For a scope template, use `scope-template name set selection-tag-list`. For a prefix, use `prefix name set selection-tags`. For a prefix template, use `prefix-template name set selection-tags`.

Defining Client-Class Hostname Properties

You can specify the hostname that each client should adopt, using the Hostname (host-name) attribute of the client-class. This can be an absolute, valid DNS value to override the one included in the DHCP client request, or can be any of these:

- @host-name-option—The server uses whatever hostname option the client sent.
- @no-host-name-option—The server ignores the hostname that the client sends. If DNS name generation is in effect, the server uses a generated name, if set up as such for dynamic DNS updating.
- @use-macaddress—The server synthesizes a hostname from the client MAC address, hyphenates the octets, then adds an x at the front. For example, if a client MAC address is 1,6:00:d0:ba:d3:bd:3b, the synthesized hostname would be x1-6-00-d0-ba-d3-bd-3b.

If you omit a value, the hostname is unspecified. You can also synthesize hostnames by using a DNS update configuration (see the “Creating DNS Update Configurations” section on page 29-5).

Related Topics

- Editing Client-Classes and Their Embedded Policies, page 25-5
- Processing Client Data Including External Sources, page 25-6
- Troubleshooting Client-Classes, page 25-8
- Subscriber Limitation Using Option 82, page 25-14
- Configuring Cisco Prime IP Express to Use LDAP, page 25-19

Editing Client-Classes and Their Embedded Policies

Editing a client-class involves the same attributes as creating a client-class. You can also add and modify an embedded policy for the client-class so that you can set its policy options. The embedded policy has no properties or DHCP options associated with it until you add them. (See also the “Creating and Editing Embedded Policies” section on page 22-7). The client-class embedded policy setting is the third priority the DHCP server uses in its policy selection, after that set for the client itself (see the “Policy Hierarchy”
Local Advanced Web UI

Step 1  Create the client-class.
Step 2  Select the client-class on the List/Add DHCP Client Classes page to open the Edit DHCP Client Class page.
Step 3  Make changes to attribute settings as required.
Step 4  To add a new embedded policy for the client-class, click Create New Embedded Policy. If there is an existing embedded policy that you want to edit, click Edit Existing Embedded Policy. Both actions open the Edit DHCP Embedded Policy for Client-Class page. (If you want to unset the existing embedded policy, click Unset on the Edit DHCP Client-Class page; this resets the button to Create New Embedded Policy.)

a. Modify the fields, options, and attributes on this page. For example, under the DHCPv4 Options, set the client lease time by choosing dhcp-lease-time [51] from the drop-down list, enter a lease interval value in the Value field, then click Add Option. If necessary, unset attribute values.

Step 5  Click Save.

CLI Commands

To check if there are any embedded policy values already set for a client-class, use client-class-policy client-class-name show. To set the attributes for the embedded policy, use client-class-policy client-class-name set attribute=value. To set the DHCP options, use client-class-policy client-class-name {setOption | setVendorOption | setV6Option | setV6VendorOption} option value. To set the lease time, use client-class-policy client-class-name setLeaseTime value.

Processing Client Data Including External Sources

Information about network hosts running DHCP clients and their users can arrive at the DHCP server from several external sources. The server can use this data as part of client-class processing, and capture it in its lease database to make it available to the Cisco Prime IP Express management system.

Recently introduced external factors that can influence client definitions are:

- A subscriber-id suboption of the relay-agent-info DHCP option (82), whereby a network administrator defines a network subscriber or client and sends this data to the DHCP server.
- RADIUS authentication server data, as part of 802.1x protocol deployments where the RADIUS data can be helpful in DHCP decision making. In this case, a device can send the data as part of radius-attribute suboption attributes in the relay-agent-info DHCP option (82).

Both these external options use DHCP option 82, as described in the “Subscriber Limitation Using Option 82” section on page 25-14. The RADIUS source can end the following attributes:

- Client user or account name (the user attribute)
- Administratively defined class string (the class attribute)
- Vendor-specific data (the vendor-specific attribute)
- Session timeout value (the session-timeout attribute)
- IP address pool to use for the client (the framed-pool attribute)
- IPv6 address pool to use for the client (the framed-ipv6-pool attribute)

Cisco Prime IP Express provides extension support for the subscriber-id suboption and the user, class, and framed-pool attributes of the RADIUS suboption, and expression support for all of the suboptions (see Chapter 26, “Using Expressions”). Additionally, the DHCP server now includes attribute settings to configure how the server handles the RADIUS class and framed-pool attributes. Cisco Prime IP Express can use the server attributes to map the RADIUS attribute value as a selection tag or client-class name, or append the value to the selection tag that it finds in its client database. For example:

```
nrcmd> dhcp set map-radius-class=append-to-tags
```

For client-classes and selection tags determined from external resources such as RADIUS, the processing order is slightly more complex than that described in the “Client-Class Process” section on page 25-2. See the following subsections. Remember that to use the client-class feature, you must enable the DHCP server client-class attribute.

**Related Topics**

Processing Order to Determine Client-Classes, page 25-7  
Processing Order to Determine Selection Tags, page 25-7

**Processing Order to Determine Client-Classes**

The order in which the DHCP server uses possible sources to determine client-class names is as follows:

1. It uses the client-class name in the extension environment dictionary.
2. If it finds a real client-entry in the database, it uses its client-class-name. (To prevent this unnecessary database read, enable the skip-client-lookup DHCP server attribute; see the “Skipping Client Entries for Client-Classing” section on page 25-13.)
3. If you map the RADIUS framed-pool value to a client-class (by using dhcp set map-radius-pool-name=map-as-class), it uses the framed-pool value.
4. If you map the RADIUS class value to a client-class (by using dhcp set map-radius-class=map-as-class), it uses the class value.
5. If you map the dhcp-user-class-id DHCP option (77) to a client-class (by using dhcp set map-user-class-id=map-as-class), it uses the option value. (Note that you can alternatively use a lookup ID expression instead of this mapping; see the “Client-Class Lookup Expression Processing” section on page 25-16.)
6. If it finds no mapping or user-class ID, it uses the default-client-class-name from the environment dictionary.
7. If it finds no default-client-class-name or client-entry, it uses the client-class-name from the client named default (if found).

**Processing Order to Determine Selection Tags**

The order in which the server uses the possible sources to determine selection tags (it uses the first nonnull source) is as follows:

1. Selection tags in the extension environment dictionary.
2. If it finds a real client-entry in the database, it uses the client-entry selection-tags. (To prevent this unnecessary database read, enable the skip-client-lookup DHCP server attribute; see the “Skipping Client Entries for Client-Classing” section on page 25-13.)
3. Selection tags in the client-class.
4. If you map an available RADIUS framed-pool value to a tag (by using `dhcp set map-radius-pool-name=map-as-tag`), it uses that tag.
5. If you map an available RADIUS class value to a tag (by using `dhcp set map-radius-class=map-as-tag`), it uses that tag.
6. If you map an available `dhcp-user-class-id` DHCP option (77) to a tag (by using `dhcp set map-user-class-id=map-as-tag`), it uses that tag.

Next, the server could append one of the following to the list of selection tags (if any):
1. If a RADIUS framed-pool value is available and you set the `map-radius-pool` DHCP attribute to append to the tags (by using `dhcp set map-radius-pool=append-to-tags`), the server appends it.
2. If a RADIUS class value is available and you set the `map-radius-class` DHCP attribute to append to the selection tags (by using `dhcp set map-radius-class=append-to-tags`), the server appends it.
3. If a `dhcp-user-class-id` is available and you set the `map-user-class-id` DHCP attribute to append to the selection tags (by using `dhcp set map-user-class-id=append-to-tags`), the server appends it.

**Troubleshooting Client-Classes**

To troubleshoot a client-class, enable client-class logging using the `log-settings` attribute on the Edit DHCP Server page of the web UI, or `dhcp set log-settings=setting` in the CLI, then reload the DHCP server (if in staged `dhcp edit` mode). The recommended settings are:

- **client-detail**—Logs a single line at the end of every client-class client lookup operation. This line shows all the data found for the client as well as the data that was found in the client-class.
- **client-criteria-processing**—Logs a message whenever the server examines a scope or prefix to find an available lease or to determine if a lease is still acceptable for a client that already has one.
- **ldap-query-detail**—Logs messages whenever the DHCP server initiates a lease state entry creation to an LDAP server, receives a response from an LDAP server, or retrieves a result or error message from an LDAP server.
- If the problem could be related to your LDAP server, also enable the LDAP `can-query` setting.

These logs will help answer these questions:

- Is the server reading the client entry from the expected database?
  The server can read the client entry from LDAP or CNRDB (the Cisco Prime IP Express internal database). The `client-detail` log shows you from where the server is reading the client entry.
- Is client-class enabled?
  If enabled but you are getting unexpected results, verify from which database is your Cisco Prime IP Express server reading clients. Is it reading from LDAP or CNRDB? The `ldap-query-detail` log tells you if it is reading from LDAP. If not, enable the DHCP `use-ldap-client-data` property.

  **Note** Using LDAP requires configuring the LDAP server for queries. Enable the LDAP `can-query` attribute. You also must configure the DHCP server to use LDAP for queries.

- Is the server providing clients the right data, but you see the wrong results from that data (for example, clients are not receiving the expected IP addresses)?
Verify the explicit relationships on your network. The client-criteria-processing log shows from which scopes or prefixes the server is getting addresses. If it does not get addresses from the expected sources, explicit relationships might be incorrectly defined. A scope that you thought was a secondary scope might not be defined that way.

- In Expert mode, did you set the include and exclude criteria for selection tags properly?

  If you define a series of selection tags to include, the tags of a scope or prefix must match those of the client. In Expert mode, you can also use a selection-criteria-excluded attribute on the client-class to exclude selection tags. If you define a series to exclude, a scope or prefix must have none of these tags defined so that the client can get configuration parameters from it. Avoid complex inclusion and exclusion scenarios as you begin working with selection tags.

Configuring Clients

DHCP client properties include the participating client-class and associated policy for a client, the action to perform, and the inclusion and exclusion criteria for selection tags. A client inherits the properties from its client-class, which you may choose to override or supplement by specifying different properties for the client.

Local Basic or Advanced Web UI

**Step 1**  
From the Design menu, choose Clients under the DHCP Settings submenu to open the List/Add DHCP Clients page.

**Step 2**  
Click the Add Clients icon in the Clients pane to open the Add DHCP Client dialog box, enter the client identity, typically a MAC address, but it can also be a DUID or lookup key. (Note that you can set up the DHCP server to validate the client name as a MAC address by enabling the server attribute validate-client-name-as-mac.)

You can also create a client named default that does not have a specific client configuration. For example, you can have a client always use its MAC address for its hostname.

**Step 3**  
Select a client-class name, if desired, from the drop-down list of predefined client-classes.

**Step 4**  
Click Add DHCP Client. This opens the Edit DHCP Client page.

The critical step in creating a client is defining its selection criteria so that you can associate the client with a scope or prefix (unless the selection criteria were already set up for the client-class associated with the client).

Use the selection-criteria attribute under the Attribute list (see also Table 25-1 on page 25-4). You can enter multiple selection tags by separating them with commas. The values have to match the selection tags set for the desired scope or prefix (see the “Setting Selection Tags on Scopes and Prefixes” section on page 25-4).

**Note**  
If you chose a client-class for the client, this page does not appear, and the client name is listed on the List/Add Client page.

**Step 5**  
Set other attributes as desired. For example:

- Set the host-name attribute to @no-host-name-option to provide provisional addresses to unknown clients. See the “Allocating Provisional Addresses” section on page 25-12.

- Set the domain name of the zone to use when performing dynamic DNS updates.
Chapter 25  Configuring Client-Classes and Clients

Configuring Clients

- Set the policy and action for the client. With the exclude action, the server ignores all communication from this client (no packets are shown); with the one-shot action, the server does not renew or re-offer a lease to this client.
- Choose the number of time units (seconds, minutes, hours, days, weeks), or UNIX style date (such as Mar 24 12:00:00 2002) to indicate when the authentication expires, or use forever.

Step 6  Click Save at the bottom of the page.
Step 7  Debug as needed. To debug client errors, set the DHCP log settings to client-criteria-processing.
Step 8  To delete a client, click the Delete icon in the Clients pane, and confirm the deletion.

CLI Commands

To create a client, use client name create. To associate a client-class with the client, use client name set client-class-name=value. To set selection criteria for scopes or prefixes, use client name set selection-criteria. To set other attributes, use client name set attribute=value.

To display client properties, use client name [show]. To display properties for all the clients, use client list, or client listnames to list just the names. To debug clients, use dhcp set log-settings=client-detail. To delete a client, use client name delete.

Related Topics

Editing Clients and Their Embedded Policies, page 25-10
Setting Windows Client Properties, page 25-11
Allocating Provisional Addresses, page 25-12
Skipping Client Entries for Client-Classing, page 25-13
Limiting Client Authentication, page 25-13
Setting Client Caching Parameters, page 25-13

Editing Clients and Their Embedded Policies

Editing a client involves the same attributes as creating a client. You can also add and modify an embedded policy for the client so that you can set its policy options. The embedded policy has no properties or DHCP options associated with it until you add them. (See also the “Creating and Editing Embedded Policies” section on page 22-7.) The client embedded policy setting is the first priority the DHCP server uses in its policy selection (see the “Policy Hierarchy” section on page 22-3).

Local Basic or Advanced Web UI

Step 1  Create the client.
Step 2  Select the client from the Clients pane on the List/Add DHCP Clients page to open the Edit DHCP Client page.
Step 3  Make changes to attribute settings as required.
Chapter 25  Configuring Client-Classes and Clients

Configuring Clients

Step 4  
To add a new embedded policy for the client-class, click Create New Embedded Policy. If there is an existing embedded policy that you want to edit, click Edit Existing Embedded Policy. Both actions open the Edit DHCP Embedded Policy for Client page. (This page is almost identical to the Edit DHCP Embedded Policy for Client-Class page.)

a. Modify the fields, options, and attributes on the Edit DHCP Embedded Policy for Client page. For example, under the DHCPv4 Options, set the client lease time by choosing dhcp-lease-time [51] from the drop-down list, enter a lease interval value in the Value field, then click Add Option. If necessary, unset attribute values.

If you want to unset the existing embedded policy, click Unset on the Edit DHCP Client page; this resets the button to Create New Embedded Policy.

Step 5  
Click Save.

CLI Commands

To see if there are any embedded policy values already set for a client, use client-policy client-name show. To create an embedded policy, use client-policy client-name set attribute=value. To set the DHCP options, use client-policy client-name {setOption | setVendorOption | setV6Option | setV6VendorOption} option value. To set the lease time, use client-class-policy client-name setLeaseTime value.

Setting Windows Client Properties

Windows clients support class-based provisioning. You can set certain properties that relate to client-class processing. These are:

- Look up the client entry to determine the default client for client-class processing.
- Map the user class ID to the client-class or selection tag.
- Set whether to append the class ID to the selection tag name.

Settings in Windows Clients

On the Windows client host, use ipconfig /setclassid to set the class ID. If you plan to map this client ID to a client-class or selection tag, it must have the same name. Then confirm by using ipconfig /showclassid. For example:

DOS> ipconfig /setclassid adapter engineering
DOS> ipconfig /showclassid adapter

Settings in DHCP Servers

You must set Windows client properties in the DHCP server.

Use DHCP server attributes in the local cluster web UI or dhcp set command attributes in the CLI to set the Windows client properties for the server. If you set the skip-client-lookup attribute to true (the default is false), the DHCP server skips the client entry for client-class processing. (See the “Skipping Client Entries for Client-Classing” section on page 25-13.) Use one of the map-user-class-id attribute settings:

- 0—I ignore the user class ID (the default)
- 1—Map the user class ID to the selection tag
- 2—Map the user class ID to the client-class
• 3—Append the user class ID to the list of selection tags

### Allocating Provisional Addresses

You can provide provisional addresses to clients.

#### Provisional Addresses for Unknown Clients

The DHCP server can allocate provisional addresses to unknown clients for a short time on a one-shot basis. The server gives an address to the unknown client only as long as its lease period (which should be set short), and the client cannot renew the lease. Once the lease expires, the client cannot obtain a new lease until after the grace period expires (this locks the client out of network access). The idea is to give the client a short time to register and prevent it from using the network if it does not register in that time.

**Step 1**
Create an unknown policy, for example (the name is arbitrary).

**Step 2**
Use the Grace period field on the Edit DHCP Policy page, or the policy unknown create grace-period=extended-time setting in the CLI.

**Step 3**
Use the default client to set the Policy name value to unknown, and the action attribute value to one-shot on the Edit DHCP Client page, or use client default create policy-name=unknown action=one-shot in the CLI, to give provisional addresses to unknown clients.

**Note**
Provisioning unknown clients is not supported in DHCPv6.

#### Using One-Shot Action

Use the one-shot action to allocate provisional addresses. This is useful when you want a client to have an address for only a short time. Configure the default client (or the client-class that the default client specifies) by setting the action attribute to one-shot.

The server then gives a lease to an unknown client, but does not allow it to renew the lease. When the lease expires, the server does not respond to that client during the lease grace period, and only responds when a different client uses the lease. The grace period, therefore, is the minimum period during which the client cannot obtain a lease.

You can allow the client a relatively short lease time, such as one day, and specify a long grace period, such as two weeks. This way you can offer an incentive to the client to register with some authority and become a known client, while not re-allocating the lease to another client. After the lease expires, the client cannot get another address for the lease for the two-week grace period.

You can configure the lease and grace period differently for each scope or prefix, so that provisional leases can have different lease and grace periods than nonprovisional ones. Provisional addresses are less restrictive if you use multiple DHCP servers, because each server operates its one-shot capabilities independently. With the approach described and two DHCP servers, an unregistered client can get two days of provisional address use every two weeks.
Skipping Client Entries for Client-Classing

You may not want to honor client entries for client-classing to prevent unnecessary database reads. To accomplish this, enable the `skip-client-lookup` DHCP server attribute (`dhcp enable skip-client-lookup` in the CLI).

Limiting Client Authentication

By default, client entries get unlimited authentication. Using the `authenticate-until` attribute, you can limit authenticating a client entry by specifying an expiration time.

When a client entry is no longer authenticated, the DHCP server uses the `unauthenticated-client-class-name` attribute value for the name of the client-class entry to use in answering this DHCP request. If this attribute is unset, or if there is no client-class entry in it, the DHCP server ignores the request.

The valid client authentication values are:

- `+num unit`—Time in the future, where `num` is a decimal number and `unit` is `s`, `m`, `h`, `d`, or `w` for seconds, minutes, hours, days or weeks, respectively. For example, “+3w” is three weeks in the future.
- `date`—Month, day, 24-hour, and 2-or-4-digit-year. For example, “Jun 30 20:00:00 2002.” Enter the local process time. If the server runs in another time zone, disregard the time zone and use local time instead.
- `forever`—Does not expire the authentication for this client.

An example follows of using the `authenticate-until` attribute to distinguish between clients that are authenticated and those that are not authenticated. After the authentication expires and the client requests another address, the DHCP server assigns the client an address from the unauthenticated scope range:

```
Step 1  Create an authenticated and an unauthenticated client-class. Set the selection criteria for each as appropriate.
Step 2  Create the client and include the authenticate-until expiration time. Set the `client-class-name` and `unauthenticated-client-class-name` attributes as appropriate.
Step 3  Create the authenticated and unauthenticated scopes, define their address ranges, and tie them to their respective selection tags.
Step 4  Enable client-class processing for the server.
Step 5  If necessary, reload the DHCP server.
```

Setting Client Caching Parameters

The initial request from a client for an address from a DHCP server often goes through a `DHCPDISCOVER-DHCPOFFER-DHCPREQUEST-DHCPACK` cycle. This process requires that the server must consult the database twice per request for client data. If the client caching parameters are set, the DHCP server caches client data in memory so that it only needs to consult the database once. Client caching can provide a noticeable performance improvement in systems that store client information in LDAP. Client caching is enabled by default unless you unset the applicable attributes.
You can adjust the maximum cache count and time-to-live (TTL) parameters based on the expected rate of client requests. If you expect an onslaught of requests, you might want to increase the cache count, up to a limit based on your available memory. If you expect a longer request cycle, you might want to increase the TTL. The aim is to have the server consult the client cache once during the request cycle.

To set the limit on the number of entries that the server keeps in the client cache, use the `client-cache-count` attribute on the Edit DHCP Server page, or `dhcp set client-cache-count` in the CLI. By default, the maximum number to cache is 1000 clients. To disable the cache, set the attribute to 0.

The client cache is usually valid for only ten seconds, called the cache TTL. After the TTL expires, the server reads the client information from the database, if necessary. You can adjust the TTL using the `client-cache-ttl` attribute on the Edit DHCP Server page, or `dhcp set client-cache-ttl` in the CLI.

When the client cache count reaches the specified maximum, the server cannot cache any more clients until a client-entry TTL expires, after which it reads from the database and begins caching again.

The DHCP server, by default, caches client data while processing DISCOVER message only. If you want to cache client data during REQUEST (Renew or Rebind) message, you need to set `cache-client-for-requests` attribute to true. This attribute can be configured on the Edit DHCP Server page, or using `dhcp set cache-client-for-requests` in the CLI. This attribute should be set to true only if the duration between the two REQUEST (Renew or Rebind) messages are lesser than the cache TTL.

### Subscriber Limitation Using Option 82

In many situations, service providers want to limit the number of IP addresses the DHCP server should give out to devices on customer premises. They want these devices to have “real” addresses that the DHCP server provides, but limit their number. One way is to use the client-class to register (or provision) each customer device so that the server issues IP addresses only to devices that are registered in the client-entry database. The major drawback to this approach is that it requires registering every customer device, which involves knowing its MAC address. Service providers often do not want to know about each device, but simply that there are not too many of them per customer.

Another approach is to limit customer devices on a per-subscriber basis on values in the `relay-agent-info` DHCP option (option 82, as described in RFC 3046) that the DHCP relay agent sends in a DHCPDISCOVER message. This option includes data about the port on a switch over which the customer device is attached. In a cable modem scenario, one of the option 82 suboptions usually contains the MAC address of the cable modem when the DHCP request comes from a device attached beyond the cable modem. In general, many devices that generate option 82 data place some values in its suboptions such that the value varies per subscriber on the same upstream device. In some cases, this value is unique across all possible subscribers (such as the MAC address of the cable modem). In others, it can be a port on a switch and thus unique across the other subscribers attached to that switch. However, it might not be unique across all subscribers on the switch.

Using this approach, the network administrator can configure limitations on subscriber use of the DHCP-allocated addresses without seriously impacting other DHCP server capabilities. In many environments, network administrators might want to use option 82 limitation for some class of devices and not others. A key aspect of this support is to allow network administrators to separate the devices for which they want to use option 82 limitation from those for which they do not.

### Related Topics

- General Approach to Subscriber Limitation, page 25-15
- Typical Limitation Scenario, page 25-15
- Calculating Client-Classes and Creating Keys, page 25-15
- Client-Class Lookup Expression Processing, page 25-16
General Approach to Subscriber Limitation

The current approach to client processing is to look up every client in the client-entry database. One of the goals of option 82 limitation is to remove the need explicitly to register (provision) every customer device in the client-entry database (either in the CNRDB or LDAP). However, there is still a requirement that the specific number to which a subscriber is limited should be configurable and override the default number given to all unregistered subscribers.

Note

Limitation processing is not currently available for DHCPv6 clients.

At a high level, you can configure subscriber limitation by creating an expression that the server evaluates for each incoming packet and returns the name of the client-class where you want the client to go (see Chapter 26, “Using Expressions,” for details on the use of expressions.). Each client-class allows specification of a limitation identifier (ID), a key the server determines from the incoming packet and uses in later processing to actually limit the number of devices. The server considers all devices with the same limitation ID (the limitation-id property) to come from the same subscriber.

Typical Limitation Scenario

For example, an incoming packet might be evaluated such that:

1. If the remote-id suboption of option 82 matches the client hardware address (chaddr), the subscriber is a cable modem and should be in the cm-client-class.
2. If the first six bytes in the dhcp-class-identifier option value match the string docsis, then the subscriber is a DOCSIS modem and should be in the docsis-cm-client-class.
3. If the user-class option value matches the string alternative-class, then the subscriber should be in the alternative-cm-client-class.

Calculating Client-Classes and Creating Keys

You set the expression that determines the client-class for the client-class-lookup-id attribute of the DHCP server, or dhcp set client-class-lookup-id=expression in the CLI. Include simple expressions in the attribute definition or more complex ones in a file referenced in the attribute definition (see the “Using Expressions” section on page 26-1).

Clients and client-classes also allow specifying a limitation-id value for the client or client-class. The server uses this identifier (ID) value to set the address limit on the number of devices with the identical ID on the same network or LAN segment. If a requesting client oversteps the limit of available addresses for that ID, the server assigns it to an over-limit-client-class-name (if set); otherwise, it drops the packet. The limitation-id, in effect, defines a subscriber.
Client-Class Lookup Expression Processing

The initial client-class lookup is to allow you to decide whether the client should participate in some sort of limitation. Configure an expression server-wide with the client-class-lookup-id attribute. The server executes this expression on every incoming packet with the goal of determining the client-class of the packet.

The expression should return a string that is the client-class name for the packet, or the distinguishing string <none> indicating that no client-class value was considered for the client request. Returning the <none> string is equivalent to not configuring a client-class-lookup-id value and that no client-class processing should occur. If the expression returns null or there is an error evaluating the client-class-lookup-id, the server drops the packet (with an accompanying log message).

Limitation Processing

The DHCP server limits the number of IP addresses allocated to DHCP clients with the same limitation-id value in the same network or LAN segment. In cases where the server finds that allocating another address to the client would go over the limit, it places the client packet in the overflow-client-class (if any is specified). This allows special handling for clients that are over the configured limit. Handling these clients in some self-provisioning way is one of the benefits of using limitation on the DHCP server instead of in the hardware (should it even be supported).

If there is no over-limit client-class, the server drops a packet where allocating an address for that packet would exceed the allowed limitation-count for that limitation-id. Note that the server enforces the limitation only in a single network or LAN segment. This is hardly a restriction, because network managers tend to see a single subscriber connecting only over one LAN segment at a time.

Configure the limitation-count with an identical limitation-id in a DHCP policy. The limitation code searches up the policy hierarchy for the limitation-count just as it does for any other policy item. This means that you can configure the limitation-count in a client-class embedded or named policy, a scope embedded or named policy, or the system system_default_policy.

When you configure a limitation-id on a client-class, you thereby signal to pursue limitation processing for the client-class. When you do not configure a limitation-id, you thereby signal not to pursue it. When executing the expression to determine the limitation-id, if the expression returns null, this signals that limitation processing should occur and to use the limitation-id saved in the lease state database.

Expression Processing for Subscriber Limitation

Expressions exist in several places in the limitation processing. Each expression evaluates to null or a string (typically to determine a client-class name when looking up a client-class), or to a series of bytes (a blob) when creating a limitation-id. You can use expressions in these places:

- Looking up a client-class
- Creating the key to limit clients of the same subscriber (the limitation-id)
- Creating the key to look up in the client-entry database (the client-lookup-id)
Configuring Option 82 Limitation

Step 1  If you do not register clients explicitly, do not enable client-class as a DHCP server property when using option 82 data.

Step 2  Determine if you want to limit some clients and not others. If you want to limit some clients:
   a. Find some method to distinguish these clients from the others, based on some values contained in the DHCP requests from each class of clients.
   b. Determine the names of the client-classes into which you want to put the clients that are not limited, and the selection tag and scope or scopes you want to use for these unlimited clients.

Step 3  Decide if you want to put clients that are over-limit into a different client-class or just drop their packets. If you want to put them in an over-limit client-class, determine the client-class name and the selection tag and scope or scopes into which you want to put the over-limit clients.

Step 4  Determine the client-class into which you want to put clients that you intend to limit and the selection tags and scope or scopes you want to use for these clients.

Step 5  Create all these selection tags, client-classes, and scopes.

Step 6  Configure the limitation-count in a policy, probably the named policy associated with the client-class for the clients to limit.

Step 7  Write the expression to separate the incoming clients into those to be limited and those not to be limited. Configure it on the DHCP server by setting the client-class-lookup-id attribute.

Step 8  Write the expression to determine the limitation ID for the devices to limit, and configure it on the client-class for clients to limit by setting the limitation-id.

Lease Renewal Processing for Option 82 Limitation

Only packets that the DHCP client broadcasts arrive at the server with option 82 data attached. The BOOTP or DHCP relay agent adds the option 82 data in the first upstream router from the client device. A DHCPRENEW packet is unicast to the server and arrives without option 82 data. This can pose a problem when trying to configure the server for subscriber limitation.

There are generally two approaches to take when dealing with renewals:

- Place all packets that do not have option 82 data in a client-class with no associated selection tags. This is equivalent to a wildcard selection and means that any packet with no option 82 data is accepted.
- Place a DHCPRENEW in the same client-class as you would place a packet that has option 82 data, and have its limitation-id evaluate to null. This is a signal that when checking for limitation, the DHCP server should use a previously stored limitation-id instead of one from the packet.

Both approaches work. The second one can be more secure, but in practice, it is not much better than the first. This is because you have to use an IP address for the DHCP server to respond to a DHCPRENEW, and most clients would not do this unless the server loses some of its state. In this case, you would want it to give the client the address. In the case of a malicious client, it would still have to use the address to get the server to give the address to the client, thereby limiting the exposure for this case.
Administering Option 82 Limitation

Whenever a client is involved in limitation because of its inclusion in a client-class with a limitation-id, the limitation ID used appears in the DHCP log file whenever client data logging occurs as “... LID: nnn:nnn:nnn...” The data is logged only for clients with active leases that are currently occupying one of the limitation-count counts.

You can determine all the clients using a limitation-id in a subnet. On the Manage DHCP Server page, click the Run icon in the Commands column to open the DHCP Server Commands page. Enter at least the IP address of the currently active lease in the IP Address field, then click the Run icon. You can also enter the limitation-id itself in the form nn:nn:nn or as a string (“nnnn”), in which case the IP address becomes the network in which to search. In the CLI, use dhcp limitationList:

```bash
nrcmd> dhcp limitationList ipaddr [limitation-id] show
```

If you specify both the ipaddr and limitation-id, the ipaddr value, the server uses it just like a giaddr to determine the subnet. You can use any IP address that could appear in any scope (primary or secondary) for the network to specify a subnet. If you specify only the ipaddr, it must be an address that the DHCP server serves, and the command returns all the clients and corresponding leases they use.

If a client is denied service due to a limitation-count overflow, a message such as this appears in the DHCP server log file:

```
Warning Server 0 05646 Could not add Client MAC: ‘1,6,01:02:03:04:0c:03’ with limitation-id: 01:02:03 using Lease: 10.0.0.23, already 3 Clients with that id. No over-limit client class specified! Dropping packet!
```

You can determine which clients are extended beyond the limitation-count, thus causing a denial of service for any new client, by using dhcp limitationList. The ipaddr value in the command should be the “using Lease:” value, and the limitation-id should be the “limitation-id:” value, in the log file. Using the log file example, the command would be:

```bash
nrcmd> dhcp limitationList 10.0.0.23 01:02:03 show
```

Troubleshooting Option 82 Limitation

There are several ways that you can debug limitation support. First, you might want to turn on packet tracing by setting the DHCP server debug value to VX=1 (or using dhcp setDebug VX=1). (The VX=0 debug value disables packet tracing.) Then, you probably want to enable client-class debugging by adding client-criteria-processing and client-detail to your log settings.

There is also a server-wide expression trace level, expression-trace-level, that you can set to various levels. Setting it to 6 gives you a details trace of every expression evaluation. This can take a bit of space in the log, and slows down the server considerably as well, but is invaluable in the process of getting familiar with expression evaluation. See the “Debugging Expressions” section on page 26-26.

When things seem to be going strangely, or when submitting log files to report a problem, it is important to enable some additional tracing by setting the DHCP server debug value to QR57=9 (or using dhcp setDebug QR57=9). (The QR57=0 debug value disables this tracing). Note that the Q and R are both uppercase. The Q is client-class debugging and the R is response debugging (required to get the flow of control clear in the log). The 5 is expression processing and the 7 is client-class-lookup processing. This generates a page or so of output for each packet, which will help you understand what is going on inside the server.
Expression Examples

See the “Expression Examples” section on page 26-22.

Configuring Cisco Prime IP Express to Use LDAP

The Lightweight Directory Access Protocol (LDAP) provides directory services to integrate Cisco Prime IP Express client and lease information. By building on your existing standard schema for objects stored in LDAP directories, you can handle information about DHCP client entries. Thus, instead of maintaining client information in the DHCP server database, you can ask the Cisco Prime IP Express DHCP server to issue queries to one or more LDAP servers for data in response to DHCP client requests, or write lease data to an LDAP server.

Cisco Prime IP Express on Windows use the Oracle Directory Server Enterprise Edition LDAP Software Development Kit (SDK) version 5.0. Linux uses their OpenLDAP client distribution.

Related Topics

About LDAP Directory Servers, page 25-19
Adding and Editing LDAP Remote Servers, page 25-19
Configuring DHCP Client Queries in LDAP, page 25-20
Configuring DHCP LDAP Update and Create Services, page 25-23
Troubleshooting LDAP, page 25-29

About LDAP Directory Servers

LDAP directory servers provide a way to name, manage, and access collections of attribute/value pairs. You can enter information into your LDAP server in any number of ways, because Cisco Prime IP Express does not depend on specific LDAP object classes or schema:

- You can store DHCP client information in unused attributes. For example, you could use the givenname attribute to hold the DHCP client-class name value.
- You can add new attributes to an object class without altering your LDAP schema if you disable LDAP schema checking. For example, you could add the client-class-name attribute to the organizational person object class.
- You can create a new object class and define the appropriate attributes. For example, you can create the DHCP client object class and define the client attributes that you want to use.

When you configure the DHCP server to read from LDAP, a query dictionary tells the server which LDAP attributes to query for. The server converts the resulting data into DHCP client data attributes.

Tip

You can configure Cisco Prime IP Express to generate SNMP traps when an LDAP server stops responding or resumes responding to requests from the DHCP server.

Adding and Editing LDAP Remote Servers

You must add a remote LDAP server so that you can begin using the LDAP services.
Chapter 25 Configuring Client-Classes and Clients

Configuring Cisco Prime IP Express to Use LDAP

Local and Regional Advanced Web UI

From the Deploy menu, choose LDAP under the DHCP submenu to open the List/Add LDAP Remote Servers page. Click the Add LDAP icon in the LDAP pane to open the Add DHCP LDAP Server dialog box. To edit the remote server, select the LDAP in the LDAP pane to open the Edit LDAP Remote Server page.

On this page, provide at least the name and fully qualified domain name of the LDAP server. The username and password are required for successful operation.

Note

The Query Settings and the Create Settings will be used in the local for DHCP lease while the same settings will be used in the regional for BYOD client creation.

CLI Commands

Use `ldap name create domain-name`. For example:

```
nrcmd> ldap ldap-1 create ldap.example.com
```

Configuring DHCP Client Queries in LDAP

You can configure and unprovision DHCP client queries, and configure embedded policies, in an LDAP client entry.

Configuring DHCP-Server-to-LDAP Client Queries

To enable the DHCP server to query your LDAP server for client data, perform the following steps. Like local client entries, LDAP client entries are keyed by the client MAC address.

Note

When connecting to an LDAP server, use the distinguished name (dn) of the user. It uniquely identifies an object in the LDAP schema, and is like a unique key in a database or a fully qualified path name for a file. For example, a dn for a person might be dn: cn=Beth Jones, ou=Marketing, o=Example Corporation. In this company, there may be many people named Beth and many people named Jones, but no one else named Beth Jones works in Marketing at Example Corporation.

Step 1
Supply a hostname for the LDAP server. On the Add LDAP Remote Server page, enter a value in the name field. In the local CLI, use this command:

```
nrcmd> ldap ldap-1 create ldap.example.com
```

Later, if you need to delete the server, use `ldap server delete`.

Step 2
Configure the connection credentials. Use the distinguished name (dn) for the user. Enter a value in the username field. In the CLI, use this command, for example:

```
nrcmd> ldap ldap-1 set username="cn=joe,o=Example Corp,c=US" password=access
```

Step 3
Set the search path (and, if necessary, the search scope). The path is a point in the directory from which to start searches. If the search scope is:

- SUBTREE, the server searches all the children of the search path.
- ONELEVEL, the server searches only the immediate children of the base object.
BASE, the server searches only the base object itself.

This example sets the base of the search to be the organization Example Corp and the country US, with a subtree search scope. Enter a value in the search-path field. In the CLI, use this command, for example:

```
nrcmd> ldap ldap-1 set search-path="o=Example Corp,c=US" search-scope=SUBTREE
```

**Step 4** Set the search filter to be the attribute for which DHCP will substitute the clients’ MAC addresses. In this example, the attribute is the common name (cn). Enter a value in the search-filter field. In the CLI, use this command, for example:

```
nrcmd> ldap ldap-1 set search-filter=(cn=%s)
```

**Step 5** Configure a query dictionary that contains all the LDAP-to-DHCP mappings. Use `ldap servername setEntry` to set these mappings.

a. Retrieve the DHCP surname from the `sn` LDAP attribute:

```
nrcmd> ldap ldap-1 setEntry query-dictionary sn=host-name
```

b. Retrieve the client-class name from the first `givenname` LDAP attribute:

```
nrcmd> ldap ldap-1 setEntry query-dictionary givenname=client-class-name
```

c. Retrieve the domain name from the `localityname` LDAP attribute:

```
nrcmd> ldap ldap-1 setEntry query-dictionary localityname=domain-name
```

d. If you need to unset any of the entries, use `ldap server unsetEntry attribute key`. You can also check any of the settings using `ldap server getEntry attribute key`.

**Step 6** Enable queries for the LDAP server. This example enables queries for myserver. Set the `can-query` attribute to enabled. In the CLI, use this command:

```
nrcmd> ldap ldap-1 enable can-query
```

**Step 7** Enable client-class processing for the DHCP server. On the Edit DHCP Server page, set the `client-class` attribute to enabled. In the CLI, use this command:

```
nrcmd> dhcp enable client-class
```

**Step 8** Enable the DHCP server to use LDAP for client entry queries. On the Manage DHCP Server page, set the `client-class` attribute to enabled. In the CLI, use this command:

```
nrcmd> dhcp enable use-ldap-client-data
```

**Step 9** If you have more than one LDAP server configured, you can also set them to operate in round-robin or failover mode:

- **round-robin**—The LDAP servers’ preference values are ignored and all servers that are configured to handle client queries and accept lease state updates are treated equally.

- **failover**—The DHCP server uses the active LDAP server with the highest preference (lowest preference number). If the preferred server loses its connection or fails, the DHCP server uses the next LDAP server of lower preference (increasing preference number). If the preference values are the same (or not set), the DHCP reverts to round-robin mode with these servers.

Set the LDAP server mode by setting the `ldap-mode` on the Edit DHCP Server page. LDAP failover mode actually performs preferential load balancing. The DHCP server assesses the LDAP connection and error states and how fast the LDAP server responds. In an optimal state, the DHCP server uses the LDAP server with the highest assigned preference (lowest preference number). In a less-than-optimal state, the DHCP server uses the next LDAP server of lower preference (increasing preference number). If the preference values are the same (or unset), the DHCP server reverts to round-robin mode.
In the CLI, use `dhcp set ldap-mode` to set the mode, and `ldap server set preference` to set the server preferences; for example:

```
ncmd> dhcp set ldap-mode=failover
ncmd> ldap ldap-1 set preference=1
ncmd> ldap ldap-2 set preference=2
```

Note also that, depending on how many threads you have open, as set by using the `connections` attribute (see the “Recommended Values for LDAP” section on page 25-30) between the DHCP and LDAP servers, the DHCP server opens only as many threads as it can before the `query-timeout` expires. The LDAP server might be processing these threads, but it is not servicing the request, because the failover server has now taken over.

**Step 10** Show or list the LDAP configuration. Go to the List/Add LDAP Remote Servers page. In the CLI, use:

```
nrcmd> ldap ldap-1
nrcmd> ldap list
nrcmd> ldap listnames
```

**Step 11** Reload the DHCP server.

---

### Unprovisioning Client Entries

You can unprovision LDAP client entries so that the client information remains in LDAP, but the DHCP server treats the client as if that information does not exist. The DHCP server then supplies the client with the default behavior. Configure the search filter set in Step 4 of the preceding section so that the LDAP server does not return a client entry containing a specified attribute with a value.

If you want to unprovision the LDAP entry `givenname`, configure the search filter accordingly. For example:

```
nrcmd> ldap ldap-1 set search-filter=(&(cn=%s)(!(givenname=unprovision)))
```

Whenever the `givenname` attribute in the LDAP client entry is set to the “unprovision” string, the LDAP server does not return the client entry to the DHCP server. In other words, the DHCP server treats the client as if it has no LDAP client entry. This procedure has no measurable performance impact on either the DHCP or the LDAP server.

### Configuring Embedded Policies in LDAP

**Step 1** Configure an LDAP server for the DHCP server, naming it myserver, for example.

**Step 2** Map the LDAP attribute that you want the DHCP server to interpret as the embedded policy to the internal `embedded-policy` property. This example maps the `businessCategory` LDAP attribute:

```
nrcmd> ldap myserver setEntry query-dictionary businessCategory=embedded-policy
```

**Step 3** Add a string to the LDAP attribute that the DHCP server can interpret as an embedded policy. The most practical way to determine what this string should look like is to create a dummy client in the Cisco Prime IP Express database and extract data from the client embedded policy setup. Note that this dummy client will never be used, because you are using LDAP, and you can subsequently delete it. Have the embedded policy include the option data types that you need.

a. For example, create an embedded client policy for dummy client `1,6,00:d0:ba:d3:bd:3b`. Add some reply options and a multivalue option (routers) with an IP address data type:

```
nrcmd> client 1,6,00:d0:ba:d3:bd:3b create
nrcmd> client-policy 1,6,00:d0:ba:d3:bd:3b set v4-reply-options=routers
nrcmd> client-policy 1,6,00:d0:ba:d3:bd:3b set v4-reply-options=routers
```
b. Get the client embedded policy data so that you can display the values:

```
nrcmd> client 1,6,00:d0:ba:d3:bd:3b get embedded-policy
100 Ok
embedded-policy="((ClassName Policy)(name client-policy:00:d0:ba:d3:bd:3b)(option-list
  (((ClassName Option)(number 3)(option-definition-set-name dhcp-config)(value
  01:02:03:04:05:06:07:08)))(v4-reply-options [routers ]))"
```

c. Copy what is between the quotes in the client output in the previous substep and paste it in for the definition of the `businessCategory` LDAP attribute:

```
businessCategory:((ClassName Policy)(name client-policy:00:d0:ba:d3:bd:3b)(option-list
  (((ClassName Option)(number 3)(option-definition-set-name dhcp-config)(value
  01:02:03:04:05:06:07:08)))(v4-reply-options [routers ]))
```

The option values are translated into hexadecimal field syntax, including multiple values that were originally comma-separated.

d. Use the syntax as a model for each new embedded policy entry in LDAP. To see how other option data types appear in the LDAP string, add these options to the client or create further dummy clients with them. Once you extract the data, you can delete the dummy client:

```
nrcmd> client 1,6,00:d0:ba:d3:bd:3b delete
nrcmd> save
```

Here is another example with multiple option definitions:

1. Create a dummy client 1,6,00:d0:ba:d3:bd:3b and an embedded policy attached to that client, with the following options and values:

   - 3 routers 10.1.1.1,10.2.1.1
   - 66 tftp-server tftp-server.com
   - 67 bootfile device-boot-file.txt

2. Save the changes to the embedded policy, save the client, then extract the following output string into an LDAP client configuration:

```
nrcmd> client 1,6,00:d0:ba:d3:bd:3b get embedded-policy
100 Ok
embedded-policy="((ClassName Policy)(name client-policy:00:d0:ba:d3:bd:3b)(option-list
  (((ClassName Option)(number 3)(option-definition-set-name dhcp-config)(value
  0a:01:01:01:0a:02:01:01))((ClassName Option)(number 66)(option-definition-set-name dhcp-config)(value
  Option)(number 67)(option-definition-set-name dhcp-config)(value
```

### Configuring DHCP LDAP Update and Create Services

You can configure the Cisco Prime IP Express DHCP server to write lease and client data to an LDAP server. The DHCP server can use the client data when responding to DHCP client requests, through the use of the `query` configuration. You can configure the DHCP LDAP service to copy lease state data to attributes on client objects in the LDAP server. The DHCP server converts the lease state data to string form, and uses an update dictionary to map the DHCP data values to the LDAP attributes.
Each time the lease state changes, the DHCP server writes the change to the LDAP server that you configured to store the data. The lease data that the DHCP server writes to LDAP is “write-only” in that it is a copy of the authoritative data in the lease state database.

Related Topics

Lease State Attributes, page 25-24
Configuring DHCP to Write Lease States to LDAP, page 25-25
Using LDAP Updates, page 25-26
Configuring LDAP State Updates, page 25-26
Configuring LDAP Entry Creation, page 25-28

Lease State Attributes

You can store any of these attributes about the lease state information in your LDAP server:

- **address**—IP address of this lease.
- **client-dns-name**—Name the DHCP server attempted to enter into the DNS server for this client.
- **client-domain-name**—Domain into which to put the client name.
- **client-flags**—A variety of flags relating to the client.
- **client-host-name**—DNS name that the client requested the DHCP server to place in the DNS server.
- **client-id**—Client-id specified by the client, or one synthesized by the DHCP server for this client.
- **client-mac-addr**—MAC address that the client presented to the DHCP server.

Note: Although the MAC addresses in LDAP have to be formatted exactly the way they are formatted by Cisco Prime IP Express when it creates local client-entries, they are separate instances and thus unique to lease data.

- **expiration**—The time at which the lease expires.
- **flags**—Flags for the lease (reserved or deactivated).
- **lease-renewal-time**—The earliest time in which the client is expected to issue a lease renewal. You can have Cisco Prime IP Express save this as part of the lease state by using `dhcp enable save-lease-renewal-time` (it is not saved by default).
- **start-time-of-state**—The time at which the state last changed to its current value.
- **state**—The lease state can be:
  - Available (1)
  - Deferred (2)
  - Leased (3)
  - Expired (4)
  - Unavailable (5)
  - Released (6)
  - Other_available (7)
  - Disconnected (8)
  - Deleted (9)
• **vendor-class-identifier**—The name of the vendor, used by clients and servers to exchange vendor-specific information.

Not every lease has all these attributes. The `client-mac-addr` and `client-id` lease state attribute are not present if a client releases its lease or is forced available through Cisco Prime IP Express. In addition, the `lease-renewal-time` attribute may not be present if the `save-lease-renewal-time` property is disabled through DHCP. Similarly, the `vendor-class-identifier` property may not be present if the `save-vendor-class-id` property is disabled through DHCP, using the CLI.

### Configuring DHCP to Write Lease States to LDAP

To have DHCP write lease state updates to LDAP:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Choose the LDAP lease state update scheme.</td>
</tr>
<tr>
<td>2</td>
<td>Add entries to the directory or modify existing entries to store the lease state information. You may need to extend entries through the addition of attributes or custom object classes.</td>
</tr>
<tr>
<td>3</td>
<td>Configure Cisco Prime IP Express to perform the updates.</td>
</tr>
</tbody>
</table>

Given the flexibility of directories, there are many different ways in which you could choose to store a copy of lease state attributes in a directory. For example, you could choose to store the lease state data as part of an existing entry, or you could store the lease state data independently.

### Storing Lease State Data as Part of Existing Entries

You can store lease state data as part of an existing entry. It is even possible to store the client entry, lease state, and employee data in the same entry. As part of the setup for this method, you must decide how you want to store the lease data attributes. You can store data attributes using these methods:

- Map attributes from the entry
- Add attributes to the entry
- Extend the entry by creating a new object class

The advantage is that lease data is stored directly with other client information. The disadvantage is that there are scenarios, albeit unlikely, related to client-class and reservations that could result in stale data being in the directory for a short period of time when the server moves a client off a lease.

**Note**

If the lease whose state is being updated does not have a client, it will not have an associated MAC address. This situation occurs when a client gets a lease, and then is moved off that lease by client-class processing. It can also occur when a client has a pre-existing lease and a reservation for a different lease in the same LAN segment. When the reserved lease is available, the server moves the client off its existing lease and onto the reservation. Both of these transfers result in an LDAP update for the old lease without a client MAC address. This is generally not a problem, because the update for the new lease (which has an associated MAC address) should come through.

Also, this method requires two LDAP interactions to write the lease information. When updating lease state information, the DHCP LDAP service contacts the directory twice because when updating an entry it is not enough just to know how to find the entry. You must specifically know the `dn` of the entry.
The DHCP LDAP service first finds the appropriate entry in the directory by using one of the lease state attributes that you chose (preferably the MAC address) as the search criteria. This is necessary because none of the lease state attributes is part of the \textit{dn} of the entry. When the DHCP LDAP service locates the entry, the \textit{dn} is returned. The DHCP LDAP service then updates that same entry with the appropriate information. For an example how to use this method, see the “Configuring LDAP State Updates” section on page 25-26.

**Storing Lease State Data Independently**

You can store lease state data by IP address in its own entries. This method results in a copy of the server lease database in a directory, and is the most straightforward way to configure the database. As part of the setup for this method, create new entries for each IP address that the server can serve. The advantage to this method is that there are no scenarios in which the lease state data in the directory will be stale. The disadvantage is that lease data is not stored directly with other associated client information.

To update the lease state information, the DHCP LDAP service contacts the directory service once. When performing the update, the service uses the IP address to construct the \textit{dn}.

**Using LDAP Updates**

There are two ways you can use the LDAP update feature:

- Keep track of clients that use LDAP client entry information and to associate some of the attributes of that LDAP host with lease state attributes.
- Create and update objects that can be located by their IP address. When Cisco Prime IP Express creates these objects, it can make a level of LDAP objects that matches (or is) the DHCP server lease state.

When using Cisco Prime IP Express, you should be aware that:

- The DHCP server only reads from a single object and writes to a single object. You can use separate objects to hold the client entry data read and the lease state date written, but Cisco Prime IP Express cannot read some attributes from one object and some from another.
- The performance of LDAP queries, like all database access, depends on indexed attributes. If you did not index the attributes that you configure to use in query filters, you will experience poor performance.
- LDAP attributes must either come preconfigured in the LDAP schema at server installation or be created by some other means outside Cisco Prime IP Express.

**Configuring LDAP State Updates**

There are two options available for performing a lease state update to an LDAP server:

- \texttt{update-search-path}—The DHCP server first queries to locate the \textit{dn} for an update.
- \texttt{dn-format}—The server is provided with the \textit{dn} for an update. In other words, the DHCP performs a direct update without having to query before an update.

**Option 1: Using the update-search-path Option**

The following example illustrates the first option, \texttt{update-search-path}. It shows what to do when the distinguished name (\textit{dn}) of an LDAP object cannot be constructed from data that is available in the lease state. The DHCP server creates an LDAP query based on the \texttt{update-search-xxx} information, locates the LDAP object, and uses its \textit{dn} to issue an LDAP update.
The example shown in Table 25-2 on page 25-27 assumes that you are using the standard LDAP organizational person object class attributes to hold lease update data.

**Table 25-2  LDAP-to-DHCP Mapping Example**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>DHCP Lease Entry Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>address (IP address)</td>
</tr>
<tr>
<td>carlicense</td>
<td>state (lease state)</td>
</tr>
</tbody>
</table>

**Step 1**  
Tell DHCP about the LDAP server by supplying the server hostname in the LDAP configuration.

**Step 2**  
Configure the credentials to use when connecting to the LDAP server. This CLI example sets the administrator to joe and his password to access. Use the distinguished name (dn) for the user:

```
nrcmd> ldap myserver set username="cn=joe,o=Example Corporation,c=US" password=access
```

**Step 3**  
Configure the `update-search-path` attribute, which is the starting point in the directory for the objects that the DHCP server will update. You can also set the update search scope. This CLI example sets the search path to begin at the organizational unit (ou) IT, the organization Example Corporation, and country US. The update search scope is set to SUBTREE:

```
nrcmd> ldap myserver set update-search-path="ou=IT,o=Example Corp,c=US" update-search-scope=SUBTREE
```

**Step 4**  
Set the ID of the attribute you want to use to search for the LDAP object that will be updated. This CLI example sets the search attribute to be the client MAC address:

```
nrcmd> ldap myserver set update-search-attribute=client-mac-addr
```

**Step 5**  
Configure a filter expression into which the `update-search-attribute` attribute should be formatted. This expression must contain a “%s,” which indicates where the search attribute data should be substituted. Here is a CLI example:

```
nrcmd> ldap myserver set update-search-filter=(cn=%s)
```

**Step 6**  
Configure the `update-dictionary` attribute, which allows you to identify the LDAP attributes that you want set with the values of the corresponding lease state attributes. This example specifies that the LDAP UID should be updated to contain the IP address, and that the `carlicense` attribute should be updated to contain the DHCP lease state information. Using the CLI:

```
nrcmd> ldap myserver setEntry update-dictionary uid=address carlicense=state
```

**Step 7**  
Enable updates for the new LDAP server. Here is a CLI example:

```
nrcmd> ldap myserver enable can-update
```

**Step 8**  
Reload the DHCP server.

---

**Option 2: Using the dn-format Option**

This example illustrates using the second option, `dn-format`:

**Step 1**  
Tell DHCP about the LDAP server by supplying the server hostname in the LDAP configuration.

**Step 2**  
Configure the credentials to use when connecting to the LDAP server. This CLI example sets the administrator to joe and his password to access. Use the `dn` for the user:
Configuring LDAP Entry Creation

This section explains how to create LDAP entries. LDAP entry creation provides the ability to locate entries and update them with current lease information. Entries are created only if a state update operation fails because it cannot locate an entry.

After performing the steps in the previous example, follow these steps in the CLI:

Step 1  Set the dn-attribute property for the LDAP server for the lease object attribute, such as the client-mac-addr field, and set the dn-format string. Here is a CLI example:

```
nrcmd> ldap myserver set dn-attribute=client-mac-addr
dn-format="cn=\"%s\",ou=IT,o=Example Corp,c=US"
```

This step is required only if you configure the lease state updates using the update-search-path option. (See the “Option 1: Using the update-search-path Option” section on page 25-26). Skip this step if you configure lease state updates using the dn-format string. (See the “Option 2: Using the dn-format Option” section on page 25-27.)

Step 2  Specify the dn of the entry to be created when combined with the existing dn-attribute property. Here is a CLI example:

```
nrcmd> ldap myserver set dn-create-format="cn=\"%s\",ou=IT,o=Example Corp,c=US"
```

The Cisco Prime IP Express client-mac-addr field uses the form 1,6:xx:xx:xx:xx:xx. Since the comma character is a special separator in LDAP, you must use the " characters to quote the dn.

Step 3  Using the create-dictionary property, establish mappings between LDAP attributes and lease state attributes by entering a series of name=value pairs. The LDAP attributes indicate the entry attributes set to the value of their corresponding lease state attributes. In the CLI:

```
nrcmd> ldap myserver setEntry create-dictionary sn=client-host-name
nrcmd> ldap myserver setEntry create-dictionary givenname=client-class-name
nrcmd> ldap myserver setEntry create-dictionary localityname=client-domain-name
```
**Step 4**  Using the `create-object-classes` property, specify the object classes to be used when creating the entry. Here is a CLI example:

```
nrcmd> ldap myserver set create-object-classes= "top,person,organizationalPerson/inetorganperson"
```

**Step 5**  Enable entry creation for the LDAP server myserver. Here is a CLI example:

```
nrcmd> ldap myserver enable can-create
```

**Note**  Enable the `can-update` attribute before you enable the `can-create` attribute. For an example, see the “Configuring LDAP State Updates” section on page 25-26.

**Step 6**  Reload the DHCP server.

**Step 7**  To see if creation, queries, and updates were successful, view the LDAP log settings.

---

**Troubleshooting LDAP**

The following sections include some advice on fine-tuning and detecting failures of the LDAP server.

**Related Topics**

**LDAP Connection Optimization**, page 25-29

**Recommended Values for LDAP**, page 25-30

**LDAP Connection Optimization**

You can optimize LDAP connections by using separately tunable read and write objects. This CLI example tunes write (create and update) operations, which require longer server processing:

```
nrcmd> ldap LDAP-Write create csrc-ldap password=changeme port=389 preference=1
nrcmd> ldap LDAP-Write setEntry query-dictionary csrcclientclasses=client-class-name
nrcmd> ldap LDAP-Write set
search-filter(&(macaddress=%s)(|(crscclassname=Computer)(crscclassname=Modem)))
```

```
nrcmd> ldap LDAP-Write set search-path=csrcprogramname=csrc,o=NetscapeRoot
nrcmd> ldap LDAP-Write set
username=uid=admin,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot
nrcmd> ldap LDAP-Write disable can-query
nrcmd> ldap LDAP-Write enable can-create
nrcmd> ldap LDAP-Write enable can-update
nrcmd> ldap LDAP-Write enable limit-requests
nrcmd> ldap LDAP-Write set connections=2 max-requests=8 timeout=10s
```

This CLI example tunes read (query) operations, which require shorter server processing:

```
nrcmd> ldap LDAP-Read create csrc-ldap password=changeme port=389 preference=1
nrcmd> ldap LDAP-Read setEntry query-dictionary csrcclientclasses=client-class-name
nrcmd> ldap LDAP-Read set
search-filter(&(macaddress=%s)(|(crscclassname=Computer)(crscclassname=Modem)))
```

```
nrcmd> ldap LDAP-Read set search-path=csrcprogramname=csrc,o=NetscapeRoot
nrcmd> ldap LDAP-Read set
username=uid=admin,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot
nrcmd> ldap LDAP-Read enable can-query
nrcmd> ldap LDAP-Read disable can-create
```
Configuring Cisco Prime IP Express to Use LDAP

nrcmd> ldap LDAP-Read disable can-update
nrcmd> ldap LDAP-Read enable limit-requests
nrcmd> ldap LDAP-Read set connections=3 max-requests=12 timeout=4s

Recommended Values for LDAP

Table 25-3 shows recommended values for some key LDAP attributes.

<table>
<thead>
<tr>
<th>Attribute and Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connections=5 to 25</td>
<td>Number of connections that the server should make to an LDAP server. This is primarily a performance tuning parameter. The default is one connection. In some cases, more than one connection can improve overall throughput. The amount depends on the load on the LDAP server. With many applications using LDAP, five connections would be appropriate; with just Cisco Prime IP Express using LDAP, 25 would be appropriate.</td>
</tr>
<tr>
<td>threadwaittime=2</td>
<td>Interval (in milliseconds) at which each LDAP client connection polls for results, if it has outstanding queries or updates.</td>
</tr>
<tr>
<td>query-timeout=3</td>
<td>Cisco Prime IP Express DHCP servers fail over at the query-timeout interval if failover and can-query are set. The default setting is 3 seconds and is recommended (in that it is less than the default 4-second drop-old-packets value for the DHCP server, after which time the connection is considered inactive and the LDAP server as “unhealthy”).</td>
</tr>
<tr>
<td>timeout=10</td>
<td>Number of seconds an LDAP request remains in a connection queue before being declared stale and timing out. Any response the DHCP client receives after the client timeout period is stale. The default is 10 seconds, which is recommended. Cisco Prime IP Express DHCP servers fail over at the timeout interval if failover and can-update or can-create are enabled.</td>
</tr>
</tbody>
</table>
Using Expressions

Cisco Prime IP Express provides enhanced client-class support. You can now place a request into a client-class based on the contents of the request, without having to register the client in the client database. Also, you can now place requests in a client-class based on the number of the active leases of a subscriber, allowing limitations on the level of service offered to various subscribers. This is possible through the special DHCP options processing using expressions.

You can set the limitation on subscriber addresses based on values in the DHCP relay-agent-info option (option 82, as described in RFC 3046). These values do not need to reveal any sensitive addresses. You can create values that relate an individual to a subscriber by creating an expression that evaluates the incoming DHCPDISCOVER request packets against option 82 suboptions (remote-id or circuit-id) or other DHCP options. The expression is a series of if statements that return different values depending on what is evaluated in the packet. This, in effect, calculates the client-class in which the subscriber belongs, and limits address assignment to the scope of that client-class.

**Note**

Expressions are not the same as DHCP extensions. Expressions are commonly used to create client identities or look up clients. Extensions (see Chapter 30, “Using Extension Points”) are used to modify request or response packets. The expressions described here are also not the same as regex.

**Related Topics**

- Using Expressions, page 26-1
- Entering Expressions, page 26-2
- Creating Expressions, page 26-3
- Expression Examples, page 26-22
- Debugging Expressions, page 26-26

**Using Expressions**

Expression processing is used in several places:

- **Calculating a client-class**—client-class-lookup-id. This expression determines the client-class based on the contents of the incoming packet.

- **Creating the key to look up in the client-entry database**—client-lookup-id. This accesses the client-entry database with the key resulting from the expression evaluation.

- **Creating the ID to use to limit clients of the same subscriber**—limitation-id. This is the ID to use to check if any other clients are associated with this subscriber.
This kind of processing results in this scenario:

1. The DHCP server tries to get a client-class based on a `client-class-lookup-id` expression. If it cannot calculate the client-class, it uses the usual MAC address method to look up the client.

2. If the server can calculate the client-class, it determines if it needs to do a client-entry lookup, based on evaluating a `client-lookup-id` expression that returns a `client-lookup-id`. If it has such an ID, it uses it to look up the client. If it does not have such an ID, it uses the calculated client-class value to assign addresses.

3. If the server uses the `client-lookup-id` and finds a client-entry, it uses the data for the client. If it cannot find a client-entry, it uses the calculated or default client-class data.

You set the upper limit on assigned addresses to clients on a network or LAN segment having an identical `limitation-id` value on the policy level. Set this upper limit as a positive integer using the `limitation-count` attribute for the policy.

The values to set for limiting IP addresses to subscribers are:

- For a policy, set the `limitation-count` attribute to a positive integer.
- For a client-class, set the `limitation-id` and `client-lookup-id` attributes to an expression, and set the `over-limit-client-class-name` attribute to a client-class.
- For a client, set the `over-limit-client-class-name` attribute to a client-class.

The expressions to use are described in the “Creating Expressions” section on page 26-3.

**Entering Expressions**

You can include simple expressions as such in the attribute definition, or include more complex ones in an expression file and reference the file in the attribute definition. Either way, the maximum allowable characters is 16 KB.

Simple expressions must adhere to these rules when you enter them in the CLI:

- They must be limited to a single command line.
- The entire expression must be enclosed in double quotes ("").
- Embedded double quotes must be escaped with a backslash (\).

Here is an example of a simple expression to set the `client-class-lookup-id`:

```
"\"limit\""
```

Here is a slightly more extensive example to set the client-class `limitation-id`:

```
"(request option 82 \"circuit-id\")"
```

You must enter any more complex expressions, that are not limited to one line or that you want to format for comprehension, in a file and reference it in the attribute definition prefixed by the “at” symbol (@):

```
@cclookup.txt
```

The syntax of the expression in the file does not have the extra requirements (as to spacing and escaping of characters) of the simple expression. It can also include comment lines, prefixed by the pound sign (#), double-slash (/), or a semicolon (;), and terminated at the end of line.

For example, in the cclookup.txt file:

```
// Expression to calculate client-class based on remote-id
(try (if (equal (request option "relay-agent-info" "remote-id") (request chaddr))
  "cm-client-class"
```
Chapter 26      Using Expressions

Creating Expressions

Creating Expressions

Using DHCP expressions, you can retrieve, process, and make decisions based on data in incoming DHCP packets. You can use them for determining the client-class of an incoming packet, and create the equivalence key for option 82 limitation support. They provide a way to get information out of a packet and individual options, a variety of conditional functions to allow decisions based on information in the packet, and data synthesis capabilities where you can create a client-class name or key.

The expression to include in an expression file that would describe the example in the “Typical Limitation Scenario” section on page 25-15 would be:

```
// Begins the try function
(try
 (or (if (equal (request option "relay-agent-info" "remote-id") (request chaddr))
      "cm-client-class")
       (if (equal (substring (request option "dhcp-class-identifier") 0 6) "docsis")
      "docsis-cm-client-class")
       (if (equal (request option "user-class") "alternative-class")
      "alternative-cm-client-class")
    )
 "<none>"
)
// Ends the try function
```

The expression uses the or function and evaluates three if functions. In a simpler form, you can calculate a client-class and include this expression in the cclookup.txt file.

```
// Expression to calculate client-class based on remote-id
(try (if (equal (request option "relay-agent-info" "remote-id") (request chaddr))
      "cm-client-class"
       "cpe-client-class")
 "<none>>"
)
```

The IPv6 version of the previous example (using option numbers) is:

```
// Expression to calculate client-class based on DOCSIS 3.0 cm-mac-address
(try (if (equal (request option 17 enterprise-id 4491 36)
      (or (request relay option 17 enterprise-id 4491 1026) "none"))
      "v6-cm-client-class"
       "v6-cpe-client-class")
 "<none>"
)
```

You can also write the previous expression by substituting option names in place of numbers:

```
// Expression to calculate client-class based on DOCSIS 3.0 cm-mac-address
(try (if (equal (request option "vendor-opts" enterprise-id "dhcp6-cablelabs-config" "device-id"
      (or (request relay option "vendor-opts" enterprise-id "dhcp6-cablelabs-config"
      "cm-mac-address") "none"))
      "v6-cm-client-class"
       "v6-cpe-client-class")
 "<none>"
)
```

The or function in the example ensures that if the packet was not relayed or if the relay agent did not add the option, then the server assumes the client to be a CPE and not a cable modem (CM).
Refer to this file to use the expression to set the client-class lookup ID for the server:

```
nrcmd> dhcp set client-class-lookup-id=@cclookup.txt
```

You can generate a limitation key by trying to get the \textit{remote-id} suboption from option 82, and if unable, to use a standard MAC blob key. Include an expression in a file and set the limitation ID to it in the \texttt{cclimit.txt} file:

```
// Expression to use remote-id or standard MAC
(try (request option "relay-agent-info" "remote-id") 00:d0:ba:d3:bd:3b)
```

### Related Topics

- Expression Syntax, page 26-4
- Expression Datatypes, page 26-4
- Literals in Expressions, page 26-5
- Expressions Return Typed Values, page 26-5
- Expressions Can Fail, page 26-6
- Expression Functions, page 26-6
- Datatype Conversions, page 26-21

```
nrcmd> client-class name limitation-id=@cclimit.txt
```

### Expression Syntax

Expressions consist solely of functions and literals. Its syntax is similar to that of Lisp. It follows many of the same rules and uses Lisp functions names where possible. The basic syntax is:

```
(function argument-0 ... argument-n)
```

A more useful example is:

```
(try (if (equal (request option "relay-agent-info" "remote-id") (request chaddr))
    "cm-client-class"
    "cpe-client-class")
    "<none>")
```

This example compares the \textit{remote-id} suboption of the \textit{relay-agent-info} option (option 82) with the MAC address in the packet, and if they are the same, returns “cm-client-class,” and if they are different, returns “cpe-client-class.” (If the expression cannot evaluate the data, the \textit{try} function returns a “<none>” value—see the “Expressions Can Fail” section on page 26-6.) The intent is to determine if the device is a cable modem (where, presumably, the \textit{remote-id} equals the MAC address) and, if so, put it into a separate client-class than the customer premise equipment or PC. Note that both functions and literals are expressions. The previous example shows a function as an expression. For literals, see the “Literals in Expressions” section on page 26-5.

### Expression Datatypes

The datatypes that expressions support are:

- **Blob**—Counted series of bytes, with a minimum supported length of 1 KB.
- **String**—Counted series of NVT ASCII characters, not terminated by a zero byte, with a minimum supported length of 1 KB.
Creating Expressions

- **Signed integer**—32-bit signed integer.
- **Unsigned integer**—32-bit unsigned integer.

Note that there is no IP address datatype; an IP address is a 4-byte blob. All numbers are in network byte order. See the “Datatype Conversions” section on page 26-21.

### Literals in Expressions

A variety of literals are included in the expression capability:

- **Signed integers**—Normal numbers that must fit in 32 bits.
- **Unsigned integers**—Normal unsigned numbers that must fit in 32 bits.
- **Blobs**—Hex bytes separated by colons. For example, 01:02:03:04:05:06 is a 6-byte blob with the bytes 1 through 6 in it. This is distinct from “01:02:03:04:05:06” (a 17-byte string). The string is related to the blob by being the text representation of the blob. For example, the expression `(to-blob "01:02:03")` returns the blob 01:02:03. Note that you cannot create a literal representation of a one-byte blob, as 01 will turn into an integer. To get a one-byte blob containing a 1, you would use the expression `(substring (to-blob 1) 3 1)`. The 3 indicates the offset to extract the fourth byte of the 4-byte integer (00:00:00:01), with the 1 being the number of bytes extracted, with a result of “01.”
- **String**—Characters enclosed in double quotes. For example, “example.com” is a string, as is “01:02:03:04:05:06.” To place a quote in a literal string, escape it with a backslash (\), for example: "this has one "quote"

Integer literals (signed and unsigned) are assumed to be in base10. If they start with a 0, they are considered octal; if they start with 0x, they are considered hexadecimal. Some examples of literals:

- “hello world” is a string literal (and a perfectly valid expression).
- 1 is an unsigned integer literal (also a perfectly valid expression). It contains 4 bytes, the first three of which are zero, and the last of which contains a 1 in the least significant bit.
- 01:02:03 is a blob literal containing three bytes, 01, 02, and 03.
- –10 is a signed integer literal containing four bytes with the twos-complement representation of decimal -10.

### Expressions Return Typed Values

With few exceptions, the point of an expression is to return a value. The expression configured to determine a client-class is configured in the DHCP server property `client-class-lookup-id`. When this expression is evaluated, the DHCP server expects it to return a string containing the name of a client-class, or the string `<none>`.

Every function returns a value. The datatype of the value may depend on the datatype of the argument or arguments. Some expressions only accept arguments of a certain datatype; for example:

`(+ argument0 argument1)`

In most cases, a function that requires a certain datatype for a particular argument tries to convert the argument that it gets to the proper datatype. For example, `(+:1*2)` returns 3, because it successfully converts the string literal “1” into a numeric 1. However, `(+:one*2)` causes an error, because “one” does not convert successfully into a number. In general, the expression evaluator tries to do the right thing as much as possible when making datatype conversion decisions.
Expressions Can Fail

While some of the functions that make up an expression operate correctly on any datatype or value, many do not. In the previous section, the + function would not convert the string literal “one” into a valid number, so the evaluation of that function failed. When a function fails to evaluate, its calling function also fails, and so on, until the entire expression fails. A failed expression evaluation has different consequences depending on the expression involved. In some cases, it can cause the packet to be dropped, while in others it only generates a warning message.

You can prevent the evaluation from failing by using the (try expression failure-expression) function. The try function evaluates the expression and, if successful, the value of the function is the value of the expression. If the evaluation fails (for whatever reason), the value of the function is the value of the failure-expression. The only situation where a try function itself fails is if the failure-expression evaluation fails. Thus, you should be careful what expression you define as a failure-expression. A string literal is a safe bet. Thus, protecting the evaluation of the client-class-lookup-id with a try function is a good idea. The previously cited example shows how this can work:

(try (if equal (request option "relay-agent-info" "remote-id") (request chaddr))
    "cm-client-class"
    "cpe-client-class")
"<none>"

If evaluating the if function fails in this case, the value of the client-class-lookup-id expression is <none>. It could have been a client-class name instead, of course.

Expression Functions

Table 26-1 lists the expression functions. Expressions must be enclosed in parentheses.

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(+ arg1 ... argn)</td>
<td>(+ 1 2 3 4)</td>
<td>10</td>
</tr>
<tr>
<td>(– arg1 ... argn)</td>
<td>(– 10 5 2)</td>
<td>3</td>
</tr>
<tr>
<td>(* arg1 ... argn)</td>
<td>(* 3 4 5)</td>
<td>60</td>
</tr>
<tr>
<td>(/ arg1 ... argn)</td>
<td>(/ 20 2 5)</td>
<td>2</td>
</tr>
<tr>
<td>(% arg1 arg2)</td>
<td>(% 12 7)</td>
<td>An error 5 (12/7=1*7+5)</td>
</tr>
</tbody>
</table>

Arithmetic operations on a signed integer or an expression is convertible to a signed integer. Any argument that cannot convert to a signed integer (and is not null) returns an error. Any argument that evaluates to null is ignored (except that the first argument for – and / must not evaluate to null). These functions always return signed integers (note that overflow and underflow are currently not caught):

- + sums the arguments; if no arguments, the result is 0.
- – negates the value of a single argument or, if multiple arguments, successively subtracts the values of the remaining ones from the first one; for example, (– 3 4 5) becomes –6.
- * takes the product of the argument values; if no arguments, the result is 1.
- / successively divides the first argument by all of the others; for example, (/ 100 4 5) becomes 5. If any argument other than the first equals 0, an error is returned.
- % is the modulo arithmetic operator to determine the remainder of the result of the first argument divided by the second one; for example, (% 12 7) becomes 5 (12 / 7 = 1 * 7 + 5).
Creating Expressions

Table 26-1  Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(and arg1 ... argn)</td>
<td>(and &quot;hello&quot; &quot;world&quot;)</td>
<td>(and (request option 82 1) (request option 82 2)) returns option-82 sub-option 2 if both option-82 sub-option 1 and sub-option 2 are present in the request.</td>
</tr>
</tbody>
</table>

Returns a value that is the datatype of argn or null. It evaluates its arguments in order from left to right (the arguments can evaluate to a datatype). If any argument evaluates to null, it stops evaluating the arguments and returns null. Otherwise, it returns the value of the last argument, argn.


Treats expr as if it were a blob. If expr evaluates to a string, the bytes that make up the string become the bytes of the blob that is returned. If expr evaluates to a blob, that blob is returned unmodified. If expr evaluates to either kind of integer, a 4-byte blob containing the bytes of the integer is returned. (See Table 26-2 on page 26-21.)

(as-sint expr) | (as-sint ff:ff:ff:ff) | returns -1 |

Treats expr as if it were a signed integer. If expr evaluates to a string or blob of 4 bytes or less, the function returns a signed integer constructed out of those bytes (if longer than 4 bytes, it returns an error). If expr evaluates to a signed integer, it returns the value unchanged; if an unsigned integer, it returns a signed integer with the same bit value. (See Table 26-2 on page 26-21.)

(as-string expr) | (as-string 97) | returns "a" |

Treats expr as if it were a string. If expr evaluates to a string, it returns that string. If expr evaluates to a blob, it returns a string constructed from the bytes in the blob, unless they are nonprintable ASCII values, which returns an error. If expr evaluates to an integer, it considers its value to be the ASCII value for a single character and returns a string consisting of that one character, unless it is nonprintable, which returns an error. (See Table 26-2 on page 26-21.)

(as-uint expr) | (as-uint -2147483648) | returns the unsigned integer 2147483648 |

Treats expr as if it were an integer. If expr evaluates to a string or blob of 4 bytes or less, it returns an unsigned integer constructed from those bytes; if longer than 4 bytes, it returns an error. If the result is an unsigned integer, it returns the argument unchanged; if a signed integer, it returns an unsigned integer with the same bit value (see Table 26-2 on page 26-21).
### Table 26-1 Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ash expr shift)</td>
<td>(ash 00:01:00 1)</td>
<td>returns the blob 00:02:00</td>
</tr>
<tr>
<td>(lshift expr shift)</td>
<td>(lshift 00:01:00 -1)</td>
<td>returns the blob 00:00:80</td>
</tr>
<tr>
<td></td>
<td>(ash 1)</td>
<td>returns the unsigned integer 2</td>
</tr>
</tbody>
</table>

Returns an integer or blob with the bits shifted by the `shift` amount. The `expr` can evaluate to an integer, blob or string. If `expr` evaluates to a string, this function tries to convert it to a signed integer, and if that fails, to a blob. If both fail, it returns an error. The `shift` must evaluate to something that is convertible to a signed integer. If `shift` is positive, the shift is to the left; if negative, the shift is to the right. If `expr` results in a signed integer, the right shift is with sign extension. If `expr` results in an unsigned integer or blob, a right shift shifts zero bits in on the most significant bits.

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(bit-and arg1 arg2)</td>
<td>(bit-and 00:20 00:ff)</td>
<td>returns 00:20</td>
</tr>
<tr>
<td>(bit-or arg1 arg2)</td>
<td>(bit-or 00:20 00:ff)</td>
<td>returns 00:ff</td>
</tr>
<tr>
<td>(bit-xor arg1 arg2)</td>
<td>(bit-xor 00:20 00:ff)</td>
<td>returns 00:df</td>
</tr>
<tr>
<td></td>
<td>(bit-andc1 00:20 00:ff)</td>
<td>returns 00:df</td>
</tr>
</tbody>
</table>

Return the result of a bit-wise boolean operation on the two arguments. The data type of the result is a signed integer if both arguments result in either kind of integer, otherwise the result is a blob. The `arg1` and `arg2` arguments must evaluate to two integers, two blobs of equal length, or one integer and one blob of length 4. If either argument evaluates to a string, the function tries to convert the string to a signed integer, and if that fails, to a blob. After this conversion, the results must match the criteria mentioned above. If these conditions are not met, it returns an error. Operations with `c1` and `c2` indicate that the first and second arguments, respectively, are complemented before the operation.

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(bit-not expr)</td>
<td>(bit-not ff:ff)</td>
<td>returns 00:00</td>
</tr>
<tr>
<td></td>
<td>(bit-not 1)</td>
<td>returns 4294967295</td>
</tr>
<tr>
<td></td>
<td>(bit-not &quot;hello world&quot;)</td>
<td>returns an error</td>
</tr>
</tbody>
</table>

Returns a value that is the bit-by-bit complement of `expr`. The datatype of the result is the same as the result of evaluating `expr` and any subsequent conversions, if the result was a string. The expression must evaluate to an integer of either type, or a blob. If it evaluates to a string, the function tries to convert it to a signed integer, and if that fails, to a blob, and if that fails, returns an error.

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(byte arg1)</td>
<td>(byte 150)</td>
<td>returns 0x96</td>
</tr>
<tr>
<td></td>
<td>(byte 0x96)</td>
<td>returns 0x96</td>
</tr>
</tbody>
</table>

Eases creation of one-byte blobs. It returns this blob depending on the data type:

- **sint, uint**—Returns a low-order byte of type integer.
- **blob**—Returns the last byte in the blob.
- **string**—Returns the last byte in the string.

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(comment comment expr1… exprm)</td>
<td>(comment &quot;this is a comment that won’t get lost&quot; (request option 82 1))</td>
<td>inserts a comment string into an expression and returns the value of the last expression (exprm).</td>
</tr>
</tbody>
</table>
Table 26-1  Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation of Expressions</td>
<td>(concat arg1 ... argn)</td>
<td>(concat &quot;hello &quot; &quot;world&quot;) returns 'hello world'</td>
</tr>
<tr>
<td></td>
<td>(concat -1 &quot;world&quot;)</td>
<td>(concat -1 00:01:02) returns an error</td>
</tr>
<tr>
<td></td>
<td>(concat -1 00:01:02)</td>
<td>ff:ff:ff:00:01:02</td>
</tr>
</tbody>
</table>

Concatenates the values of the arguments into a string or blob (ignoring null arguments). The first argument (arg1) must evaluate to a string or a blob; if it evaluates to an integer, the function converts it to a blob. The datatype of arg1 (after any conversion) determines the datatype of the result. The function converts all subsequent arguments to the datatype of the result, and if this conversion fails, returns an error.

**concat**

Returns the datatype of the result of the expression (expr). If the expression cannot evaluate expr, it returns an error, otherwise it returns the datatype as a string, which can be:

- "unset" (internal, considered as null)
- "null"
- "uint"
- "sint"
- "string"
- "blob"

**dotimes**

Creates an environment with a single local integer variable, var, which is initially set to zero, and evaluates exp1 through expn. It then increments var by one, and if it is less than count-exp, evaluates exp1 through expn again. When var is equal to or greater than count-exp, the function evaluates result-exp and returns it as the result of the entire dotimes. If there is no result-exp, the function returns null.

The var defines a local variable, and must be an alphabetic name. The count-exp must evaluate to an integer or be convertible to one. The exp1 through expn are expressions that can evaluate to any data type. The result-exp is optional, and if it appears, it can evaluate to any data type. When the function evaluates count-exp, var is not bound and cannot appear in count-exp. Alternatively, var is bound for the evaluation of result-exp and has the value of count-exp. If result-exp is omitted, the function returns null.

**Note**  Be careful changing the value of var in exp1 through expn, because you can easily create an infinite loop (see the example).
Chapter 26  Using Expressions

Creating Expressions

```
(environmentdictionary
  {get | put val | delete}
attr)
```

```
nrcmd> dhcp set
  initial-environment-dictionary=first=one,
  second=2
  (environmentdictionary get "first")
  returns "one"
  (environmentdictionary get "second")
  returns "2" (note string 2)
  (environmentdictionary put "two"
  "second") returns "second"
  (environmentdictionary delete "first")
  returns null
```

Gets, puts, or deletes a DHCP extension environment dictionary attribute value. The `val` is the value of the attribute and `attr` is the attribute name. Both are converted to a string regardless of their initial datatype. The initial environment dictionary cannot be changed, but it can be shadowed (you can redefine something that is in the initial dictionary, but if you remove it, then the original initial value is still there). Note that the get keyword is not optional for a “get.”

```
(equal expr1 expr2
expr3)
```

```
(equali expr1 expr2
expr3)
```

The `equal` function evaluates the equivalency of the result of evaluating `expr1` and `expr2`. If they are equal, it returns:

1. The value of `expr3`, if specified, else
2. The value (and datatype, after possible string conversion) of `expr2`, as long as `expr2` is not null, else
3. The string “*T*” (since returning null would incorrectly indicate a failed comparison).

If `expr1` and `expr2` are not equal, the function returns null.

The arguments can be any datatype. If different, the function converts them to strings (which cannot fail) before comparing them. Note that any string conversion is performed using the equivalent of `to-string ...`). Thus, the blob 61:62 is not equal to the “ab” string. Note also that a one-byte blob 01 is not equal to a literal integer 1 (both are converted to strings, and the “01” and “1” strings are not equal).

The `equali` function is identical to the `equal` function, except that if the comparison is for strings (either because string arguments were used or because the arguments were converted to strings), a case insensitive comparison is used.

Table 26-1  Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
</table>
| (environmentdictionary
  {get | put val | delete}
attr) | nrcmd> dhcp set
  initial-environment-dictionary=first=one,
  second=2
  (environmentdictionary get "first")
  returns "one"
  (environmentdictionary get "second")
  returns "2" (note string 2)
  (environmentdictionary put "two"
  "second") returns "second"
  (environmentdictionary delete "first")
  returns null | | |

The `equali` function is identical to the `equal` function, except that if the comparison is for strings (either because string arguments were used or because the arguments were converted to strings), a case insensitive comparison is used.
Creating Expressions

Table 26-1  Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(error)</td>
<td>Returns a &quot;no recovery&quot; error that causes the entire expression evaluation to fail unless there is a try function above the error function evaluation.</td>
<td></td>
</tr>
<tr>
<td>(if cond [then else])</td>
<td>(if (equali (substring (request option &quot;dhcp-class-identifier&quot;) 0 6) &quot;docsis&quot;) (request option 82 1)) returns sub-option 1 of option 82 if the first six characters of the dhcp-class-identifier are &quot;docsis&quot; in any case; otherwise returns null</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Evaluates the condition expression cond in an if-then-else sense. If cond evaluates to a value that is nonnull, it returns the result of evaluating the then argument; otherwise it returns the result of evaluating the else argument. Both then and else are optional arguments. If you omit the then and else arguments, the function simply returns the results of evaluating the cond argument. If you omit the else argument and cond evaluates to null, the function returns null. There are no restrictions on the data types of any of the three arguments.</td>
<td></td>
</tr>
<tr>
<td>(ip-string blob)</td>
<td>(ip-string 01:02:03:04) returns &quot;1.2.3.4&quot; (ip-string -1) returns &quot;255.255.255.255&quot; (ip-string (as-blob &quot;hello world&quot;)] returns &quot;104.101.108.108&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returns the string representation of the four-byte IP address blob in the form &quot;a.b.c.d&quot;. The single argument blob must evaluate to a blob or be convertible into one. If the blob exceeds four bytes, the function uses only the first four to create the IP address string. If the blob has fewer bytes, the function considers the right-most bytes as zero when it creates the IP address string.</td>
<td></td>
</tr>
<tr>
<td>(ip6-string blob)</td>
<td>(ip6-string (as-blob &quot;hello world&quot;)] returns &quot;6865:6c6c:6f20:776f:726c:6400::&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returns the string representation of a 16-byte IPv6 address blob in the form &quot;a:b:c:d:e:f:g:h&quot;. The single argument blob must evaluate to a blob or be convertible into one. If the blob exceeds 16 bytes, the function uses only the first 16 to create the IPv6 address string. If the blob has fewer bytes, the function considers the right-most bytes as zero when it creates the IPv6 string.</td>
<td></td>
</tr>
<tr>
<td>(is-string expr)</td>
<td>(is-string 01:02:03:04) returns null (is-string &quot;hello world&quot;) returns &quot;hello world&quot; (is-string 68:65:6c6c:6f20:77:6f:72:6c64) returns the blob</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returns the value of expr, if the result of evaluating expr is a string or can be used as a string, this function, otherwise it returns null. That is, if as-string does not return an error, then is-string returns the value of expr.</td>
<td></td>
</tr>
<tr>
<td>(length expr)</td>
<td>(length 1) returns 4 (length 01:02:03) returns 3 (length &quot;hello world&quot;) returns 11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returns an integer whose value is the length, in bytes, of the value of expr. The argument expr can evaluate to any datatype. Integers always have length 4. The length of a string does not include any zero byte that may terminate the string.</td>
<td></td>
</tr>
</tbody>
</table>
Table 26-1 Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(let (var1 ... varn) expr1 ... exprn)</td>
<td>(let (x) (setq x (substring (request option &quot;dhcp-class-identifier&quot;) 0 6)) (if (equali x &quot;docsis&quot;) &quot;client-class-1&quot;) (if (equali x &quot;something else&quot;) &quot;client-class-2&quot;))</td>
<td></td>
</tr>
</tbody>
</table>

Creates an environment with local variables var1 through varn, which are initialized to a null value (you can give them other values by using the `setq` function). Once the local variables are initialized to null, the function evaluates expressions expr1 through exprn in order. It then returns the value of its last expression, exprn. The benefit of this function is that you can use it to calculate a value once, assign it to a local variable, then reuse that value in other expressions without having to recalculate it. Variables are case-sensitive.

(log severity expr)

Logs the result of converting expr to a string. The severity and expr must be a string and are converted to one if they do not evaluate to one. The severity can also be null; if a string, it must have one of these values:

- "debug"
- "activity" (the default if severity is null)
- "info"
- "warning"
- "error"

Note Logging consumes considerable server resources, so limit the number of log function evaluations you put in an expression. Even if "error" severity is logged, the log function does not return an error. This only tags the log message with an error indication. See the error function to return an error as part of a function evaluation.

(mask-blob mask-size length)

Returns a blob that contains the mask of length mask-size starting from the high-order bit of the blob, with a blob length of length. The mask-size is an expression that evaluates to an integer or must be convertible to one. Likewise the length, which cannot be smaller than the mask-size, but has no fixed limit except that it must be zero or positive. If mask-size is less than zero, it denotes a mask length calculated from the right end of the blob.

(mask-int mask-size)

Returns an integer mask of length mask-size bits starting from the high-order bit of the integer. The mask-size is an expression that evaluates to an integer or must be convertible to one. Any number over 32 is meaningless and is treated as though a value of 32 was used. If mask-size is less than zero, it denotes a mask length calculated from the right end of the integer.

(not expr)

Evaluates a string, blob, or integer expression to nonnull if it is null, and null if it is nonnull. The nonnull value returned when the value of expr is null is not guaranteed to remain the same over two calls.
### Table 26-1 Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(null [expr1 ... exprn])</td>
<td></td>
<td>Returns null and does not evaluate any of its arguments.</td>
</tr>
<tr>
<td>(or arg1 ... argn)</td>
<td>(or (request option 82 1) (request option 82 2) 01:02:03:04)</td>
<td>returns the value of sub-option 1 in option 82, and if that does not exist, returns the value of sub-option 2, and if that does not exist, returns 01:02:03:04</td>
</tr>
<tr>
<td>(pick-first-value arg1 ... argn)</td>
<td></td>
<td>Evaluates the arguments sequentially. When evaluating an arg returns a nonnull value, the first nonnull argument value is returned. Otherwise, returns the value of the last argument, argn. The datatypes need not be the same.</td>
</tr>
<tr>
<td>(progn arg ... argn)</td>
<td>(progn (log (null) &quot;I was here&quot;) (request option 82 1)) (request option 82 1) (request option 82 1)</td>
<td>Evaluates arguments sequentially and returns the value of the last argument, argn.</td>
</tr>
<tr>
<td>(return-last arg ... argn)</td>
<td></td>
<td>(progn (log (null) &quot;I was here&quot;) (request option 82 1)) (request option 82 1)</td>
</tr>
</tbody>
</table>
Creating Expressions

Returns the value of the option from the packet. The keywords are:

- **get**—Optional and assumed if omitted.
- **get-blob**—Returns the data as a blob, providing direct access to the option bytes.
- **relay**—Applies to IPv6 packets only, otherwise returns an error. Requests a relay option instead of a client option. The \( n \) indicates the \( n \)th closest relay agent to the client; if omitted, 0 (the relay agent nearest to the client) is assumed.
- **option**—Options are specified with the \( opt \) argument, which must evaluate to an integer or a string. If it does not evaluate to one of these, the function does not convert it and returns an error. Valid string values for the \( opt \) specifier are the same as those used for extensions.
- **enterprise-id**—After an option or suboption, and for DHCPv4 and DHCPv6, returns only the data bytes after the given enterprise-id in the packet, instead of the entire data of an option.
- **vendor**—After an option or suboption, requests that the vendor custom option definition be used for decoding the data in the option. Does not apply to DHCPv6 options. Note that if no definition exists for the specified vendor string, no error is issued and the standard definition of an option is used (or, if none, it is assumed to be a blob).
- **instance**—Selects the \((n+1)\)th instance of the preceding option or suboption. Instances start at 0. (You cannot use the instance and instance-count together in a single request function.)
- **instance-count**—Returns the number of instances of the preceding option or suboption, and is usually used to loop through all instances of it.
- **index**—Selects the \((n+1)\)th value in an option that contains multiple values. For example, **index 0** returns the first value and **index 1** returns the second value.
- **count**—Returns the number of relevant data items in the preceding option, and is usually used with the **index** keyword to loop through all data values for an option or suboption.

The only string-valued suboption names defined for the **subopt** (suboption) specifier are for the relay-agent-info option (82) and are:

- **circircuit-id**
- **remote-id**
- **device-class**
- **subnet-selection**
- **subscriber-id**
- **radius-attributes** (which includes the following encapsulated attributes that can be specified as subsuboptions: 1—"radius-user", 6—"radius-class", 88—"radius-framed-pool-name", 26—"radius-vendor-specific", 27—"radius-session-timeout", 100—"radius-framed-ipv6-pool")
- **authentication**
- **v-i-vendor-class**
- **cisco-subnet-selection**
- **cisco-vpn-id**
- **cisco-server-id-override**
- **vpn-id**
- **server-id-override**
The `request option` function returns a value with a datatype depending on the option requested. This shows how the datatypes in the table correspond to the datatypes returned by the `request` function:

- `blob` —> `blob`
- `IP address` —> `4-byte blob`
- `string` —> `string`
- `8-bit unsigned integer` —> `uint`
- `16-bit unsigned integer` —> `uint`
- `32-bit unsigned integer` —> `uint`
- `integer` —> `sint`
- `byte-valued boolean` —> `sint=1` if true, null if false

Valid values for `packetfield` are:

- `op` (blob 1)
- `htype` (blob 1)
- `hlen` (blob 1)
- `hops` (blob 1)
- `xid` (uint)
- `secs` (uint)
- `flags` (uint)
- `ciaddr` (blob 4)
- `yiaddr` (blob 4)
- `siaddr` (blob 4)
- `giaddr` (blob 4)
- `chaddr` (blob `hlen`)
- `sname` (string)
- `file` (string)

The `request packetfield` function returns the value of the named field from the request packet. DHCP request packets contain named fields as well as options in an option area. This form of the request function is used to retrieve specific named fields from the request packet. The `relay` keyword is described in the earlier, more general `request` function.

The `packetfield` values defined in RFC 2131 are listed at the left. There are several `packetfield` values that can be requested which do not appear in exactly these ways in the raw DHCP packet. These take data that appears in the packet and combine it in commonly used ways. In these explanations, the packet contents assumed are:

- `hlen` = 1
- `htype` = 6
- `chaddr` = 01:02:03:04:05:06

- `macaddress-string` (string) — Returns the MAC address in `hlen,htype,chaddr` format (for example, “1,6,01:02:03:04:05:06”)
- `macaddress-blob` (blob) — Returns the MAC address in `hlen:htype:chaddr` format (for example, 01:06:01:02:03:04:05:06)
- `macaddress-clientid` (blob) — Returns a client-id created from the MAC address in the Microsoft `htype:chaddr` client-id format (for example, 01:01:02:03:04:05:06)
Table 26-1  Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Valid values for the DHCPv6 packetfield are:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msg-type (uint)</td>
<td></td>
<td>The msg-type packet field for DHCPv6 describes the current relay or client message type, and has the values:</td>
</tr>
<tr>
<td>msg-type-name (string)</td>
<td></td>
<td>1=SOLICIT, 2=ADVERTISE, 3=REQUEST, 4=CONFIRM,</td>
</tr>
<tr>
<td>xid (uint)</td>
<td></td>
<td>5=RENEW, 6=REBIND, 8=RELEASE, 9=DECLINE,</td>
</tr>
<tr>
<td>relay-count (uint)</td>
<td></td>
<td>11=INFORMATION-REQUEST, 12=RELAY-FORWARD</td>
</tr>
<tr>
<td>hop-count (uint)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link-address (blob 16)</td>
<td></td>
<td>The xid is the 24-bit client transaction ID, and the relay-count is the number of relay messages in the request.</td>
</tr>
<tr>
<td>peer-address (blob 16)</td>
<td></td>
<td>If a DHCPv6 packet field is requested from a DHCPv4 packet, an error is returned. The inverse is also true.</td>
</tr>
</tbody>
</table>

(request dump)

Dumps the current request packet to the log file, after the function evaluates the expression. Note that not all expression evaluations support the dump keyword, and when unsupported, it is ignored.

(requestdictionary {get | put val | delete} attr)

Gets, puts, or deletes a DHCP extension request dictionary attribute value, val is the value of the attribute and attr is the attribute name. Both are converted to a string regardless of their initial datatype. Note that the get keyword is not optional for a “get.”

(response [get | get-blob] [relay [n]] option opt [[enterprise-id n] | {vendor string}] [instance n] [[subopt | {option opt}] [[enterprise-id n] | {vendor string}] [instance n] [[sub-subopt | {option opt}] [[enterprise-id n] | {vendor string}] [instance n] [instance-count | {index n} | count])

Returns the value of the option from the packet. The keywords are identical to those for the request function.

(response [get | get-blob] [relay [number]] packetfield)

Returns the value of the named packetfield from the response packet. The description and valid values are identical to those for the request packetfield function.

(response dump)

Dumps the current response packet to the log file after the function evaluates the expression. Note that not all expression evaluations support the dump keyword, and when unsupported, it is ignored.

(responsedictionary {get | put val | delete} attr)

Gets, puts, or deletes a DHCP extension response dictionary attribute value. The val is the value of the attribute and attr is the attribute name. Both are converted to a string regardless of their initial datatype. Note that the get keyword is not optional for a “get.”
Table 26-1 Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(search arg1 arg2 fromend)</td>
<td>(search &quot;test&quot; &quot;this is a test&quot;) returns 9</td>
<td>(search &quot;test&quot; &quot;this test test test&quot; &quot;true&quot;) returns 15</td>
</tr>
</tbody>
</table>

Searches arg1 for a subsequence in arg2 that exactly matches. If found, it returns the index of the element in arg2 where the subsequence begins (unless you set the fromend argument to “true” or some other arbitrary value); otherwise it returns null. (If arg1 is null, it returns 0; if arg2 is null, it returns null.) The function does an implicit as-blob conversion on both arguments. Thus, it compares the actual byte sequences of strings and blobs, and sints and uints become 4-byte blobs for the purpose of comparison.

A nonnull fromend argument returns the index of the leftmost element of the rightmost matching subsequence.

(setq var expr)

Sets var to the value of expr. You must precede it with the let function.

(starts-with expr prefix-expr)

Returns the value of expr if the prefix-expr value matches the beginning of expr, otherwise null. If prefix-expr is longer than expr, it returns null. The function returns an error if prefix-expr cannot be converted to the same datatype as expr (string or blob), or if expr evaluates to an integer. (See Table 26-2 on page 26-21.)

(substring expr offset len)

Returns len bytes of expression expr, starting at offset. The expr can be a string or blob; if an integer, converts to a blob. The result is a string or a blob, or null if any argument evaluates to null. If:

- offset is greater than the length len, the result is null.
- offset plus len is data beyond the end of expr, the function returns the rest of the data in expr.
- offset is less than zero, the offset is from the end of the data (the last character is index –1, because –0=0, which references the first character).
- This references data beyond the beginning of data, the offset is considered to be zero.
### Chapter 26  Using Expressions

#### Creating Expressions

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(synthesize-host-name method namestem)</td>
<td>(synthesize-host-name) returns &quot;dhcp-rhfxxi5pkjp6c&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(synthesize-host-name &quot;duid&quot; &quot;test&quot;) returns &quot;test-00030001010203040506&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(synthesize-host-name &quot;client-id&quot; &quot;test&quot;) returns &quot;test-00030001010203040506&quot;</td>
<td></td>
</tr>
</tbody>
</table>

Generates a hostname based on the configured method (if none is specified), or the specified `method` and `namestem`.

The `method` argument can have the value `configured` to specify the configured method (thus allowing a `namestem` specification), `default`, or one of the v6-synthetic-name-generator enumeration values (if IPv6) of the DNS update configuration (`hashed-duid`, `duid`, `cablelabs-device-id`, or `cablelabs-cm-mac-addr`; see the “Generating Synthetic Names in DHCPv4 and DHCPv6” section on page 29-3).

The `method` argument can have the value `configured` to specify the configured method (thus allowing a `namestem` specification), `default`, or one of the v4-synthetic-name-generator enumeration values (if IPv4) of the DNS update configuration (`address`, `hashed-client-id`, or `client-id`; see the “Generating Synthetic Names in DHCPv4 and DHCPv6” section on page 29-3).

The `namestem` argument specifies the `synthetic-name-stem` value of the DNS update configuration (see the “Creating DNS Update Configurations” section on page 29-5).

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(to-blob expr)</td>
<td>(to-blob 1) returns 00:00:00:01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(to-blob &quot;01:02&quot;) returns 01:02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(to-blob 02:03) returns 02:03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(to-blob &quot;this is not in blob format&quot;) return an error</td>
<td></td>
</tr>
</tbody>
</table>

Converts an expression to a blob. If:

- `expr` evaluates to a string it must be in “nn:nn:nn” format. This function returns a blob that is the result of converting the string to a blob.
- The function cannot convert the string to a blob, it returns an error.
- `expr` evaluates to a blob, it returns that blob.
- `expr` evaluates to an integer, it returns a four-byte blob representing the bytes of the integer in network order. (See Table 26-2 on page 26-21.)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(to-ip expr)</td>
<td>(to-ip6 expr)</td>
<td></td>
</tr>
</tbody>
</table>

Converts an expression as string, blob, or integer to an IP address. If:

- A string, it must be in dotted decimal IP address format for IPv4 or colon-formatted format for IPv6. Returns the blob IP address determined by parsing the string into an IP address.
- The result is a blob, it returns the first bytes of the blob.
- The blob is less than four bytes, it pads the argument blob with zero bytes in the high order bytes.
- The result is an integer, it converts the integer (of either type) into a blob. Because the integers and blobs are in network order, no order change is required.
Creating Expressions

**Table 26-1 Expression Functions (continued)**

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(to-lower expr)</strong></td>
<td></td>
<td>Takes a string and produces a lowercase string from it. When using the <code>client-lookup-id</code> attribute to calculate a client-specifier to look up a client-entry in the CNRDB local store (as opposed to LDAP), the resulting string must be lowercase. Use this function to easily make the result of the <code>client-lookup-id</code> a lowercase string. You may or may not want to use this function when accessing LDAP using the <code>client-lookup-id</code>.</td>
</tr>
<tr>
<td><strong>(to-sint expr)</strong></td>
<td>(to-sint &quot;1&quot;) returns 1</td>
<td>Converts an expression to a signed integer.</td>
</tr>
<tr>
<td></td>
<td>(to-sint -1) returns -1</td>
<td>If <code>expr</code> evaluates to a string, it must be in a format that can be converted into a signed integer, else the function returns an error. If:</td>
</tr>
<tr>
<td></td>
<td>(to-sint 00:02) returns 2</td>
<td>• <code>expr</code> evaluates to a blob of one to four bytes, the function returns it as a signed integer.</td>
</tr>
<tr>
<td></td>
<td>(to-sint &quot;00:02&quot;) returns an error</td>
<td>• <code>expr</code> evaluates to a blob of more than 4 bytes in length, it returns an error.</td>
</tr>
<tr>
<td></td>
<td>(to-sint &quot;4294967295&quot;) returns an error</td>
<td>• <code>expr</code> evaluates to a signed integer, it returns a signed integer with the same value, unless the value of the unsigned integer was greater than the largest positive signed integer, in which case it returns an error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>expr</code> evaluates to a signed integer, it returns that value. (See Table 26-2 on page 26-21.)</td>
</tr>
<tr>
<td><strong>(to-string expr)</strong></td>
<td>(to-string &quot;hello world&quot;) returns 'hello world'</td>
<td>Converts an expression to a string. If <code>expr</code> evaluates to a string, it returns it; if a blob or integer, it returns its printable representation. It never returns an error if <code>expr</code> itself evaluates without error, because every value has a printable representation. (See Table 26-2 on page 26-21.)</td>
</tr>
<tr>
<td></td>
<td>(to-string -1) returns &quot;-1&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(to-string 02:04:06) returns &quot;02:04:06&quot;</td>
<td></td>
</tr>
<tr>
<td><strong>(to-uint expr)</strong></td>
<td>(to-uint &quot;1&quot;) returns 1</td>
<td>Converts an expression to an unsigned integer. If <code>expr</code> evaluates to a string, it must be in a format that can be converted into an unsigned integer, else the function returns an error. If:</td>
</tr>
<tr>
<td></td>
<td>(to-uint 00:02) returns 2</td>
<td>• <code>expr</code> evaluates to a blob of one to four bytes, it returns it as an unsigned integer.</td>
</tr>
<tr>
<td></td>
<td>(to-uint &quot;4294967295&quot;) returns 4294967295</td>
<td>• <code>expr</code> evaluates to a blob of more than 4 bytes in length, it returns an error.</td>
</tr>
<tr>
<td></td>
<td>(to-uint &quot;00:02&quot;) returns an error</td>
<td>• <code>expr</code> evaluates to a signed integer, it returns an unsigned integer with the same value, unless the value of the signed integer less than zero, in which case it returns an error.</td>
</tr>
<tr>
<td></td>
<td>(to-uint -1) returns an error</td>
<td>• <code>expr</code> evaluates to an unsigned integer, the function returns that value. (See Table 26-2 on page 26-21.)</td>
</tr>
</tbody>
</table>
Creating Expressions

(take expression search replace)

(translate "Hello apple and eve\" \"abcdef\" \"123456\") returns "H5llo 1ppl5 In4 5v5"

(translate \"a\&b\$c%d\" \"%\$\&\") returns "abcd"

Takes as an argument an expression that evaluates to a sequence of bytes (either a string or a blob), and replaces various characters or bytes that appear in search with corresponding values (in the same position) in replace. If:

- expr is a string or blob, the value is left as it is, otherwise it is forced to be a string. If, after processing, expr is a string, search and replace must be strings.
- expr is a blob, both search and replace must also be blobs.
- replace is shorter than search, the bytes or characters in search that do not have corresponding bytes or characters in replace are dropped from the output.
- replace does not appear, all the bytes or characters in search are removed from expr.

(try expr failure-exp)

Evaluates expr and returns the result of that evaluation if there were no errors encountered during the evaluation. If an error occurs while evaluating expr then:

- If there is a failure-exp and it evaluates without error, it returns the result of that evaluation as the result of the try function.
- If there is a failure-exp and the function encounters an error while evaluating failure-exp, it returns that error.
- If there is no failure-exp, the try returns null.

### Table 26-1 Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| (translate expr search replace) | (translate \"Hello apple and eve\" \"abcdef\" \"123456\") returns "H5llo 1ppl5 In4 5v5" (translate \"a\&b\$c%d\" \"%\$\&\") returns "abcd" | Takes as an argument an expression that evaluates to a sequence of bytes (either a string or a blob), and replaces various characters or bytes that appear in search with corresponding values (in the same position) in replace. If:  
- expr is a string or blob, the value is left as it is, otherwise it is forced to be a string. If, after processing, expr is a string, search and replace must be strings. 
- expr is a blob, both search and replace must also be blobs. 
- replace is shorter than search, the bytes or characters in search that do not have corresponding bytes or characters in replace are dropped from the output. 
- replace does not appear, all the bytes or characters in search are removed from expr. |

(try (try (expr) (complex-failure-exp))  
"string-constant" ensures that the outer try never returns an error (because evaluating "string-constant" cannot fail).  
(try (error) 01:02:03) always returns 01:02:03  
(try 1 01:02:03) always returns 1  
(try (request option 82) "failure") never returns "failure" because (request option 82) turns null if there is no option-82 in the packet and does not return an error  
(try (request option "junk") "failure") returns "failure" because "junk" is not a valid option-name. | Evaluates expr and returns the result of that evaluation if there were no errors encountered during the evaluation. If an error occurs while evaluating expr then:  
- If there is a failure-exp and it evaluates without error, it returns the result of that evaluation as the result of the try function.  
- If there is a failure-exp and the function encounters an error while evaluating failure-exp, it returns that error.  
- If there is no failure-exp, the try returns null. |
Table 26-1  Expression Functions (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(validate-host-name hostname)</td>
<td>(validate-host-name &quot;a b c d e f&quot;)</td>
<td>returns &quot;a-b-c-d-e-f&quot;</td>
</tr>
<tr>
<td></td>
<td>(validate-host-name &quot;a b c d e f&quot;)</td>
<td>returns &quot;a-b-c-d-e-f&quot;</td>
</tr>
<tr>
<td></td>
<td>(validate-host-name &quot;abcdef&quot;)</td>
<td>returns &quot;abcdef&quot;</td>
</tr>
<tr>
<td></td>
<td>(validate-host-name &quot;a&amp;b*c#d@!e()f&quot;)</td>
<td>returns &quot;abcdef&quot;</td>
</tr>
</tbody>
</table>

Takes the hostname string and returns a validated hostname, which can be the same as the input hostname or modified as follows:

- Space and underscore characters mapped to a hyphen.
- Invalid hostname characters removed. Valid characters are A-Z, a-z, 0-9, and hyphen.
- Null labels removed (".." changed to ".").
- Each label in the hostname truncated to 63 characters.

Datatype Conversions

When a function needs an argument of a particular datatype, it tries to convert a value into that datatype. Sometimes this can fail, often causing the entire function to fail. Datatype conversion is also performed by the to-string, to-blob, to-sint, and to-uint functions. Whenever a function needs an argument in a specific datatype, it calls the internal version of these externally available functions.

There are also as-string, as-blob, as-sint, and as-uint conversion functions, where the data in a value are simply relabeled as the desired datatype, although some checking does go on. The conversion matrix for both function sets appears in Table 26-2 on page 26-21.

Table 26-2  Datatype Conversion Matrix

<table>
<thead>
<tr>
<th>Function</th>
<th>String</th>
<th>Blob</th>
<th>Signed Integer</th>
<th>Unsigned Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>as-blob</td>
<td>Cannot fail; relabels ASCII characters as blob bytes.</td>
<td>—</td>
<td>Cannot fail; produces a 4-byte blob from the 4 bytes of the integer.</td>
<td>Cannot fail; produces a 4-byte blob from the 4 bytes of the integer.</td>
</tr>
<tr>
<td>as-sint</td>
<td>Not usually useful; converts a 1-, 2-, 3-, or 4-byte string to a blob and then packs it up into a signed integer.</td>
<td>Not usually useful; converts only 1-, 2-, 3-, or 4-byte blobs.</td>
<td>—</td>
<td>Cannot fail; converts to a signed integer, negative if a larger unsigned integer would fit into a positive signed integer.</td>
</tr>
<tr>
<td>as-string</td>
<td>—</td>
<td>Relabels as string bytes, if printable characters</td>
<td>Converts to a 4-byte blob, then processes it as a blob (which fails except for a few special integers)</td>
<td>Converts to a 4-byte blob, then processes as a blob (which fails except for a few special integers)</td>
</tr>
</tbody>
</table>
Expression Examples

You must include the expression in a text file if you want to include it with a CLI attribute setting. The default path of this file is the current working directory. Do not enclose the expression in quotes. You can add comment lines prefixed by #, //, or ;. For example:

```
// Expression to set client-class based on remote-id
(if (equal (request option "relay-agent-info" "remote-id") (request chaddr))
 "no-limit" "limit")
```

The server reads the file when processing the command. An example of a command to include this file is:
```
nrcmd> dhcp set client-class-lookup-id=@expressionfile1.txt
```

### Expression Examples

These examples provide the maximum support for option 82 processing. They set up clients to limit, those not to limit, and those that exceed configuration limits and should be assigned to an over-limit client-class. There are separate scopes and selection tags for each of the three classes of clients:

- **Client-classes**—limit, no-limit, and over-limit.
- **Scopes**—10.0.1.0 (primary), 10.0.2.0 and 10.0.3.0 (secondaries), named for their subnets.
- **Selection tags**—limit-tag, no-limit-tag, and over-limit-tag. The scopes are named for the address pools that they represent. The selection tags are allocated to the scopes with 10.0.1.0 getting limit-tag, 10.0.2.0 getting no-limit-tag, and 10.0.3.0 getting over-limit-tag.
Limitation Example 1: DOCSIS Cable Modem

The test is to determine whether the device is considered a DOCSIS cable modem, and limit the number of customer devices behind every cable modem. The limitation ID for the limit client-class is the cable modem MAC address, included in the remote-id suboption of the relay-agent-info option.

The expression for the client-class-lookup-id attribute on the server is:

// Expression to set client-class to no-limit or limit based on remote-id
(if (equal (request option "relay-agent-info" "remote-id")
    (request chaddr))
   "no-limit"
   "limit")

The above expression indicates that if the contents of the remote-id suboption (2) of the relay-agent-info option is the same as the chaddr of the packet, then the client-class is no-limit, otherwise limit.

The limitation-id expression for the limit client-class is:

(request option "relay-agent-info" "remote-id")

Use this expression in the following steps:

---

**Step 1**
Define the client-classes.

**Step 2**
Define the scopes, their ranges and tags, and if they are primary or secondary. Note the host range for each scope, which is less likely to be misread than if they all have the same host number.

**Step 3**
Define the limitation count. It can go in the default policy; if the request does not show a limitation ID, the count is not checked.

**Step 4**
Add an expression in an expression file, cclookup1.txt, for the purpose:

// Expression to set limitation count based on remote-id
(if (equal (request option "relay-agent-info" "remote-id")
    (request chaddr)) "no-limit" "limit")

**Step 5**
Refer to the expression file when setting the client-class lookup-id attribute on the server level.

**Step 6**
Add another expression for the limitation ID for the client in a cclimit1.txt file:

// Expression to set limitation ID based on remote-id
(request option "relay-agent-info" "remote-id")

**Step 7**
Refer to this expression file when setting the limitation-id attribute for the client-class.

**Step 8**
Reload the server.

---

The result of doing this for a previously unused configuration would be to put the first two DHCP clients with a common remote-id option 82 suboption value in the limit client-class. The third client with the same value would go in the over-limit client-class. There are no limits to the number of devices a
subscriber can have in the no-limit client-class, because it has no configured limitation ID. Any device with a MAC address equal to the value of the remote-id suboption is ignored for the purposes of limitation, and goes in the no-limit client class, for which there is no limitation ID configured.

**Limitation Example 2: Extended DOCSIS Cable Modem**

This example is an extension to the example described in the “Limitation Example 1: DOCSIS Cable Modem” section on page 26-23. In the latter example, all of the cable modems allowed only two client devices beyond them, since a limitation count of two was defined for the default policy. In this example, specific cable-modems are configured to allow a different number of devices to be granted IP addresses from the scopes that use the limit-tag selection tag.

In this case, you need to explicitly configure any cable modem with more than two addresses behind it in the client-class database. This requires enabling client-class processing server-wide, so that you can look up the client entry for a cable modem in the Cisco Prime IP Express or LDAP database. Not finding the cable modem limits the number of devices to two; finding it uses the limitation count from the policy configured for the cable modem.

This example requires just one additional policy, five, which allows five devices.

**Step 1** Enable client-class processing server-wide.
**Step 2** Create the five policy with a limitation count of five devices.
**Step 3** As in the previous example, use an expression to set a limitation ID for the limit client-class. Put the limitation ID in a cclimit2.txt file, and the lookup ID in a cclookup2.txt file:

**cclimit2.txt file:**

// Expression to set limitation ID
(request option "relay-agent-info" "remote-id")

**cclookup2.txt file:**

// Expression to set client-class lookup ID
(concat "1,6," (to-string (request option "relay-agent-info" "remote-id")))

**Step 4** Refer to these files when setting the appropriate attributes.
**Step 5** Define some cable modem clients and apply the five policy to them.
**Step 6** Reload the server.

**Limitation Example 3: DSL over Asynchronous Transfer Mode**

This example shows how to use expressions to configure Digital Subscriber Line (DSL) access for a subscriber to a service provider using asynchronous transfer mode (ATM) routed bridge encapsulation (RBE). Service providers are increasingly using ATM RBE to configure a DSL subscriber. The DHCP Option 82 support for routed bridge encapsulation feature as of Cisco IOS Release 12.2(2)T enables those service providers to use DHCP to assign IP addresses and option 82 to implement security and IP address assignment policies.

In this scenario, DSL subscribers are identified as individual ATM subinterfaces on a Cisco 7401ASR router. Each customer has their own subinterface in the router and each subinterface has its own virtual channel identifier (VCI) and virtual path identifier (VPI) to identify the next destination of an ATM cell as it passes through ATM switches. The 7401ASR router routes up to a Cisco 7206 gateway router.
Step 1  Set up the DHCP server and interfaces for the router using IOS. This is a typical IOS configuration:

```bash
Router#ip dhcp-server 170.16.1.2
Router#interface Loopback0
Loopback0(config)#ip address 11.1.1.129 255.255.255.192
Loopback0(config)#exit
Router#interface ATM4/0
ATM4/0(config)#no ip address
ATM4/0(config)#exit
Router#interface ATM4/0.1 point-to-point
ATM4/0.1(config)#ip unnumbered Loopback0
ATM4/0.1(config)#ip helper-address 170.16.1.2
ATM4/0.1(config)#atm route-bridged ip
ATM4/0.1(config)#pvc 88/800
ATM4/0.1(config)#encapsulation aal5snap
ATM4/0.1(config)#exit
Router#interface Ethernet5/1
Ethernet5/1(config)#ip address 170.16.1.1 255.255.0.0
Ethernet5/1(config)#exit
Router#router eigrp 100
eigrp(config)#network 11.0.0.0
eigrp(config)#network 170.16.0.0
eigrp(config)#exit
```

Step 2  In IOS, enable the system to insert the DHCP option 82 data in forwarded BOOTREQUEST messages to a Cisco IOS DHCP server:

```bash
Router#ip dhcp relay information option
```

Step 3  In IOS, specify the IP address of the loopback interface on the DHCP relay agent that is sent to the DHCP server using the option 82 remote-id suboption (2):

```bash
Router#rbe nasip Loopback0
```

Step 4  In Cisco Prime IP Express, enable client-class processing server-wide.

Step 5  Create the one policy with a limitation count of one device.

Step 6  Put the packets in the right client-class. All the packets should be in the limit client-class. Create a lookup file containing just the value limit, then set the client-class lookup ID. In the cclookup3.txt file:

```bash
// Sets client-class to limit
"limit"
```

Step 7  Use an expression to ensure that those packets that are limited have the right limitation ID. Put the expression in a file and refer to that file to set the limitation ID. The substring function gets the VPI/VCI by extracting bytes 10 through 12 of the option 82 suboption 2 (remote-id) data field. In the cclimit3.txt file:

```bash
// Sets limitation ID
(substring (request option 82 2) 9 3)
```

Step 8  Reload the server.
Debugging Expressions

If you are having trouble with expressions, examine the DHCP log file at server startup. The expression is printed in such a way as to clarify the nesting of functions, and can help in confirming your intentions. Pay special attention to the `equal` function and any datatype conversions of arguments. If the arguments are not the same datatype, they are converted to strings using code similar to the `to-string` function.

You can set various debug levels for expressions by using the `expression-trace-level` attribute for the DHCP server. All executed expressions are traced to the degree set by the attribute. The highest trace level is 10. If you set the level to at least 2, any nonworking expression is retried again at level 10.

The trace levels for `expression-trace-level` are (use the number value):

- 0—No tracing
- 1—Failures, including those protected by `(try ...)`
- 2—Total failure retries (with trace level = 6 for retry)
- 3—Function calls and returns
- 4—Function arguments evaluated
- 5—Print function arguments
- 6—Datatype conversions (everything)

The trace levels for `expression-configuration-trace-level` are (use the number value):

- 0—No additional tracing
- 1—No additional tracing
- 2—Failure retry (the default)
- 3—Function definitions
- 4—Function arguments
- 5—Variable lookups and literal details
- 6—Everything

To trace expressions you have trouble configuring, there is also an `expression-configuration-trace-level` attribute that you can set to any level from 1 through 10. If you set the level to at least 2, any expression that does not configure is retried again with the level set to 6. Gaps in the numbering are to accommodate future level additions.
Managing DHCPv6 Addresses

Cisco Prime IP Express supports the following IPv6 addressing for DHCP (DHCPv6):

- **Stateless autoconfiguration (RFC 3736)**—The DHCPv6 server does not assign addresses, but instead provides configuration parameters, such as DNS server data, to clients.
- **Stateful autoconfiguration (RFC 3315)**—The DHCPv6 server assigns nontemporary or temporary addresses and provides configuration parameters to clients.
- **Prefix Delegation (RFC 3633)**—The DHCPv6 server delegates prefixes to clients (routers).

The DHCPv6 service provides these capabilities:

- **Allocation groups**—Allows multiple prefixes to be treated as one from an allocation standpoint, and provides control over the order in which the prefixes are used.
- **Client-classing**—You can classify clients and select prefixes based on known clients or packet-based expressions.
- **DNS Updates**—DNS server updates of DHCP activity (over IPv4).
- **Extensions**—Extend the DHCP server processing by using C/C++ and Tcl extensions.
- **Failover**—You can configure a DHCP failover pair so that if one cannot provide leases to requesting clients, another one can take over.
- **LDAP**—Allows client entry lookups in an LDAP repository (external to Cisco Prime IP Express) and these clients may specify client reservations. LDAP writes/updates are not supported for DHCPv6.
- **Leasequery**—Offers leasequery support.
- **Links and prefixes**—Similar to DHCPv4 networks and scopes that define the network topology. Each link can have one or more prefixes.
- **Policies and options**—You can assign attributes and options to links, prefixes, and clients.
- **Prefix Stability**—Clients can retain the delegated prefix when they change their location, that is even when they move from one CMTS to another or move within an address space. Prefix Stability, with appropriate infrastructure support (CMTS, routers), allows the subscriber to be moved or move without requiring a different delegated prefix.
- **SNMP traps**—Generate traps for events, such as if the number of leases in a prefix exceeds a certain limit (or drops below a certain limit) or if the server detects duplicate addresses.
- **Reservations**—Clients can receive predetermined addresses.
- **Statistics collection and logging**—Provides server activity monitoring.
- **VPN support**—Provides multiple address spaces (virtual private networks).

The DHCPv6 service requires that the server operating system support IPv6 and that you configure at least one interface on the system for IPv6.
DHCPv6 Concepts

The following subsections describe the concepts related to DHCPv6 operation:

- **IPv6 Addressing**
- **Links and Prefixes**
- **DHCPv6 Clients and Leases, page 27-7**
- **DHCPv6 Policy Hierarchy, page 27-12**
- **DHCPv6 Options, page 27-13**

**IPv6 Addressing**

IPv6 addresses are 128 bits long and are represented as a series of 16-bit hexadecimal fields separated by colons (:). The A, B, C, D, E, and F in hexadecimal are case insensitive. For example:

```
2001:db8:0000:0000:0000:0000:0000:0000
```

A few shortcuts to this addressing are:

- Leading zeros in a field are optional, so that you can write 09c0 as 9c0, and 0000 as 0.
- You can represent successive fields of zeros (any number of them) by a double colon (::), but only once in an address (because, if used more than once, the address parser has no way of identifying the size of each block of zeros). This reduces the length of addresses; for example, 2001:db8:0000:0000:0000:0000:0000:0000 can be written:

```
2001:db8::
```

Link-local addresses have a scope limited to the link, and use the prefix fe80::/10. Loopback addresses have the address ::1. Multicast addresses have the prefix ff00::/8 (there are no broadcast addresses in IPv6).

The IPv4-compatible addresses in IPv6 are the IPv4 decimal quad addresses prefixed by ::. For example, you can write the IPv4 address interpreted as ::c0a8:1e01 in the form ::192.168.30.1.

**Links and Prefixes**

The explicit DHCPv6 configuration objects are links and prefixes:

- **Link**—Network segment that can have one or more prefixes, and adds an additional layer at which policies can be applied for DHCPv6 clients.
- **Prefix**—Equates to a scope in IPv4. The link associated with a prefix is similar to a primary scope, except that it names a link and not another prefix.
Chapter 27      Managing DHCPv6 Addresses

DHCPv6 Concepts

Just as with scopes, you can create multiple prefix objects for the same IPv6 prefix. However, rather than supporting multiple ranges with explicit start and end addresses, prefixes support only a single range that must be an IPv6 prefix with a length the same as, or longer than, the prefix object. For example, if you define a 2001::/64 prefix with a 2001::/96 range, the server can assign addresses from 2001:0:0:0:0:0:0:0 through 2001:0:0:0:0:0:ffff:ffff only. The range:

- Is limited to powers of 2.
- Must be unique (cannot be duplicated by any other range, except in a different VPN).
- Cannot be contained in, or contain, another range, except for prefix delegation prefixes, as explained below.
- Is the full IPv6 prefix if not specified, except for prefix delegation prefixes, as explained below.

If a prefix delegation prefix object is defined with an unspecified range, it may contain non prefix-delegation prefixes, and the effective range is either:

- The full IPv6 prefix if no other prefixes exist with the same IPv6 prefix, or
- The prefixes that remain when all other ranges for prefix objects with the same IPv6 prefix are removed from the IPv6 prefix.

You create a link only if more than one prefix object with a different IPv6 prefix exists on a link. When the server loads the configuration, if a prefix has no explicit link, the server searches for or creates an implicit link with the name Link-[vpn.name]/prefix. All prefix objects with the same IPv6 prefix must either not specify a link or explicitly specify the same link.

The DHCPv6-enabled server supports VPN address spaces for DHCPv6. Both the link and prefix objects may be assigned to a VPN. But all prefixes on a link must use the same VPN ID. Because there is presently no DHCPv6 VPN option, clients can only be assigned addresses from a VPN by using the client or client-class override-vpn attribute.

Related Topics

- Determining Links and Prefixes
- Generating Addresses, page 27-4
- Generating Delegated Prefixes, page 27-5
- Prefix Stability, page 27-5
- Prefix Allocation Groups, page 27-6

Determining Links and Prefixes

When the DHCPv6 server receives a DHCPv6 message, it determines the links and prefixes it uses to service the request. The server:

1. Finds the source address:
   a. If the client message was relayed, the server sets the source address to the first nonzero link-address field starting with the Relay-Forward message closest to the client (working outwards). If the server finds a source address, it proceeds to step 2.
   b. Otherwise, if the message source address is a link-local address, the server sets the source address to the first address for the interface on which it received the message for which a prefix exists (or 0 if it finds no prefix for any address). It then proceeds to step 2.
   c. Otherwise, the server sets the source address to the message source address.

2. Locates the prefix for the source address. If the server cannot find a prefix for the source address, it cannot service the client and drops the request.
3. Locates the link for the prefix. This always exists and is either an explicitly configured link or the implicitly created link based on the prefix address. The link must be a topological link (see the “Prefix Stability” section on page 27-5 section below).

Now that the server can determine the client link, it can process the client request. Depending on whether the client request is stateful or prefix-delegated, and on the selection criteria and other factors, the server might use one or more prefixes for the link to service the client request.

This is one area of difference between DHCPv4 and DHCPv6. In DHCPv4, the server selects only one of the scopes from the network to service the client request. In DHCPv6, the server can use all the prefixes for the link. Thus, the server might assign a client an address, or delegate a prefix, from multiple prefixes for the link (subject to selection criteria and other conditions). (See the “Configuring Links” section on page 27-14.)

Generating Addresses

IPv6 addresses are 128-bit addresses (as compared to 32-bit addresses for IPv4). In most cases, DHCPv6 servers assign 64 of those bits, the interface-identifier (EUI-64) portion (see RFC 4291). You can generate addresses by using the client 64-bit interface-identifier or a random number generator. The interface-identifier emulates how stateless autoconfiguration assigns addresses to clients. Unfortunately, there are privacy concerns regarding its use, and it is limited to one address per prefix for the client.

By default, Cisco Prime IP Express generates an address using an algorithm similar to that described in RFC 4941 to generate a random interface identifier. These random interface identifiers have a zero value for the universal/local bit to distinguish them from EUI-64-based identifiers. The server also skips randomly generated interface identifiers from ::0 to ::ff so that you can use identifiers for infrastructure devices (such as routers). You can configure whether to assign the interface-identifier (if available) first for each prefix (through the interface-identifier flag of the prefix allocation-algorithms attribute). (See the “Creating and Editing Prefixes” section on page 27-25.) If you specify use of the interface-identifier, the server might still use randomly generated addresses if the address is not available to the client, or the client requests multiple addresses on a prefix.

The server generates addresses based on the prefix-configured range (or the prefix address if there is no range). If the range prefix length is shorter than 64, the server supplies only 64 bits and places them in the address interface-identifier field. If the prefix length is longer than 64, the server supplies only the remaining bits of the address. Thus, a /96 range uses 96 bits from the specified range followed by 32 bits of either the client interface-identifier or a randomly generated value. If the resulting address is not available (such as if it is already leased to another client, or to the same client, but on a different binding), the server tries to generate another address. It repeats this process up to at most 500 times.

When DHCP failover is configured, the server generated addresses are always odd addresses on the main and even addresses on the backup.

Note

The DHCP server tests only the randomly generated interface identifier for values from ::0 to ::ff, not the resulting address. Thus, a randomly generated address may end up using an xxxxxxxxxxxxxxxxxxx:0 through xxxxxxxxxxxxxxxxxxx:ff address if the length of the prefix is longer than /64 and the prefix bits that extend beyond the /64 boundary are all zero.

Tip

You can also choose from additional address generation algorithms for a prefix and prefix template; see the “Creating and Editing Prefix Templates” section on page 27-20.
Generates Delegated Prefixes

The DHCPv6 server uses the best first-fit algorithm when generating delegated prefixes. The server uses the first longest available prefix of the length configured or requested.

For DHCP failover, each server only considers the delegated prefix leases in the available state. When the server is in the PARTNER-DOWN state, the server can also use leases in the other-available or pending-available states after certain time restrictions have passed.

Prefix Stability

Prefix Stability is to let you control prefix delegation independent of the network topology. A new link attribute type specifies the type of link.

There are three different link types:

- Topological—A client on a topological link is allocated leases based on the network segment it is connected to.
- Location independent—This link type is introduced to support the CableLabs DOCSIS 3.0 concept of CMTS prefix stability. It supports service provider load balancing and reconfiguration events within a group of CMTS (such as in a central office). A subscriber that is moved from one CMTS to another on a location-independent link can retain a delegated prefix. This link type allows movement within a single DHCP server.
- Universal—This link type is introduced to let subscribers retain a delegated prefix anywhere in the network. Use of this link type requires administrative assignment of the delegated prefixes and use of client or lease reservations. It can be deployed across multiple DHCP servers.

Note

Use of prefix stability has routing implications and requires appropriate support from relay agents (that is, CMTS) in order to advertise the routes. For CMTS prefix stability, these are localized to the CMTS group. The implications are greater for universal prefix stability as routes need to be advertised throughout the service providers network.

CMNTS Prefix Stability

Location independent links implement the CableLabs DOCSIS 3.0 requirements for CMTS prefix stability. CMTS prefix stability is possible as long as all prefixes are serviced by a single DHCP server.

If you want to introduce CMTS prefix stability in a particular area, you need to:

- Modify existing links to specify the same link group name across all of the links within the group. Each CMTS (or CMTS bundle) will have a separate link, but all of these links within the area for which CMTS Prefix Stability is desired need to be made part of the same link group.
- Create a new link, flagged as location-independent and made part of this link group. Create or move one or more prefix delegation prefixes under this location-independent link - these are the prefixes from which the stable prefixes will be allocated.
- Remove any prefix delegation prefixes from the existing links that are no longer needed. Note that stateful prefixes (dhcp-type of dhcp) should not be removed.

Note

You can have only one location independent link in a group.
When a client request is received, the server locates the link by checking for the longest matching prefix and using the link of the prefix. However, if this topological link is part of a link group and that group has a location-independent link, the prefixes under the location-independent link will be checked first for possible leases requested by the client. Only if no leases are available from this location-independent link will the topological link be used. This is used for each binding requested by the client.

Any leasing mechanism (lease or client reservations, first best-fit, or extension generated/supplied) may be used with CMTS Prefix Stability as the leases are only known within the single server that services the CMTS group.

**Universal Prefix Stability**

Universal Prefix Stability lets you retain a delegated prefix regardless of where you connect. To use this feature, you must configure reservations for the delegated prefixes. Either client and lease reservations can be used.

Client reservations let you specify the delegated prefixes in a central LDAP repository that the DHCP servers access dynamically (see “Using Client Reservations” section on page 23-10). Lease reservations are managed centrally on the CCM regional server, and are pushed to each local DHCP with the universal link. Because the complete list of reservations is replicated on each server when using lease reservations, you should consider client reservations for larger deployments.

**Note**

You can have only one universal link in a particular VPN address space.

If a link is configured with the universal link type, the prefixes in that link are considered first when attempting to allocate a lease for a client. If no lease is available, the prefixes in the location-independent link type from the link group (if any) is used. Finally, the prefixes in the topological link are used.

**Note**

You can enable both CMTS Prefix Stability and Universal Prefix Stability at the same time, though only one will apply to a subscriber.

**Prefix Allocation Groups**

The prefix allocation groups let you define multiple prefixes that do not result in multiple lease assignments to clients, and control the order in which the prefixes are used. The allocation-group and allocation-group-priority attributes are introduced to specify this behavior.

All prefixes on a link with the same allocation group name belong to that allocation group. A prefix with no allocation group name is in its own allocation group. At most one lease per binding is allocated across all the prefixes in the same allocation group.

The allocation-group-priority setting controls which prefixes are used. Lower numeric values have higher priority, except for 0 (the default), which has the lowest possible priority. Prefixes with the same priority are ordered by the active lease count, where the prefix with the lowest count will have the highest priority.

**Note**

The allocation-group name is only specific to the link. Different links can reuse the same allocation group names.
To control the number of leases a client can obtain from an allocation group prefix you can set the \textit{max-leases-per-binding} attribute for the DHCP policy. For example, if you set \textit{max-leases-per-binding} as 1, the client can obtain only one lease from an allocation group prefix. In addition, if more than one lease is already allocated from the same allocation group prefix then the additional leases are revoked (usually the oldest lease is revoked).

\textbf{DHCPv6 Clients and Leases}

The DHCPv6 server supports clients and leases that are similar to those for DHCPv4. The key differences are:

- The server identifies DHCPv6 clients by their DHCP Unique Identifier (DUID), which is the DHCPv4 concept of hardware addresses and client IDs consolidated into one unique client identifier.

- DHCPv6 clients can have multiple leases. This means that if multiple prefixes are on a single link and are not grouped using the \textit{allocation-group} attribute, the server assigns the client a lease from each prefix that it is allowed to use, not just from one scope, as in DHCPv4. If multiple prefixes on a single link are grouped using the \textit{allocation-group} attribute, then the server assigns the client only one lease per allocation group from the prefix with highest priority within the prefix allocation group (see “Prefix Allocation Groups” section on page 27-6).

- The server first creates a DHCPv6 client when it associates the first lease with it, and deletes the client when it no longer has any leases associated with it. This is identical to DHCPv4 behavior, except that a DHCPv4 client can only have a single lease.

- DHCPv6 leases are dynamically created. The server does not create all leases that it can potentially use at configuration time, because there potentially could be billions of these leases.

Leases can be for:

- **Nontemporary addresses**—Standard IPv6 unicast addresses with likely long (and renewable) lifetimes.

- **Temporary addresses**—Standard IPv6 unicast addresses, but with very limited (and nonrenewable) lifetimes. Temporary addresses solve a privacy issue with IPv6 (see RFC 3041).

- **Delegated prefixes**—Used for prefix delegation (see RFC 3633).

Leases have both a preferred and valid lifetime:

- **Preferred lifetime**—Primarily for the use of the client, the length of time that a valid address is preferred. When the preferred lifetime expires, the address becomes deprecated.

- **Valid lifetime**—Used by both client and server, it is the length of time an address remains in the valid state. The valid lifetime must be greater than or equal to the preferred lifetime. When the valid lifetime expires, the address becomes invalid. A lease is eligible to be deleted once the valid lifetime expires. This is essentially the same as the DHCPv4 lease time.

\textbf{Related Topics}

- DHCPv6 Bindings
- Lease Affinity
- IPv6 Lease States, page 27-8
- Lease Life Cycle, page 27-9
- DHCPv6 Lease Reservations, page 27-10
DHCPv6 Concepts

DHCPv6 Client Reservations, page 27-12
Searching for Leases, page 27-12
Querying Leases for DHCPv6, page 27-12

DHCPv6 Bindings

Bindings are new to DHCPv6 and allow multiple groups of addresses to be allocated to a client. A client binding consists of one of three types:

- Nontemporary (IA_NA)
- Temporary (IA_TA)
- Prefix delegation (IA_PD)

A binding also consists of a unique Identity Association Identifier (IAID). Leases always exist under a binding. Clients, therefore, have one or more bindings, and bindings have one or more leases. The server creates bindings when it first adds the lease, and removes the binding when it has no more leases. The server creates clients when adding the first binding, and removes them when it has no more bindings.

Lease Affinity

For DHCPv4, when a lease expires or the server releases it, the server remembers the client for an address as long as it is not assigned to another client. For DHCPv6, because of the large IPv6 address space and depending on the address generation technique, eons could pass before an address needs reassignment to another client. Therefore, Cisco Prime IP Express provides an affinity-period attribute so that the client can get the same address even if not requesting a renewal before expiration.

The affinity period is desirable in some environments, but not in others where the affinity time would be zero or very small. During the affinity period, the lease is in the AVAILABLE state and still associated with the client that last leased it. If the client requests a lease during this period, the server grants it the same lease (or, if renewals are inhibited, the client explicitly does not get that lease).

IPv6 Lease States

A lease can be in one of the states described in Table 27-1.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available</td>
<td>IP address available to be leased.</td>
</tr>
<tr>
<td>Offered</td>
<td>Offered to the client.</td>
</tr>
<tr>
<td>Leased</td>
<td>Held by a client.</td>
</tr>
<tr>
<td>Expired</td>
<td>Available when the lease grace period expires.</td>
</tr>
<tr>
<td>Unavailable</td>
<td>Not leasable. It was made unavailable because of some conflict.</td>
</tr>
<tr>
<td>Released</td>
<td>The client has released the lease, but the server is configured to apply a grace period to the lease. The lease will not be made available until the grace period expires.</td>
</tr>
<tr>
<td>Other available</td>
<td>Failover-related. Available for allocation by the failover partner but not available for allocation by this server.</td>
</tr>
</tbody>
</table>
Lease Life Cycle

Leases have a life cycle controlled by states. A lease only exists while it is associated with a client and the server deletes it once it is no longer associated with that client. The life cycle and state transitions are:

1. A lease is born and associated with an address when the server:
   a. Creates a reservation for a lease, which puts the lease in the AVAILABLE state and marks it as RESERVED. No timer is associated with this state and the server does not delete the lease as long as it is RESERVED.
   b. Sends an ADVERTISE message to a client, which puts the lease in OFFERED state. The lease transitions to DELETED state after the offer timeout.
   c. Sends a REPLY message to a client (for a REQUEST, RENEW, or REBIND), which puts the lease in LEASED state. The lease transitions to EXPIRED state after the valid lifetime for the lease elapses.

2. An OFFERED lease transitions to:
   a. LEASED state when the server receives a REQUEST message, and then transitions to EXPIRED state after the valid lifetime for the lease elapses.
   b. DELETED state if the offered-time expires.

3. A LEASED lease:
   a. Is renewed when the server receives a REQUEST, RENEW, or REBIND message. The lease transitions to EXPIRED state after the new valid lifetime for the lease elapses (note that the new valid lifetime could be 0).
   b. Transitions to RELEASED state when the server receives a RELEASE message. The lease transitions to AVAILABLE state after the release-grace-period elapses.
   c. Transitions to UNAVAILABLE state when the server receives a DECLINE message. The server deletes the lease after the unavailable timeout period elapses.

4. An EXPIRED lease transitions to either AVAILABLE state after the grace-period. The server deletes the lease after the affinity-period elapses.

5. An AVAILABLE lease:
   d. Transitions to DELETE state and the server deletes it from memory and the lease database after the affinity-period elapses.
   e. Cannot be deleted if it is RESERVED, and it remains AVAILABLE.

6. The server can reoffer a LEASED, EXPIRED, RELEASED, or AVAILABLE lease to a client, but it remains in its current state, although the server extends the timeout to at least the offer-timeout.
The DHCP failover complicates some of the state transitions as these transitions can generally not occur until the partner acknowledges them. The additional life cycle and state transitions (failover related) are as follows:

- Transitioning into the AVAILABLE (or OTHER AVAILABLE) state requires that the partner to acknowledge the transition and hence the PENDING AVAILABLE state is used until the acknowledgement is received from the partner.
- Disassociating a lease from a client also requires an acknowledgement from the partner and hence the PENDING DELETE state is used until the partner has acknowledged the state change.

**DHCPv6 Lease Reservations**

Reservations apply to nontemporary addresses and delegated prefixes only. They are associated with a prefix in the configuration, and must always be for an address (or prefix) under a configured prefix object.

The reservation can be outside the object range of the prefix, provided it is not in object range of another prefix. However, this has implications when you add a new prefix object. If a reservation that is contained in the new range of the prefix exists, the prefix will not be added. This results in an EX_CONFLICT status. For details, see the “Creating Lease Reservations” section on page 23-13.

---

**Note**

The operations for DHCPv4 reservations are similar to DHCPv6 reservations, except that the addresses are v6 addresses, not v4 addresses. Also, the main identity for a DHCPv6 client is a client DUID, and not the mac-address. DHCPv6 reservations include addresses and delegated prefixes.

Any change you make in the v6 reservation list modifies the parent prefix to indicate that a server reload is required. On the regional server, if the DHCP edit mode is synchronous and the parent prefix has been assigned to a local cluster, changes are automatically forwarded to the local cluster. A server reload is required, before these changes take effect.

---

**Caution**

If multiple DHCP servers distribute IP addresses on the same prefix, the reservations must be identical. If not, a client for whom a reservation exists can receive offers of different IP addresses from different servers.

A lease reservation pairs an IP address with a lookup key. A lookup key can be a string value or binary blob.

---

**Note**

If a new prefix delegation reservation is added that has a shorter or longer prefix that conflicts (is contained by or contained in) an existing lease when the server is reloaded, the reservation will prevent the existing leases from being loaded.

**Local Advanced Web UI**

To view the reservations for DHCPv6 prefixes, do the following:

**Step 1**
To view the DHCPv6 lease reservations, from the Design menu, choose Prefixes under the DHCPv6 submenu to open the List/Add DHCPv6 Prefixes page.

**Step 2**
Select the prefix on the Prefixes pane and click the Reservations tab.
To configure the reservations directly for DHCPv6 prefixes, do the following:
In the advanced mode, if a valid parent prefix is not specified, the CCM server automatically sets the appropriate parent prefix.

Step 1 From the Design menu, choose Reservations under the DHCPv6 submenu to open the List/Add DHCPv6 Reservations page.

Step 2 To create a reservation, click the Add Reservations icon in the Reservations pane, enter the IP address you want to reserve for lease, and enter a lookup key in the Lookup Key field.

Step 3 Click the String radio button, if you entered string value or click the Binary radio button, if you entered binary value in the Lookup Key field.

Step 4 Click Add v6 Reservation.

Step 5 On the Reservations pane, choose a filter type from the Filter Type drop-down list. Enter a value in the Filter Value field. Click Set Filter. To set Filter Type as None, click Clear Filter. The lease IP address, Lookup Key and Prefix details are displayed in the List/Add DHCPv6 Reservations page.

**CLI Commands**

The reservation6 command lets you access the global list of DHCPv6 reservations of Cisco Prime IP Express.

A matching prefix must exist for each reservation in the global list, otherwise the edit is rejected as invalid.

Create a new address by using, `reservation6 [vpn-name/]address create lookup-key [blob | string]`

For example:

```
nrcmd> reservation6 white/2001:db8::1 create 00:03:00:01:01:02:03:04:05:06
```

Delete an address by using, `reservation6 [vpn-name/]address delete`

For example:

```
nrcmd> reservation6 white/2001:DB8::1 delete
```

Get an address by using, `reservation6 [vpn-name/]address get value`

For example:

```
nrcmd> reservation6 white/2001:DB8::1 get value
```

Set an attribute by using, `reservation6 [vpn-name/]address set prefix=<name-of-prefix>`

For example:

```
nrcmd> reservation6 white/2001:DB8::1 set prefix=cm_prefix
```

Unset an attribute by using, `reservation6 [vpn-name/]address unset value`

For example:

```
nrcmd> reservation6 white/2001:DB8::1 unset value
```
DHCPv6 Client Reservations

Cisco Prime IP Express supports both Lease and Client Reservations for DHCPv6. For details on Client Reservations, see “Using Client Reservations” section on page 23-10.

Searching for Leases

For details on searching for leases in the configured DHCPv6 network, see the “Searching Server-Wide for Leases” section on page 23-8.

Querying Leases for DHCPv6

For details on the DHCLEASEQUERY implementation for DHCPv6, see the “Leasequery for DHCPv6” section on page 23-31.

DHCPv6 Policy Hierarchy

DHCPv6 uses the existing policy objects, with additional DHCPv6 specific attributes (that are mostly analogous to those in DHCPv4). For DHCPv6, the hierarchy is:

1. Client embedded policy
2. Client named policy
3. Client-class embedded policy
4. Client-class named policy
5. Prefix embedded policy
6. Prefix named policy
7. Link embedded policy
8. Link named policy
9. system_default_policy

For attributes, the default value for the most local policy applies. This hierarchy is the same as for DHCPv4, except for the additional link policies and the fact that the prefix policies replace the scope policies. (For a comparison with the DHCPv4 policy hierarchy, see the “Policy Hierarchy” section on page 22-3.)

The hierarchy applies to most policy attributes, which the server processes in the context of a single prefix. However, the server processes a few attributes (specifically allow-rapid-commit, reconfigure, v6-reply-option, v6-options, and v6-vendor-options) in the context of multiple prefixes. In these cases, the processing at the prefix levels (steps 5 and 6) is a bit different:

- For the reconfigure attribute that controls whether the server requires, allows, or disallows client reconfiguration, the server checks the embedded and named policies of all prefixes on the link that the client is allowed to use (based on selection tags). If any of the prefix policies have the reconfigure attribute set to disallow or require, the server uses that setting. Otherwise, if at least one policy has
it set to allow. Reconfigure is allowed. Otherwise, the server checks the remaining policies in the hierarchy. (See the “Reconfigure Support” section on page 27-35 for details.)

- If the client requests Rapid Commit (see the “Editing DHCPv6 Server Attributes” section on page 27-32), the server checks the embedded and named policies of all prefixes on the link that the client is allowed to use (based on selection tags). If one of these policies has allow-rapid-commit disabled, the server processes the client request as if Rapid Commit were not part of the request. If at least one policy has allow-rapid-commit enabled, the client can use Rapid Commit. If no prefix policy has the attribute configured, processing continues at step 7.

- For the options-related attributes (see the “Setting DHCPv6 Options” section on page 27-34), the server also does special handling at steps 5 and 6. The server checks the embedded and then named policy of each prefix on the link. It then uses the first one with the configured v6-reply-option attribute, or the first one with the configured value for the v6-options or v6-vendor-options.

- The server checks the prefixes in case-insensitive alphabetical order.

- The server ignores any policies related to the location-independent and/or universal link and the prefixes under those. Only topological links (and prefixes under those links) are considered.

---

**Tip**

In configurations with multiple prefixes on a link, avoid setting the Rapid Commit and option properties for the prefix policy, but rather set them on the link policy or other policy instead.

### DHCPv6 Options

DHCPv6 options do not use DHCPv4 options; they are unique and separate. There are currently about 46 DHCPv6 options. Most of these options are the DHCPv6 protocol infrastructure options and are not user-definable. They use a 16-bit option code and 16-bit length (DHCPv4 uses only 8 bits for both of these). Configuring options and the behavior of configured options in policies are similar to those for DHCPv4. See the “Setting DHCPv6 Options” section on page 27-34 for details about client processing as it relates to the policy hierarchy.

### DHCPv6 Configuration

The following sections describe how to configure DHCPv6 in Cisco Prime IP Express:

- Viewing IPv6 Address Space
- Configuring Links
- Configuring Prefixes, page 27-20
- Viewing DHCPv6 Networks, page 27-32
- Editing DHCPv6 Server Attributes, page 27-32
- Configuring DHCPv6 Policies, page 27-32
- Configuring DHCPv6 Client-Classes, page 27-33
- Configuring DHCPv6 Clients, page 27-34
- Setting DHCPv6 Options, page 27-34
- Reconfigure Support, page 27-35
Chapter 27  Managing DHCPv6 Addresses

DHCPv6 Configuration

Viewing IPv6 Address Space

From the Design menu, choose Address Tree under the DHCPv6 submenu, in the local advanced or regional web UI, to view the DHCP v6 Address Tree page. This page is like the DHCP Address Tree page for IPv4 (see the “Viewing Address Space” section on page 9-2). On the View Unified v6 Address Space page you can:

- Set a VPN for the address space from the username drop-down list on the top right of the window.
- Add a prefix by clicking the Add Address Tree icon in the Address Tree pane, enter the prefix name, address and choosing a DHCP type and possible template. Click Add IPv6 Prefix (see the “Creating and Editing Prefixes” section on page 27-25).
- Edit a prefix by selecting its name in the Address Tree pane. This opens the Edit IPv6 Prefix page (see the “Creating and Editing Prefixes” section on page 27-25).
- View the current usage of the prefix space (see the “Viewing Address Utilization for Prefixes” section on page 27-30).

Configuring Links

You can configure DHCPv6 links directly, or you can create link templates for them first. See the following subsections:

- Creating and Editing Link Templates
- Creating and Editing Links, page 27-18

Creating and Editing Link Templates

You can create links from predefined templates. The attributes you can set for a link template are as follows (for the expression syntax, see the “Using Expressions in Link Templates” section on page 27-16):

- **name**—User-assigned name for the link template.
- **description**—Description of the link template itself.
- **policy**—Shared policy used when replying to clients, as applied to the link.
- **owner**—Owner of the link.
- **region**—Region for this link.
- **link-name-expr**—Expression to define the name of the link once the template is applied.
- **link-description-expr**—Expression to define the description on the link once applied.
- **prefix-expr**—Expression to create the list of associated prefixes once the template is applied. For example, you can specify creating prefixes based on defining prefix-expr as @link-prefix-expr.txt to point to the file that contains this expression (and assuming that the cm-prefix, cpe-address-prefix, and cpe-pd-prefix templates exist):

```
(list
 (create-prefix "cm-prefix" (create-prefix-range 32 0x1))
 (create-prefix "cpe-address-prefix" (create-prefix-range 32 0x2))
 (create-prefix "cpe-pd-prefix" (create-prefix-range 16 0x1))
)
```

- **options-expr**—Expression to define the list of embedded policy options to create with the link.
• `free-addr-config`—Trap that captures unexpected free address events on this link
• `type`—Type of the link (topological, location-independent, universal).
• `group-name`—Link group to which the link belongs.

Local Advanced and Regional Web UI

**Step 1**
From the Design menu, choose Link Templates under the DHCPv6 submenu. The List/Add DHCP v6 Link Templates page appears. The page displays the existing templates.

**Step 2**
Click the Add Link Templates icon in the Link Templates pane to open the Add Link Template dialog box.

**Step 3**
Enter a link template name and click Add Link Template.

**Step 4**
Enter an optional description, and optionally choose a preconfigured policy from the drop-down list.

**Step 5**
Add expressions for the `link-name-expr`, `link-description-expr`, `prefix-expr`, or `options-expr` field attributes (see the “Using Expressions in Link Templates” section on page 27-16).

**Step 6**
If the link template is for Prefix Stability, select the link type (type) and specify a link group name (group-name). You can find these attributes in the Prefix Stability block in the Add DHCP v6 Link Template page (see the “Prefix Stability” section on page 27-5 for details on link types and link groups).

**Step 7**
Click Save.

**Step 8**
In the regional web UI, you can pull replica link templates or push templates to local clusters:

- Click Pull Replica Link Template to open the Select DHCPv6 Link Template Data to Pull page. Choose a pull mode for the cluster (ensure, replace, or exact), then click Pull All Link Templates. On the Report Pull DHCPv6 Link Template page, click OK.
- Click Push Link Template for a specific template (or Push All Link Templates) to open the Push DHCPv6 Link Template Data to Local Cluster page. Choose a data synchronization mode (ensure, replace, or exact), move the desired cluster or clusters to the Selected table, then click Push Data to Clusters.

CLI Commands

To create the link template, use `link-template name create`. For example:

```
nrcmd> link-template example-link-template create [attribute=value]
```

You can set and enable the aforementioned expression setting attributes in the usual way, and you can show and list link templates. For example, to set a prefix expression for the link template, use the following file definition and pointer to the file (and assuming that the cm-prefix, cpe-address-prefix, and cpe-pd-prefix templates exist):

```
> type link-prefix-expr.txt
(list (create-prefix "cm-prefix" (create-prefix-range 32 0x1))
 (create-prefix "cpe-address-prefix" (create-prefix-range 32 0x2))
 (create-prefix "cpe-pd-prefix" (create-prefix-range 16 0x1)) )
```

```
nrcmd> link-template example-link-template set prefix-expr=@link-prefix-expr.txt
```

In addition:

- To clone a link template, use `link-template name create clone=name`. 
To apply a template to one or more links, use `link-template name apply-to { all | link[link,...] }`. You can create prefixes by using `link-template name apply-to link [prefix]`, but only with one link specified. The link-template includes an embedded-policy object. The link-template-policy CLI command and the Web UI supports the embedded policy on the link-template page.

### Using Expressions in Link Templates

You can specify expressions in a link template to dynamically create prefix names, IP address ranges, and embedded options when creating a link. Expressions can include context variables and operations. Expressions are not the same as DHCP extensions. Expressions are commonly used to create client identities or look up clients. Extensions (see Chapter 30, “Using Extension Points”) are used to modify request or response packets.

When a template is applied to a link, if the link-template has an embedded policy, it is copied to the link. This embedded policy may or may not have options. As the entire link-template’s embedded policy is used (if it exists), it will wipe out any existing options in the link. If the link-template has no embedded policy, the link’s embedded policy is retained. Next the link-template’s option expression, if any, is evaluated and the options are added to the embedded policy options in the link (if no embedded policy exists, one is created).

Table 27-2 lists the link template predefined variables and Table 27-3 lists the link template operators. Note that these variables and operators are not case-sensitive. Table 27-5 lists the prefix template operators. The link template operators table and prefix template operations table both have same operators, except that only a link template can use Create Prefix Operator and prefix template can not use the operator.

### Table 27-2  Link Template Expression Predefined Variables

<table>
<thead>
<tr>
<th>Predefined Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mask-length</td>
<td>Number of prefix mask bits (with a template-root-prefix defined).</td>
</tr>
<tr>
<td>prefix</td>
<td>Network number and length (with a template-root-prefix defined).</td>
</tr>
<tr>
<td>prefix-addr</td>
<td>Address portion of the prefix (with a template-root-prefix defined).</td>
</tr>
<tr>
<td>prefix-length</td>
<td>Number of prefix address bits (with a template-root-prefix defined).</td>
</tr>
<tr>
<td>template.attribute</td>
<td>Attribute of the link template.</td>
</tr>
<tr>
<td>vpn</td>
<td>VPN of the link.</td>
</tr>
</tbody>
</table>

### Table 27-3  Link Template Expression Operators

<table>
<thead>
<tr>
<th>Expression Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arithmetic Operators</strong> (unsigned integer arguments only)</td>
<td></td>
</tr>
<tr>
<td>(+ arg1 arg2)</td>
<td>Adds the two argument values, such as (+ 2 3).</td>
</tr>
<tr>
<td>(– arg1 arg2)</td>
<td>Subtracts the second argument value from the first one.</td>
</tr>
<tr>
<td>(* arg1 arg2)</td>
<td>Multiplies the values of two arguments.</td>
</tr>
<tr>
<td>(/ arg1 arg2)</td>
<td>Divides the value of the first argument by that of the second one (which cannot be zero).</td>
</tr>
</tbody>
</table>
### Table 27-3  Link Template Expression Operators (continued)

<table>
<thead>
<tr>
<th>Expression Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(% arg1 arg2)</td>
<td>Modulo arithmetic operator to determine the remainder of the result of the</td>
</tr>
<tr>
<td></td>
<td>first argument divided by the second one.</td>
</tr>
<tr>
<td>(concat arg1 ... argn)</td>
<td>Concatenates the arguments into a string.</td>
</tr>
<tr>
<td>(list oper1 ... opern)</td>
<td>Creates an options list or list of prefixes. Required if you need more than</td>
</tr>
<tr>
<td></td>
<td>one option for a link or prefix, or more than one prefix for a link. All</td>
</tr>
<tr>
<td></td>
<td>arguments must be create-v6-option operation. Nesting is not supported. For</td>
</tr>
<tr>
<td></td>
<td>example:</td>
</tr>
<tr>
<td></td>
<td>(list (create-prefix &quot;cm-prefix&quot; (create-prefix-range 32 0x1))</td>
</tr>
<tr>
<td></td>
<td>(create-prefix &quot;cpe-address-prefix&quot; (create-prefix-range 32 0x2))</td>
</tr>
<tr>
<td></td>
<td>(create-prefix &quot;cpe-pd-prefix&quot; (create-prefix-range 16 0x1)) )</td>
</tr>
<tr>
<td>(create-prefix template</td>
<td>Creates a prefix based on a predefined prefix template name and the prefix,</td>
</tr>
<tr>
<td>prefix)</td>
<td>including the link VPN (assuming that a template-root-prefix is defined).</td>
</tr>
<tr>
<td></td>
<td>The prefix argument can be the prefix name, but also the create-prefix-addr</td>
</tr>
<tr>
<td></td>
<td>or create-prefix-range operator value. You can use the list function to</td>
</tr>
<tr>
<td></td>
<td>combine multiple operations. For example:</td>
</tr>
<tr>
<td></td>
<td>(create-prefix &quot;cm-prefix&quot; (create-prefix-range 32 0x1))</td>
</tr>
<tr>
<td>(create-prefix-addr prefix</td>
<td>Creates an IPv6 address string (assuming that a template-root-prefix is</td>
</tr>
<tr>
<td>interface-id)</td>
<td>defined) based on the prefix name and interface ID (an IPv6 address that you</td>
</tr>
<tr>
<td></td>
<td>can specify as a string), which is the lower 64-bit address in the prefix</td>
</tr>
<tr>
<td></td>
<td>(which need not be contained in the parent prefix). Used in the prefix-expr</td>
</tr>
<tr>
<td></td>
<td>and options-expr attributes.</td>
</tr>
<tr>
<td>(create-prefix-range size n)</td>
<td>Creates an address range (child) for the prefix, used in the prefix-expr</td>
</tr>
<tr>
<td></td>
<td>attribute. The prefix value that the function is based on is either the</td>
</tr>
<tr>
<td></td>
<td>template-root-prefix if applying a link template to a link, or the prefix</td>
</tr>
<tr>
<td></td>
<td>address, if applying a prefix template to a prefix.</td>
</tr>
<tr>
<td></td>
<td>Range value—An increase in the prefix length.</td>
</tr>
<tr>
<td></td>
<td>Size—The number of bits by which you can increase the prefix length. Must</td>
</tr>
<tr>
<td></td>
<td>be a value from 1 through 32. Must be less than the parent prefix length.</td>
</tr>
<tr>
<td></td>
<td>n—The nth occurrence of the child prefix. Value can be 0, but is limited to</td>
</tr>
<tr>
<td></td>
<td>less than two to the power of the size. Must be less than or equal to the</td>
</tr>
<tr>
<td></td>
<td>size. The size and n must be greater than zero.</td>
</tr>
<tr>
<td></td>
<td>The n must be less than or equal to the size, and the size must by less than</td>
</tr>
<tr>
<td></td>
<td>the parent prefix length. For example:</td>
</tr>
<tr>
<td></td>
<td>(create-prefix-range 32 0x1)</td>
</tr>
</tbody>
</table>
DHCPv6 Configuration

Chapter 27  Managing DHCPv6 Addresses

Table 27-3  Link Template Expression Operators (continued)

<table>
<thead>
<tr>
<th>Expression Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Option Operator</td>
<td></td>
</tr>
<tr>
<td>(create-option opt val)</td>
<td>Creates a DHCPv6 option, used in the options-expr attribute. The opt can be</td>
</tr>
<tr>
<td></td>
<td>the literal string or integer identifying the option. The val is the string</td>
</tr>
<tr>
<td></td>
<td>representation of the option value, as defined by the option TLV value.</td>
</tr>
<tr>
<td></td>
<td>You can use custom defined and unknown options. For undefined options, the</td>
</tr>
<tr>
<td></td>
<td>option number must be specified and the data is used as is (as blob data).</td>
</tr>
<tr>
<td></td>
<td>If the data is a string, the string is used as is and if the data is a number</td>
</tr>
<tr>
<td></td>
<td>or address, it is used as is.</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>(list (create-option &quot;dns-servers&quot; (create-prefix-addr prefix &quot;::2&quot;))</td>
</tr>
<tr>
<td></td>
<td>(create-option &quot;domain-list&quot; &quot;sales.example.com,example.com&quot;))</td>
</tr>
<tr>
<td>Note</td>
<td>(create-v6-option opt val) is a synonym for (create-option) and can be used</td>
</tr>
<tr>
<td></td>
<td>instead; but we recommend that you use (create-option).</td>
</tr>
<tr>
<td>Create Vendor Option Operation</td>
<td></td>
</tr>
<tr>
<td>(create-vendor-option set-name opt val)</td>
<td>Creates a DHCPv6 vendor option, used in the options-expr attribute. The</td>
</tr>
<tr>
<td></td>
<td>set-name specifies the option definition set for the vendor option. The opt</td>
</tr>
<tr>
<td></td>
<td>can be the literal string or integer identifying the vendor option in the set.</td>
</tr>
<tr>
<td></td>
<td>The val is representation of the option value.</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>(list (create-option &quot;dns-servers&quot; (create-prefix-addr prefix &quot;::2&quot;))</td>
</tr>
<tr>
<td></td>
<td>(create-vendor-option &quot;dhcp6-cablelabs-config&quot; 17</td>
</tr>
<tr>
<td></td>
<td>&quot;{enterprise-id 4491{(tftp-servers 32 3800:0:0:180::6)</td>
</tr>
<tr>
<td></td>
<td>(config-file-name 33 modem_ipv6.bin)(syslog-servers 34</td>
</tr>
<tr>
<td></td>
<td>3800:0:0:180::8)(rfc868-servers 37 3800:0:0:180::6)(time-offset 38 -5h)</td>
</tr>
<tr>
<td></td>
<td>(cablelabs-client-configuration 2170 (primary-dhcp-server 1</td>
</tr>
<tr>
<td></td>
<td>10.38.1.5)(secondary-dhcp-server 2 10.38.1.6))))&quot;))</td>
</tr>
<tr>
<td>Note</td>
<td>(create-v6-vendor-option opt val) is a synonym for (create-vendor-option)</td>
</tr>
<tr>
<td></td>
<td>and can be used instead; but we recommend that you use (create-vendor-option).</td>
</tr>
</tbody>
</table>

Creating and Editing Links

You can create links directly. The attributes you can set for the link are:

- **name**—User-assigned name for the link.
- **vpn-id**—VPN that contains the link.
- **description**—Descriptive text for the link.
- **policy**—Shared policy used when replying to clients.
- **owner**—Owner of the link.
Chapter 27      Managing DHCPv6 Addresses

DHCPv6 Configuration

- **region**—Region for this link.
- **free-address-config**—Identifies which trap captures unexpected free address events on this prefix. If not configured, the server looks at its $v6$-$default$-$free$-$address$-$config$ attribute.
- **type**—Type of link (topological, location-independent, universal).
- **group-name**—Link group to which the link belongs.

Local Advanced and Regional Web UI

Step 1  From the Design menu, choose Links under the DHCPv6 submenu. The List/Add DHCP v6 Links page displays the existing links.

Step 2  To add a link, click the Add Link icon in the Links pane.

Step 3  Enter the desired name for the link.

Step 4  If the link is for Prefix Stability, select the link type (type) and specify a link group name (group-name). The Link Type is $topological$, by default.

You can also find these attributes in the Prefix Stability area in the Edit DHCP v6 Link Template page (see the “Prefix Stability” section on page 27-5 for details on link types and link groups).

Note  You can have only one location independent link in a link group and one universal link in a VPN address space. Also, you cannot assign a link of type universal to a link group.

Step 5  Click Add Link.

Step 6  In the Edit Link page of the new link, choose the predefined prefixes for the link by moving them from the Available field to the Selected field.

Step 7  To add new prefixes for the link, enter each prefix name and its address at the bottom of the page, indicate a range, choose the DHCP type and template (if needed), then click Apply Prefix for each one.

Step 8  Click Save.

Step 9  In the regional web UI, you can push links to local clusters and reclaim links on the Edit DHCP v6 Link page and pull replica IPv6 address space on the List/Add DHCP v6 Links page:

- To push the link, click Push (at the top of the page) to open the Push DHCP v6 Link page. Choose the cluster or link template to which you want to push the link, then click Push Link. When the link is pushed, all prefixes on the link, and all reservations on the prefixes are also be pushed.

- To reclaim the prefix, click Reclaim (at the top of the page) to open the Reclaim DHCP v6 Link page. Choose the cluster or link template to which you want to reclaim the link, then click Reclaim Link. When the link is reclaimed, the reservations, prefixes, and link is deleted from the local cluster, provided there are no active leases. If active leases are found, prefixes are deactivated instead. The force option lets you remove the link and its prefixes when there are active leases.

Note  Only universal links can be pushed to more than one cluster.

- To pull replica IPv6 address space, click the Pull Replica icon (at the top of the links pane on the left) to open Select Pull Replica IPv6 Address Space. Choose the data synchronization mode (update, complete, or exact) and click Report. The local changes will not take effect until the next server reload.
Chapter 27      Managing DHCPv6 Addresses

CLI Commands

Use **link name create**. (The **link** command is a synonym for the **dhcp-link** command from previous releases.) For example:

```
nrcmd> link example-link create [attribute=value]
```

To apply a link template during link creation, use **link name create template**=**name** [**template-root-prefix**=**address**], with the **template-root-prefix** specified if the template could create more than one prefix. To apply a template to an existing link definition, use **link name applyTemplate template-name** [**template-root-prefix**].

You can set and enable the aforementioned attributes in the usual way, and you can show and list links. To list prefixes or prefix names associated with a link, use **link name listPrefixes** or **link name listPrefixNames**.

Configuring Prefixes

You can configure DHCPv6 prefixes directly, or you can create prefix templates for them first. See the following subsections:

- Creating and Editing Prefix Templates
- Creating and Editing Prefixes, page 27-25
- Viewing Address Utilization for Prefixes, page 27-30

Creating and Editing Prefix Templates

You can create prefixes from predefined templates. The attributes you can set for a prefix template are the following (for the expression syntax, see the “Using Expressions in Prefix Templates” section on page 27-23):

- **name**—User-assigned name for the prefix template.
- **description**—Descriptive text for the prefix template.
- **dhcp-type**—Defines how DHCP manages address assignment for a prefix:
  - **dhcp** (preset value)—Uses the prefix for stateful address assignment.
  - **stateless**—Uses the prefix for stateless option configuration.
  - **prefix-delegation**—Uses the prefix for prefix delegation.
  - **infrastructure**—Uses the prefix to map a client address to a link, when the prefix does not have an address pool.
- **policy**—Shared policy to use when replying to clients.
- **owner**—Owner of this prefix, referenced by name.
- **region**—Region for this prefix, referenced by name.
- **prefix-name-expr**—Expression that evaluates to a string value to use for the name of the prefix created. For example, you can have the prefix name prepended by **CM** if you define **prefix-name-expr** as (**concat** "**CM**-" **prefix**). In the CLI, you would include the expression in a file and point to that file:

  ```
  > type prefix-name.txt
  (concat "CM-" prefix)
  ```
Chapter 27      Managing DHCPv6 Addresses

DHCPv6 Configuration

prefix-template ex-template create prefix-name-expr=@prefix-name.txt

- prefix-description-expr—Expression that evaluates to a string value to apply to the description on the prefix created when using the template.

- range-expr—Expression that evaluates to an IPv6 prefix value to create an address range. In the CLI, you must use a file reference. For example:

  nrcmd> prefix-template ex-template set range-expr=@subprefix-expr.txt

- options-expr—Expression that evaluates to embedded policy options to create. (Use the list function to create multiple options.)

- allocation-algorithms—One or more algorithms the server uses to select a new address or prefix to lease to a client. The available algorithms are:
  - client-request (preset to off)—Controls whether the server uses a client-requested lease.
  - reservation (preset to on)—Controls whether the server uses an available reservation for the client.
  - extension (preset to on)—Controls whether the server calls extensions attached at the generate-lease extension point to generate an address or prefix for the client. When you use generate-lease extension point with DHCPv6 failover, the server uses the address or delegated prefix that the extension returns and does not perform a hash on this address or prefix as it does with the randomly generated addresses. If the extension is using some algorithmic method to generate the address or delegated prefix then the extension must be failover aware (extension will be able to determine if failover configuration is enabled and the role of the failover server).
    For details on extensions, see Chapter 30, “Using Extension Points.”
  - interface-identifier (preset to off)—Controls whether the server uses the interface-identifier from the client (link-local) address to generate an address; ignored for temporary addresses and prefix delegation.
  - random (preset to on)—Controls whether the server generates an address using an RFC 3041 algorithm; ignored for prefix delegation.
  - best-fit (preset to on)—Controls whether the server delegates the first, best-fit available prefix; ignored for addresses.

    When the server needs an address to assign to a client, it processes the flags in the following order until it finds a usable address: client-request, reservation, extension, interface-identifier, and random. When the server needs to delegate a prefix to a client, it processes the flags in the following order until it finds a usable prefix: client-request, reservation, extension, and best-fit.

- restrict-to-reservations—Controls whether the prefix is restricted to client (or lease) reservations.

- max-leases—Maximum number of nonreserved leases allowed on the prefix. When a new lease needs to be created, the server does so only if the limit is not exceeded. When the limit is exceeded, the server cannot create or offer new leases to clients. If you also enable SNMP traps, the max-leases value also calculates the percentage of used and available addresses.

Note Be sure to set the max-leases value to the expected maximum so that the SNMP address traps can return meaningful results.
• **ignore-declines**—Controls whether the server responds to a DHCPv6 DECLINE message that refers to an IPv6 address or a delegated prefix from this prefix. If enabled, the server ignores all declines for leases in this prefix. If disabled (the preset value) or unset, the server sets to UNAVAILABLE every address or delegated prefix requested in a DECLINE message if it is leased to the client.

• **deactivated**—Controls whether a prefix extends leases to clients. A deactivated prefix does not extend leases to any clients and treats all addresses in its ranges as if they were individually deactivated. The preset value is false (activated).

• **expiration-time**—Time and date at which a prefix expires. After this date and time, the server neither grants new leases nor renews existing leases from this prefix. Enter a value in the format "[weekday] month day hh:mm:ss year"; for example, "Dec 31 23:59 2006". The reason for an expiration time is to support network renumbering events. The general idea is a new prefix is added and the old is taken away sometime at or after the expiration-time. Clients will be given leases on both prefixes. The server will automatically stop giving new clients leases once the configured valid lifetime before the expiration-time is reached. At this time, new clients will not get a lease on the prefix. Existing clients will continue to be able to use an existing lease, but will get shorter and shorter lifetimes (preferred and valid). The delta between the preferred and valid is always maintained. Thus if the preferred is 1 day and the valid 2 days, new clients will stop getting leases 2 days before the expiration-time, existing clients will continue to be able to renew leases with preferred lifetimes lesser than 1 day and valid lifetimes greater than 2 days. 1 day before the expiration-time, clients will get a 0 preferred lifetime.

• **free-addr-config**—Trap that captures unexpected free address events on the prefix.

• **reverse-zone-prefix-length**—Prefix length of the reverse zone for ip6.arpa updates. (See the “Determining Reverse Zones for DNS Updates” section on page 29-4 for details.)

• **max-pd-balancing-length**—Controls the maximum prefix-delegation prefix length that the failover pool balancing will consider in balancing a prefix-delegation prefix. The default value is 64 and it should never be longer than the longest prefix length allowed for the prefix delegation.

• **selection-tags**—List of selection tags associated with the prefix.

• **allocation-group**—Allocation group to which the prefix belongs.

• **allocation-group-priority**—Priority of the prefix over other prefixes in the same allocation group. The default value is zero.

**Local Advanced and Regional Web UI**

**Step 1** From the Design menu, choose **Prefix Templates** under the DHCPv6 submenu. The List/Add DHCPv6 Prefix Templates page shows the existing templates.

**Step 2** Click the **Add Prefix Templates** icon in the **Prefix Templates** pane to open the Add Prefix Template dialog box.

**Step 3** Enter the prefix template name and click **Add Prefix Template**.

**Step 4** To edit a prefix template, select its name on the Prefixes pane. Set the attributes and add expressions for the templates that require expressions (see the “Using Expressions in Prefix Templates” section on page 27-23).

**Step 5** On the Edit DHCPv6 Prefix Template page, edit the template attributes, such as adding a selection tag, assigning a group and setting priorities, then click **Save**.

**Step 6** In the regional web UI, you can pull replica prefix templates or push templates to local clusters:
• Click **Pull Replica Prefix Template** to open the Select DHCPv6 Prefix Template Data to Pull page. Choose a pull mode for the cluster (ensure, replace, or exact), then click **Pull All Prefix Templates**. On the Report Pull DHCPv6 Prefix Template page, click **OK**.

• Click **Push Prefix Template** for a specific template (or **Push All Prefix Templates**) to open the Push DHCPv6 Prefix Template Data to Local Cluster page. Choose a data synchronization mode (ensure, replace, or exact), move the desired cluster or clusters to the Selected table, then click **Push Data to Clusters**.

### CLI Commands

To create the prefix template, use `prefix-template name create`. For example:

```
nrcmd> prefix-template example-prefix-template create [attribute=value]
```

You can set and enable the aforementioned attributes in the usual way, and you can show and list prefix templates. In addition:

• To clone a prefix template, use `prefix-template name create clone=name`.

• To apply a template to one or more prefixes, use `prefix-template name apply-to {all | prefix[prefix,...]}`.

• The prefix-template includes an embedded-policy object. The prefix-template-policy CLI command and the Web UI supports the embedded policy on the prefix-template page.

### Using Expressions in Prefix Templates

You can specify expressions in a prefix template to dynamically create prefix names, IP address ranges, and embedded options when creating a prefix. Expressions can include context variables and operations.

---

**Note**

Expressions are not the same as DHCP extensions. Expressions are commonly used to create client identities or look up clients. Extensions (see Chapter 30, “Using Extension Points”) are used to modify request or response packets.

When a template is applied to a prefix, if the prefix-template has an embedded policy, it is copied to the prefix. This embedded policy may or may not have options. As the entire prefix-template’s embedded policy is used (if it exists), it will wipe out any existing options in the prefix. If the prefix-template has no embedded policy, the prefix’s embedded policy is retained. Next the prefix-template’s option expression, if any, is evaluated and the options are added to the embedded policy options in the prefix (if no embedded policy exists, one is created).

**Table 27-4** lists the prefix template predefined variables and **Table 27-5** on page 27-24 lists the operators. Note that these variables and operators are not case-sensitive.

### Table 27-4  Prefix Template Expression Predefined Variables

<table>
<thead>
<tr>
<th>Predefined Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix</td>
<td>Network number and length, based on the template root prefix if applying a link template to a link, or the prefix address if applying a prefix template to a prefix.</td>
</tr>
<tr>
<td>vpn</td>
<td>VPN of the prefix.</td>
</tr>
<tr>
<td>prefix-addr</td>
<td>Address portion of the prefix.</td>
</tr>
</tbody>
</table>
Table 27-4  Prefix Template Expression Predefined Variables (continued)

<table>
<thead>
<tr>
<th>Predefined Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix-length</td>
<td>Number of prefix address bits.</td>
</tr>
<tr>
<td>mask-length</td>
<td>Number of prefix mask bits.</td>
</tr>
<tr>
<td>template.attribute</td>
<td>Attribute of the prefix template.</td>
</tr>
</tbody>
</table>

Table 27-5  Prefix Template Expression Operators

<table>
<thead>
<tr>
<th>Expression Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arithmetic Operators</strong></td>
<td>(unsigned integer arguments only)</td>
</tr>
<tr>
<td>(+ arg1 arg2)</td>
<td>Adds the two argument values, such as (+ 2 3).</td>
</tr>
<tr>
<td>(– arg1 arg2)</td>
<td>Subtracts the second argument value from the first one, such as with ping-timeout defined as 100, (– template.ping-timeout 10) yields 90.</td>
</tr>
<tr>
<td>(* arg1 arg2)</td>
<td>Multiplies the values of two arguments.</td>
</tr>
<tr>
<td>(/ arg1 arg2)</td>
<td>Divides the value of the first argument by that of the second one (which cannot be zero).</td>
</tr>
<tr>
<td>(‰ arg1 arg2)</td>
<td>Modulo arithmetic operator to determine the remainder of the result of the first argument divided by the second one.</td>
</tr>
<tr>
<td><strong>Concatenation Operator</strong></td>
<td>Concatenates the arguments into a string.</td>
</tr>
<tr>
<td>(concat arg1 ... argn)</td>
<td>Concatenates the arguments into a string.</td>
</tr>
<tr>
<td><strong>List Operator</strong></td>
<td>Creates an options list or list of prefixes. Required if needing more than one option for a prefix. All arguments must be create-v6-option or create-prefix-range operations. Nesting is not supported.</td>
</tr>
<tr>
<td>(list oper1 ... opern)</td>
<td>Creates an options list or list of prefixes. Required if needing more than one option for a prefix. All arguments must be create-v6-option or create-prefix-range operations. Nesting is not supported.</td>
</tr>
<tr>
<td><strong>Create IP Operator</strong></td>
<td>Creates an IPv6 address string based on the prefix name and interface ID (an IPv6 address that you can specify as a string), which is the lower 64-bit address in the prefix (which need not be contained in the parent prefix). Used in the range-expr and options-expr attributes.</td>
</tr>
<tr>
<td>(create-prefix-addr prefix-name interface-id)</td>
<td>Creates an IPv6 address string based on the prefix name and interface ID (an IPv6 address that you can specify as a string), which is the lower 64-bit address in the prefix (which need not be contained in the parent prefix). Used in the range-expr and options-expr attributes.</td>
</tr>
<tr>
<td><strong>Create Range Operator</strong></td>
<td>Creates an address range (child) for the prefix, used in the range-expr attribute. The prefix value that the function is based on is either the template-root-prefix if applying a link template to a link, or the prefix address if applying a prefix template to a prefix.</td>
</tr>
<tr>
<td>(create-prefix-range size n)</td>
<td>Creates an address range (child) for the prefix, used in the range-expr attribute. The prefix value that the function is based on is either the template-root-prefix if applying a link template to a link, or the prefix address if applying a prefix template to a prefix.</td>
</tr>
<tr>
<td>Range value—An increase in the prefix length.</td>
<td>Range value—An increase in the prefix length.</td>
</tr>
<tr>
<td>Size—The number of bits by which you can increase the prefix length. Must be a value from 1 through 32. Must be less than the parent prefix length.</td>
<td>Size—The number of bits by which you can increase the prefix length. Must be a value from 1 through 32. Must be less than the parent prefix length.</td>
</tr>
<tr>
<td>n—The nth occurrence of the child prefix. Value can be 0, but is limited to less than two to the power of the size. Must be less than or equal to the size.</td>
<td>n—The nth occurrence of the child prefix. Value can be 0, but is limited to less than two to the power of the size. Must be less than or equal to the size.</td>
</tr>
<tr>
<td>The size and n must be greater than zero. The n must be less than or equal to the size, and the size must by less than the parent prefix length. For example:</td>
<td>The size and n must be greater than zero. The n must be less than or equal to the size, and the size must by less than the parent prefix length. For example:</td>
</tr>
<tr>
<td>(create-prefix-range 32 0x1)</td>
<td>(create-prefix-range 32 0x1)</td>
</tr>
</tbody>
</table>
Chapter 27      Managing DHCPv6 Addresses

Table 27-5   Prefix Template Expression Operators (continued)

<table>
<thead>
<tr>
<th>Expression Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Option Operation</td>
<td></td>
</tr>
<tr>
<td>(create-option opt val)</td>
<td>Creates a DHCPv6 option, used in the options-expr attribute. The opt can be the literal string or integer identifying the option. The val is the string representation of the option value, as defined by the option TLV value. You can use custom defined and unknown options. For undefined options, the option number must be specified and the data is used as is (as blob data). If the data is a string, the string is used as is and if the data is a number or address, it is used as is. For example:</td>
</tr>
<tr>
<td>(list (create-option &quot;dns-servers&quot; (create-prefix-addr prefix &quot;:2&quot;))) (create-option &quot;domain-list&quot; &quot;sales.example.com,example.com&quot;)</td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>(create-v6-option opt val) is a synonym for (create-option) and can be used instead.</td>
</tr>
<tr>
<td>Create Vendor Option Operator</td>
<td></td>
</tr>
<tr>
<td>(create-vendor-option set-name opt val)</td>
<td>Creates a DHCPv6 vendor option, used in the options-expr attribute. The set-name specifies the option definition set for the vendor option. The opt can be the literal string or integer identifying the vendor option in the set. The val is representation of the option value. For example:</td>
</tr>
</tbody>
</table>
| (list (create-option "dns-servers" (create-prefix-addr prefix ":2"))) (create-vendor-option "dhcp6-cablelabs-config" 17 
*{enterprise-id 4491{(tftp-servers 32 3800:0:0:180::6) (config-file-name 33 modem_ipv6.bin) (syslog-servers 34 3800:0:0:180::8) (rfc868-servers 37 3800:0:0:180::6) (time-offset 38 -5h) (cablelabs-client-configuration 2170 (primary-dhcp-server 1 10.38.1.5) (secondary-dhcp-server 2 10.38.1.6))})*) |
| Note | (create-v6-vendor-option opt val) is a synonym for (create-vendor-option) and can be used instead. |

We recommend that you use create-option and create-vendor-option for v4 and v6.

Creating and Editing Prefixes

You can create prefixes directly (and optionally apply an existing template to it; see the “Creating and Editing Prefix Templates” section on page 27-20). These are the prefix attributes that you can set:

- **name**—Assigns a name to this prefix.
- **vpn-id**—VPN that contains the prefix.
• **description**—Describes the prefix.

• **dhcp-type**—Defines how DHCP manages address assignment for a prefix:
  - dhcp (preset value)—Uses the prefix for stateful address assignment.
  - stateless—Uses the prefix for stateless option configuration.
  - prefix-delegation—Uses the prefix for prefix delegation.
  - infrastructure—Uses the prefix to map a client address to a link, when the prefix does not have an address pool.
  - parent—Do not have DHCP use the prefix. But, use it as a container object to group child prefixes. Parent prefixes appear only in the IPv6 address space listing in the web UI, not in the prefixes listing.

• **address**—Prefix (subnet) to which an interface belongs to, using the high-order bits of an IPv6 address.

• **owner**—Owner of the prefix.

• **region**—Region for the prefix.

• **reverse-zone-prefix-length**—Prefix length of the reverse zone for ip6.arpa updates. (See the “Determining Reverse Zones for DNS Updates” section on page 29-4 for details.)

• **range**—Subrange the server can use to configure prefixes for address assignment. The prefix used depends on the value set for the dhcp-type attribute. If unset, the prefix address applies. This value can specify a longer prefix than the prefix address to limit the range of addresses or prefixes available for assignment. (See the “Links and Prefixes” section on page 27-2 for details.)

• **link**—Link associated with the prefix (subnet), used to group prefixes that are on a single link.

• **policy**—Shared policy to use when replying to clients.

• **selection-tags**—List of selection tags associated with the prefix.

• **allocation-algorithms**—One or more algorithms the server uses to select a new address or prefix to lease to a client. The available algorithms are:
  - client-request (preset to off)—Controls whether the server uses a client requested lease.
  - reservation (preset to on)—Controls whether the server uses an available reservation for the client.
  - extension (preset to on)—Controls whether the server calls extensions attached at the generate-lease extension point to generate an address or prefix for the client.
    When you use generate-lease extension point with DHCPv6 failover, the server uses the address or delegated prefix that the extension returns and does not perform a hash on this address or prefix as it does with the randomly generated addresses. If the extension is using some algorithmic method to generate the address or delegated prefix then the extension must be failover aware (extension will be able to determine if failover configuration is enabled and the role of the failover server). For details on extensions, see Chapter 30, “Using Extension Points.”
  - interface-identifier (preset to off)—Controls whether the server uses the interface-identifier from the client (link-local) address to generate an address; ignored for temporary addresses and prefix delegation.
  - random (preset to on)—Controls whether the server generates an address using an RFC 3041 algorithm; ignored for prefix delegation.
  - best-fit (preset to on)—Controls whether the server delegates the first, best-fit available prefix; ignored for addresses.
When the server needs an address to assign to a client, it processes the flags in the following order until it finds a usable address: client-request, reservation, extension, interface-identifier, and random. When the server needs to delegate a prefix to a client, it processes the flags in the following order until it finds a usable prefix: client-request, reservation, extension, and best-fit.

- **restrict-to-reservations**—Controls whether the prefix is restricted to client (or lease) reservations.
- **max-leases**—Maximum number of nonreserved leases allowed on the prefix. When a new lease needs to be created, the server does so only if the limit is not exceeded. When the limit is exceeded, the server cannot create or offer new leases to clients. If you also enable SNMP traps, the `max-leases` value also calculates the percentage of used and available addresses.

**Tip**

Set the `max-leases` value to the expected maximum so that the SNMP address traps can return meaningful results.

- **ignore-declines**—Controls whether the server responds to a DHCPv6 DECLINE message that refers to an IPv6 address or a delegated prefix from this prefix. If enabled, the server ignores all declines for leases in this prefix. If disabled (the preset value) or unset, the server sets to UNAVAILABLE every address or delegated prefix requested in a DECLINE message if it is leased to the client.
- **expiration-time**—Time and date at which a prefix expires. After this date and time, the server neither grants new leases nor renews existing leases from this prefix. Enter a value in the format "[weekday] month day hh:mm:ss year"; for example, "Dec 31 23:59 2006". See the explanation for `expiration-time` attribute under “Creating and Editing Prefix Templates” section on page 27-20.
- **free-address-config**—Identifies which trap captures unexpected free address events on this prefix. If not configured, the server looks for the `free-address-config` attribute value for the parent link. If that attribute is not configured, the server looks at its `v6-default-free-address-config` attribute.
- **deactivated**—Controls whether a prefix extends leases to clients. A deactivated prefix does not extend leases to any clients and treats all addresses in its ranges as if they were individually deactivated. The preset value is false (activated).
- **max-pd-balancing-length**—Controls the maximum prefix-delegation prefix length that the failover pool balancing will consider in balancing a prefix-delegation prefix. The default value is 64 and it should never be longer than the longest prefix length allowed for the prefix delegation.
- **allocation-group**—Allocation group to which this prefix belongs.
- **allocation-group-priority**—Priority of this prefix over other prefixes in the same allocation group. The default value is zero.
- **embedded-policy**—Policy embedded in the prefix.

### Local Advanced and Regional Web UI

**Step 1**

From the Design menu, choose **Prefixes** under the **DHCPv6** submenu. The List/Add DHCP v6 Prefixes page shows the existing prefixes.

**Step 2**

To create the prefix:

a. If creating it in other than the current VPN, choose a VPN from the `username` drop-down list on the top right of the window at the top right of the window.

b. Click the **Add Prefixes** icon in the Prefixes pane, enter a prefix name and address, and choose a prefix length from the drop-down list.

c. If you want a range of addresses for the prefix, enter the subnet address and choose a prefix length.
d. Choose a DHCP type (see the attribute descriptions at the top of this section). The default is DHCP.

e. If you want to apply a preconfigured prefix template, choose it from the drop-down list. (Note that
the attribute values of an applied template overwrite the ones set for the prefix.)

f. Click Add IPv6 Prefix, which should add the prefix to the list.

g. Reload the DHCP server. When you return to the List/Add DHCPv6 Prefixes page, a message
indicates how many prefixes are synchronized.

Step 3  
To create a reverse zone from the prefix, click the Reverse Zone tab. On this tab, you can select a zone
template, and click Report, then Run.

Step 4  
Once you create a prefix, you can view and manage the leases for the prefix by clicking the Leases tab.
On the Leases tab, you can view the leases for the client lookup key and manage each lease separately
by clicking its name.

Step 5  
You can view and manage the reservations for the prefix by clicking the Reservations tab. Add each
reservation IP address and lookup key and whether the lookup key is a string or binary, then click
Add Reservation.

Step 6  
To edit a prefix, click its name on the Prefixes pane. On the Edit Prefix page, edit the prefix attributes,
assign prefix to a group and set priorities, or create a new or edit an existing embedded policy.

To assign the prefix to a group and set priorities:

a. Enter the name of the group in the allocation-group attribute field.

b. Enter the priority value in the allocation-group-priority attribute field. If you do not enter any value
here, it will be allotted the default value (0) and this prefix will have the lowest priority in the group.
You can find these attributes under Allocation Group in Advanced mode (see “Prefix Allocation
Groups” section on page 27-6).

To manage an embedded policy:

a. Click Create New Embedded Policy or Edit Existing Embedded Policy to open the Edit DHCP
Embedded Policy for Prefix page.

b. Modify the embedded policy properties (see the “DHCPv6 Policy Hierarchy” section on
page 27-12).

c. Click Modify Embedded Policy. The next time the Edit DHCPv6 Prefix page appears, you can edit
the embedded policy for the prefix.

d. Click Save.

Step 7  
In the regional web UI, you can push prefixes to local clusters and reclaim prefixes on the List/Add
DHCPv6 Prefixes page:

- To push the prefix, click Push to open the DHCPv6 Push Prefix page. Choose the cluster or prefix
template to which you want to push the prefix, then click Push Prefix. When the prefix is pushed,
the reservations on the prefix is pushed with the prefix. Also, if the prefix is on a link, the parent
prefix is pushed if it is not already present on the local cluster.

- To reclaim the prefix, click Reclaim to open the DHCPv6 Reclaim Prefix page. Choose the cluster
or prefix template to which you want to reclaim the prefix, then click Reclaim Prefix. When the
prefix is reclaimed, the reservations are deleted with the prefix, if there are no active leases, or if the
force option is specified. Otherwise the prefix is deactivated.
If the prefix is on a universal link, it can be pushed to more than one cluster and that local changes will not take effect until the next server reload.

**CLI Commands**

Use `prefix name create ipv6address/length`. (The `prefix` command is a synonym for the `dhcp-prefix` command from previous releases.) Reload the DHCP server. For example:

```
nrcmd> prefix example-prefix create 2001:0db8::/32 [attribute=value]
nrcmd> dhcp reload
```

To apply a prefix template during prefix creation, use `prefix name create ipv6address/length template=name`. To apply a template to an existing prefix definition, use `prefix name applyTemplate template-name`. For example:

```
nrcmd> prefix example-prefix create 2001:0db8::/64 template=preftemp-1
nrcmd> prefix example-prefix applyTemplate template=preftemp-1
nrcmd> dhcp reload
```

You can set and enable the aforementioned attributes in the usual way. Add reservations by using `prefix name addReservation ipv6address/length lookup-key [–blob | –string]`. List leases by using `prefix name listLeases`.

**Tip**

See the “Reconfigure Support” section on page 27-35 for additional syntax.

You can get an exact count of the total prefixes and links for the DHCP server by using `dhcp getPrefixCount [vpn name | all]`. You can specify a VPN or all VPNs. Omitting the `vpn name` returns a count for the current VPN.

**Viewing IPv6 Leases**

You can view the leases for prefixes.

**Local Basic or Advanced Web UI**

To view leases, choose Prefix from the Design menu to open the List/Add DHCP v6 Prefixes page, then click the Leases tab for the prefix. This opens the page, where you can click each lease to manage it.

See the “IPv6 Lease States” section on page 27-8 for a description of the values in the State column. For guidelines as to the lease expiration time, see the “Guidelines for Lease Times” section on page 23-3.

To manage the DHCP leases, click the lease IP address.

**CLI Commands**

Use `lease6 ipaddr show` to show the properties of a particular lease based on its IP address. To manage DHCPv6 leases use the following commands:

```
nrcmd> lease6 (vpn-id/ | vpn-name/)ip6address[/prefix-length] activate
nrcmd> lease6 (vpn-id/ | vpn-name/)ip6address[/prefix-length] deactivate
nrcmd> lease6 (vpn-id/ | vpn-name/)ip6address[/prefix-length] force-available
nrcmd> lease6 (vpn-id/ | vpn-name/)ip6address[/prefix-length] get attribute
```
DHCPv6 Configuration

Chapter 27  Managing DHCPv6 Addresses

DHCPv6 Configuration

nrcmd> lease6 {vpn-id/ | vpn-name/}ip6address[/prefix-length] show
nrcmd> lease6 list

Note  The lease6 list command lists available and other-available prefix delegation leases that are not bound to a client.

Viewing Address Utilization for Prefixes

You can view the current address utilization for prefixes.

Local Advanced and Regional Web UI

The function is available on the DHCP v6 Address Tree page (see the “Viewing Address Space” section on page 9-2).

Tip  You can use the DHCP v6 Address Tree page to push and reclaim prefixes. Click the Push or Reclaim icon for the desired prefix. (See in the “Creating and Editing Prefixes” section on page 27-25 for details.)

When you click the Current Usage tab, the Utilization Detail appears.

Note  To ensure the proper subnet-to-server mapping on this page, you must update the regional address space view so that it is consistent with the relevant local cluster. Do this by pulling the replica address space, or reclaiming the subnet to push to the DHCP server. Also ensure that the particular DHCP server is running.

The other columns under the Current Usage tab identify:

- **Range**—Address range of the prefix.
- **Type**—Whether the address space is a prefix or link.
- **Active Dynamic**—Addresses that are part of a dynamic range managed by DHCP and that are currently leased, but not reserved.
- **Allocation Group**—Allocation group to which the prefix belongs.

The items under Utilization Detail are expandable so that you can view the prefix or parent prefix data. The Utilization Detail page is a read-only page that shows detailed address utilization attributes for the prefix or the parent prefix (identified as Totals). The address utilization attributes are described in Table 27-6.
## Table 27-6  Address Utilization Attributes

<table>
<thead>
<tr>
<th>Utilization Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Addresses</strong></td>
<td></td>
</tr>
<tr>
<td>active-dynamic</td>
<td>Total number of dynamic leases in active use (leased, offered, released, expired, or revoked. The Active Dynamic category shows the states of these leases.</td>
</tr>
<tr>
<td>total-reserved</td>
<td>Total number of reserved leases.</td>
</tr>
<tr>
<td><strong>Active Dynamic</strong></td>
<td></td>
</tr>
<tr>
<td>offered</td>
<td>Number of dynamic (unreserved) leases that are currently offered to clients, but not yet acknowledged as being leased.</td>
</tr>
<tr>
<td>leased</td>
<td>Number of dynamic leases that are currently acknowledged as leased to clients.</td>
</tr>
<tr>
<td>expired</td>
<td>Number of dynamic leases that are past the lease expiration period, but will not be available for other clients (except after the policy grace-period expires).</td>
</tr>
<tr>
<td>revoked</td>
<td>Number of dynamic leases that the client can no longer use, but that some other client could be using.</td>
</tr>
<tr>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>reserved-active</td>
<td>Number of reserved leases that clients are actively using.</td>
</tr>
<tr>
<td>reserved-inactive</td>
<td>Number of reserved leases that clients are not actively using.</td>
</tr>
<tr>
<td><strong>Unavailable</strong></td>
<td></td>
</tr>
<tr>
<td>unavail</td>
<td>Number of unreserved dynamic leases that a client declines or the server marks with an address conflict (usually indicating configurations that need correcting).</td>
</tr>
<tr>
<td>reserved-unavail</td>
<td>Number of reserved leases that a client declines or the server marks with an address conflict (usually indicating configurations that need correcting).</td>
</tr>
<tr>
<td><strong>Deactivated</strong></td>
<td></td>
</tr>
<tr>
<td>deactivated</td>
<td>Number of dynamic and reserved leases that clients are actively leasing (that are not offered, expired, or released), but that an administrator deactivated.</td>
</tr>
<tr>
<td>leased-deactivated</td>
<td>Number of dynamic leases that an administrator deactivated.</td>
</tr>
<tr>
<td>reserved-leased-</td>
<td>Number of reserved leases that an administrator deactivated.</td>
</tr>
<tr>
<td>deactivated</td>
<td></td>
</tr>
<tr>
<td><strong>Prefix Delegation Lease Counts</strong></td>
<td></td>
</tr>
<tr>
<td>max-pd-balancing-length</td>
<td>Prefix length used for counting the prefix delegation leases.</td>
</tr>
<tr>
<td>prefixes-in-use</td>
<td>Number of prefixes, of the max-pd-balancing-length prefix length, that are in use.</td>
</tr>
<tr>
<td>prefixes-available</td>
<td>Number of prefixes, of the max-pd-balancing-length prefix length, available to any client on this server.</td>
</tr>
<tr>
<td>prefixes-other-available</td>
<td>Number of prefixes, of the max-pd-balancing-length prefix length, available to any client on the failover partner.</td>
</tr>
<tr>
<td>prefixes-in-transition</td>
<td>Number of prefixes, of the max-pd-balancing-length prefix length, in transition between the failover partners.</td>
</tr>
<tr>
<td><strong>Failover Related</strong></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 27  Managing DHCPv6 Addresses

DHCPv6 Configuration

Viewing DHCPv6 Networks

To view the networks in the DHCPv6 address space, choose Networks from the Design > DHCPv6 menu, to open the DHCPv6 Network Tree page. On this page you can add DHCPv6 links using a template and a template root prefix, as you would on the List/Add DHCPv6 Links page. Adding a link opens the Add DHCPv6 Link page. After creating the link, you can select it on the View DHCPv6 Networks page for editing.

Tip
You can use the DHCP v6 Network Tree page to push and reclaim links. Click the Push or Reclaim icon for the desired link. (See in the “Creating and Editing Links” section on page 27-18 for details.)

Editing DHCPv6 Server Attributes

You can edit DHCP server attributes related to DHCPv6. These attributes are:

- `v6-client-class-lookup-id`—Expression that determines a client-class based on the DHCPv6 client request and returns a string with either the name of a configured client-class or <none> (if the expression does not wish to provide a client-class). The attribute has no preset value.
- `max-client-leases`—Maximum number of leases a DHCPv6 client can have on a link. Do not use this attribute to limit clients to one lease only. The preset is 50.

Local Basic or Advanced Web UI

From the Deploy menu, choose DHCP Server under the DHCP submenu to open the Manage DHCP Server page. Click the Local DHCP Server link to open the Edit DHCP Server page, modify the aforementioned DHCPv6 attribute values, then click Save.

CLI Commands

Use `dhcp` to show the aforementioned DHCPv6 server attributes, then modify them by using `dhcp set`.

Configuring DHCPv6 Policies

You can edit DHCPv6 policy attributes, which are:

- `affinity-period`—See the “Lease Affinity” section on page 27-8 (no preset value).
Chapter 27 Managing DHCPv6 Addresses

**DHCPv6 Configuration**

- **allow-non-temporary-addresses**—Enable or disable DHCPv6 clients requesting nontemporary (IA_NA) addresses (preset value enable).
- **allow-rapid-commit**—With Rapid Commit enabled, clients receive information (when solicited) on committed addresses, which are then more quickly committed with a client request (preset value disable). Use Rapid Commit only if one DHCP server is servicing clients, otherwise it might seem like the client is receiving multiple addresses. (See the “DHCPv6 Policy Hierarchy” section on page 27-12 for special handling of this attribute, and Reconfigure support, when used in an embedded or named policy for a prefix.)
- **allow-temporary-addresses**—Enable or disable DHCPv6 clients requesting temporary (IA_IA) addresses (preset value enable).
- **default-prefix-length**—For prefix delegation, default prefix length of the delegated prefix if the client or router does not explicitly request it (or allow-client-hints is disabled); must always be less than or equal to the prefix range prefix length (preset value 64 bytes).
- **preferred-lifetime**—Default and maximum preferred lifetime for leases (preset value 1 week).
- **v6-reply-options**—DHCPv6 options returned in replies to clients (no preset value). (See the “DHCPv6 Policy Hierarchy” section on page 27-12 for special handling of this attribute when used in an embedded or named policy for a prefix.)
- **valid-lifetime**—Default and maximum valid lifetime for leases (preset value 2 weeks).

**Tip**

For details on the Reconfigure attributes, see the “Reconfigure Support” section on page 27-35.

**Local Advanced Web UI**

**Step 1**
From the **Design** menu, choose **Policies** under the **DHCP Settings** submenu to open the List/Add DHCP Policies page.

**Step 2**
Click the **Add Policy** icon in the Policies pane to add a new policy or select an existing policy to open the Edit DHCP Policy page.

**Step 3**
The DHCP Policies page has DHCPv4 and DHCPv6 options sections. Add (or delete) options and set attributes as desired, then click **Save**.

**CLI Commands**

Use **policy list** or **policy name show** to show the aforementioned policy attributes, then modify them by using **policy name set** or **enable**.

**Configuring DHCPv6 Client-Classes**

You can configure DHCPv6 client-class attributes, which are:

- **v6-client-lookup-id**—Key value to use to look up the DHCPv6 client in the client database (locally or through LDAP), specified as an expression that evaluates to a string (or a blob as a valid string).
- **v6-override-client-id**—Value that replaces any client-identity value in an incoming packet, specified as an expression that evaluates to a blob.
Local Advanced Web UI

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>From the <strong>Design</strong> menu, choose <strong>Client Classes</strong> under the <strong>DHCP Settings</strong> to open the List/Add DHCP Client Classes page.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Click an existing client-class to open the Edit DHCP Client Class page, or click the <strong>Add Client Classes</strong> icon in the Client Classes pane to add a new client-class on the Add DHCP Client Class dialog box. Both pages include the aforementioned attributes.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Click <strong>Save</strong>.</td>
</tr>
<tr>
<td>Step 4</td>
<td>To generate clients, be sure that <em>validate-client-name-as-mac</em> is disabled for the DHCP server. This attribute appears on the Edit DHCP Server page under the Client-Class attributes.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Reload the DHCP server.</td>
</tr>
</tbody>
</table>

CLI Commands

Use `client-class list` or `client-class name show` to show the aforementioned client-class attributes, then modify them using `client-class name set`. To generate clients, be sure that *validate-client-name-as-mac* is disabled for the DHCP server.

Configuring DHCPv6 Clients

You can configure DHCPv6 clients.

Local Advanced Web UI

From the **Design** menu, choose **Clients** under the **DHCP Settings** submenu to open the List/Add DHCP Clients page. Select an existing client to open the Edit DHCP Client page or click the **Add Clients** icon on Clients pane to add a new client-class on the List/Add DHCP Client page, choose the client-class that includes the DHCPv6 attributes that were set (see the “Configuring DHCPv6 Client-Classes” section on page 27-33), then click **Save**.

**Tip**

Disable the *validate-client-name-as-mac* attribute for the DHCP server.

CLI Commands

Use `client list` or `client name show` to show the existing clients. To set the client-class name for the client, use `client name set client-class-name=value`. Also ensure that the *validate-client-name-as-mac* attribute is disabled for the DHCP server.

Setting DHCPv6 Options

Set DHCPv6 options and vendor options when you create or edit policies (embedded or named) for prefixes. (See the “DHCPv6 Policy Hierarchy” section on page 27-12 for special handling of the *v6-options* and *v6-vendor-options* policy attributes when used in an embedded or named policy on a prefix.)
Local Advanced Web UI

The DHCPv6 options coexist along with the DHCPv4 options on the List/Add DHCP Policies or Edit DHCP Policy page. Note that the vendor options appear only if you create these options (see the “Creating DHCP Option Definition Sets and Option Definitions” section on page 22-8).

You can select the options from the drop-down lists. If option descriptions exist, they appear under the Name and Number headings, which you can click to sort the entries.

CLI Commands

Use `policy name setV6Option` or `policy name setV6VendorOption`. The option settings require an option name (or ID) and a value. For example:

```
nrcmd> policy dhcpv6-policy setV6Option dns-servers 2222::1,2222::2
nrcmd> policy foo setV6VendorOption 17 dhcp6-cablelabs-config "(32 2222::3,2222::4)"
```

Reconfigure Support

For DHCPv6, a server can send a RECONFIGURE message to a client to inform the client that the server has new or updated configuration parameters. If so authorized and through proper authentication, the client then immediately initiates a Renew, Rebind, or Information-request reply transaction with the server so that the client can retrieve the new data. Without this support, a client must wait until it renews its lease to get configuration updates.

You can have the server unicast the Reconfigure packet or deliver it through a relay agent. If you do not specify either way, the client’s client-class policy, requested lease’s prefix or link policies, or system_default_policy (but not the client policy) determines the preferred method. If the unicast method is not available (the client has no valid address lease), the server uses the relay agent; with no relay agent, the server tries to unicast; failing both results in an error. With the unicast method, if the specified lease is not usable, the server selects the lease with the longest valid lifetime.

The server and client negotiate Reconfigure support through the added security of a reconfigure key. The internal process is basically:

1. The client sends the server a REQUEST, SOLICIT, or ADVERTISE packet that includes the `reconfigure-accept` option (20) to indicate that the client wants to accept Reconfigure messages. (Conversely, the DHCP server can send a `reconfigure-accept` option to the client about whether the client should accept Reconfigure messages.) This option is required for Reconfigure support.

2. If the Cisco Prime IP Express policy for the client has the `reconfigure` attribute set to `allow` or `require` (rather than `disallow`), the DHCP server accepts the packet and generates a reconfigure key for the client. (The server records the key value and its generation time in the `client-reconfigure-key` and `client-reconfigure-key-generation-time` attributes for the DHCPv6 lease.)

3. The server sends a Reply packet to the client with the reconfigure key in the `auth` option (11) along with the `reconfigure-accept` option.

4. The client records the reconfigure key to authenticate Reconfigure messages from the server.

5. When the server wants to reconfigure the client, it sends a Reconfigure packet with the `reconfigure-message` option (19) and an `auth` option containing a hash generated from the packet and the reconfigure key. The `reconfigure-message` option indicates in the `msg-type` field whether the client should respond with a Renew or an Information-request packet.
6. Upon receiving the packet, the client validates that the auth option contains the valid hash, then returns a Renew, Rebind, or Information-request packet. This packet includes an Option Request (oro) option (6) to indicate specific option updates. (If the server does not receive a reply from the client in a preconfigured timeout value of 2 seconds, the server retransmits the Reconfigure message at most 8 times, then aborts the reconfigure process for the client.)

7. The server sends the client a Reply packet that includes options for configuration parameters. The packet might also include options containing addresses and new values for other configuration parameters, even if the client did not request them. The client records these changes.

Note For details on how Reconfigure support affects particular DHCP extension points, see the “Extension Dictionaries” section on page 30-23.

Local Advanced Web UI

The List/Add DHCP Leases for Prefix page includes a Reconfigure button for each lease so that you can initiate a reconfiguration request for that particular lease.

CLI Commands

To support Reconfigure, Cisco Prime IP Express includes the following syntax for the lease6 command:

```
lease6 ipaddr reconfigure [renew | rebind | information-request] [-unicast | -via-relay]
```

The options determine whether to have the client respond to the Reconfigure message with a Renew, Rebind, or Information-request packet, and whether the server should unicast or go through a relay agent. The lease6 list and show commands also display values for these related attributes:

- **client-reconfigure-key**—128-bit key that the server generates for Reconfigure messages to the client.
- **client-reconfigure-key-generation-time**—Time at which the server generated the client-reconfigure-key.

The policy command includes two related attribute settings:

- **reconfigure**—Whether to allow (1), disallow (2), or require (3) Reconfigure support; the preset value is allow (1).
- **reconfigure-via-relay**—Whether to allow reconfiguration over a relay agent; the preset value is false, whereby reconfiguration notification is by unicasting from the server.

DNS Update for DHCPv6

For details on enabling and configuring DNS update for DHCPv6 clients, see the “DNS Update for DHCPv6” section on page 29-2.
Managing DHCP Failover

Cisco Prime IP Express failover protocol is designed to allow a backup DHCP server to take over for a main server if the main server is taken offline for any reason. This protocol is TCP based and supports both DHCPv4 and DHCPv6. The DHCP failover supports the following features:

- DHCPv4 addresses
- DHCPv6 addresses (non-temporary and temporary)
- DHCPv6 prefix delegation

The DHCP failover is not applicable to DHCPv4 on-demand address pools.

Related Topics

How DHCP Failover Works, page 28-1
DHCP Simple Failover, page 28-2
DHCPv6 Failover, page 28-3
Setting Up Failover Server Pairs, page 28-3
Configuring Failover Parameters Based on Your Scenario, page 28-8
Recovering from a DHCP Failover, page 28-14
Maintaining Failover Server Pair, page 28-20
Restoring a Standalone DHCP Failover Server - Tutorial, page 28-21
Changing Failover Server Roles, page 28-27
Troubleshooting Failover, page 28-29
Supporting BOOTP Clients in Failover, page 28-31

How DHCP Failover Works

DHCP failover is based on a server-partner relationship. The partner must have identical DHCPv4 scopes, DHCPv6 prefixes, DHCPv6 links, reservations, policies, and client-classes, as the server. After the servers start up, they contact each other. The main server provides its partner with a DHCPv4 addresses and DHCPv6 delegated prefixes, and updates its partner with every client operation. If the main server fails, then the partner takes over offering and renewing leases, using its DHCPv4 addresses and DHCPv6 delegated prefixes. When the main server becomes operational again, it re-integrates with its partner without administrative intervention. These servers are in a relationship known as a failover pair.
The failover protocol keeps DHCP operational, if:

- **The main server fails**—The partner takes over services during the time the main server is down. The servers cannot generate duplicate addresses, even if the main server fails before updating its partner.

- **Communication fails**—A partner can operate correctly even though it cannot tell whether it was the other server or the communication with it that failed. The servers cannot issue duplicate addresses, even if they are both running and each can communicate with only a subset of clients.

After a failover pair is configured:

1. The partners connect.
2. The main server supplies data about all existing leases to its partner.
3. The backup server requests a pool of backup addresses from the main server.
4. The main server replies with a percentage of available addresses from each scope or prefix to its partner.
5. The backup server ignores all DHCPDISCOVER and Solicit requests, unless it senses that the main server is down or load balancing is enabled for the failover pair. In normal operation, the backup server handles only some renewal and rebinding requests.
6. The main server updates its partner with the results of all client operations.

You can automatically synchronize the configuration of the servers in a failover pair. The two servers dynamically rebalance the available leases; if the main server hands out a large portion of its available leases, it can reclaim leases from its partner.

## DHCP Simple Failover

Cisco Prime IP Express supports only simple failover configuration. Simple failover involves a single main server and a single backup server pair (see Figure 28-1 on page 28-3). In the example, main server A has three scopes or prefixes that must be configured identically on backup server B.
DHCPv6 Failover

DHCPv6 failover works very similar to the DHCPv4 simple failover configuration. The DHCPv6 failover partners keep each other updated on stateful address and delegated prefix leases that are granted, perform synchronization when communication is restored, and generally follow and adhere to the DHCPv4 failover protocol requirements (except the differences between DHCPv4 and DHCPv6).

The maximum client lead time (MCLT) and lease time restrictions are applied to DHCPv6 leases and both the valid lifetime and preferred lifetime of leases are limited to MCLT defined for the failover pair. Only when the longest lease time allowed by the failover pair exceeds the configured preferred lifetime and if the configured preferred lifetime is less than the configured valid lifetime, the preferred lifetime and valid lifetime of the lease can be different. The delegated prefixes are managed and balanced similar to DHCPv4 addresses.

The most significant difference is that the DHCPv6 failover servers do not balance the available addresses on each prefix but instead use an algorithm to determine which new addresses each server can lease. The algorithm uses the least significant bit of the address and the main server assigns odd addresses whereas the backup server assigns even addresses and is not applicable if:

- A lease is already assigned to the client.
- A reservation exists for the client.
- The allocation-algorithms interface-identifier is set and is used. In this case, the interface-identifier (EUI-64) bit is assumed to be unique, and as the global bit is set, these addresses do not conflict with randomly generated addresses as these never have the global bit set.
- Client reservations are configured on the prefix.
- An extension supplies the address.

Setting Up Failover Server Pairs

You can create and synchronize failover pairs at the local and regional clusters.

A failover pair has two main elements, its configuration and the state information that the servers maintain. The key configuration attributes are the name of the failover pair, the role of the local server (main or backup), and the address of the partner. The failover state is defined when you reload the server and the server processes this state data at startup.
## Adding Failover Pairs

Create the DHCP failover pair based on the cluster of the main and backup servers. Then synchronize the configuration of the failover pair so that the scopes, prefixes, policies, and other DHCP properties match between the servers.

To add a failover pair:

### Local and Regional Web UI

1. **Step 1** From the **Deploy** menu, choose **Failover Pair** under the **DHCP** submenu to open the List/Add DHCP Failover Pairs page.
2. **Step 2** Click the **Add Failover Pair** icon in the Failover pane.
3. **Step 3** On the Add DHCP Failover Pair dialog box, add a failover pair name.
   - This is required and can be any distinguishing name. (See the “Changing Failover Pair Server Addresses” section on page 28-20.)
4. **Step 4** Choose the cluster for the main DHCP server. This can be localhost or some other cluster you define.
   - If you change the IP address of your local host, you must modify the localhost cluster (on the Edit Cluster page) to change the address in the IP Address field. Do not set the value to 127.0.0.1.
5. **Step 5** Choose the cluster for the backup DHCP server. This cannot be the same as the main server cluster, but it must be localhost if the main cluster is not localhost. Whatever you select here becomes the IP address value for the `dhcp-backup-server` attribute (Expert mode attribute) once you add the failover pair, then click **Add DHCP Failover Pair**.
6. **Step 6** You can set additional attributes, such as the maximum client lead time (`mclt`) or backup percentage (`backup-pct`). Most of the default values are optimized. Leave the `failover` attribute enabled by default unless you want to temporarily disable failover for the pair.
   - You must specify both the `main-ip6address` and `backup-ip6address` if you want to use TCP over IPv6 address for failover pair communication. The `main-server` address must be changed only if a multihomed host and the management address of the cluster is not appropriate.
7. **Step 7** Click **Save**. You can edit the failover pair properties.

### Related Topics

- Adding Failover Pairs, page 28-4
- Synchronizing Failover Pairs, page 28-5
- Failover Checklist, page 28-8

### Changing Failover Pair Server Addresses

- Changing Failover Pair Server Addresses, page 28-20
- Synchronizing Failover Pairs, page 28-5
- Restarting the Failover Servers, page 28-21
CLI Commands

Use `failover-pair name create main-server-address backup-server-address`. If you want to synchronize between the servers, you must also specify the main and backup clusters by setting the `main` and `backup` attributes. For example:

```
nrcmd> failover-pair example-fo-pair create 192.168.50.1 192.168.60.1 main=Example-cluster backup=Boston-cluster
```

Synchronizing Failover Pairs

Once you create the failover pairs, you must synchronize the failover pair configuration.

Web UI

**Step 1** From the Deploy menu, choose Failover Pairs under the DHCP submenu to open the List/Add DHCP Failover Pairs page.

**Step 2** Select the failover pair on the Failover pane.

**Step 3** On the List/Add DHCP Failover Pairs page, click the Synchronize DHCP Failover Pair tab.

For synchronization in the regional web UI, see the “Managing DHCP Failover Pairs” section on page 6-20.

**Step 4** Choose the direction of synchronization. The direction of synchronization can be either from main to backup server or from backup to main server.

**Step 5** Choose the synchronization operation, depending on the degree to which you want the main server objects to replace those of the backup server. The following are the basic synchronization operations that can be performed on the servers:

- **Update operation**—This is the default and least radical operation. It is appropriate for update synchronizations in that it has the least effect on the unique properties of the backup server.
- **Complete operation**—This operation is appropriate for all initial synchronizations. It is more complete than an update operation, while still preserving many of the backup server unique properties.
- **Exact operation**—This operation is appropriate for simple failover configuration. It makes the two servers mirror images of each other, as much as possible, although this operation retains the unique DHCP server, and extension points on the backup server.

**Note** For initial failover configurations, use the Exact or Complete operation.

For a better understanding of the functions that are performed on the classes of the objects, consider the following example. Here, we have a main server and its backup server with the following objects:

<table>
<thead>
<tr>
<th>On the main server</th>
<th>On the backup server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name1=A</td>
<td>Name2=B</td>
</tr>
<tr>
<td>Name2=C</td>
<td>Name3=D</td>
</tr>
</tbody>
</table>
Chapter 28      Managing DHCP Failover

Setting Up Failover Server Pairs

Note In this example, we consider failover synchronization from the main server to the backup server.

Each operation performs a different mix of functions on the classes of objects. The following are the four functions that are performed on the objects based on the operation selected.

- **no change**—Makes no change to the list of properties or their values on the backup server.
  
  For example, the result would be Name2=B, Name3=D.

- **ensure**—Ensures that a copy of the main server object exists on the backup. The target server objects with the same name as main server objects are left unchanged, the objects that are not on the target server are added to it, and the objects only on the target server are left unchanged.
  
  For example, the result would be Name1=A, Name2=B, Name3=D.

- **replace**—Ensures that the existing object in the target server is replaced by the main server object of the same name. Also the objects that are not on the target server are added to it and the objects only on the target server are left unchanged. The only exceptions to this are for policies and option definition sets, where the option lists are extracted to compare the list entries.
  
  For example, the result would be Name1=A, Name2=C, Name3=D.

- **exact**—Puts an exact copy of the main server object on the backup server and removes the unique ones. That is, the objects of target server are made identical to the objects of main server.
  
  For example, the result would be Name1=A, Name2=C.

  For more information, see Table 28-1 on page 28-6. This table provides the information on the functions (no change, ensure, replace, or exact) that are performed on the objects based on the operations (Update, Complete, Exact) you select.

**Table 28-1   Failover Pair Synchronization Functions**

<table>
<thead>
<tr>
<th>Data Description</th>
<th>Update</th>
<th>Complete</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP Server:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client-Class Properties</td>
<td>replace</td>
<td>replace</td>
<td>replace</td>
</tr>
<tr>
<td>Client Host Name Processing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic DNS Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Failover Tuning Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All other properties</td>
<td>no change</td>
<td>replace</td>
<td>replace</td>
</tr>
<tr>
<td>LDAP Remote Server</td>
<td>ensure</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>Policy:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option List Properties</td>
<td>ensure</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>Packet Boot File Properties</td>
<td>ensure</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>All other properties</td>
<td>replace</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>Client</td>
<td>replace</td>
<td>replace</td>
<td>exact</td>
</tr>
</tbody>
</table>
Table 28-1  Failover Pair Synchronization Functions (continued)

<table>
<thead>
<tr>
<th>Data Description</th>
<th>Update</th>
<th>Complete</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client-Class</td>
<td>replace</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>Scopes</td>
<td>exact</td>
<td>exact</td>
<td>exact</td>
</tr>
<tr>
<td>Links</td>
<td>exact</td>
<td>exact</td>
<td>exact</td>
</tr>
<tr>
<td>Prefixes</td>
<td>exact</td>
<td>exact</td>
<td>exact</td>
</tr>
<tr>
<td>DNS Update Configuration</td>
<td>replace</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>Trap Configuration</td>
<td>ensure</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>VPN</td>
<td>replace</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>Key</td>
<td>replace</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>Extensions</td>
<td>ensure</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>(You must copy extension files.)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extension Point</td>
<td>replace</td>
<td>replace</td>
<td>replace</td>
</tr>
<tr>
<td>Option Information:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom options list</td>
<td>ensure</td>
<td>replace</td>
<td>exact</td>
</tr>
<tr>
<td>Vendor options list</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHCP Listener Configuration</td>
<td>ensure</td>
<td>replace</td>
<td>exact</td>
</tr>
</tbody>
</table>

Step 6 Click **Report** on the Synchronize DHCP Failover Pair page:

- When you click **Report**, the resulting View DHCP Failover Pair Sync Report page shows what change entries the synchronization will apply if you run the synchronization. You have the option to choose the direction of synchronization and also the option to check the desired mode of synchronization operation (Update, Complete, Exact). Check the desired values and click **Report**.

Step 7 Click **Save** on the View DHCP Failover Pair Sync Report page.

Step 8 On the List/Add DHCP Failover Pairs page, click the **Manage Failover Servers** tab.

Step 9 Click the **Restart Server** icon to reload the backup server.

Step 10 Try to get a lease.

Step 11 On the Manage Failover Servers tab, look at the health of the servers. Also, click the **Logs** tab to view the log entries on the Log for Server page, and ensure that the servers are in NORMAL failover mode. The log file should contain an item similar to the following:

```
06/19/2003  9:41:19 name/dhcp/1 Info Configuration 0 04092 Failover is enabled server-wide. Main server name: '192.168.0.1', backup server name: '192.168.0.110', mclt = 3600, backup-pct = 10, dynamic-bootp-backup-pct = 0, use-safe-period: disabled, safe-period = 0.
```

CLI Commands

Use `failover-pair name sync {update | complete | exact} {main-to-backup | backup-to-main}`:

```
nrcmd> failover-pair example-fo-pair sync exact main-to-backup
```
Failover Checklist

Once you create the failover pair, you must synchronize the configuration of the failover servers. Use this checklist to prepare for an effective failover configuration:

- Duplicate the DHCPv4 scope, DHCPv6 prefix, DHCPv6 links, reservations (IPv4 and IPv6), selection tags, policy, DHCP option, IP addresses, client-classes, dynamic DNS updates, dynamic BOOTP, VPN, DHCP extensions, DHCP extensions, LDAP server, and address configurations on the partner servers by synchronizing a failover server pair for a simple failover scenario.

- Ensure that both partners are configured with a wide enough range of addresses so that the backup server can provide leases for a reasonable amount of time while the main server is down.

- If you use BOOTP (DHCP) relay agents (IP helpers), configure all BOOTP relay agents to point to both partners. Cisco Prime IP Express does not automatically detect this. You can detect BOOTP configuration errors only by performing live tests in which you periodically take the main server out of service to verify that the backup server is available to DHCP clients.

Configuring Failover Parameters Based on Your Scenario

Following are the advanced failover properties that are important to set:

- Backup percentage (see the “Setting Backup Percentages” section on page 28-8)
- Backup allocation boundaries (see the “Setting Backup Allocation Boundaries” section on page 28-20)
- Maximum client lead time (MCLT) (see the “Setting the Maximum Client Lead Time” section on page 28-10)
- Safe period (see the “Using the Failover Safe Period to Move Servers into PARTNER-DOWN State” section on page 28-11)
- Request and response packet buffers (see the “Setting DHCP Request and Response Packet Buffers” section on page 28-13)
- Load balancing (see the “Setting Load Balancing” section on page 28-13)

Setting Backup Percentages

To keep failover partners operating despite a network partition (when both servers can communicate with clients, but not with each other), allocate more addresses than the addresses for a single server. Configure the main server to allocate a percentage of the currently available addresses in each scope and prefix delegation prefixes to the backup server. This makes these addresses unavailable to the main server. The backup server uses these addresses when it cannot talk to the main server and cannot tell if it is down.

If the main server detects that the address pool is significantly out of balance or the server has no leases, then the pool of available or other-available leases are rebalanced even when the failover pair is functioning in the Normal state. The failover pair must be carefully monitored during failover and if the failover partner is down for an extended period then operator intervention may be required to move the failover partner to the PARTNER-DOWN state.

You can set the percentage of currently available addresses by setting the backup-pct attribute on the failover pair or DHCPv4 scope (failover-pair name set fail backup-pct or scope name set backup-pct in the CLI). The default backup percentage is 50%. DHCPv6 prefix delegation prefixes are fixed at 50% for the backup-pct equivalent.
Note that setting the backup percentage on the failover pair level sets the value for all scopes not set with that attribute. However, if set at the scope level, the backup percentage overrides the one at the failover pair level. If the `load-balancing` attribute is enabled for the failover pair (``failover-pair name enable load-balancing`` in the CLI), the backup percentage is fixed at 50% and any of the backup percentage attributes (on a failover pair or scope) are ignored.

The backup percentage should be set large enough to allow the backup server to continue serving new clients in the event that the main server fails. The backup percentage is calculated based on the number of available addresses. The backup percentage can safely be set to a larger value, if extended outages are expected, because the main server periodically reclaim addresses (once per hour) if, in the course of normal leasing activity, the main server's available address pool drops below its predefined percentage. For example, if backup percentage is set to 60%, the main server will reclaim addresses if its address pool falls below 60%.

---

**Note**

When failover load balancing is in effect, the main and backup servers actively move available leases between them to maintain the backup percentage of available leases. See the “Setting Load Balancing” section on page 28-13.

The percentage depends on the new client arrival rate and the network operator reaction time. It depends on the arrival rate of new DHCP clients and the reaction time of your network administration staff. The backup server needs enough addresses from each scope to satisfy all new clients requests arriving during the time it does not know if the main server is down. Even during PARTNER-DOWN state, the backup server waits for the maximum client lead time (MCLT) and lease time to expire before reallocating leases. See the “Setting the Maximum Client Lead Time” section on page 28-10. When these times expire, the backup server offers:

- Leases from its private pool.
- Leases from the main server pool.
- Expired leases to new clients.

During the working hours, an operator likely responds within two hours to COMMUNICATIONS-INTERRUPTED state to determine if the main server is working. The backup server then needs enough addresses to support a reasonable upper bound on the number of new clients that could arrive during those two hours.

During off-hours, the arrival rate of previously unknown clients is likely to be less. The operator can usually respond within 12 hours to the same situation. The backup server then needs enough addresses to support a reasonable upper bound on the number of clients that could arrive during those 12 hours.

The number of addresses over which the backup server requires sole control is the greater of the two numbers. You can express this number as a percentage of the currently available (unassigned) addresses in each scope. If you use client-classes, remember that some clients can only use some sets of scopes and not others.

---

**Note**

During failover, clients can sometimes obtain leases whose expiration times are shorter than the amount configured. This is a normal part of keeping the server partners synchronized. Typically this happens only for the first lease period, or during COMMUNICATIONS-INTERRUPTED state.

**Related Topics**

- BOOTP Backup Percentage, page 28-32
Setting the Maximum Client Lead Time

You can set a property for the failover pair that controls an adjustment to the lease period, the maximum client lead time (MCLT). The MCLT adjusts for a potential period of uncertain connectivity between the servers. It is the maximum time one server can grant (or extend) a lease to a client without first negotiating a longer time with its partner. This time has the following implications:

- Clients may initially (or if the partners are not communicating) only receive leases of the MCLT length. This means that they need to renew leases sooner than they might otherwise without failover. At this renewal, the client should get a full lease time (unless the partners are not communicating).

- If a server enters PARTNER-DOWN state, it must wait until the MCLT after the later of the partner-down time or the latest lease expiration time communicated with the partner gets over. The latest lease expiration time communicated to the partner is typically 1.5 times the lease time from the last client lease request before communication was interrupted.

- If a failover recovery occurs where there is uncertainty about what one partner did (such as when it loses its lease database), the partners may have to restrict leasing activity for the MCLT period after they synchronize before they can resume normal failover operations.

The default MCLT is one hour, the optimum for most configurations. As defined by the failover protocol, the lease period given for a client can never be more than the MCLT plus the most recently received potential expiration time from the failover partner, or the current time, whichever is later. That is why you sometimes see the initial lease period as only an hour, or an hour longer than expected for renewals. The actual lease time is recalculated when the main server comes back.

The MCLT is necessary because of failover use of lazy updates. Using lazy updates, the server can issue or renew leases to clients before updating its partner, which it can then do in batches of updates. If the server goes down and cannot communicate the lease information to its partner, the partner may try to reoffer the lease to another client based on what it last knew the expiration to be. The MCLT guarantees that there is an added window of opportunity for the client to renew. The way that a lease offer and renewal works with the MCLT is:

1. The client sends a DHCPDISCOVER or DHCPv6 Solicit to the server, requesting a desired lease period (for example, three days). The server responds with a DHCPOFFER or DHCPv6 Advertise with an initial lease period of only the MCLT (one hour by default). The client then requests the MCLT lease period and the server acknowledges it.

2. The server sends its partner a bind update containing the lease expiration for the client as the current time plus the MCLT. The update also includes the potential expiration time as the current time plus the client desired period plus half of the client desired period (3 + 1.5 = 4.5 days). The partner acknowledges the potential expiration, thereby guaranteeing the transaction.

3. When the client sends a renewal request halfway through its lease (in one-half hour), the server acknowledges with the client desired lease period (3 days). The server then updates its partner with the lease expiration as the current time plus the desired lease period (3 days), and the potential expiration time (4.5 days. See the description in Step 2.). The partner acknowledges this potential expiration of 4.5 days. In this way, the main server tries to have its partner always lead the client in its understanding of the client lease period so that it can always offer it to the client.

There is no one correct value for the MCLT. There is an explicit trade-off between various factors in choosing one. Most people use the preset value of one hour effectively and it works well in almost all environments. Here are some of the trade-offs between a short and long MCLT:

- Short MCLT—A short MCLT value means that after entering PARTNER-DOWN state, a server only has to wait a short time before it can start allocating its partner IP addresses to DHCP clients. Furthermore, it only has to wait a short time after a lease expires before it can reallocate that address to another DHCP client. However, the down side is that the initial lease interval that is offered to
every new DHCP client will be short, which causes increased traffic, because those clients need to send their first renewal in a half of a short MCLT time. Also, the lease extensions that a server in COMMUNICATIONS-INTERRUPTED state can give is the MCLT only after the server has been in that state for around the desired client lease period. If a server stays in that state for that long, then the leases it hands out will be short, increasing the load on that server, possibly causing difficulty.

- **Long MCLT**—A long MCLT value means that the initial lease period will be longer and the time that a server in COMMUNICATIONS-INTERRUPTED state can extend leases (after it being in that state for around the desired client lease period) will be longer. However, a server entering PARTNER-DOWN state must wait the longer MCLT before being able to allocate its partner addresses to new DHCP clients. This may mean that additional addresses are required to cover this time period. Also, the server in PARTNER-DOWN state must wait the longer MCLT from every lease expiration before it can reallocate an address to a different DHCP client.

### Using the Failover Safe Period to Move Servers into PARTNER-DOWN State

One or both failover partners could potentially move into COMMUNICATIONS-INTERRUPTED state. They cannot issue duplicate addresses while in this state. However, having a server in this state over longer periods is not a good idea, because there are restrictions on what a server can do. The main server cannot reallocate expired leases and the backup server can run out of addresses from its pool. COMMUNICATIONS-INTERRUPTED state was designed for servers to easily survive transient communication failures of a few minutes to a few days. A server might function effectively in this state for only a short time, depending on the client arrival and departure rate. After that, it would be better to move a server into PARTNER-DOWN state so it can completely take over the lease functions until the servers resynchronize.

There are two ways a server can move into PARTNER-DOWN state:

- **User action**—An administrator sets a server into PARTNER-DOWN state based on an accurate assessment of reality. The failover protocol handles this correctly. Never set both partners to PARTNER-DOWN.

- **Failover safe period expires**—When the servers run unattended for longer periods, they need an automatic way to enter PARTNER-DOWN state.

Network operators might not sense in time that a server is down or uncommunicative. Hence, the failover safe period, which provides network operators some time to react to a server moving into COMMUNICATIONS-INTERRUPTED state. During the safe period, the only requirement is that the operators determine that both servers are still running, and if so, fix the network communications failure or take one of the servers down before the safe period expires.

The length of the safe period is installation-specific, and depends on the number of unallocated addresses in the pool and the expected arrival rate of previously unknown clients requiring addresses.

The use-safe-period attribute is enabled by default for a failover pair and the default safe period is 4 hours. This ensures that if the failover partner is in COMMUNICATIONS-INTERRUPTED state for 4 hours, it will enter PARTNER-DOWN state automatically after the safe period elapses. You may need to review if this setting is appropriate for your network and adjust the safe-period based on your network requirements.

In addition, during this safe period, either server allows renewals from any existing client, but there is a major risk of possibly issuing duplicate addresses. This is because one server can suddenly enter PARTNER-DOWN state while the other is still operating. In order to prevent this problem, it is important that you do not change the default settings for use-safe-period and put operational procedures in place to alert operations personnel when the failover pair loses contact with each other. Especially, in the event
of network communications failure, operator intervention is required before the safe period elapses. Either one failover server needs to be taken offline or the use-safe-period attribute needs to be disabled on both the servers before the safe period elapses.

**Note**
The use-safe-period is enabled by default. You may want to review if this is appropriate for your network and you may want to disable the use-safe-period or adjust the safe-period based on your network requirements and monitoring.

The number of extra addresses required for the safe period should be the same as the expected total of new clients a server encounters. This depends on the arrival rate of new clients, not the total outstanding leases. Even if you can only afford a short safe period, because of a shortage of addresses or a high arrival rate of new clients, you can benefit substantially by allowing DHCP to ride through minor problems that are fixable in an hour. There is minimum chance of duplicate address allocation, and reintegration after the solved failure is automatic and requires no operator intervention.

If the failover safe period length is more than the length of the MCLT and the failover server enters into PARTNER-DOWN state because of the safe-period, the server can start allocating its partner other-available leases to DHCP clients immediately. The advantage of this is that the server has additional leases to allocate. However, the disadvantage is that operator intervention is required within the safe period in case of network communications failure. Either the failover server needs to be taken offline or the use-safe-period attribute needs to be disabled on both the servers before the safe period elapses. Without operator intervention, both failover servers will transition to PARTNER-DOWN state and start allocating its partner addresses to new DHCP clients.

Here are some guidelines to follow, to help you decide whether to use manual intervention or the safe period for transitioning to PARTNER-DOWN state:

- If your corporate policy is to have minimal manual intervention, set the safe period. Enable the failover pair attribute use-safe-period to enable the safe period. Then, set the DHCP attribute safe-period to set the duration (4 hours by default). Set this duration long enough so that operations personnel can explore the cause of the communication failure and assure that the partner is truly down.

- If your corporate policy is to avoid conflict under any circumstances, then never let either server go into PARTNER-DOWN state unless by explicit command. Allocate sufficient addresses to the backup server so that it can handle new client arrivals during periods when there is no administrative coverage. You can set PARTNER-DOWN on the Manage Failover Servers tab of the web UI, if the partner is in the Communications-interrupted failover state, you can click **Set Partner Down** in association with an input field for the PARTNER-DOWN date setting. This setting is initialized to the value of the start-of-communications-interrupted attribute. (In Normal web UI mode, you cannot set this date to be an earlier value than the initialized date. In Expert web UI mode, you can set this value to any date.)

Use **failover-pair name setPartnerDown date** in the CLI, specifying the name of the partner server. This moves all the scopes running failover with the partner into PARTNER-DOWN state immediately, unless you specify a date and time with the command. This date and time should be when the partner was last known to be operational.

If you use **setPartnerDown** in the CLI and specify the date and time when the partner was last known to be operational then the failover server calculates the MCLT from the time specified in the **setPartnerDown** command. If the date and time is not specified for the **setPartnerDown** command, then the failover server calculates the MCLT from the time the failover server moved to the COMMUNICATIONS-INTERRUPTED state. In case of network communications failure, it is important that you specify the actual time the partner was last known to be operational in the **setPartnerDown** command. Otherwise, it can result in duplicate IP addresses.
Chapter 28  Managing DHCP Failover

Configuring Failover Parameters Based on Your Scenario

There are two conventions for specifying the date:

- \(-num\ unit\) (a time in the past), where \(num\) is a decimal number and \(unit\) is \(s\), \(m\), \(h\), \(d\), or \(w\) for seconds, minutes, hours, days or weeks respectively. For example, specify \(-3d\) for three days.

- Month (name or its first three letters), day, hour (24-hour convention), year (fully specified year or last two digits). This example notifies the backup server that its main server went down at 12 o’clock midnight on October 31, 2002:

```
nrcmd> failover-pair dhcp2.example.com. setPartnerDown -3d
nrcmd> failover-pair dhcp2.example.com. setPartnerDown Oct 31 00:00:00 2002
```

Note  Wherever you specify a date and time in the CLI, enter the time that is local to the `nrcmd` process. If the server is running in a different time zone than this process, disregard the time zone where the server is running and use local time instead.

### Setting DHCP Request and Response Packet Buffers

DHCP failover allows a limited number of binding updates to be outstanding (set through the (expert mode) `max-unacked-bndupd-failover-pair` attribute). The default value is 100 but can be tuned to the capacity of the server. The server allocates additional request buffers to accommodate failover (as it must have these resources available for failover).

### Setting Load Balancing

In normal failover mode, the main DHCP server bears most of the burden of servicing clients when the failover partners are in NORMAL communication mode. The main server not only services all new client requests, but has to handle renewal and rebinding requests and expired leases from the backup partner. To distribute the load more evenly between the two servers in a simple failover configuration scenario, Cisco Prime IP Express introduced the load balancing feature (based on RFC 3074).

Failover load balancing allows both servers to actively service clients and determine which unique clients each will serve without running the risk of both servicing the same ones. Failover load balancing applies only while the servers are in NORMAL mode; in other states, both servers can respond to clients. According to RFC 3074, the servers calculate a hash value for each request that the server receives, based on the client identifier option value or hardware address. The request is serviced if the hash value is assigned to that server.

With failover load balancing enabled, the servers split the client load evenly. The main partner processes 50% of the hash values and the backup partner the other 50%.

While the failover partners periodically balance the available leases on the backup server or do so shortly after a scope or prefix is detected to be out of leases, enabling the `rebalancing-delta-pct` attribute (Expert mode) on the main server to set the percentage difference between the desired and actual available leases on the backup server that will trigger a rebalancing on the scope or prefix.

Each partner responds to all clients whenever a partner is not in NORMAL mode. Each partner responds only to the broadcast DHCPDISCOVER or SOLICIT messages from clients that are in their assigned hash values.

For broadcast DHCPREQUEST or REBIND messages, the server responds only if it is the targeted one (based on the server identifier option); so, if the targeted server is the main server and it is down, the backup does not service the client (unless you release the lease). Broadcast BOOTP, DHCPINFORM, and INFORMATION-REQUEST requests are also load-balanced.
Recovering from a DHCP Failover

During normal operation, the failover partners undergo transition between states. If one of the failover server fails, then the partner takes over offering and renewing leases, using its private pool. When the main server becomes operational again, it re-integrates with its partner without administrative intervention.

The following sections describe how to confirm a DHCP failover, monitor DHCP failover event, what happens when servers enter various states, and how the servers integrate.

Confirming Failover

To confirm the failover:

**Step 1**
Ping from one server to the other to verify TCP/IP connectivity. Make sure that routers are configured to forward clients to both servers.

**Step 2**
Check that the server is in NORMAL mode by clicking the **Related Servers** icon on the Manage DHCP Server or List/Add DHCP Failover Pairs page, or use `dhcp getRelatedServers` in the CLI.

**Step 3**
After startup, have a client attempt to get a lease.

**Step 4**
Set the log settings on the main server to include at least `failover-detail`.

**Step 5**
Confirm that the `name_dhcp_1_log` log file (in `install-path/logs`) on the main server contains `DHCPBINDACK` or `DHCPBINDUPD` messages (for IPv4) and `BINDUPD6` or `BNDACK6` messages (for IPv6) from each server.

**Step 6**
Confirm that the `name_dhcp_1_log` log file on the backup server contains messages that the backup server is dropping requests because failover is in NORMAL state.

**Step 7**
Repeat Step 2.

Related Topics

- **State Transitions During Integration**, page 28-17
- **Configuring Failover Parameters Based on Your Scenario**, page 28-8
Monitoring DHCP Failover

When the main failover server goes down, the backup server moves to COMMUNICATIONS-INTERRUPTED state. The backup server cannot determine whether the main server is down or whether it cannot contact with the backup server. Depending upon the nature of outage you should monitor situation and follow the following steps:

1. Monitor both the failover servers and take action immediately when the main server goes down.
2. When the backup server first takes over, attempt to get the main server operational.
3. If you succeed in getting the main server operational within the MCLT, then nothing more needs be done.
4. If the main server is not operational until the MCLT has expired, then move the backup server to PARTNER DOWN state. On the backup server, use `failover-pair name setPartnerDown date` in the CLI.
5. When the main server is operational, ensure that it can contact the backup server before it is restarted.

For more information, see the “State Transitions During Integration” section on page 28-17.

Failover States and Transitions

During normal operation, the failover partners undergo transition between states. They stay in their current state until all the actions for the state transition are completed. If communication fails, they stay in their current state until the conditions for the next state are fulfilled. The states and their transitions are described in Table 28-2.

<table>
<thead>
<tr>
<th>Table 28-2</th>
<th>Failover States and Transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Server Action</td>
</tr>
<tr>
<td>STARTUP</td>
<td>Tries to contact its partner to learn its state, then transitions to another state after a short time, typically seconds.</td>
</tr>
<tr>
<td>NORMAL</td>
<td>Can communicate with its partner. The main and backup servers act differently in this state:</td>
</tr>
<tr>
<td></td>
<td>• The main server responds to all client requests using its pool. If its partner requests a backup pool, the main server provides it.</td>
</tr>
<tr>
<td></td>
<td>• The backup server only responds to renewal and rebinding requests. It requests a backup pool from the main server.</td>
</tr>
</tbody>
</table>
Recovering from a DHCP Failover

Table 28-2  Failover States and Transitions (continued)

<table>
<thead>
<tr>
<th>State</th>
<th>Server Action</th>
</tr>
</thead>
</table>
| COMMUNICATIONS-INTERRUPTED | Cannot communicate with its partner, whether it or the communication with it is down. The servers cycle between this state and NORMAL state as the connection fails and recovers, or as they cycle between operational and nonoperational. During this time, the servers cannot give duplicate addresses. During this state, you usually do not need to intervene and move a server into the PARTNER-DOWN state. However, this is not practical in some cases. A server running in this state is not using the available pool efficiently. This can restrict the time a server can effectively service clients. A server is restricted in COMMUNICATIONS-INTERRUPTED state:  
  - It cannot reallocate an expired address to another client.  
  - It cannot offer a lease or renewal beyond the maximum client lead time (MCLT) longer than the current lease time. The MCLT is a small additional time added that controls how much the client lease expiration is ahead of what the backup server thinks it is.  
  - A backup server can run out of addresses to give new clients, because it normally has only a small pool, while the main server has most of them. The server is limited only by the number of addresses allocated to it and the arrival rate of new clients. With a high new client arrival or turnover rate, you may need to move the server into PARTNER-DOWN state more quickly. |
| PARTNER-DOWN          | Acts as if it were the only operating server, based on one of these facts:  
  - The partner notified it during its shutdown.  
  - The administrator put the server into PARTNER-DOWN state.  
  - The safe period expired and the partner automatically went into this state. In this state, the server ignores that the other server might still operate and could service a different set of clients. It can control all its addresses, offer leases and extensions, and reallocate addresses. The same restrictions to servers in COMMUNICATIONS-INTERRUPTED state do not apply. Either server can be in this state, but only one should be in it at a time so that the servers do not issue duplicate addresses and can properly resynchronize later on. Until then, an address is in a pending-available state. |
| POTENTIAL-CONFLICT    | Might be in a situation that does not guarantee automatic reintegration, and is trying to reintegrate with its partner. The server might determine that two clients (who might not be operating) were offered and accepted the same address, and tries to resolve this conflict. |
| RECOVER               | Has no data in its stable storage, or is trying to reintegrate after recovering from PARTNER-DOWN state, from which it tries to refresh its stable storage. A main server in this state does not immediately start serving leases again. Because of this, do not reload a server in this state. |
| RECOVER-DONE          | Can transition from RECOVER or PARTNER-DOWN state, or from COMMUNICATIONS-INTERRUPTED into NORMAL state. |
Table 28-2  Failover States and Transitions (continued)

<table>
<thead>
<tr>
<th>State</th>
<th>Server Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAUSED</td>
<td>Can inform its partner that it will be out of service for a short time. The</td>
</tr>
<tr>
<td></td>
<td>partner then transitions to COMMUNICATIONS-INTERRUPTED state and begins</td>
</tr>
<tr>
<td></td>
<td>servicing clients.</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>Can inform its partner that it will be out of service for a long time. The</td>
</tr>
<tr>
<td></td>
<td>partner then transitions to PARTNER-DOWN state to take over completely.</td>
</tr>
</tbody>
</table>

State Transitions During Integration

During normal operation, the failover partners transition between states. They stay in their current state until all the actions for the state transition are completed, and if communication fails, until the conditions for the next state are fulfilled. Table 28-3 describes what happens when servers enter various states and how they initially integrate and later reintegrate with each other under certain conditions.

Table 28-3  Failover State Transitions and Integration Processes

<table>
<thead>
<tr>
<th>Integration</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Into NORMAL state, the first time the backup server contacts the main server</td>
<td>1. The newly configured backup server contacts the main server, which starts in PARTNER-DOWN state.</td>
</tr>
<tr>
<td></td>
<td>2. Because the backup server is a new partner, it goes into RECOVER state and sends a Binding Request message to the main server.</td>
</tr>
<tr>
<td></td>
<td>3. The main server replies with Binding Update messages that include the leases in its lease state database.</td>
</tr>
<tr>
<td></td>
<td>4. After the backup server acknowledges these messages, the main server responds with a Binding Complete message.</td>
</tr>
<tr>
<td></td>
<td>5. The backup server goes into RECOVER-DONE state.</td>
</tr>
<tr>
<td></td>
<td>6. Both servers go into NORMAL state.</td>
</tr>
<tr>
<td></td>
<td>7. The backup server sends Pool Request messages.</td>
</tr>
<tr>
<td></td>
<td>8. The main server responds with the leases to allocate to the backup server based on the backup-pct configured.</td>
</tr>
</tbody>
</table>

After COMMUNICATIONS-INTERRUPTED state

<table>
<thead>
<tr>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. When a server comes back up and connects with a partner in this</td>
</tr>
<tr>
<td>state, the returning server moves into the same state and then</td>
</tr>
<tr>
<td>immediately into NORMAL state.</td>
</tr>
<tr>
<td>2. The partner also moves into NORMAL state.</td>
</tr>
</tbody>
</table>
Chapter 28      Managing DHCP Failover

Recovering from a DHCP Failover

After PARTNER-DOWN state

When a server comes back up and connects with a partner in this state, the server compares the time it went down with the time the partner went into this state.

- If the server finds that it went down and the partner subsequently went into this state:
  a. The returning server moves into RECOVER state and sends an Update Request message to the partner.
  b. The partner returns all the binding data it was unable to send earlier and follows up with an Update Done message.
  c. The returning server moves into RECOVER-DONE state.
  d. Both servers move into NORMAL state.

- If the returning server finds that it was still operating when the partner went into PARTNER-DOWN state:
  a. The server goes into POTENTIAL-CONFLICT state, which also causes the partner to go into this state.
  b. The main server sends an update request to the backup server.
  c. The backup server responds with all unacknowledged updates to the main server and finishes off with an Update Done message.
  d. The main server moves into NORMAL state.
  e. The backup server sends the main server an Update Request message requesting all unacknowledged updates.
  f. The main server sends these updates and finishes off with an Update Done message.
  g. The backup server goes into NORMAL state.

After the server loses its lease state database

A returning server usually retains its lease state database. However, it can also lose it because of a catastrophic failure or intentional removal.

1. When a server with a missing lease database returns with a partner that is in PARTNER-DOWN or COMMUNICATIONS-INTERRUPTED state, the server determines whether the partner ever communicated with it. If not, it assumes it has lost its database, moves into RECOVER state, and sends an Update Request All message to its partner.

2. The partner responds with binding data about every lease in its database and follows up with an Update Done message.

3. The returning server waits the maximum client lead time (MCLT) period, typically one hour, and moves into RECOVER-DONE state. For details on the MCLT, see the “Setting the Maximum Client Lead Time” section on page 28-10.

4. Both servers then move into NORMAL state.

Table 28-3  Failover State Transitions and Integration Processes (continued)

<table>
<thead>
<tr>
<th>Integration</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>After PARTNER-DOWN state</td>
<td>When a server comes back up and connects with a partner in this state, the server compares</td>
</tr>
<tr>
<td></td>
<td>the time it went down with the time the partner went into this state.</td>
</tr>
<tr>
<td></td>
<td>- If the server finds that it went down and the partner subsequently went into this state:</td>
</tr>
<tr>
<td></td>
<td>a. The returning server moves into RECOVER state and sends an Update Request message to</td>
</tr>
<tr>
<td></td>
<td>the partner.</td>
</tr>
<tr>
<td></td>
<td>b. The partner returns all the binding data it was unable to send earlier and follows up</td>
</tr>
<tr>
<td></td>
<td>with an Update Done message.</td>
</tr>
<tr>
<td></td>
<td>c. The returning server moves into RECOVER-DONE state.</td>
</tr>
<tr>
<td></td>
<td>d. Both servers move into NORMAL state.</td>
</tr>
<tr>
<td></td>
<td>- If the returning server finds that it was still operating when the partner went into PARTNER-DOWN state:</td>
</tr>
<tr>
<td></td>
<td>a. The server goes into POTENTIAL-CONFLICT state, which also causes the partner to go into</td>
</tr>
<tr>
<td></td>
<td>this state.</td>
</tr>
<tr>
<td></td>
<td>b. The main server sends an update request to the backup server.</td>
</tr>
<tr>
<td></td>
<td>c. The backup server responds with all unacknowledged updates to the main server and</td>
</tr>
<tr>
<td></td>
<td>finishes off with an Update Done message.</td>
</tr>
<tr>
<td></td>
<td>d. The main server moves into NORMAL state.</td>
</tr>
<tr>
<td></td>
<td>e. The backup server sends the main server an Update Request message requesting all</td>
</tr>
<tr>
<td></td>
<td>unacknowledged updates.</td>
</tr>
<tr>
<td></td>
<td>f. The main server sends these updates and finishes off with an Update Done message.</td>
</tr>
<tr>
<td></td>
<td>g. The backup server goes into NORMAL state.</td>
</tr>
<tr>
<td>After the server loses its lease state database</td>
<td>A returning server usually retains its lease state database. However, it can also lose it because of a catastrophic failure or intentional removal.</td>
</tr>
</tbody>
</table>

1. When a server with a missing lease database returns with a partner that is in PARTNER-DOWN or COMMUNICATIONS-INTERRUPTED state, the server determines whether the partner ever communicated with it. If not, it assumes it has lost its database, moves into RECOVER state, and sends an Update Request All message to its partner.

2. The partner responds with binding data about every lease in its database and follows up with an Update Done message.

3. The returning server waits the maximum client lead time (MCLT) period, typically one hour, and moves into RECOVER-DONE state. For details on the MCLT, see the “Setting the Maximum Client Lead Time” section on page 28-10.

4. Both servers then move into NORMAL state.
Chapter 28      Managing DHCP Failover

Table 28-3 Failover State Transitions and Integration Processes (continued)

<table>
<thead>
<tr>
<th>Integration</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>After a lease state database backup restoration</td>
<td>When a returning server has its lease state database restored from backup, and if it reconnects with its partner without additional data, it only requests lease binding data that it has not yet seen. This data may be different from what it expects.</td>
</tr>
<tr>
<td></td>
<td>1. In this case, you must configure the returning server with the failover-recover attribute set to the time the backup occurred.</td>
</tr>
<tr>
<td></td>
<td>2. The server moves into RECOVER state and requests all its partner data. The server waits the MCLT period, typically one hour, from when the backup occurred and goes into RECOVER-DONE state. For details on the MCLT, see the “Setting the Maximum Client Lead Time” section on page 28-10.</td>
</tr>
<tr>
<td></td>
<td>3. Once the server returns to NORMAL state, you must unset its failover-recover attribute, or set it to zero.</td>
</tr>
<tr>
<td></td>
<td>nrcmd&gt; dhcp set failover-recover=0</td>
</tr>
</tbody>
</table>

| After the operational server had failover disabled | If the operating server had failover enabled, disabled, and subsequently reenabled, you must use special considerations when bringing a newly configured backup server into play. The backup server must have no lease state data and must have the failover-recover attribute set to the current time minus the MCLT interval, typically one hour. For details on the MCLT, see the “Setting the Maximum Client Lead Time” section on page 28-10. |
|                                                    | 1. The backup server then knows to request all the lease state data from the main server. Unlike what is described in “After the server loses its lease state database” section of this table, the backup server cannot request this data automatically because it has no record of having ever communicated with the main server. |
|                                                    | 2. After reconnecting, the backup server goes into RECOVER state, requests all the main server lease data, and goes into RECOVER-DONE state. |
|                                                    | 3. Both servers go into NORMAL state. At this point, you must unset the backup server failover-recover attribute, or set it to zero. |
|                                                    | nrcmd> dhcp set failover-recover=0 |

Setting Advanced Failover Attributes

The advanced failover properties that are important to set are the following:

- Setting backup allocation boundaries (see the “Setting Backup Allocation Boundaries” section on page 28-20)
- DHCPLEASEQUERY and failover (see the “DHCPLEASEQUERY and Failover” section on page 28-20)
Setting Backup Allocation Boundaries

You can be more specific as to which addresses to allocate to the backup server by using the `failover-backup-allocation-boundary` attribute on the scope. The IP address set as this value is the upper boundary of addresses from which to allocate addresses to a backup server. Only addresses below this boundary are allocated to the backup. If there are no addresses available below this boundary, then the addresses above it, if any, are allocated to the backup. The actual allocation works down from this address, while the normal allocation for DHCP clients works up from the lowest address in the scope.

If you set `failover-backup-allocation-boundary` for the scope, you must also enable the `allocate-first-available` attribute. If `failover-backup-allocation-boundary` is unset or set to zero, then the boundary used is halfway between the first and last addresses in the scope ranges. If there are no available addresses below this boundary, then the first available address is used.

DHCPLEASEQUERY and Failover

To accommodate DHCPLEASEQUERY messages sent to a DHCP failover backup server when the master server is down, the master server must communicate the `relay-agent-info` (82) option values to its partner server. To accomplish this, the master server uses DHCP failover update messages.

Maintaining Failover Server Pair

This section describes how to maintain failover server pair and perform the following administrative tasks:

- Changing failover pair server addresses (see the “Changing Failover Pair Server Addresses” section on page 28-20)
- Restarting the failover servers (see the “Restarting the Failover Servers” section on page 28-21)

Changing Failover Pair Server Addresses

If you need to change the name of a failover pair or an address associated with a cluster involved in a failover relationship, you can ensure that the failover state information is preserved. The failover partners should not enter recover state for the MCLT period. To change the name of a failover pair, you must remove the old object and add a new object.

If the cluster is configured for simple failover (only one failover pair exists), you can remove and add the renamed failover pair and change addresses, provided the failover role (main or backup) is not changed.

If there are multiple failover pairs, change either the name or the address, not both at the same time. Reload the DHCP server and allow the failover partners to return to normal state between name and address changes.

**Note**

If a cluster role in a failover relationship is changed (main to backup or backup to main), any existing state information for that relationship is discarded.
Restoring the Failover Servers

For any failover synchronization to take effect, you must first connect to, and restart, both the main and backup failover servers.

**Step 1** On the List/Add DHCP Failover Pairs page, select the failover pair on the Failover pane.

**Step 2** On the Manage Failover Servers tab for the main server, select the server you want to restart.

**Step 3** Choose Restart Server from the Quick View menu.

Related Topics

Confirming Failover, page 28-14

Restoring a Standalone DHCP Failover Server - Tutorial

This section describes how to recreate a DHCP failover relationship between a main and backup server where a backup server was put in standalone mode. This situation does not come up very often.

An administrator may have to take the main failover server offline because of a hardware failure or manual shutdown of Cisco Prime IP Express. The failover relationship with the main server will then be turned off, and the backup server will be pressed into service, temporarily, as a standalone DHCP server. Unfortunately, restoring the previous failover relationship from this condition can be hazardous to the lease state data.

According to the DHCP Failover protocol, if either of the partners maintained in a failover relationship fails, recovery is assured because the partners resynchronize. Even with a failed backup server, putting the main server in standalone mode would not be overly complicated to recover to failover mode.

However, restoring a standalone DHCP server to backup is not straightforward.

1. The standalone server assumes the role of the main server.
2. The original main server becomes the backup server.
3. The partners then synchronize.
4. Failover relationship to be intentionally broken to reverse the server roles.
5. Partners to resynchronize in their original failover roles.

Related Topics

Background, page 28-22
Repair Procedure, page 28-22
Reversing the Failover Role on Backup Server, page 28-23
Starting with Server A Powered Off, page 28-23
Starting with Server A Powered On and DHCP Server Stopped, page 28-24
Starting with Server A Replaced, page 28-25
Transferring Current Lease State to Server A, page 28-26
Background

For the remainder of this section, the main DHCP failover server is identified as Server A (with a cluster object named cluster-A), and the backup server as Server B (with a cluster object named cluster-B). Server A is administratively or otherwise shut down or its Cisco Prime IP Express server agent gets stopped. At this point, Server B goes into the Communications-Interrupted mode.

The system administrator may then take one of the following approaches:

- **Continue running backup Server B in Communications-Interrupted mode**—The risk of running the backup server in this mode indefinitely is that it can exhaust the pool of typically 10% of the available addresses with which the backup server is allocated to service new clients.

- **Put Server B into Partner-Down mode without breaking the failover relationship**—One major caveat of giving the backup server full control of the address space, without suspending failover, is that the full transfer of the address space ownership does not occur until after the configured Maximum Client Lead Time (MCLT). The MCLT is an additional time period set on the main server, which controls the duration for which the client lease expiration is ahead of what the backup server detects it to be. The MCLT is typically 60 minutes. Until the MCLT expires, the available address pool of the backup server is limited to its allocated reserve.

- **Put Server B into Partner-Down mode and break the failover relationship**—This approach puts the backup server in standalone mode, and is the approach that the administrator chose in this scenario. The deciding factors were that the main server was expected to be offline for an extended period, and the number of new devices coming online was higher than anticipated. Because the low percentage of available addresses that the backup server could service would soon cause an outage for new devices, the administrator put Server B in standalone mode. The disadvantage of this approach is the care and effort required to preserve the original state of the network when restoring the partners to their original relationship.

The first two approaches have distinct advantages over the third. In most cases, the backup server is expected to have enough addresses to cover newly arrived clients until the MCLT expires. Pursuing the third approach can incur unnecessary administrative burden and risk.

Repair Procedure

The repair procedure is:

1. **Temporarily assign the backup Server B the role of the main failover server**—Reversing the failover partner roles effectively allows Server A to learn the current failover state from Server B.

2. **Migrate Server A and Server B back to their original failover roles**—The goal is for Server A to reacquire its original status as the main DHCP failover server.

The assumptions are:

- The Original main Server A is nonoperational and Cisco Prime IP Express is stopped.
- The Original backup Server B is operational.
- Failover between the partners is administratively disabled.
- Decision was made not to permanently reverse the failover roles of the two partners.
- Domain Name Services (DNS) is not running on either of the failover partners.

*Note*  
The IP addresses used as examples are for demonstration purposes only.
Reversing the Failover Role on Backup Server

The following steps restore failover by temporarily moving Server B into the main server mode.

On Server B (cluster-B):

**Step 1**
Ensure that failover is disabled. Modify the failover configuration, so that Server B becomes the main and Server A the backup:

```
nrcmd> failover-pair examplepair set failover=false
nrcmd> failover-pair examplepair set main=cluster-B backup=cluster-A
```

**Step 2**
Save the changes and reload the server:

```
nrcmd> save
nrcmd> dhcp reload
```

**Step 3**
Re-enable failover and reload the server again:

```
nrcmd> failover-pair examplepair set failover=true
nrcmd> dhcp reload
```

Server B is now the main failover server, ready for its partner to become operational again. Any further action that you take to prevent Server A from beginning to give out addresses in the meantime depends on its current state.

If Server A is:

- **Powered off**—See the “Starting with Server A Powered Off” section on page 28-23.
- **Powered on with the Cisco Prime IP Express DHCP not configured to start**—See the “Starting with Server A Powered On and DHCP Server Stopped” section on page 28-24.
- **Replaced by another machine**—See the “Starting with Server A Replaced” section on page 28-25.

Starting with Server A Powered Off

If Server A was powered off, you must power it on again to continue. The next steps ensure that Server A comes online while preventing IP address leakage.

On Server A (cluster-A):

**Step 1**
Before turning on the server, you must take steps to prevent it from communicating with clients. The best way to do this is to manually disconnect the network cable, then boot up the machine. You will require a local console to perform the next step. Other alternatives include reconfiguring the relay agents not to forward packets to the server or otherwise preventing DHCP traffic to be received on the machine (such as by installing a temporary filter for DHCP packets on a firewall).

**Note**
If it is not possible to prevent client traffic from reaching the server, it may provide erroneous information to clients that do attempt to communicate with it, until the DHCP server can be stopped. Therefore, you must be ready to stop the DHCP server as soon as possible after turning the server on, as described in the next steps, to reduce the number of clients that might be provided erroneous information, potentially resulting in duplicated leases.
Restoring a Standalone DHCP Failover Server - Tutorial

Step 2  Turn on the server.

Step 3  Stop the DHCP server as quickly as possible:
        `nrcmd> dhcp stop`

Step 4  Go to the “Starting with Server A Powered On and DHCP Server Stopped” section on page 28-24.

Starting with Server A Powered On and DHCP Server Stopped

Starting from a point where Server A is powered on, but the Cisco Prime IP Express DHCP server is stopped:

On Server A (cluster-A):

Step 1  Modify the failover configuration so that Server A becomes the backup server:
        `nrcmd> failover-pair examplepair set main=cluster-B backup=cluster-A`
Step 2  Stop Cisco Prime IP Express:
   • Linux—`/etc/init.d/nwreglocal stop`
   • Windows—`net stop nwreglocal`

Step 3  Examine the DHCP logs to confirm that the DHCP server is not running.

Step 4  Bring Server A back on the network. Reconnect the network cable, reconfigure the relay agents, or remove any firewall filter added in the previous section.

Step 5  Remove the lease state database and event store:
   • Linux
     
     ```
     rm /var/nwreg2/local/data/dhcpeventstore/*.*
     rm -r /var/nwreg2/local/data/dhcp/ndb/*.*
     ```
   
   • Windows
     
     ```
     cd install-path\local\data
     rmdir /s dhcpeventstore
     rmdir /s dhcp\ndb
     ```

Step 6  Start Cisco Prime IP Express:
   • Linux—`/etc/init.d/nwreglocal start`
   • Windows—`net start nwreglocal`

Step 7  Set the DHCP service to be enabled on reboot and start the DHCP server:
   ```
   nrcmd> dhcp enable start-on-reboot
   nrcmd> dhcp start
   ```

Step 8  Go to the “Transferring Current Lease State to Server A” section on page 28-26.

Starting with Server A Replaced

If Server A was decommissioned and replaced, you must install Cisco Prime IP Express and push the failover configuration from Server B to the new machine. Also, you must restore any customer configuration specific to Server A. After these steps, Cisco Prime IP Express will start but not give out addresses:

Step 1  On Server A (cluster-A), install Cisco Prime IP Express.

Step 2  Reconstruct the Cisco Prime IP Express operating environment by restoring the accompanying software, such as Cisco Broadband Access Center and its required DHCP extensions. Do not make any administrative changes to the configuration until after pushing the configuration to Server B.

Step 3  On Server B (cluster-B), by using the Cisco Prime IP Express web UI, push an exact failover configuration to Server A. This effectively makes Server A the backup partner.

Step 4  On Server A:
   a. If necessary, customize the Cisco Prime IP Express configuration as required for the operating environment, which might include making administrative changes.
   b. Reload the DHCP server:

   ```
   nrcmd> dhcp reload
   ```
Transferring Current Lease State to Server A

- At this point, the failover partnership reestablishes itself, both servers will resynchronize their states.
- Server A becomes operational as the backup server.
- The operation will pause for the MCLT period (of one hour) and both partners resume their failover operations in normal communication mode.

**Note**
Do not proceed to the “Repairing Partners to Their Original Roles” section on page 28-26 until both partners synchronize and report normal communication.

Repairing Partners to Their Original Roles

Assume that both partners are fully synchronized and report normal communication. To ensure that the failover partners can assume their original roles, you should:

**Step 1**
On **Server A** (cluster-A), stop the DHCP server:
```
nrcmd> dhcp stop
```

**Step 2**
On **Server B** (cluster-B), stop the DHCP server:
```
nrcmd> dhcp stop
```

**Step 3**
On **Server A**:

a. Disable failover, then make Server A the main server and Server B the backup:
```
nrcmd> failover-pair examplepair set failover=false
nrcmd> failover-pair examplepair set main=cluster-A backup=cluster-B
```

b. Save the changes and reload DHCP:
```
nrcmd> save
nrcmd> dhcp reload
```

c. Ensure that the configuration is in place and currently running. At this point, Server A is the sole operational DHCP server with 100% of the address pool.

d. Re-enable failover:
```
nrcmd> failover-pair examplepair set failover=true
```

e. Reload DHCP and double-check the configuration changes:
```
nrcmd> dhcp reload
```

Server A is now the failover main server awaiting Server B to become operational.

**Step 4**
On **Server B**:

a. Make Server A the main server and Server B the backup, then enable failover:
Changing Failover Server Roles

Caution

Be careful when you change the role of a failover server. Remember that all address states in a DHCPv4 scope or DHCPv6 prefix are lost from a server if it is ever reloaded without that scope or prefix in its configuration.

Related Topics

- Establishing Failover Using Standalone Server as Main, page 28-27
- Replacing Servers Having Defective Storage, page 28-28
- Removing Backup Servers and Halting Failover Operation, page 28-29
- Adding Main Servers to Existing Backup Servers, page 28-29
- Configuring Failover on Multiple Interface Hosts, page 28-29

Establishing Failover Using Standalone Server as Main

You can update an existing installation and increase the availability of the DHCP service it offers. You can use this procedure only if the standalone server never participated in failover.

Step 1

Install Cisco Prime IP Express on the machine that is to be the backup server. Note the IP address of the backup server.

Step 2

Configure the cluster. Enable failover on the standalone server, configure it to be the main server and recently installed as the backup.

nrcmd> failover-pair examplepair set main=cluster-A backup=cluster-B
nrcmd> failover-pair examplepair set failover=true

b. Save the new configuration, but do not reload the server:

nrcmd> save

c. Restart the DHCP server on Server B:

nrcmd> dhcp reload

At this point, the failover partnership reestablishes itself in its original roles, both servers will resynchronize their states, and Server B becomes operational as the backup server. The operation will pause for the MCLT period (of one hour) and both partners resume their failover operations in normal communication mode.

Step 5

On Server A and Server B:

a. Validate whether both partners are in normal failover state:

nrcmd> dhcp getRelatedservers

b. Run a report and ensure that the results match on both partners, allowing a bit of skew for the difference in running times between the partners.
Changing Failover Server Roles

To configure the cluster, use:

```
cluster name create IP address scp-port=value admin=value password=value
```

For example:

```
cluster backup create 10.65.201.23 scp-port=1234 admin=admin password=changeme
```

**Step 3** Reload the main server. It should go into PARTNER-DOWN state. It cannot locate the backup server, because it is not yet configured. There should be no change in main server operation at this point.

**Step 4** To sync the configuration use failover synchronization and do a exact sync from Main to Backup.

**Step 5** Reconfigure all operational BOOTP relays to forward broadcast packets to the main server and backup server.

**Step 6** Reload the backup server.

---

After you complete these steps:

1. The backup server detects the main server and moves into RECOVER state.
2. The backup server refreshes its stable storage with the main server lease data, and when complete, moves into RECOVER-DONE state.
3. The main server moves into NORMAL state.
4. The backup server moves into NORMAL state.
5. The backup server sends a pool request to get its pool of address.
6. After allocating these addresses, the main server allocates the IP address to backup based on backup percentage.

---

Replacing Servers Having Defective Storage

If a failover server loses its stable storage (hard disk), you can replace the server and have it recover its state information from its partner.

**Step 1** Determine which server lost its stable storage.

**Step 2** Use `failover-pair name setPartnerDown` in the CLI to tell the other server that its partner is down. If you do not specify a time, the current time is used.

**Step 3** When the server is again operational, reinstall Cisco Prime IP Express.

**Step 4** Sync the server configuration from its partner configuration using failover synchronization. However, do not recover any lease databases from an earlier backup or the partner system.

**Step 5** Reload the replacement server.

---

After you complete these steps:

1. The recovered server moves into RECOVER state.
2. Its partner sends it all its data.
3. The server moves into RECOVER-DONE state when it reaches its maximum client lead time (and any time set for `failover-recover`).
4. Its partner moves into NORMAL state.
5. The recovered server moves into NORMAL state. It can request addresses, but can allocate few new ones, because its partner already sent it all its previously allocated addresses.

Removing Backup Servers and Halting Failover Operation

Sometimes you might need to remove the backup server and halt all failover operations.

**Step 1** On the backup server, remove all the scopes or prefixes that were designated as a backup to the main server.

**Step 2** On the main server, remove the failover capability from those scopes or prefixes that were main for the backup server, or disable failover server-wide if that is how it was configured.

**Step 3** Reload both servers.

Adding Main Servers to Existing Backup Servers

You can use an existing backup server for a main server.

**Step 1** Sync the main server scopes, policies, and other configurations on the backup server using failover synchronization.

**Step 2** Configure the main server to enable failover and point to the backup server.

**Step 3** Configure the backup server to enable failover for the new scopes that point to the new main server.

**Step 4** Reload both servers. Cisco Prime IP Express performs the same steps as those described in the “Establishing Failover Using Standalone Server as Main” section on page 28-27.

Configuring Failover on Multiple Interface Hosts

If you plan to use failover on a server host with multiple interfaces, you must explicitly configure the local server name or address. This requires an additional command. For example, if you have a host with two interfaces, server A and server B, and you want to make server A the a main failover server, you must define server A as the failover-main-server before you set the backup server name (external server B). If you do not do this, failover might not initialize correctly and tries to use the wrong interface.

Set the DHCP server properties `failover-main-server` and `failover-backup-server`.

With multiple interfaces on one host, you must specify a hostname that points to only one address or a record. You cannot set up your servers for round-robin support.

Troubleshooting Failover

This section describes how to avoid failover configuration mistakes, monitor failover operations, and detect and handle network problems.
Related Topics

Monitoring Failover Operations, page 28-30
Detecting and Handling Network Failures, page 28-30

Monitoring Failover Operations

You can examine the DHCP server log files on both partner servers to verify your failover configuration. You can make a few important log and debug settings to troubleshoot failover. Set the DHCP log settings to failover-detail to track the number and details of failover messages logged. To ensure that previous messages do not get overwritten, add the failover-detail attribute to the end of the list. Use the no-failover-conflict attribute to inhibit logging server failover conflicts, or the no-failover-activity attribute to inhibit logging normal server failover activity. Then, reload the server.

You can also isolate misconfigurations more easily by clicking the Related Servers icon on the Manage DHCP Server or List/Add DHCP Failover Pairs page, or by using dhcp getRelatedServers in the CLI.

Detecting and Handling Network Failures

Table 28-4 describes some symptoms, causes, and solutions for failover problems.

<table>
<thead>
<tr>
<th>Symptom</th>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>New clients cannot get addresses</td>
<td>A backup server is in COMMUNICATIONS-INTERRUPTED state with too few addresses.</td>
<td>Increase the backup percentage on the main server.</td>
</tr>
<tr>
<td>Error messages about mismatched scopes</td>
<td>There are mismatched scope configurations between partners.</td>
<td>Reconfigure your servers.</td>
</tr>
<tr>
<td>Log messages about failure to communicate with partner</td>
<td>Server cannot communicate with its partner.</td>
<td>Check the status of the server.</td>
</tr>
</tbody>
</table>
| Main server fails. Some clients cannot renew or rebind leases. The leases expire even when the backup server is up and possibly processing some client requests. | Some BOOTP relay agent (ip-helper) was not configured to point at both servers; see the “Configuring BOOTP Relays” section on page 28-32. | • Reconfigure BOOTP relays to point at both main and backup server.  
• Run a fire drill test—Take the main server down for a day or so and see if your user community can get and renew leases. |
| SNMP trap: other server not responding | Server cannot communicate with its partner. | Check the status of the server. |
Supporting BOOTP Clients in Failover

You can configure scopes to support two types of BOOTP clients—static and dynamic.

Related Topics

Static BOOTP, page 28-31
Dynamic BOOTP, page 28-31
Configuring BOOTP Relays, page 28-32

Static BOOTP

You can support static BOOTP clients using DHCP reservations. When you enable failover, remember to configure both the main and backup servers with identical reservations.

Dynamic BOOTP

You can enable dynamic BOOTP clients by enabling the `dynamic-bootp` attribute on a scope. When using failover, however, there are additional restrictions on address usage in such scopes, because BOOTP clients get permanent addresses and leases that never expire.

When a server whose scope does not have the `dynamic-bootp` option enabled goes to PARTNER-DOWN state, it can allocate any available (unassigned) address from that scope, whether or not it was initially available to any partner. However, when the `dynamic-bootp` option is set, each partner can only allocate its own addresses. Consequently, scopes that enable the `dynamic-bootp` option require more addresses to support failover.

When using dynamic BOOTP:

- Segregate dynamic BOOTP clients to a single scope. Disable DHCP clients from using that scope by disabling the `dhcp` attribute on the scope.
- Set the `dynamic-bootp-backup-pct` failover pair attribute to allocate a greater percentage of addresses to the backup server for this scope, as much as 50 percent higher than a regular backup percentage.

#### Table 28-4 Detecting and Handling Failures (continued)

<table>
<thead>
<tr>
<th>Symptom</th>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP trap: dhcp failover configuration mismatch</td>
<td>Mismatched scope configurations between partners</td>
<td>Reconfigure your servers.</td>
</tr>
<tr>
<td>Users complain that they cannot use services or system as expected</td>
<td>Mismatched policies and client-classes between partners</td>
<td>Reconfigure partners to have identical policies; possibly use LDAP for client registration if currently registering clients directly in partners.</td>
</tr>
</tbody>
</table>
Supporting BOOTP Clients in Failover

Configuring BOOTP Relays

The Cisco Prime IP Express failover protocol works with BOOTP relay (also called IP helper), a router capability that supports DHCP clients that are not locally connected to a server.

If you use BOOTP relay, ensure that the implementations point to both the main and backup servers. If they do not and the main server fails, clients are not serviced, because the backup server cannot see the required packets. If you cannot configure BOOTP relay to forward broadcast packets to two different servers, configure the router to forward the packets to a subnet-local broadcast address for a LAN segment, which could contain both the main and backup servers. Then, ensure that both the main and backup servers are on the same LAN segment.

BOOTP Backup Percentage

For scopes for which you enable dynamic BOOTP, use the `dynamic-bootp-backup-pct` attribute rather than the `backup-pct` attribute for the failover pair. The `dynamic-bootp-backup-pct` is the percentage of available addresses that the main server should send to the backup server for use with BOOTP clients.

The `dynamic-bootp-backup-pct` is distinct from the `backup-pct` attribute, because if you enable BOOTP on a scope, a server, even in PARTNER-DOWN state, never grants leases on addresses that are available to the other server. Cisco Prime IP Express does not grant leases because the partner might give them out using dynamic BOOTP, and you can never safely assume that they are available again.

| Note | You must define the dynamic BOOTP backup percentage on the main server. If you define it on the backup server, Cisco Prime IP Express ignores it (to enable duplicating configuration through scripts). If you do not define it, Cisco Prime IP Express uses the default `backup-pct` for the failover pair or scope. |

To properly support dynamic BOOTP while using the failover protocol, do this on every LAN segment in which you want BOOTP support:

- Create one scope for dynamic BOOTP
- Enable BOOTP and dynamic BOOTP
- Disable DHCP for that scope
Configuring DNS Update

The DNS Update protocol (RFC 2136) integrates DNS with DHCP. The latter two protocols are complementary; DHCP centralizes and automates IP address allocation, while DNS automatically records the association between assigned addresses and hostnames. When you use DHCP with DNS update, this configures a host automatically for network access whenever it attaches to the IP network. You can locate and reach the host using its unique DNS hostname. Mobile hosts, for example, can move freely without user or administrator intervention.

This chapter explains how to use DNS update with Cisco Prime IP Express servers, and its special relevance to Windows client systems.

Related Topics

DNS Update Process, page 29-1
Special DNS Update Considerations, page 29-2
DNS Update for DHCPv6, page 29-2
Creating DNS Update Configurations, page 29-5
Creating DNS Update Maps, page 29-7
Configuring Access Control Lists and Transaction Security, page 29-8
Configuring DNS Update Policies, page 29-15
Confirming Dynamic Records, page 29-19
Scavenging Dynamic Records, page 29-19
Troubleshooting DNS Update, page 29-21
Transitioning to DHCID RR for DHCPv4, page 29-21
Configuring DNS Update for Windows Clients, page 29-22
Configuring GSS-TSIG, page 29-34

DNS Update Process

To configure DNS updates, you must:

1. Create a DNS update configuration for a forward or reverse zone or both. See the “Creating DNS Update Configurations” section on page 29-5.

2. Use this DNS update configuration in either of two ways:
   - Specify the DNS update configuration on a named, embedded, or default DHCP policy. See the “Creating and Applying DHCP Policies” section on page 22-3.
   - Define a DNS update map to autoconfigure a single DNS update relationship between a Cisco Prime IP Express DHCP server or failover pair and a DNS server or High-Availability (HA) pair. Specify the update configuration in the DNS update map. See the “Creating DNS Update Maps” section on page 29-7.

C H A P T E R  2 9
Special DNS Update Considerations

Consider these two issues when configuring DNS updates:

- For security purposes, the Cisco Prime IP Express DNS update process does not modify or delete a name an administrator manually enters in the DNS database.
- If you enable DNS update for large deployments, and you are not using HA DNS (see Chapter 19, “Configuring High-Availability DNS Servers”), divide primary DNS and DHCP servers across multiple clusters. DNS update generates an additional load on the servers.

DNS Update for DHCPv6

Cisco Prime IP Express currently supports DHCPv6 DNS update over IPv4 only. For DHCPv6, DNS update applies to nontemporary stateful addresses only, not delegated prefixes.

DNS update for DHCPv6 involves AAAA and PTR RR mappings for leases. Cisco Prime IP Express supports server- or extension-synthesizing fully qualified domain names and the DHCPv6 client-fqdn option (39).

Because Cisco Prime IP Express is compliant with RFCs 4701, 4703, and 4704, it supports the DHCID resource record (RR). All RFC-4703-compliant updaters can generate DHCID RRs and result in data that is a hash of the client identifier (DUID) and the FQDN (per RFC 4701). Nevertheless, you can use AAAA and DHCID RRs in update policy rules.

DNS update processing for DHCPv6 is similar to that for DHCPv4 except that a single FQDN can have more than one lease, resulting in multiple AAAA and PTR RRs for a single client. The multiple AAAA RRs can be under the same name or a different name; however, PTR RRs are always under a different name, based on the lease address. RFC-4703-compliant updaters use the DHCID RR to avoid collisions among multiple clients.

Note

If the DNS server is down and the DHCP server cannot complete the DNS updates to remove RRs added for a DHCPv6 lease, the lease continues to exist in the AVAILABLE state. Only the same client reuses the lease.
Related Topics

DHCPv6 Upgrade Considerations, page 29-3
Generating Synthetic Names in DHCPv4 and DHCPv6, page 29-3
Determining Reverse Zones for DNS Updates, page 29-4
Using the Client FQDN, page 29-4

DHCPv6 Upgrade Considerations

If you use any policy configured prior to Cisco Prime IP Express that references a DNS update object for DHCPv6 processing (see the “DHCPv6 Policy Hierarchy” section on page 27-12), after the upgrade, the server begins queuing DNS updates to the specified DNS server or servers. This means that DNS updates might automatically (and unexpectedly) start for DHCPv6 leases.

Generating Synthetic Names in DHCPv4 and DHCPv6

If clients do not supply hostnames, DHCPv4 and DHCPv6 includes a synthetic name generator. The v6-synthetic-name-generator attribute for the DNS update configuration allows appending a generated name to the synthetic-name-stem based on the:

- Hash of the client DHCP Unique Identifier (DUID) value (the preset value).
- Raw client DUID value (as a hex string with no separators).
- CableLabs cablelabs-17 option device-id suboption value (as a hex string with no separators, or the hash of the client DUID if not found).
- CableLabs cablelabs-17 option cm-mac-address suboption value (as a hex string with no separators, or the hash of the client DUID if not found).

Caution

Some generation methods might cause privacy issues if the domain is accessible from the Internet.

The v4-synthetic-name-generator attribute for the DNS update configuration allows appending a generated name to the synthetic-name-stem based on the:

- address—Identifies the v4 address of client.
- client-id—Client-id or DUID given by DHCPv4 client in its request (Option 61).
- hashed-client-id—The hashed client-id which is a 13-character base 32 encoded string formed of the right part 64-bits of the SHA-256 hash appended with the forward zone name.

See the “Creating DNS Update Configurations” section on page 29-5 for how to create a DNS update configuration with synthetic name generation.

In the CLI, an example of this setting is:

```
nrcmd> dhcp-dns-update example-update-config set v6-synthetic-name-generator=hashed-duid
nrcmd> dhcp-dns-update example-update-config set v4-synthetic-name-generator=client-id
```
Determining Reverse Zones for DNS Updates

The DNS update configuration uses the prefix length value in the specified `reverse-zone-prefix-length` attribute to generate a reverse zone in the ip6.arpa domain. You do not need to specify the full reverse zone, because you can synthesize it by using the ip6.arpa domain. You set this attribute for the reverse DNS update configuration (see the “Creating DNS Update Configurations” section on page 29-5). Here are some rules for `reverse-zone-prefix-length`:

- Use a multiple of 4 for the value, because ip6.arpa zones are on 4-bit boundaries. If not a multiple of 4, the value is rounded up to the next multiple of 4.
- The maximum value is 124, because specifying 128 would create a zone name without any possible hostnames contained therein.
- A value of 0 means none of the bits are used for the zone name, hence ip6.arpa is used.
- If you omit the value from the DNS update configuration, the server uses the value from the prefix or, as a last resort, the prefix length derived from the `address` value of the prefix (see the “Configuring Prefixes” section on page 27-20).

Note that to synthesize the reverse zone name, the `synthesize-reverse-zone` attribute must remain enabled for the DHCP server. Thus, the order in which a reverse zone name is synthesized for DHCPv6 is:

1. Use the full `reverse-zone-name` in the reverse DNS update configuration.
2. Base it on the ip6.arpa zone from the `reverse-zone-prefix-length` in the reverse DNS update configuration.
3. Base it on the ip6.arpa zone from the `reverse-zone-prefix-length` in the prefix definition.
4. Base it on the ip6.arpa zone from the prefix length for the `address` in the prefix definition.

In the CLI, an example of setting the reverse zone prefix length is:

```
nrcmd> dhcp-dns-update example-update-config set reverse-zone-prefix-length=32
```

To create a reverse zone for a prefix in the web UI, the List/Add Prefixes page includes a Create Reverse Zone button for each prefix. (See the “Creating and Editing Prefixes” section on page 27-25.)

The CLI also provides the `prefix name createReverseZone [–range]` command to create a reverse zone for a prefix (from its address or range value). Delete the reverse zone by using `prefix name deleteReverseZone [–range].`

You can also create a reverse zone from a DHCPv4 subnet or DHCPv6 prefix by entering the subnet or prefix value when directly configuring the reverse zone. See the “Adding Primary Reverse Zones” section on page 14-11 for details.

Using the Client FQDN

The existing DHCP server `use-client-fqdn` attribute controls whether the server pays attention to the DHCPv6 client FQDN option in the request. The rules that the server uses to determine which name to return when multiple names exist for a client are in the following order of preference:

1. The server FQDN that uses the client requested FQDN if it is in use for any lease (even if not considered to be in DNS).
2. The FQDN with the longest valid lifetime considered to be in DNS.
3. The FQDN with the longest valid lifetime that is not yet considered to be in DNS.
Creating DNS Update Configurations

A DNS update configuration defines the DHCP server framework for DNS updates to a DNS server or HA DNS server pair. It determines if you want to generate forward or reverse zone DNS updates (or both). It optionally sets TSIG keys for the transaction, attributes to control the style of autogenerated hostnames, and the specific forward or reverse zone to be updated. You must specify a DNS update configuration for each unique server relationship.

For example, if all updates from the DHCP server are directed to a single DNS server, you can create a single DNS update configuration that is set on the server default policy. To assign each group of clients in a client-class to a corresponding forward zone, set the forward zone name for each in a more specific client-class policy.

Local Advanced and Regional Web UI

Step 1
From the Deploy menu, choose DNS Update Configs under the DNS Updates submenu to open the List/Add DNS Update Configurations page.

Step 2
Click the Add DNS Update Configs icon in the DNS Update Configs pane to open the Add DnsUpdateConfig dialog box.

Step 3
Enter a name for the update configuration in the Name attribute field.

Step 4
Click Add DnsUpdateConfig to add the DNS update configuration.

Step 5
Select the name of update configuration to open the Edit DNS Update Configuration page.

Step 6
Click the appropriate dynamic-dns setting under the Update Settings block:

- **update-none** — Do not update forward or reverse zones.
- **update-all** — Update forward and reverse zones (the default value).
- **update-fwd-only** — Update forward zones only.
- **update-reverse-only** — Update reverse zones only.

Step 7
Click the appropriate dns-client-identity setting under the Update Settings block:

- **txt** — The server uses TXT RR for DHCPv4 DNS updates and DHCID RR for DHCPv6 DNS updates.
- **dhcid** — The server uses DHCID RR for both DHCPv4 and DHCPv6 DNS updates.
- **transition-to-dhcid** — The server uses DHCID RR for new records in the DNS server and updates existing entries to use the DHCID RR when the next DNS update is done.
- **regress-to-txt** — The server uses the TXT RR for new entries in the DNS server and upgrades existing entries to use the TXT RR when the next DNS update is done.

*Note* The dns-client-identity attribute is also available as part of the DHCP server-wide settings which will be taken into consideration if the attribute of the individual DNS update config was not configured.

Step 8
Set the other attributes appropriately:

- If necessary, enable synthesize-name and set the synthetic-name-stem value.
You can set the stem of the default hostname to use if clients do not supply hostnames, by using `synthetic-name-stem`. For DHCPv4, enable the `synthesize-name` attribute to trigger the DHCP server to synthesize unique names for clients based on the value of the `synthetic-name-stem`. The resulting name is the name stem appended with the hyphenated IP address. For example, if you specify a `synthetic-name-stem` of `host` for address 192.168.50.1 in the `example.com` domain, and enable the `synthesize-name` attribute, the resulting hostname is `host-192-168-50-1.example.com`. The preset value for the synthetic name stem is `dhcp`.

The `synthetic-name-stem` must:
- Be a relative name without a trailing dot.
- Include alphanumeric values and hyphens (`-`) only. Space characters and underscores become hyphens and other characters are removed.
- Include no leading or trailing hyphen characters.
- Have DNS hostnames of no more than 63 characters per label and 255 characters in their entirety. The algorithm uses the configured forward zone name to determine the number of available characters for the hostname, and truncates the end of the last label if necessary.

For DHCPv6, see the “Generating Synthetic Names in DHCPv4 and DHCPv6” section on page 29-3.

b. Set `forward-zone-name` to the forward zone, if updating forward zones. Note that the policy `forward-zone-name` takes precedence over the one set in the DNS update configuration.

For DHCPv6, the server ignores the client and client-class policies when searching for a `forward-zone-name` value in the policy hierarchy. The search for a forward zone name begins with the prefix embedded policy.

c. For DHCPv4, set `reverse-zone-name` to the reverse (in-addr.arpa) zone to be updated with PTR and TXT records. If unset and the DHCP server `synthesize-reverse-zone` attribute is enabled, the server synthesizes a reverse zone name based on the address of each lease, scope subnet number, and DNS update configuration (or scope) `dns-host-bytes` attribute value.

The `dns-host-bytes` value controls the split between the host and zone parts of the reverse zone name. The value sets the number of bytes from the lease IP address to use for the hostname; the remaining bytes are used for the in-addr.arpa zone name. A value of 1 means use just one byte for the host part of the domain and the other three from the domain name (reversed). A value of 4 means use all four bytes for the host part of the address, thus using just the in-addr.arpa part of the domain. If unset, the server synthesizes an appropriate value based on the scope subnet size, or if the `reverse-zone-name` is defined, calculates the host bytes from this name.

The `one-a-rr-per-dns-name` controls the DHCPv4 DNS updates to allow either one or multiple A RRs per name. The introduction of the DUID support and DHCID RR, multi-connection clients will have multiple A RRs.

For DHCPv6, see the “Determining Reverse Zones for DNS Updates” section on page 29-4.

d. Set `server-addr` to the IP address of the primary DNS server for the forward zone (or reverse zone if updating reverse zones only).

e. Set `server-key` and `backup-server-key` if you are using a TSIG key to process all DNS updates (see the “Transaction Security” section on page 29-10).

f. Set `gss-tsig-config` if the Generic Security Service (GSS) method of the secure key exchange has to be used (see the “Configuring GSS-TSIG” section on page 29-34).

g. Set `backup-server-addr` to the IP address of the backup DNS server, if HA DNS is configured.
Creating DNS Update Maps

A DNS update map facilitates configuring DNS updates so that the update properties are synchronized between HA DNS server pairs or DHCP failover server pairs, based on an update configuration, so as to reduce redundant data entry. The update map applies to all the primary zones that the DNS pairs service, or all the scopes that the DHCP pairs service. You must specify a policy for the update map. To use this function, you must be an administrator assigned the server-management subrole of the dns-management or central-dns-management role, and the dhcp-management role (for update configurations).

Local and Regional Web UI

Step 1 From the Deploy menu, choose Update Maps under the DNS Updates submenu to open the List/Add DNS Update Maps page. The option is available if the server is configured with authoritative service.

Step 2 Click the Add DNS Update Map icon in the Update Maps pane to open the Add DNS Update Map dialog box.

Step 3 Enter a name for the update map in the Name field.

Step 4 Choose the The DNS server or HA pair associated with this configuration.

Step 5 Choose the DHCP server or DHCP failover pair associated with this configuration.

Step 6 Enter the DNS update configuration from the previous section in the dns-config field.

Step 7 Set the kind of policy selection you want for the dhcp-policy-selector attribute. The choices are:
Step 8 If using update ACLs (see the “Configuring Access Control Lists and Transaction Security” section on page 29-8) or DNS update policies (see the “Configuring DNS Update Policies” section on page 29-15), set either the `dns-update-acl` or `dns-update-policy-list` attribute. Either value can be one or more addresses separated by commas. The `dns-update-acl` takes precedence over the `dns-update-policy-list`.

If you omit both values, a simple update ACL is constructed whereby only the specified DHCP servers or failover pair can perform updates, along with any `server-key` value set in the update configuration specified for the `dns-config` attribute.

Step 9 Click Add DNS Update Map.

Step 10 At the regional level, you can also push update maps to the local clusters, or pull them from the replica database on the List/Add DNS Update Maps page.

CLI Commands

Specify the name, cluster of the DHCP and DNS servers (or DHCP failover or HA DNS server pair), and the DNS update configuration when you create the update map, using `dns-update-map name create dhcp-cluster dns-cluster dns-config`. For example:

```bash
nrcmd> dns-update-map example-update-map create Example-cluster Boston-cluster example-update-config
```

Set the `dhcp-policy-selector` attribute value to use-named-policy, use-client-class-embedded-policy, or use-scope-embedded-policy. If using the use-named-policy value, also set the `dhcp-named-policy` attribute value. For example:

```bash
nrcmd> dns-update-map example-update-map set dhcp-policy-selector=use-named-policy
nrcmd> dns-update-map example-update-map set dhcp-named-policy=example-policy
```

Configuring Access Control Lists and Transaction Security

ACLs are authorization lists, while transaction signatures (TSIG) is an authentication mechanism:

- ACLs enable the server to allow or disallow the request or action defined in a packet.
- TSIG ensures that DNS messages come from a trusted source and are not tampered with.

For each DNS query, update, or zone transfer that is to be secured, you must set up an ACL to provide permission control. TSIG processing is performed only on messages that contain TSIG information. A message that does not contain, or is stripped of, this information bypasses the authentication process.

For a totally secure solution, messages should be authorized by the same authentication key. For example, if the DHCP server is configured to use TSIG for DNS updates and the same TSIG key is included in the ACL for the zones to be updated, then any packet that does not contain TSIG information fails the authorization step. And if the DHCP server is configured to use GSS-TSIG, the TKEY negotiation happens involving the DHCP server and DNS server to create...
a security context which is used to create and verify the transaction signatures of the server messages. This secures the update transactions and ensures that messages are both authenticated and authorized before making zone changes.

ACLs and TSIG play a role in setting up DNS update policies for the server or zones, as described in the “Configuring DNS Update Policies” section on page 29-15.

**Related Topics**

- Access Control Lists, page 29-9
- Configuring Zones for Access Control Lists, page 29-10
- Transaction Security, page 29-10
- GSS-TSIG, page 29-13

**Access Control Lists**

You assign ACLs on the DNS Caching server or zone level. ACLs can include one or more of these elements:

- **IP address**—In dotted decimal notation; for example, 192.168.1.2.
- **Network address**—In dotted decimal and slash notation; for example, 192.168.0.0/24. In this example, only hosts on that network can update the DNS server.
- **Another ACL**—Must be predefined. You cannot delete an ACL that is embedded in another one until you remove the embedded relationship. You should not delete an ACL until all references to that ACL are deleted.
- **Transaction Signature (TSIG) key**—The value must be in the form **key value**, with the keyword **key** followed by the secret value. To accommodate space characters, the entire list must be enclosed in double quotes. For TSIG keys, see the “Transaction Security” section on page 29-10.

For Generic Security Service (GSS)-TSIG, the security context is created automatically as explained in the RFC 2930 and RFC 3645.

You assign each ACL a unique name. However, the following ACL names have special meanings and you cannot use them for regular ACL names:

- **any**—Anyone can perform a certain action
- **none**—No one can perform a certain action
- **localhost**—Any of the local host addresses can perform a certain action
- **localnets**—Any of the local networks can perform a certain action

Note the following:

- If an ACL is not configured, **any** is assumed.
- If an ACL is configured, at least one clause must allow traffic.
- The negation operator (!) disallows traffic for the object it precedes, but it does not intrinsically allow anything else unless you also explicitly specify it. For example, to disallow traffic for the IP address 192.168.50.0 only, use **!192.168.50.0, any**.
Local Advanced Web UI

From the Design menu, choose ACLs under the Security submenu to open the List/Add Access Control Lists page. Click the Add ACLs icon in the ACLs pane and enter an ACL name and match list and click Add ACL. Note that a key value pair should not be in quotes. At the regional level, you can additionally pull replica ACLs or push ACLs to local clusters.

CLI Commands

Use acl name create match-list, which takes a name and one or more ACL elements. The ACL list is comma-separated, with double quotes surrounding it if there is a space character. The CLI does not provide the pull/push function.

For example, the following commands create three ACLs. The first is a key with a value, the second is for a network, and the third points to the first ACL. Including an exclamation point (!) before a value negates that value, so that you can exclude it in a series of values:

```
nrcmd> acl sec-acl create "key h-a.h-b.example.com."
nrcmd> acl dyn-update-acl create "!192.168.2.13,192.168.2.0/24"
nrcmd> acl main-acl create sec-acl
``` 

Configuring Zones for Access Control Lists

To configure ACLs for the DNS server or zones, set up a DNS update policy, then define this update policy for the zone (see the “Configuring DNS Update Policies” section on page 29-15).

Transaction Security

Transaction Signature (TSIG) RRs enable the DNS server to authenticate each message that it receives, containing a TSIG. Communication between servers is not encrypted but it becomes authenticated, which allows validation of the authenticity of the data and the source of the packet.

When you configure the Cisco Prime IP Express DHCP server to use TSIG for DNS updates, the server appends a TSIG RR to the messages. Part of the TSIG record is a message authentication code.

When the DNS server receives a message, it looks for the TSIG record. If it finds one, it first verifies that the key name in it is one of the keys it recognizes. It then verifies that the time stamp in the update is reasonable (to help fight against traffic replay attacks). Finally, the server looks up the key shared secret that was sent in the packet and calculates its own authentication code. If the resulting calculated authentication code matches the one included in the packet, then the contents are considered to be authentic.

Supported TSIG algorithms will be GSS and HMAC-MD5. To use GSS-TSIG algorithm, please see Configuring GSS-TSIG, page 29-34 for further details.

Related Topics

Creating TSIG Keys, page 29-11
Generating Keys, page 29-11
Considerations for Managing Keys, page 29-12
Adding Supporting TSIG Attributes, page 29-12
Creating TSIG Keys

If you want to enable key authentication for Address-to-User Lookup (ATUL) support, you must also define a key identifier (id attribute value). See the “Setting DHCP Forwarding” section on page 24-24.

Local Advanced Web UI

From the Design menu, choose Keys under the Security submenu to open the List/Add Encryption Keys page.

For a description of the Algorithm, Security Type, Time Skew, Key ID, and Secret values, see Table 29-1 on page 29-11. See also the “Considerations for Managing Keys” section on page 29-12.

To edit a TSIG key, click its name on the List/Add Encryption Keys page to open the Edit Encryption Key page.

At the regional level, you can additionally pull replica keys, or push keys to local clusters.

CLI Commands

Use key name create secret. Provide a name for the key (in domain name format; for example, hosta-hostb-example.com.) and a minimum of the shared secret as a base-64 encoded string (see Table 29-1 on page 29-11 for a description of the optional time skew attribute). An example in the CLI would be:

```plaintext
nrcmd> key hosta-hostb-example.com. create secret-string
```

Generating Keys

It is recommended that you use the Cisco Prime IP Express cnr_keygen utility to generate TSIG keys so that you add them or import them using import keys.

Execute the cnr_keygen key generator utility from a DOS prompt, or Linux shell:

- On Windows, the utility is in the install-path\bin folder.
- On Linux, the utility is in the install-path/usrbin directory.

An example of its usage (on Linux) is:

```plaintext
> /opt/nwreg2/local/usrbin/cnr_keygen -n a.b.example.com. -a hmac-md5 -t TSIG -b 16 -s 300
   key "a.b.example.com." { algorithm hmac-md5; secret "xGVCsFZ0/6e0N97HGF50eg=="; # cnr-time-skew 300; # cnr-security-type TSIG;
};
```

The only required input is the key name. The options are described in Table 29-1.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n name</td>
<td>Key name. Required. The maximum length is 255 bytes.</td>
</tr>
<tr>
<td>-a hmac-md5</td>
<td>Algorithm. Optional. Only hmac-md5 is currently supported.</td>
</tr>
</tbody>
</table>
You can also redirect the output to a file if you use the right-arrow (>) or double-right-arrow (>>) indicators at the end of the command line. The > writes or overwrites a given file, while the >> appends to an existing file. For example:

```
> /opt/nwreg2/local/usrbin/cnr_keygen -n example.com > keyfile.txt
> /opt/nwreg2/local/usrbin/cnr_keygen -n example.com >> addtokeyfile.txt
```

You can then import the key file into Cisco Prime IP Express using the CLI to generate the keys in the file. The key import can generate as many keys as it finds in the import file. The path to the file should be fully qualified. For example:

```
nrcmd> import keys keydir/keyfile.txt
```

### Considerations for Managing Keys

If you generate your own keys, you must enter them as a base64-encoded string (See RFC 4648 for more information on base64 encoding). This means that the only characters allowed are those in the base64 alphabet and the equals sign (=) as pad character. Entering a nonbase64-encoded string results in an error message.

Here are some other suggestions:

- Do not add or modify keys using batch commands.
- Change shared secrets frequently; every two months is recommended. Note that Cisco Prime IP Express does not explicitly enforce this.
- The shared secret length should be at least as long as the keyed message digest (HMAC-MD5 is 16 bytes). Note that Cisco Prime IP Express does not explicitly enforce this and only checks that the shared secret is a valid base64-encoded string, but it is the policy recommended by RFC 2845.

### Adding Supporting TSIG Attributes

To add TSIG support for a DNS update configuration (see the “Creating DNS Update Configurations” section on page 29-5).

Set these attributes to use HMAC-MD5 security algorithm in TSIG:

- `server-key`
- `backup-server-key`
Set the below attribute to use GSS-TSIG security algorithm in TSIG:

- `gss-tsig-config`

**GSS-TSIG**

RFC 3645 proposed extending TSIG to allow the Generic Security Service (GSS) method of secure key exchange, eliminating the need for manually distributing keys to all GSS clients. It defines an algorithm to use with TSIG, which is based on the Generic Security Service Application Program Interface (GSS API), as specified in RFC2743.

GSS-TSIG provides the secure DDNS updates and secure Zone Transfers utilizing the Kerberos security mechanism.

Client and Server use GSS API calls to establish a limited lifetime security context for authentication, integrity and confidentiality. Establishing a security context involves the passing of opaque tokens between the client and server until the negotiation is complete. The TKEY resource record [RFC2930] is used as the vehicle to transfer tokens between client and server. Once the security context is established it is used to generate and verify signatures using GSS API calls. These signatures are exchanged by the Client and Server as a part of the TSIG records exchanged in DNS messages sent between the Client and Server, as described in [RFC2845].

Client and Server MUST be locally authenticated with Kerberos server before using this protocol. Generally the initial TGT(ticket to get ticket) ticket is available in cache through system logon or obtained using utility like kinit. DHCP/DNS Client will request Kerberos server for the service ticket using the principal name(DNS/<hostname>). Client provides the service ticket to prove authentication when interacting, securely, with DNS server. The service ticket will be encrypted by the Kerberos server using service key, which can be decrypted only by the application server using the same service key.

Please see the Configuring GSS-TSIG, page 29-34 for the configuration required on the DHCP Server and DNS Server.

---

**Note**

By default, CPIPE will support HMAC-MD5 based secure TSIG updates. To enable the GSS based secure updates, user has to disable-all HMAC-MD5 configuration in the DNS server.

**DHCP Server and Secondary DNS Server Configuration in Linux**

Configure the KDC Server information in `/etc/krb5.conf`. Use kinit utility to get the initial ticket from KDC.

**DHCP Server and Secondary DNS Server Configuration in Windows**

The server machine should be under the AD domain and not workgroup.

---

**Note**

To enable GSS-TSIG in DHCP server/Secondary DNS server, ensure that `gss-tsig-config` attribute is configured in DNS Update Config/Secondary Zone page respectively.

**Troubleshooting DHCP Server and Secondary DNS Server Configuration**

Client-related errors that can occur while getting the initial credentials:

- **CLOCK SKEW ERROR** - Ensure the Kerberos client and server and synchronized in time if not synchronize with ntp.
Creating GSS-TSIG Configuration

DNS/DHCP shall support a TKEY table. This will be a non-persistent hash table. The TKEY table shall not actually store TKEY RRs but the following data:

**key:**
- The GSS Client shall store the target server name as key name.
- The GSS Server shall store TKEY name (client unique global name) from TKEY RR.

**entry:**
- security context: For negotiated or negotiating keys
- expiration-time: time which the associated context should be expired.
- exchange-count: the number of exchanges done so far during key negotiation (i.e. number of TKEY queries received from this client). Only GSS Server maintains this count.
- state:
  - negotiating
  - established
  GSS client stores only established security context in TKEY table.
- Other required data such as TKEY name to be used by GSS Client.
- SID information: PAC information from Microsoft Clients.

Local Advanced Web UI

Choose **GSS-TSIG** from **Design > Security** to open the List/Add GSS-TSIG Configuration page.
GSS-TSIG attributes

- **tkey-max-exchanges** - Per recommendation from RFC 3645 to prevent endless looping, the DNS server shall impose a maximum number of TKEY exchanges (i.e. number TKEY queries received from a particular client) in the attempt to negotiate a particular key. This attribute shall specify this limit. A TKEY table record maintains the exchange-count. If exchange-count exceeds tkey-max-exchanges during key negotiation, the DNS server shall abort the key negotiation.

- **tkey-table-max-size** - This attribute bounds the size of the TKEY table.

- **tkey-table-purge-interval** - The time interval at which purging of expired keys from TKEY table should happen.

To edit a GSS-TSIG configuration, click its name on the List/Add GSS-TSIG Configuration page to open the Edit GSS-TSIG Configuration page.

At the regional level, you can additionally pull or push GSS-TSIG configuration to local clusters.

**CLI Commands**

Use `gss-tsig <name> create <attr=value...>`. Provide a name for the GSS-TSIG configuration object. For example:

```
nrcmd> gss-tsig gss create tkey-max-exchanges=6 tkey-table-max-size=500 tkey-table-purge-interval=90
```

**Configuring DNS Update Policies**

DNS update policies provide a mechanism for managing update authorization at the RR level. Using update policies, you can grant or deny DNS updates based on rules that are based on ACLs as well as RR names and types. ACLs are described in the “Access Control Lists” section on page 29-9.

**Related Topics**

- Compatibility with Cisco Prime IP Express Releases, page 29-15
- Creating and Editing Update Policies, page 29-16
- Defining and Applying Rules for Update Policies, page 29-16

**Compatibility with Cisco Prime IP Express Releases**

Cisco Prime IP Express releases used static RRs that administrators entered, but that DNS updates could not modify. This distinction between static and dynamic RRs no longer exists. RRs can now be marked as protected or unprotected (see the “Protecting Resource Record Sets” section on page 16-3).

Administrators creating or modifying RRs can now specify whether RRs should be protected. A DNS update cannot modify a protected RR set, even if an RR of the given type does not yet exist in the set.

**Note**

Previous releases allowed DNS updates only to A, TXT, PTR, CNAME and SRV records. This was changed to allow updates to all but SOA and NS records in unprotected name sets. To remain compatible with a previous release, use an update policy to limit RR updates.
Creating and Editing Update Policies

Creating an update policy initially involves creating a name for it.

Local Advanced Web UI

Step 1 From the Design menu, choose Update Policies under the Security submenu to open the List/Add DNS Update Policies page. The option is available if the server is configured with authoritative service.

Step 2 Click the Add Update Policies icon in the Update Policies pane to open the Add DNS Update Policy dialog box.

Step 3 Enter a name for the update policy.

Step 4 Click Add DNS Update Policy.

Step 5 Proceed to the “Defining and Applying Rules for Update Policies” section on page 29-16.

CLI Commands

Use `update-policy name create`; for example:

```
nrcmd> update-policy policy1 create
```

Defining and Applying Rules for Update Policies

DNS update policies are effective only if you define rules for each that grant or deny updates for certain RRs based on an ACL. If no rule is satisfied, the default (last implicit) rule is to deny all updates ("deny any wildcard * *").

Related Topics

Defining Rules for Named Update Policies, page 29-16
Applying Update Policies to Zones, page 29-18

Defining Rules for Named Update Policies

Defining rules for named update policies involves a series of Grant and Deny statements.

Local Advanced Web UI

Step 1 Create an update policy, as described in the “Creating and Editing Update Policies” section on page 29-16, or edit it.

Step 2 On the List/Add DNS Update Policies or Edit DNS Update Policy page:

a. Enter an optional value in the Index field.

b. Click Grant to grant the rule, or Deny to deny the rule.

c. Enter an access control list in the ACL List field.

d. Choose a keyword from the Keyword drop-down list.
e. Enter a value based on the keyword in the Value field. This can be a RR or subdomain name, or, if the wildcard keyword is used, it can contain wildcards (see Table 29-2).

<table>
<thead>
<tr>
<th>Wildcard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Matches zero or more characters. For example, the pattern <code>example*</code> matches all strings starting with <code>example</code>, including <code>example-</code>.</td>
</tr>
<tr>
<td>?</td>
<td>Matches a single character only. For example, the pattern <code>example?.com</code> matches <code>example1.com</code> and <code>example2.com</code>, but not <code>example.com</code>.</td>
</tr>
<tr>
<td>/[ ]/</td>
<td>Matches any characters in the (escaped) brackets; for example, <code>/[abc]/</code>. Each square bracket must be escaped using a slash (/). The characters can also be in a range; for example, <code>/[0–9]/</code> and <code>/[a–z]/</code>. If a pattern should include a hyphen, make the hyphen the first character; for example, <code>example/[–a–z]/</code>.</td>
</tr>
</tbody>
</table>

Table 29-2  Wildcard Values for Update Policy Rules

f. Enter one or more RR types, separated by commas, in the RR Types field, or use * for “all RRs.” You can use negated values, which are values prefixed by an exclamation point; for example, !PTR.

g. Click Save.

**Step 3**  At the regional level, you can also push update policies to the local clusters, or pull them from the replica database on the List/Add DNS Update Policies page.

**Step 4**  To edit an update policy, click the name of the update policy on the List/Add DNS Update Policies page to open the Edit DNS Update Policy page, make changes to the fields, then click Save.

**CLI Commands**

Create or edit an update policy (see the “Creating and Editing Update Policies” section on page 29-16, then use `update-policy name rules add rule`, with `rule` being the rule. (See Table 29-2 on page 29-17 for the rule wildcard values.) For example:

```
nrcmd> update-policy policy1 rules add "grant 192.168.50.101 name host1 A,TXT" 0
```

The rule is enclosed in quotes. To parse the rule syntax for the example:

- **grant**—Action that the server should take, either `grant` or `deny`.
- **192.168.50.101**—The ACL, in this case an IP address. The ACL can be one of the following:
  - Name—ACL created by name, as described in the “Access Control Lists” section on page 29-9.
  - IP address, as in the example.
  - Network address, including mask; for example, `192.168.50.0/24`.
  - TSIG key—Transaction signature key, in the form `key=key`, (as described in the “Transaction Security” section on page 29-10.
  - One of the reserved words:
    - any—Any ACL
    - none—No ACL
    - localhost—Any local host addresses
    - localnets—Any local network address
You can negate the ACL value by preceding it with an exclamation point (!).

- **name**—Keyword, or type of check to perform on the RR, which can be one of the following:
  - **name**—Name of the RR, requiring a name value.
  - **subdomain**—Name of the RR or the subdomain with any of its RRs, requiring a name or subdomain value.
  - **wildcard**—Name of the RR, using a wildcard value (see Table 29-2 on page 29-17).

- **host1**—Value based on the keyword, in this case the RR named host1. This can also be a subdomain name or, if the **wildcard** keyword is used, can contain wildcards (see Table 29-2 on page 29-17).

- **A,TXT**—RR types, each separated by a comma. This can be a list of any of the RR types described in Appendix A, “Resource Records.” You can negate each record type value by preceding it with an exclamation point (!).

- Note that if this or any assigned rule is not satisfied, the default is to deny all RR updates.

Tacked onto the end of the rule, outside the quotes, is an index number, in the example, 0. The index numbers start at 0. If there are multiple rules for an update policy, the index serves to add the rule in a specific order, such that lower numbered indexes have priority in the list. If a rule does not include an index, it is placed at the end of the list. Thus, a rule always has an index, whether or not it is explicitly defined. You also specify the index number in case you need to remove the rule.

To replace a rule, use `update-policy name delete`, then recreate the update policy. To edit a rule, use `update-policy name rules remove index`, where `index` is the explicitly defined or system-defined index number (remembering that the index numbering starts at 0), then recreate the rule. To remove the second rule in the previous example, enter:

```
nrcmd> update-policy policy1 rules remove 1
```

### Applying Update Policies to Zones

After creating an update policy, you can apply it to a zone (forward and reverse) or zone template if you have configured the DNS server with authoritative services.

#### Local Advanced Web UI

**Step 1** From the Design menu, choose Forward Zones under the Auth DNS submenu to open the List/Add Forward Zones page.

**Step 2** Click the name of the zone to open the Edit Zone page.

**Tip** You can also perform this function for zone templates on the Edit Zone Template page, and primary reverse zones on the Edit Primary Reverse Zone page (see Chapter 14, “Managing Zones.”).

**Step 3** Enter the name or (comma-separated) names of one or more of the existing named update policies in the `update-policy-list` attribute field.

**Note** The server processes the `update-acl` before it processes the `update-policy-list`.

```java
```
Step 4  Click Save.

CLI Commands

Use `zone name set update-policy-list`, equating the `update-policy-list` attribute with a quoted list of comma-separated update policies, as defined in the “Creating and Editing Update Policies” section on page 29-16. For example:

```
nr cmd> zone example.com set update-policy-list="policy1,policy2"
```

Confirming Dynamic Records

The Cisco Prime IP Express DHCP server stores all pending DNS update data on disk. If the DHCP server cannot communicate with a DNS server, it periodically tests for re-established communication and submits all pending updates. This test typically occurs every 40 seconds.

Local Advanced Web UI

From the Design menu, choose Forward Zones under the Auth DNS submenu. Click the Resource Records tab to open the Edit Zone page.

CLI Commands

Use `zone name listRR dns`.

Scavenging Dynamic Records

Microsoft Windows DNS clients that get DHCP leases can update (refresh) their Address (A) records directly with the DNS server. Because many of these clients are mobile laptops that are not permanently connected, some A records may become obsolete over time. The Windows DNS server scavenges and purges these primary zone records periodically. Cisco Prime IP Express provides a similar feature that you can use to periodically purge stale records.

Scavenging is normally disabled by default, but you should enable it for zones that exclusively contain Windows clients. Zones are configured with `no-refresh` and `refresh` intervals. A record expires once it ages past its initial creation date plus these two intervals. Figure 29-1 shows the intervals in the scavenging time line.

![Address Record Scavenging Time Line Intervals](image-url)
Scavenging Dynamic Records

Chapter 29 Configuring DNS Update

The Cisco Prime IP Express process is:

1. When the client updates the DNS server with a new A record, this record gets a timestamp, or if the client refreshes its A record, this may update the timestamp (“Record is created or refreshed”).

2. During a no-refresh interval (a default value of seven days), if the client keeps sending the same record without an address change, this does not update the record timestamp.

3. Once the record ages past the no-refresh interval, it enters the refresh interval (also a default value of seven days), during which time DNS updates refresh the timestamp and put the record back into the no-refresh interval.

4. A record that ages past the refresh interval is available for scavenging when it reaches the scavenge interval.

**Note**

Only unprotected RRs are scavenged. To keep RRs from being scavenged, set them to protected. However, top-of-zone (@) RRs, even if unprotected, are not scavenged.

The following DNS server attributes affect scavenging:

- **scvg-interval**—Period during which the DNS server checks for stale records in a zone. The value can range from one hour to 365 days. You can also set this for the server (the default value is one week), although the zone setting overrides it.

- **scvg-no-refresh-interval**—Interval during which actions, such as dynamic or prerequisite-only DNS updates, do not update the record timestamp. The value can range from one hour to 365 days. The zone setting overrides the server setting (the default value is one week).

- **scvg-refresh-interval**—Interval during which DNS updates increment the record timestamp. After both the no-refresh and refresh intervals expire, the record is a candidate for scavenging. The value can range from one hour to 365 days. The zone setting overrides the server setting (the default value is one week).

- **scvg-ignore-restart-interval**—Ensures that the server does not reset the scavenging time with every server restart. Within this interval, Cisco Prime IP Express ignores the duration between a server down instance and a restart, which is usually fairly short.

   The value can range from two hours to one day. With any value longer than that set, Cisco Prime IP Express recalculates the scavenging period to allow for record updates that cannot take place while the server is stopped. The zone setting overrides the server setting (the default value is 2 hours).

Enable scavenging only for zones where a Cisco Prime IP Express DNS server receives updates exclusively from Windows clients (or those known to do automatic periodic DNS updates). Set the attributes listed above. The Cisco Prime IP Express scavenging manager starts at server startup. It reports records purged through scavenging to the changeset database. Cisco Prime IP Express also notifies secondary zones by way of zone transfers of any records scavenged from the primary zone. In cases where you create a zone that has scavenging disabled (the records do not have a timestamp) and then subsequently enable it, Cisco Prime IP Express uses a proxy timestamp as a default timestamp for each record.

You can monitor scavenging activity using one or more of the log settings scavenge, scavenge-details, ddns-refreshes, and ddns-refreshes-details.

**Local Advanced Web UI**

On the Manage DNS Authoritative Server page, click the Commands button to open the DNS Commands dialog box. Click the Run icon next to Scavenge all zones.
To scavenge a particular forward or reverse zone only, go to the Zone Commands for Zone page, which is available by clicking the Run icon on the List/Add Forward Zones page or List/Add Reverse Zones page. Click the Run icon again next to Scavenge zone on the Zone Commands for Zone page. To find out the next time scavenging is scheduled for the zone, click the Run icon next to Get scavenge start time.

**CLI Commands**

Use **dns scavenge** for all zones that have scavenging enabled. Use the **getScavengeStartTime** action on a zone to find out the next time scavenging is scheduled to start.

**Troubleshooting DNS Update**

You can use a standard DNS tool such as **dig** and **nslookup** to query the server for RRs. The tool can be valuable in determining whether dynamically generated RRs are present. For example:

$ nslookup
default Server: server2.example.com
Address: 192.168.1.2
> leasehost1.example.com
Server: server2.example.com
Address: 192.168.1.100
> set type=ptr
> 192.168.1.100
Server: server2.example.com
Address: 192.168.1.100

100.40.168.192.in-addr.arpa name = leasehost1.example.com
40.168.192.in-addr.arpa nameserver = server2.example.com

You can monitor DNS updates on the DNS server by setting the **log-settings** attribute to **ddns**, or show even more details by setting it to **ddns-details**.

**Transitioning to DHCID RR for DHCPv4**

As networks make the transition from the IPv4 to IPv6 addressing, a lot of network devices will use both IPv4 and IPv6 addresses. These devices may be using multiple interfaces on the same host, using different networks, or using different DHCP versions. These devices need to be identified consistently with respect to DHCP server and accordingly the DHCP server will update the DNS server.

DHCPv4 uses TXT RR or DHCID RR and DHCPv6 uses DHCID RR for DNS updates. The default value of DHCP server-wide settings attribute **dns-client-identity** is txt and the attribute is not configured for individual DNS update config objects. You can configure the DNS updates in one of the following ways:

- **TXT RR for DHCPv4 and DHCID for DHCPv6**—To enable this configuration set the **dns-client-identity** to txt. The server will use the TXT RR in DHCPv4 DNS updates and DHCID RR for DHCPv6 DNS updates.

- **DHCID RR for both DHCPv4 and DHCPv6**—To enable this configuration set the **dns-client-identity** to dhcid. The server will use the DHCID RR for both DHCPv4 and DHCPv6 DNS updates. This setting should be used to support dual stack clients and can only be used if all DHCP servers doing DNS updates to the zones for this configuration support and are configured to use the DHCID RR.
• **Transition to DHCID RR**—To enable this configuration set the `dns-client-identity` to `transition-to-dhcid`. Set the `force-dns-update` attribute to true. Reload the server. For the zones that need to be upgraded, set the `dns-client-identity` attribute to `dhcid` and restore the `force-dns-update` attribute to its earlier value, after the longest lease time configured in the server.

**Note** You must set the transition-to-dhcid attribute until all the DHCPv4 resource records are updated to DHCID RR. For more information, see Transitioning to DHCID RR for DHCPv4, page 29-21.

• **Regress to TXT RR**—To enable this configuration set the `dns-client-identity` to `regress-to-txt`. Set the `force-dns-update` attribute to true. Reload the server. For the zones that need to be upgraded, set the `dns-client-identity` attribute to `txt` and restore the `force-dns-update` attribute to its earlier value, after the longest lease time configured in the server.

**Local Advanced and Regional Web UI**

**Step 1** From the **Deploy** menu, choose **DNS Update Configs** under the **DNS Updates** submenu to open the List/Add DNS Update Configurations page.

**Step 2** Select the name of the update configuration to open the **Edit DNS Update Configuration** page.

**Step 3** In the DNS update settings, set `transition-to-dhcid` as `dns-client-identity` in DNS update settings.

**Step 4** Optionally set `force-dns-update` to true. Using this setting will expedite the process of transitioning from TXT RR to DHCID RR.

**Step 5** Set scavenging settings attributes in forward or reverse zones to the following values:

- Set `scvg-enabled` to true.

**Step 6** Set scavenging settings attributes in DNS server to the following values:

- Set `scvg-interval` to longest lease time.
- Set `scvg-refresh-interval` to longest lease time.
- Set `scvg-no-refresh-interval` to 0.

**Step 7** Verify that all TXT RRs are converted to DHCID RRs in the RRs for the zones.

You must set the transition-to-dhcid attribute until all the DHCPv4 resource records are updated to DHCID RR. If some TXT RRs entries do not transition to DHCID RR, you may need to remove these DNS entries manually by using the Cisco Prime IP Express single-record dynamic RR removal feature.

**Step 8** Click **Save**.

**Configuring DNS Update for Windows Clients**

The Windows operating system rely heavily on DNS and, to a lesser extent, DHCP. This reliance requires careful preparation on the part of network administrators prior to wide-scale Windows deployments. Windows clients can add entries for themselves into DNS by directly updating forward zones with their address (A) record. They cannot update reverse zones with their pointer (PTR) records.

**Related Topics**

- Client DNS Updates, page 29-23
- Dual Zone Updates for Windows Clients, page 29-25
- DNS Update Settings in Windows Clients, page 29-25
Client DNS Updates

It is not recommended that clients be allowed to update DNS directly.

For a Windows client to send address record updates to the DNS server, two conditions must apply:

- The Windows client must have the **Register this connection’s addresses in DNS** box checked on the **DNS** tab of its TCP/IP control panel settings.
- The DHCP policy must enable direct updating (Cisco Prime IP Express policies do so by default).

The Windows client notifies the DHCP server of its intention to update the A record to the DNS server by sending the **client-fqdn** DHCP option (81) in a DHCPREQUEST packet. By indicating the fully qualified domain name (FQDN), the option states unambiguously the client location in the domain namespace. Along with the FQDN itself, the client or server can send one of these possible flags in the **client-fqdn** option:

- 0 — Client should register its A record directly with the DNS server, and the DHCP server registers the PTR record (done through the policy **allow-client-a-record-update** attribute being enabled).
- 1 — Client wants the DHCP server to register its A and PTR records with the DNS server.
- 3 — DHCP server registers the A and PTR records with the DNS server regardless of the client request (done through the policy **allow-client-a-record-update** attribute being disabled, which is the default value). Only the DHCP server can set this flag.

The DHCP server returns its own **client-fqdn** response to the client in a DHCPACK based on whether DNS update is enabled. However, if the 0 flag is set (the **allow-client-a-record-update** attribute is enabled for the policy), enabling or disabling DNS update is irrelevant, because the client can still send its updates to DNS servers. See Table 29-3 on page 29-23 for the actions taken based on how various properties are set.

**Table 29-3 Windows Client DNS Update Options**

<table>
<thead>
<tr>
<th>DHCP Client Action</th>
<th>DNS Update</th>
<th>DHCP Server Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checks <strong>Register this connection’s addresses in DNS</strong> and sends <strong>client-fqdn</strong>; DHCP server enables <strong>allow-client-a-record-update</strong></td>
<td>Enabled or disabled</td>
<td>Responds with <strong>client-fqdn</strong> that it allows the client to update its A records (sets flag 0), but the DHCP server still updates the PTR records.</td>
</tr>
<tr>
<td>Checks <strong>Register...</strong> and sends <strong>client-fqdn</strong>; DHCP disables <strong>allow-client-a-record-update</strong></td>
<td>Enabled</td>
<td>Responds with <strong>client-fqdn</strong> that it does not allow the client to update the DNS server directly (sets flag 3), and updates the A and PTR records.</td>
</tr>
<tr>
<td></td>
<td>Disabled</td>
<td>Does not respond with <strong>client-fqdn</strong> and does not update the DNS server.</td>
</tr>
<tr>
<td>Unchecks <strong>Register...</strong> and sends <strong>client-fqdn</strong></td>
<td>Enabled</td>
<td>Responds with <strong>client-fqdn</strong> that it is updating the A and PTR records.</td>
</tr>
<tr>
<td></td>
<td>Disabled</td>
<td>Does not respond with <strong>client-fqdn</strong> and does not update the DNS server.</td>
</tr>
</tbody>
</table>
A Windows DHCP server can set the client-fqdn option to ignore the client request. To enable this behavior in Cisco Prime IP Express, create a policy for Windows clients and disable the allow-client-a-record-update attribute for this policy.

The following attributes are enabled by default in Cisco Prime IP Express:

- **Server use-client-fqdn**—The server uses the client-fqdn value on incoming packets and does not examine the host-name. The DHCP server ignores all characters after the first dot in the domain name value, because it determines the domain from the defined scope for that client. Disable use-client-fqdn only if you do not want the server to determine hostnames from client-fqdn, possibly because the client is sending unexpected characters.

- **Server use-client-fqdn-first**—The server examines client-fqdn on incoming packets from the client before examining the host-name option (12). If client-fqdn contains a hostname, the server uses it. If the server does not find the option, it uses the host-name value. If use-client-fqdn-first is disabled, the server prefers the host-name value over client-fqdn.

- **Server use-client-fqdn-if-asked**—The server returns the client-fqdn value in the outgoing packets if the client requests it. For example, the client might want to know the status of DNS activity, and hence request that the DHCP server should present the client-fqdn value.

- **Policy allow-client-a-record-update**—The client can update its A record directly with the DNS server, as long as the client sets the client-fqdn flag to 0 (requesting direct updating). Otherwise, the server updates the A record based on other configuration properties.

The hostnames returned to client requests vary depending on these settings (see Table 29-4 on page 29-24).

### Table 29-3  Windows Client DNS Update Options (continued)

<table>
<thead>
<tr>
<th>DHCP Client Action</th>
<th>DNS Update</th>
<th>DHCP Server Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not send client-fqdn</td>
<td>Enabled</td>
<td>Does not respond with client-fqdn, but updates the A and PTR records.</td>
</tr>
<tr>
<td></td>
<td>Disabled</td>
<td>Does not respond with client-fqdn and does not update the DNS server.</td>
</tr>
</tbody>
</table>

### Table 29-4  Hostnames Returned Based on Client Request Parameters

<table>
<thead>
<tr>
<th>Client Request</th>
<th>With Server/Policy Settings</th>
<th>Resulting Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>Includes host-name (option 12)</td>
<td>use-host-name=true use-client-fqdn=false (or use-client-fqdn-first=false) trim-host-name=true (same except:) trim-host-name=false</td>
<td>host-name trimmed at first dot. Example: host-name host1.bob is returned host1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>host-name. Example: host-name host1.bob is returned host1.bob.</td>
</tr>
<tr>
<td>Includes client-fqdn (option 81)</td>
<td>use-client-fqdn=true use-host-name=false (or use-client-fqdn-first=true)</td>
<td>client-fqdn trimmed at first dot. Example: client-fqdn host1.bob is returned host1.</td>
</tr>
</tbody>
</table>
Table 29-4  Hostnames Returned Based on Client Request Parameters (continued)

<table>
<thead>
<tr>
<th>Client Request</th>
<th>With Server/Policy Settings</th>
<th>Resulting Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>Omits host-name (option 12) and client-fqdn (option 81)</td>
<td>Or: use-host-name=false use-client-fqdn=false (same as the previous except:) hostname is undefined in the client/policy hierarchy, with synthesize-name=true (same as the previous except:) synthesize-name=false</td>
<td>Set by client/policy hierarchy. Synthesized following the synthesizing rule, which is to append the hyphenated IP address of the host after the specified synthetic-name-stem. Undefined.</td>
</tr>
</tbody>
</table>

Dual Zone Updates for Windows Clients

Windows DHCP clients might be part of a DHCP deployment where they have A records in two DNS zones. In this case, the DHCP server returns the client-fqdn so that the client can request a dual zone update. To enable a dual zone update, enable the policy attribute allow-dual-zone-dns-update.

The DHCP client sends the 0 flag in client-fqdn and the DHCP server returns the 0 flag so that the client can update the DNS server with the A record in its main zone. However, the DHCP server also directly sends an A record update based on the client secondary zone in the behalf of the client. If both allow-client-a-record-update and the allow-dual-zone-dns-update are enabled, allowing the dual zone update takes precedence so that the server can update the secondary zone A record.

DNS Update Settings in Windows Clients

The Windows client can set advanced properties to enable sending the client-fqdn option.

**Step 1**  On the Windows client, go to the Control Panel and open the TCP/IP Settings dialog box.

**Step 2**  Click the Advanced tab.

**Step 3**  Click the DNS tab.

**Step 4**  To have the client send the client-fqdn option in its request, leave the Register this connection’s addresses in DNS box checked. This indicates that the client wants to do the A record update.

Windows Client Settings in DHCP Servers

You can apply a relevant policy to a scope that includes the Windows clients, and enable DNS updates for the scope.

**Step 1**  Create a policy for the scope that includes the Windows clients. For example:

a. Create a policywin2k. You have to specify the forward or reverse zone name, main and backup server IP addresses when you create a policy.
b. Create a win2k scope with the subnet 192.168.1.0/24 and policywin2k as the policy. Add an address range of 192.168.1.10 through 192.168.1.100.

Step 2  
Set the zone name, server address (for A records), reverse zone name, and reverse server address (for PTR records), as described in the “Creating DNS Update Configurations” section on page 29-5.

Step 3  
If you want the client to update its A records at the DNS server, enable the policy attribute allow-client-a-record-update (this is the preset value). There are a few caveats to this:

• If allow-client-a-record-update is enabled and the client sends the client-fqdn with the update bit enabled, the host-name and client-fqdn returned to the client match the client client-fqdn. (However, if the override-client-fqdn is also enabled on the server, the hostname and FQDN returned to the client are generated by the configured hostname and policy domain name.)

• If, instead, the client does not send the client-fqdn with the update bit enabled, the server does the A record update, and the host-name and client-fqdn (if requested) returned to the client match the name used for the DNS update.

• If allow-client-a-record-update is disabled, the server does the A record updates, and the host-name and client-fqdn (with the update bit disabled) values returned to the client match the name used for the DNS update.

• If allow-dual-zone-dns-update is enabled, the DHCP server always does the A record updates. (See the “Dual Zone Updates for Windows Clients” section on page 29-25.)

• If use-dns-update-prereqs is enabled (the preset value) for the DHCP server or DNS update configuration and update-dns-first is disabled (the preset value) for the update configuration, the hostname and client-fqdn returned to the client are not guaranteed to match the DNS update, because of delayed name disambiguation. However, the lease data will be updated with the new names. According to RFC 2136, update prerequisites determine the action the primary master DNS server takes based on whether an RR set or name record should or should not exist. Disable use-dns-update-prereqs only under rare circumstances.

Step 4  
Reload the DHCP server.

SRV Records and DNS Updates

Windows relies heavily on the DNS protocol for advertising services to the network. Table 29-5 on page 29-27 describes how Windows handles service location (SRV) DNS RRs and DNS updates.

You can configure the Cisco Prime IP Express DNS server so that Windows domain controllers can dynamically register their services in DNS and, thereby, advertise themselves to the network. Because this process occurs through RFC-compliant DNS updates, you do not need to do anything out of the ordinary in Cisco Prime IP Express.
### Table 29-5: Windows SRV Records and DNS Updates

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRV records</td>
<td>Windows domain controllers use the SRV RR to advertise services to the network. This RR is defined in the RFC 2782, “A DNS RR for specifying the location of services (DNS SRV).” The RFC defines the format of the SRV record (DNS type code 33) as: <code>_service._protocol.name ttl class SRV priority weight port target</code></td>
</tr>
</tbody>
</table>
|                 | There should always be an A record associated with target of the SRV record, so that the client can resolve the service back to a host. In the Windows implementation of SRV records, the records might look like this:  
|                 | myserver.example.com A 10.100.200.11  
|                 | _ldap._tcp.example.com SRV 0 0 389 myserver.example.com  
|                 | _kdc._tcp.example.com SRV 0 0 88 myserver.example.com  
|                 | _ldap._tcp.dc_msds.example.com SRV 0 0 88 myserver.example.com  |
| How SRV records are used | When a Windows client boots up, it tries to initiate the network login process to authenticate against its Windows domain controller. The client must first discover where the domain controller is, and they do so using the dynamically generated SRV records.  
|                 | Before launching the net-login process, the client queries DNS with a service name; for example, _kdc._tcp.dc_msds.example.com. The DNS server SRV record target, for example, is my-domain-controller.example.com. The Windows client then queries DNS with the hostname my-domain-controller.example.com. DNS returns the host address and the client uses this address to find the domain controller. The net-login process fails without these SRV records.  |
| DNS updates     | When a Windows server is configured as a domain controller, you statically configure the name of the domain it manages through the Active Directory management console. This Windows domain should have a corresponding DNS zone associated with it. The domain controller should also have a series of DNS resolvers configured in its TCP/IP properties control panel.  
|                 | When the Windows domain controller boots up, it performs these steps to register itself in DNS and advertise its services to the network:  
|                 | 1. Queries DNS asking for the start of authority (SOA) record for the DNS domain that mostly closely encapsulates its Windows domain.  
|                 | 2. Identifies the primary DNS server for the DNS zone (from the SOA record) that mostly closely encapsulates its Windows domain name.  
|                 | 3. Creates a series of SRV records in this zone using the RFC 2136 DNS Update protocol.  |
Chapter 29      Configuring DNS Update

Configuring DNS Update for Windows Clients

To configure Cisco Prime IP Express to accept these dynamic SRV record updates:

Step 1  Determine the IP addresses of the devices in the network that need to advertise services through DNS.

Step 2  If they do not exist, create the appropriate forward and reverse zones for the Windows domains.

Step 3  Enable DNS updates for the forward and reverse zones.

Step 4  Set up a DNS update policy to define the IP addresses of the hosts to which you want to restrict accepting
        DNS updates (see the “Configuring DNS Update Policies” section on page 29-15). These are usually the
        DHCP servers and any Windows domain controllers. (The Windows domain controllers should have
        static IP addresses.)

        If it is impractical or impossible to enter the list of all the IP addresses from which a DNS server must
        accept updates, you can configure Cisco Prime IP Express to accept updates from a range of addresses,
        although Cisco does not recommend this configuration.

Step 5  Reload the DNS and DHCP servers.

Issues Related to Windows Environments

Table 29-5 describes the issues concerning interoperability between Windows and Cisco Prime IP
Express. The information in this table is intended to inform you of possible problems before you
encounter them in the field. For some frequently asked questions about Windows interoperability, see
the “Frequently Asked Questions About Windows Integration” section on page 29-32.
**Table 29-6  
Issues Concerning Windows and Cisco Prime IP Express Interoperability**

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
</table>
| Invisible dynamically created RRs | Cisco Prime IP Express, when properly configured, accepts DNS updates from both DHCP and Windows servers. You can use the CLI to access the dynamic portion of the DNS zone for viewing and deleting records. Enter this command to view all DNS RRs in a given zone:  
```
  nrcmd> zone myzone listRR dynamic myfile
```
  This redirects the output to the myfile file (see Example 29-1 on page 29-31). You can delete dynamically generated records by entering this command:  
```
  nrcmd> zone myzone removeDynRR myname [type]
```
  You can also use `nslookup` to verify their existence, and you can use version 5.x (shipped with Windows) to view dynamic SRV records. In this version, use `set type=SRV` to enable viewing SRV records.                                                                 |
| Domain controller registration | A Windows domain controller has to register itself in DNS using DNS updates. The DNS RFCs dictate that only a primary DNS server for a particular zone can accept edits to the zone data. Hence, the Windows domain controller has to discover which DNS server is the primary for the zone that includes its Windows domain name.  
  The domain controller discovers this by querying the first DNS server in its resolver list (configured in the TCP/IP properties control panel). The initial query is for the SOA record of the zone that includes the Windows domain of the domain controller. The SOA record includes the name of the primary server for the zone. If no zone exists for the domain name, the domain controller keeps removing the left-most label of the domain name and sends queries until it finds an SOA record with a primary server included in that domain. Once the domain controller has the name of the primary DNS server for its domain, it sends it DNS updates to create the necessary SRV records. Ensure that the name of the zone primary DNS server is in its SOA record. |
| Failure of A record DNS updates | When a Windows domain controller tries to advertise itself to the network, it sends several DNS update requests to the DNS server of record for its domain. Most of these update requests are for SRV records. However, the domain controller also requests an update for a single A record of the same name as the Windows domain.  
  If the Cisco Prime IP Express DNS server is also authoritative for a zone identical to this Windows domain, it rejects registering its A record, because the DNS A record update conflicts with the static SOA and NS records. This is to prevent possible security infractions, such as a dynamic host registering itself and spoofing Web traffic to a site.  
  For example, the domain controller might control the w2k.example.com Windows zone. If a zone with the same name exists on the Cisco Prime IP Express DNS server, these RRs could be part of that zone. (Example follows.) |
Chapter 29      Configuring DNS Update

Configuring DNS Update for Windows Clients

w2k.example.com. 43200 SOA nameserver.example.com.
hostmaster.example.com.
{ 98011312 ;serial
3600 ;refresh
3600 ;retry
3600000 ;expire
43200 ;minimum
w2k.example.com.86400 NS nameserver.example.com
}

The domain controller would try to add an additional record; for example:
w2k.example.com. 86400 A 192.168.2.1

Cisco Prime IP Express does not allow a DNS update to conflict with any
statically configured name in the zone, even if the record type associated with
that name is different. In the above example, an attempt to add an A record
associated with the name w2k.example.com collides with the SOA and NS
records.

When the domain controller boots up, a DNS log file entry such as this
appears:
08/10/2000 16:35:33 name/dns/1 Info Protocol 0 Error - REFUSED -
Update of static name "w2k.example.com", from address
[10.100.200.2]

This is how Cisco Prime IP Express responds to DNS updates of static DNS
data. Additionally, you can ignore this DNS update failure. Windows clients
do not use this A record. Allocation of domain controllers happens through
SRV records. Microsoft added the A record to accommodate legacy NT clients
that do not support SRV records.

Note that failing to register the controller A record slows down the domain
controller bootup process, affecting the overall login of worker clients. As
mentioned earlier, the workaround is to define the Windows domain as a
subdomain of the authoritative zone, or enable the DNS server
simulate-zone-top-dynupdate attribute. If this is not possible, contact the Cisco
Technical Assistance Center for help.

Table 29-6    Issues Concerning Windows and Cisco Prime IP Express Interoperability (continued)

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
</table>
| w2k.example.com. 43200 SOA nameserver.example.com.
hostmaster.example.com.
{ 98011312 ;serial
3600 ;refresh
3600 ;retry
3600000 ;expire
43200 ;minimum
w2k.example.com.86400 NS nameserver.example.com |

Windows RC1 DHCP clients

Microsoft released Windows build 2072 (known as RC1) with a broken DHCP
client. This client sends a misformed packet that Cisco Prime IP Express
cannot parse. Cisco Prime IP Express drops the packet and cannot serve the
client, logging this error:
08/10/2000 14:56:23 name/dhcp/1 Activity Protocol 0 10.0.0.15 Lease
offered to Host:'My-Computer' CID: 01:00:a0:24:1a:b0:d8 packet'R15'

08/10/2000 14:56:23 name/dhcp/1 Warning Protocol 0 Unable to find
necessary Client information in packet from MAC
address:'1,6,00:d0:ba:d3:bd:3b'. Packet dropped!

Cisco Prime IP Express includes error checking specifically designed to deal
with errors such as this improperly built FQDN option. However, if you
encounter this problem, install the Microsoft patch to the RC1 client on the
DHCP client. You must obtain this patch from Microsoft.
Chapter 29 Configuring DNS Update

Configuring DNS Update for Windows Clients

Example 29-1 Output Showing Invisible Dynamically Created RRs

Dynamic Resource Records

| _ldap._tcp.test-lab._sites 600 IN SRV 0 100 389 CNR-MKT-1.w2k.example.com. |
| _ldap._tcp.test-lab._sites.gc._msdcs 600 IN SRV 0 100 3268 CNR-MKT-1.w2k.example.com. |
| _kerberos._tcp.test-lab._sites.dc._msdcs 600 IN SRV 0 100 88 CNR-MKT-1.w2k.example.com. |
| _ldap._tcp.test-lab._sites.dc._msdcs 600 IN SRV 0 100 389 CNR-MKT-1.w2k.example.com. |
| _ldap._tcp.dc._msdcs 600 IN SRV 0 100 88 CNR-MKT-1.w2k.example.com. |
| _ldap._tcp.gc._msdcs 600 IN SRV 0 100 3268 CNR-MKT-1.w2k.example.com. |
| _kerberos._tcp.test-lab._sites.dc._msdcs 600 IN SRV 0 100 88 CNR-MKT-1.w2k.example.com. |
| _ldap._tcp.pdc._msdcs 600 IN SRV 0 100 389 CNR-MKT-1.w2k.example.com. |
| _ldap._tcp.gc._msdcs 600 IN SRV 0 100 3268 CNR-MKT-1.w2k.example.com. |
| _ldap._tcp.1ca176bc-86bf-46f1-8a0f-235ab891bcd2.domains._msdcs 600 IN SRV 0 100 389 CNR-MKT-1.w2k.example.com. |
| e5b0e667-27c8-44f7-bd76-6b8385c74bd7._msdcs 600 IN CNAME CNR-MKT-1.w2k.example.com. |

Table 29-6 Issues Concerning Windows and Cisco Prime IP Express Interoperability (continued)

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows plug-and-play network interface card (NIC) configuration</td>
<td>If configured to use DHCP, a Windows system tries to obtain a DHCP lease on startup. If no DHCP server is available, Windows may automatically configure the computer interface with a plug-and-play IP address. This address is not one that the network administrator or DHCP server configured or selected. These plug-and-play addresses are in the range 169.254.0.0/16. If you see devices in this address range on a network, it means that Windows autoconfigured the interfaces because it could not obtain a lease from a DHCP server. This can cause significant network and troubleshooting problems. The Windows system no longer informs the user that the DHCP client could not obtain a lease. Everything appears to function normally, but the client cannot route packets off its local subnet. Additionally, you may see the DHCP client trying to operate on the network with an address from the 169.254.0.0/16 network. This may lead you to think that the Cisco Prime IP Express DHCP server is broken and handing out the wrong addresses.</td>
</tr>
<tr>
<td>Scavenging Windows client address records</td>
<td>Windows clients do not clean up after themselves, potentially causing their dynamic record registration to remain indefinitely. This leaves stale address records on the DNS server. To ensure that these stale records are periodically removed, you must enable scavenging for the zone (see the “Scavenging Dynamic Records” section on page 29-19).</td>
</tr>
</tbody>
</table>
Chapter 29 Configuring DNS Update

Configuring DNS Update for Windows Clients

Frequently Asked Questions About Windows Integration

These questions are frequently asked about integrating Cisco Prime IP Express DNS services with Windows:

Q. What happens if both Windows clients and the DHCP server are allowed to update the same zone? Can this create the potential for stale DNS records being left in a zone? If so, what can be done about it?

A. The recommendation is not to allow Windows clients to update their zones. Instead, the DHCP server should manage all the client dynamic RR records. When configured to perform DNS updates, the DHCP server accurately manages all the RRs associated with the clients that it served leases to. In contrast, Windows client machines blindly send a daily DNS update to the server, and when removed from the network, leave a stale DNS entry behind.

Any zone being updated by DNS update clients should have DNS scavenging enabled to shorten the longevity of stale RRs left by transient Windows clients. If the DHCP server and Windows clients are both updating the same zone, three things are required in Cisco Prime IP Express:

a. Enable scavenging for the zone.

b. Configure the DHCP server to refresh its DNS update entries as each client renews its lease. By default, Cisco Prime IP Express does not update the DNS record again between its creation and its final deletion. A DNS update record that Cisco Prime IP Express creates lives from the start of the lease until the lease expires. You can change this behavior using a DHCP server (or DNS update configuration) attribute, force-dns-updates. For example:

```bash
nrcmd> dhcp enable force-dns-updates
100 Ok
force-dns-updates=true
```

c. If scavenging is enabled on a particular zone, then the lease time associated with clients that the DHCP server updates that zone on behalf of must be less than the sum of the no-refresh-interval and refresh-interval scavenging settings. Both of these settings default to seven days. You can set the lease time to 14 days or less if you do not change these default values.

Q. What needs to be done to integrate a Windows domain with a pre-existing DNS domain naming structure if it was decided not to have overlapping DNS and Windows domains? For example, if there is a pre-existing DNS domain called example.com and a Windows domain is created that is called w2k.example.com, what needs to be done to integrate the Windows domain with the DNS domain?

A. In the example, a tree in the Windows domain forest would have a root of w2k.example.com. There would be a DNS domain named example.com. This DNS domain would be represented by a zone named example.com. There may be additional DNS subdomains represented in this zone, but no subdomains are ever delegated out of this zone into their own zones. All the subdomains will always reside in the example.com. zone.
Q. In this case, how are DNS updates from the domain controllers dealt with?

A. To deal with the SRV record updates from the Windows domain controllers, limit DNS updates to the example.com zone to the domain controllers by IP address only. (Later, you will also add the IP address of the DHCP server to the list.) Enable scavenging on the zone. The controllers will update SRV and A records for the w2k.example.com subdomain in the example.com zone. There is no special configuration required to deal with the A record update from each domain controller, because an A record for w2k.example.com does not conflict with the SOA, NS, or any other static record in the example.com zone.

The example.com zone then might include these records:

```dns
example.com. 43200 SOA ns.example.com. hostmaster.example.com. (98011312 ;serial
3600 ;refresh
3600 ;retry
3600000 ;expire
43200 ) ;minimum
example.com 86400 NS ns.example.com
ns.example.com. 86400 A 10.0.0.10
_ldap._tcp.w2k.example.com. IN SRV 0 0 389 dc1.w2k.example.com
w2k.example.com 86400 A 10.0.0.25
...
```

Q. In this case, how are zone updates from individual Windows client machines dealt with?

A. In this scenario, the clients could potentially try to update the example.com zone with updates to the w2k.example.com domain. The way to avoid this is to close down the zone to updates except from trusted sources. For Cisco Prime IP Express, you can use transaction signatures (TSIG) between the DHCP server and the primary DNS server for the example.com zone.

Configure the DHCP server to do DNS updates to the example.com zone and the appropriate reverse zone for each client, and use option 81 to prevent the clients from doing DNS updates themselves.

Q. Has security been addressed in this case?

A. By configuring the forward and reverse zone to accept only updates from trusted IP addresses, you close the zone to updates from any other device on the network. Security by IP is not the most ideal solution, as it would not prevent a malicious attack from a spoofed IP address source. You can secure updates from the DHCP server by configuring TSIG between the DHCP server and the DNS server.

Q. Is scavenging required in this case?

A. No. Updates are only accepted from the domain controllers and the DHCP server. The DHCP server accurately maintains the life cycle of the records that they add and do not require scavenging. You can manage the domain controller dynamic entries manually by using the Cisco Prime IP Express single-record dynamic RR removal feature.

Q. What needs to be done to integrate a Windows domain that shares its namespace with a DNS domain? For example, if there is a pre-existing DNS zone called example.com and a Windows Active Directory domain called example.com needs to be deployed, how can it be done?

A. In this example, a tree in the Windows domain forest would have a root of example.com. There is a pre-existing domain that is also named example.com that is represented by a zone named example.com.

Q. In this case, how are DNS updates from individual Windows client machines dealt with?

A. To deal with the SRV record updates, create subzones for:
Limit DNS updates to those zones to the domain controllers by IP address only. Enable scavenging on these zones.

To deal with the A record update from each domain controller, enable a DNS server attribute, `simulate-zone-top-dynupdate`.

```
nrcmd> dns enable simulate-zone-top-dynupdate
```

It is not required, but if desired, manually add an A record for the domain controllers to the example.com zone.

**Q.** In this case, how are zone updates from individual Windows client machines dealt with?

**A.** In this scenario, the clients could potentially try to update the example.com zone. The way to avoid this is to close down the zone to updates except from trusted sources. For Cisco Prime IP Express, you can use transaction signatures (TSIG) between the DHCP server and the primary DNS server for the example.com zone.

Configure the DHCP server to do DNS updates to the example.com zone and the appropriate reverse zone for each client, and use option 81 to prevent the clients from doing DNS updates themselves.

**Q.** Has security been addressed in this case?

**A.** By configuring the forward and reverse zone to accept only updates from trusted IP addresses, you close the zone to updates from other devices on the network. Security by IP is not the most ideal solution, as it would not prevent a malicious attack from a spoofed source. Updates from the DHCP server are more secure when TSIG is configured between the DHCP server and the DNS server.

**Q.** Has scavenging been addressed in this case?

**A.** Yes. The subzones `_tcp.example.com`, `_sites.example.com`, `_msdcs.example.com`, `_msdcs.example.com`, and `_udp.example.com` zones accept updates only from the domain controllers, and scavenging was turned on for these zones. The example.com zone accepts DNS updates only from the DHCP server.

---

**Configuring GSS-TSIG**

**CPIPE DNS Configuration to integrate with AD**

To integrate AD with CPIPE DNS configuration, follow these steps:

1. **Install CPIPE DNS on a Workgroup machine.**
2. **Create a zone (same as the domain of AD).**
   - Install AD on a windows server using `dcpromo.exe` and integrate with CPIPE DNS.
3. **Ensure the SRV records are added in CPIPE DNS.**
Chapter 29    Configuring DNS Update

Configuring GSS-TSIG

DCHOSTNAME. DOMAIN.COM A AD-IP-ADDRESS
 ldap._tcp.DOMAIN.COM. SRV 0 0 389 DCHOSTNAME.DOMAIN.COM.
 kerberos._tcp.DOMAIN.COM. SRV 0 0 88 DCHOSTNAME.DOMAIN.COM.
 ldap._tcp.dc._msdcs.DOMAIN.COM. SRV 0 0 389 DCHOSTNAME.DOMAIN.COM.
 kerberos._tcp.dc._msdcs.DOMAIN.COM. SRV 0 0 88 DCHOSTNAME.DOMAIN.COM.

Note    DCHOSTNAME refers to AD host name and DOMAIN.COM is the domain that exists in AD.

Bring CPIPE DNS and AD under the same domain in the windows environment:

Step 1    Change the domain, Computer > Properties > Computer Name > change the member of domain (same as the domain of AD).
Step 2    Control Panel > Network and Internet > Network and Sharing Center > Local Area Connection > Properties > TCP/IPV4 > Preferred DNS (CPIPE DNS running IP).
Step 3    Restart the computer, and login with the User that exists in AD.
Step 4    Login to AD,
    • check the DNS active Hostname is added into, AD Server Manager > Computers
    • associate the principal name to the server using the command:
      setspn -s DNS/ <hostname of the DNS server> <Computer Name>

In Linux, the primary DNS server is integrated to AD-KDC:

Step 1    Ensure the /etc/krb5.conf or DNS server with SRV record is configured to reach the required AD

    krb5.conf configuration
     [libdefaults]
     ticket_lifetime = 24h
     default_realm = <AD REALM>
     default_tkt_enctypes = rc4-hmac
     default_tgs_enctypes = rc4-hmac
     dns_lookup_realm = true
     dns_lookup_kdc = false
     forwardable = true
     <AD REALM> = {
         kdc = <AD-HOSTNAME>:88
         admin_server = <AD-HOSTNAME>:749
         default_domain = <AD REALM>
     }

Note    Ensure that the AD-HOSTNAME is resolvable.

Step 2    Create a service account in the Windows Server Active Directory:
    1. Use the Active Directory Users and Computers Administrative Tool to create a new user account.
       • Assign a user name to the account without any space.
       • Assign a password to the account
Chapter 29  Configuring DNS Update

Configuring GSS-TSIG

Note  Whenever the password expires/changed, the keytab file needs to be generated with a new associated kvno.

2. Assign a Service Principal Name (SPN) to the account utilizing the SETSPN.EXE. An SPN is the service-name/hostname/domain depending on the deployment. There can be multiple SPNs assigned to a single account.

For example, specify a <service-name> and a <hostname> where the service-name is DNS and the hostname is the fully qualified domain name of the machine on which the DNS server will be running.

setspn -s DNS/<DNS running Computer Name> <Computer Name>

3. Get the kvno details:

ldifde -f <Filename> -d "DC=<DOMAIN>,DC=com" -1 *,msDS-KeyVersionNumber -r "(serviceprincipalname=<service-principal name>)" -p subtree OR kvno.exe
<service-principal name>@<REALM>

4. Generate the Keytab file using the ktpass.exe command:

ktpass -out<filename>  -princ <Principal name>  -mapuser <User>  -pass <password associated with the user>   -crypto all -ptype KRB5_NT_PRINCIPAL -kvno <Kvno details>

   Transfer the keytab file to the Linux machine and run Kutil to add the Keytab entry to the existing Keytab file:

   > ktutil
   ktutil:  rkt <keytab file name>
   ktutil:  wkt /etc/krb5.keytab
   ktutil:  q

5. Display the keytab entry using:

klist -k -t -e /etc/krb5.keytab

Primary DNS Server on Linux Integrated to MIT-KDC

To associate the service-principal name to MIT KDC:

1. Login to the Linux DNS server and use kadmin utility to add the principal name to the MIT-KDC:

   > kadmin
   Authenticating as principal <MIT-KDC USER@REALM> with password.
   Password for <MIT-KDC USER@REALM.COM> : <Enter the associated Password>
   kadmin:  addprinc -randkey DNS/<hostname of the DNS server>
   WARNING: no policy specified for DNS/<hostname of the DNS server>@REALM; defaulting to no policy
   add_principal: Principal or policy already exists while creating ' DNS/<hostname of the DNS server>@REALM'.
   kadmin:  ktadd -randkey DNS/<hostname of the DNS server>
   kadmin: Principal -randkey does not exist.
   Entry for principal DNS/<hostname of the DNS server> with kvno x, encryption type AES-256 CTS mode with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5.keytab.
   Entry for principal DNS/<hostname of the DNS server> with kvno x, encryption type AES-128 CTS mode with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5.keytab.
2. Display the keytab entry using:
   
   ```
   klist -k -t -e /etc/krb5.keytab
   ```

3. Login to the MIT-KDC running LINUX server and check the added principal name has the same kvno associated as above using the command:
   
   ```
   K<system> DNS/<hostname of the DNS server>
   ```

### Troubleshooting GSS-TSIG Configuration

To get the details of GSS/SSPI failure and major/minor status, enable the DEBUG options in the DNS server and set the value of g=3.

- **“The key version number for the principal in the key table is incorrect.”**

  The Kvno returned by, `klist -k -t -e /etc/krb5.keytab` in the DNS running machine should be the same kvno in KDC.

  Verification of kvno in AD-KDC:

  ```
  ldifde -f c:\spn1_out.txt -d "DC=TIG,DC=com" -l *,msDS-KeyVersionNumber -r "(serviceprincipalname=DNS/WIN-CPNUV*)" -p subtree
  ```

  Verification of kvno is MIT-KDC:

  ```
  Kvno <principal name>
  ```

- **“Wrong Principal Name”**

  Ensure that the GSS Client and the server are using the same service-key that is used to encrypt/decrypt the service ticket.
Configuring GSS-TSIG
Using Extension Points

You can write extensions to affect how Cisco Prime IP Express handles and responds to DHCP requests, and to change the behavior of a DHCP server that you cannot normally do using the user interfaces. This chapter describes the extension points to which you can attach extensions for DHCPv4 and DHCPv6.

**Related Topics**

- Using Extensions, page 30-1
- Language-Independent API, page 30-4
- Tcl Extensions, page 30-7
- C/C++ Extensions, page 30-9
- DHCP Request Processing Using Extensions, page 30-12
- Extension Dictionaries, page 30-23
- Extension Point Descriptions, page 30-28

**Using Extensions**

You can alter and customize the operation of the Cisco Prime IP Express DHCP server by using extensions, functions that you can write in Tcl or C/C++.

Follow this process to create an extension for use in the DHCP server:

1. Determine the task to perform. What DHCP packet process do I want to modify?
2. Determine the approach to use. How do I want to modify the packet process?
3. Determine the extension point to which to attach the extension.
4. Choose the language (Tcl or C/C++).
5. Write (and possibly compile and link) the extension.
6. Add the extension to the DHCP server configuration.
7. Attach the extension to the extension point.
8. Reload the DHCP server so that it recognizes the extension.
9. Test and debug the results.
Creating, Editing, and Attaching Extensions

You can create, edit, and attach extensions.

You can associate multiple extensions per extension point. Each extension executes in the order specified by the sequence number used when the attachment was created. In the web UI, the sequence is the order in which the extensions appear per extension point on the List DHCP Extension Points page. In the CLI, you use the `sequence-number` value with the `dhcp attachExtension` command.

For more on multiple extensions per extension point, see the “Multiple Extension Considerations” section on page 30-6.

Local Advanced Web UI

Creating and Attaching Extensions:

**Step 1** From the **Deploy** menu, choose **Extensions** from the DHCP submenu to open the List/Add DHCP Extensions page.

**Step 2** Click the **Add Extensions** icon to open the Add DHCP Server Extension dialog box.

**Step 3** After you create an extension, you can attach it to one or more of the extension points on this page. To show the extension points where you can attach extensions, on the List/Add DHCP Extensions page, click **DHCP Extension Points** tab.

**Step 4** If you attach more than one extension for each extension point, you can change the sequence in which they are processed by clicking the arrow keys to rearrange the entries. To remove the extension, click the **Delete** icon.

CLI Commands

Use the `extension` command, which requires this syntax:

```
nrcmd> extension name create language extension-file entry-point
```

The `entry-point` is the name of the entry point in the `extension-file`. You can also set an optional `init-entry` attribute value for the initial entry point each time the DHCP server loads the file (see the “init-entry” section on page 30-29). You can call this function from any extension point bound to this module. You can also list the extensions using `extension list`.

To attach and detach an extension, use `dhcp attachExtension` and `dhcp detachExtension` for the DHCP server, which require this syntax:

```
nrcmd> dhcp attachExtension extension-point extension-name [sequence-number]
nrcmd> dhcp detachExtension extension-point [sequence-number]
```

The `sequence-number` applies if you attach multiple extensions per extension point, in increasing sequence order ranging from 1 through 32. If omitted, it defaults to 1.
Chapter 30      Using Extension Points

Determining Tasks

The task to which to apply an extension is usually some modification of the DHCP server processing so that it better meets the needs of your environment. You can apply an extension at each of these DHCP server processing points, from receiving a request to responding to the client:

1. Receive and decode the packet.
2. Look up, modify, and process any client-class.
3. Build a response type.
4. Determine the subnet (or link, in the case of DHCPv6).
5. Find any existing leases.
6. Serialize the lease requests.
7. Determine the lease acceptability for the client.
8. Gather and encode the response packet.
10. Return the packet.

A more complete list of these steps (along with the extension points to use at each step) appears in the “DHCP Request Processing Using Extensions” section on page 30-12.

For example, you might have an unusual routing hub that uses BOOTP configuration. This device issues a BOOTP request with an Ethernet hardware type (1) and MAC address in the chaddr field. It then sends out another BOOTP request with the same MAC address, but with a hardware type of Token Ring (6).

Specifying two different hardware types causes the DHCP server to allocate two IP addresses to the device. The DHCP server normally distinguishes between a MAC address with hardware type 1 and one with type 6, and considers them different devices. In this case, you might want to write an extension that prevents the DHCP server from handing out two different addresses to the same device.

Deciding on Approaches

Many solutions are often available to a single problem. When choosing the type of extension to write, you should first consider rewriting the input DHCP packet. This is a good approach, because it avoids having to know the internal processing of the DHCP server.

For the problem described in the “Determining Tasks” section on page 30-3, you can write an extension to solve it in either of these ways:

- Drop the Token Ring (6) hardware type packet.
- Change the packet to an Ethernet packet and then switch it back again on exit.

Although the second way involves a more complex extension, the DHCP client could thereby use either reply from the DHCP server. The second approach involves rewriting the packet, in this case using the post-packet-encode extension point (see the “post-packet-encode” section on page 30-38). Other approaches require other extensions and extension points.
Choosing Extension Languages

You can write extensions in Tcl or C/C++. The capabilities of each language, so far as the DHCP server is concerned, are similar, although the application programming interface (API) is slightly different to support the two very different approaches to language design:

- **Tcl**—Although scripting in Tcl might be somewhat easier than scripting in C/C++, it is interpreted and single-threaded, and might require more resources. However, you might be less likely than in C/C++ to introduce a serious bug, and there are fewer chances of a server failure. Cisco Prime IP Express currently supports Tcl version 8.4.

- **C/C++**—This language provides the maximum possible performance and flexibility, including communicating with external processes. However, the C/C++ API is more complex than the Tcl API. Also, the possibility of a bug in the extension causing a server failure is also more likely in C/C++.

Language-Independent API

The following concepts are independent of whether you write your extensions in Tcl or C/C++.

Related Topics

- Routine Signature, page 30-4
- Dictionaries, page 30-5
- Utility Methods in Dictionaries, page 30-5
- Configuration Errors, page 30-5
- Communicating with External Servers, page 30-6
- Recognizing Extensions, page 30-6
- Multiple Extension Considerations, page 30-6

Routine Signature

You need to define the extension as a routine in a file, which can contain multiple extension functions. You then attach the extension to one or more of the DHCP server extension points. When the DHCP server reaches that extension point, it calls the routine that the extension defines. The routine returns with a success or failure. You can configure the DHCP server to drop a packet on an extension failure.

You can configure one file—Tcl source file or C/C++ .dll or .so file—as multiple extensions to the DHCP server by specifying a different entry point for each configured extension.

The server calls every routine entry point with at least three arguments, the three dictionaries—request, response, and environment. Each dictionary contains many data items, each being a key-value pair:

- The extension can retrieve data items from the DHCP server by performing a get method on a dictionary for a particular data item.
- The extension can alter data items by performing a put or remove operation on many of the same named data items.

Although you cannot use all dictionaries at every extension point, the calling sequence for all routines is the same for every extension point. The extension encounters an error if it tries to reference a dictionary that is not present at a particular extension point. (See the “Extension Dictionaries” section on page 30-23.)
Dictionaries

You access data in the request, response, and server through a dictionary interface. Extension points include three types of dictionaries—request, response, and environment:

- **Request dictionary**—Information associated with the DHCP request, along with all that came in the request itself. Data is string-, integer-, IP address-, and blob-valued.
- **Response dictionary**—Information associated with generating a DHCP response packet to return to the DHCP client. Data is string-, integer-, IP address-, and blob-valued.
- **Environment dictionary**—Information passed between the DHCP server and extension.

For a description of the dictionaries, see the “Extension Dictionaries” section on page 30-23.

You can also use the environment dictionary to communicate between an extension attached at different extension points. When encountering the first extension point at which some extension is configured, the DHCP server creates an environment dictionary. The environment dictionary is the only one in which the DHCP server does not fix the names of the allowable data items. You can use the environment dictionary to insert any string-valued data item.

Every extension point in the flow of control between the request and response for the DHCP client (all extension points except lease-state-change, depending on the cause of the change) share the same environment dictionary. Thus, an extension can determine that some condition exists, and place a sentinel in the environment dictionary so that a subsequent extension can avoid determining the same condition.

In the previous example, the extension at the post-packet-decode extension point determines that the packet was the interesting one—from a particular manufacturer device, BOOTP, and Token Ring—and then rewrites the hardware type from Token Ring to Ethernet. It also places a sentinel in the environment dictionary and then, at a very simple extension at the post-packet-encode extension point, rewrites the hardware type back to Token Ring.

Utility Methods in Dictionaries

Each dictionary has associated utility methods with which you can reset the trace level for an extension and log values to an output file.

Configuration Errors

Extensions can fail for many reasons. For example:

- The server cannot find the file.
- The entry point or init-entry point does not appear in the file.
- The extension itself can return a failure from an init-entry call.

By itself, an extension failure is not fatal and does not prevent the DHCP server from starting. However, if you configure that failed extension at any extension point, the server will not start. Therefore, to debug the configuration process, you can configure the extension at the init-entry point (see the “init-entry” section on page 30-29) without attaching it to an extension point. When you complete that process successfully, you can attach your extension to an extension point.
Communicating with External Servers

You can write extensions that communicate with external servers or databases to affect the client class or validate incoming DHCP client requests. Writing such extensions is a complex task, requiring considerable skill and debugging expertise. Such extensions must be multithreaded, and need to communicate very swiftly with the external server if the DHCP server performance is to remain at an acceptable level.

Performance degradations can result from extensions stalling the threads that are processing requests. A thread stalls while an extension communicates with an external server. If this interaction takes more than 50 to 100 milliseconds, this severely affects server performance. This might or might not impact you in the particular environment in which you deploy this extension.

One way to avoid having to communicate with an external server synchronously (that is, stalling the incoming DHCP client request processing to communicate with the external server) is to avoid performing this communication while processing the DHCP client request. This sounds obvious, and it also sounds, on the face of it, impossible. However, due to the nature of the DHCP client-server protocol, there is a way to decouple the access to the external server from the DHCP client request processing.

To avoid this bottleneck, use a caching mechanism as part of the extension. When the server calls the extension for a request, have it check a cache (with proper locking to avoid multithreading problems) for the client data. If the client is:

- In the cache (and did not expire), have the extension accept or reject the request depending on the data in the cache.
- Not in the cache, have the extension queue a request to the external server (preferably over UDP), and then drop the DHCP client request. By the time the client retransmits the request, the data should be in the cache.

This caching mechanism requires the extension to have a receiver thread (started and stopped in the init-entry extension point). This thread reads the socket and updates the cache with the response. This thread (or a separate one) would also need to time out and remove old items from the cache. Using a single thread, however, might require setting a larger receive socket buffer size.

These techniques are only necessary if the load on the DHCP server is high and the speed of the external server is not high enough to support the required performance of the DHCP server load. However, this situation turns out to be all too common in practice. And, consider what can happen if the external server is unreachable (when connection timeouts are likely to be for minutes and not seconds).

Recognizing Extensions

The DHCP server only recognizes extensions when it initially configures itself at start or reload time. You can change an extension or the configuration for extensions in general. However, until you reload or restart the server, the changes have no effect. Forgetting to reload the DHCP server can be a frequent source of errors while debugging extensions.

The reason Cisco Prime IP Express requires a reload is to ensure minimum processing impact by preloading extensions and getting them ready at server configuration time. While this approach is useful in production mode, it might cause some frustration when you debug extensions.

Multiple Extension Considerations

You can register multiple extensions at any extension point. The DHCP server runs all the extensions attached to an extension point before resuming processing, the conditions being:
An extension should not explicitly set a data item unless the extension explicitly requires that behavior. For example (as described for the drop environment dictionary data item in Table 30-5 on page 30-25), extensions can request dropping the client packet at most extension points.

The server calls the first extension registered at an extension point with drop set to False. One or more extensions can set this to True or False. If all extensions were to explicitly set drop to either True or False, then the server would take whatever action the last extension to run requested.

This might not be the desired behavior. Thus, for this data item, it is better for an extension to set drop to True only if it wants the packet to be dropped. That way, if all extensions play by this rule, the packet would be dropped if any of the extensions request it.

An extension might want to return immediately if drop is True, as there may not be a need for the extension to do its processing if another one desires the packet to be discarded.

If the environment dictionary is used to store items for use in later extension points, those data item names might want to use a prefix or suffix that is unique to that extension. This reduces the chance for data item name conflicts.

At least one environment dictionary data item, the user-defined-data (see Table 30-5 on page 30-25) that you can use to store data with a lease (for DHCPv4) or client (DHCPv6), requires special attention.

This data item can be difficult for more than one extension to use, unless those extensions take special care to preserve and recognize each other’s values. Thus, it might not be possible for more than one extension to assume it can use this data item.

You must specify whether the extensions should be run first, or last, if such a need exists. For example, you should run extensions that cause the server to drop certain packets first, because this reduces the processing burden on the server (assuming the remaining extensions return immediately if drop is true).

### Tcl Extensions

If you choose to write your extensions in Tcl, you should understand the Tcl API, how to handle errors and Boolean variables, and how to initialize Tcl extensions. Cisco Prime IP Express uses Tcl version 8.4.

#### Related Topics

- [Tcl Application Program Interface](#)
- [Dealing with Tcl errors](#)
- [Handling Boolean Variables in Tcl](#)
- [Configuring Tcl Extensions](#)
- [Init-Entry Extension Point in Tcl](#)

#### Tcl Application Program Interface

Every Tcl extension has the same routine signature:

```tcl
proc yourentry { request response environ } { # your-code }
```

To operate on the data items in any dictionary, you must treat these arguments as commands. Thus, to get the giaddr of the input packet, you would write:

```tcl
set my_giaddr [ $request get giaddr ]
```
This sets the Tcl variable `my_giaddr` to the string value of the `giaddr` in the packet; for example, 10.10.1.5 or 0.0.0.0.

You could rewrite the `giaddr` in the input packet by using this Tcl statement:

```
$request put giaddr "1.2.3.4"
```

To configure one routine entry for multiple extension points and to alter its behavior depending on the extension point from which the server calls it, the DHCP server passes the ASCII name of the extension point in the environment dictionary under the key `extension-point`.

For some sample Tcl extensions, see the Cisco Prime IP Express directory; by default:

- **Linux** — `/opt/nwreg2/local/examples/dhcp/tcl`
- **Windows** — `\Program Files (x86)\Cisco Prime IP Express\Local\examples\dhcp\tcl`

### Dealing with Tcl errors

You generate a Tcl error if you:

- Reference a dictionary that is not available.
- Reference a dictionary data item that is not available.
- Request a put operation on an invalid data item, for example, an invalid IP address.

In these cases, the extension immediately fails unless you surround the statement with a catch error statement:

```
catch { $request put giaddr "1.2.3.a" } error
```

### Handling Boolean Variables in Tcl

In the environment dictionary, the Boolean variables are string-valued and have a value of `true` or `false`. The DHCP server expects an extension to set the value to `true` or `false`. However, in the request or response dictionaries, Boolean values are single-byte numeric format, and `true` is 1 and `false` is 0. While more efficient for the C/C++ extensions, this approach does make the Tcl API a bit more complex.

### Configuring Tcl Extensions

To configure a Tcl extension, write it and place it in the extensions directory. For UNIX and Linux, this is the `/opt/nwreg2/extensions/dhcp/tcl` directory. For Windows, this is the `\Program Files\Cisco Prime IP Express\extensions\dhcp\tcl` directory.

When the DHCP server configures an extension during startup, it reads the Tcl source file into an interpreter. Any syntax errors in the source file that would render Tcl interpreter unable to load the file would also fail the extension. Typically, the DHCP server generates an error traceback in the log file from Tcl to help you to find the error.
Init-Entry Extension Point in Tcl

Tcl extensions support the `init-entry` extension point (see the “init-entry” section on page 30-29), and the arguments supplied in the `init-args` parameter to the command line appear in the environment dictionary associated with the key `arguments`.

Multiple Tcl interpreters can run in the DHCP server, for performance purposes, each in its own Tcl context. The server calls the Tcl extension once at the `init-entry` point for every Tcl context (interpreter) it runs. Ensure that your Tcl extension `init-entry` is robust, given multiple calls.

Information cannot flow between the Tcl contexts, but the `init-entry` can initialize global Tcl variables in each Tcl interpreter that any Tcl extension can access, regardless of the interpreter.

Note that all Tcl extensions share the Tcl interpreters. If your Tcl extension initializes global variables or defines procedures, ensure that these do not conflict with some other Tcl extension global variables or procedure names.

C/C++ Extensions

All DHCP C/C++ extensions are `dex` extensions, short for DHCP Extension.

Related Topics

- C/C++ API, page 30-9
- Using Types in C/C++, page 30-10
- Building C/C++ Extensions, page 30-10
- Using Thread-Safe Extensions in C/C++, page 30-10
- Configuring C/C++ Extensions, page 30-11
- Debugging C/C++ Extensions, page 30-11

C/C++ API

The routine signature for both the `entry` and `init-entry` routines for the C/C++ API is:

```c
typedef int (DEXAPI * DexEntryPointFunction)(
    int iExtensionPoint,
    dex_AttributeDictionary_t* pRequest,
    dex_AttributeDictionary_t* pResponse,
    dex_EnvironmentDictionary_t* pEnviron );
```

Along with pointers to three structures, the integer value of the extension point is one of the parameters of each routine.

The C/C++ API is specifically constructed so that you do not have to link your shared library with any Cisco Prime IP Express DHCP server files. You configure the entry to your routine when you configure the extension. The necessary call-back information for the operations to perform on the request, response, and environment dictionaries is in the structures that comprise the three dictionary parameters passed to your extension routine.

The DHCP server returns all binary information in network order, which is not necessarily properly aligned for the executing architecture.
Using Types in C/C++

Many C/C++ routines are available that use types, for example, `getByType()`. These routines are designed for use in performance-sensitive environments. The reasoning behind these routines is that the extension can acquire pointers to types once, for example, in the init-entry point, and thereafter use the pointers instead of string-valued names when calling the routines of the C/C++ API. Using types in this manner removes one hash table lookup from the extension processing flow of execution, which should improve (at least fractionally) the performance of any extension.

Building C/C++ Extensions

The directories `/opt/nwreg2/examples/dhcp/dex (UNIX and Linux)` and `\Program Files\Cisco Prime IP Express\examples\dhcp\dex (Windows)` contain sample C/C++ extension code, as well as a short makefile designed to build the sample extensions. To build your own extensions, you need to modify this file. It has sections for Microsoft Visual C++, GNU C++, and SunPro C++. Simply move the comment lines to configure the file for your environment.

Your extension needs to reference the include file `dex.h`. This file contains the information your program needs to use the C/C++ API. When building C/C++ extensions on Windows, remember to add your entry points to the .def file.

After you build the .dll (Windows) or .so (UNIX) file (all dex extensions are shared libraries), you need to move them into the `/opt/nwreg2/extensions/dhcp/dex` directory (UNIX), or the `\Program Files\Cisco Prime IP Express\extensions\dhcp\dex` directory (Windows). You can then configure them.

Using Thread-Safe Extensions in C/C++

The DHCP server is multithreaded, so any C/C++ extensions written for it must be thread-safe. Multiple threads, and possibly multiple processors, must be capable of calling these extensions simultaneously at the same entry point. You should have considerable experience writing code for a multithreaded environment before designing C/C++ extensions for Cisco Prime IP Express.

Caution

All C/C++ extensions must be thread-safe. If not, the DHCP server will not operate correctly and will fail in ways that are extremely difficult to diagnose. All libraries and library routines that these extensions use must also be thread-safe.

On several operating systems, you must ensure that the runtime functions used are really thread-safe. Check the documentation for each function. Special thread-safe versions are provided (often `functionname_r`) on several operating systems. Because Windows provides different versions of libraries for multithreaded applications that are thread-safe, this problem usually does not apply.

Be aware that if any thread makes a non-thread-safe call, it affects any of the threads that make up the safe or locked version of the call. This can cause memory corruptions, server failures, and so on.

Diagnosing these problems is extremely difficult, because the cause of these failures are rarely apparent. To cause a server failure, you need very high server loads or multiprocessor machines with many processes. You might need running times of several days. Often, problems in extension implementation might not appear until after sustained periods of heavy load.

Because some runtime or third-party libraries might make non-thread-safe calls that you cannot detect, check your executables as to what externals are being linked (`nm` on UNIX).
If the routines of a library call the routines without the _r suffixes, displayed in the following table, the library is not thread-safe, and you cannot use it. The interfaces to the thread-safe versions of these library routines can vary based on operating system.

```
asctime_r  getgrid_r  getnetent_r  getrpcbyname_r  lgamma_r
ctermid_r  getgrnam_r  getprotobyname_r  getrcent_r  localtime_r
ctime_r   gethostbyaddr_r  getprotobynumber_r  getservbyname_r  nis_sperror_r
fgetgrent_r  gethostbyname_r  getprotoent_r  getservbyport_r  rand_r
fgetpwent_r  gethostent_r  getpwnam_r  getservent_r  readdir_r
fgetspent_r  getlogin_r  getpwent_r  getspent_r  strtok_r
gamma_r  getnetbyaddr_r  getpwuid_r  getspnam_r  tmpnam_r
getgrent_r  getnetbyname_r  getrpcbyname_r  gmtime_r  ttyname_r
```

### Configuring C/C++ Extensions

Because the .dll and .so files are active when the server is running, it is not a good idea to overwrite them. After you stop the server, you can overwrite the .dll and .so files with newer versions.

### Debugging C/C++ Extensions

Because your C/C++ shared library runs in the same address space as the DHCP server, and receives pointers to information in the DHCP server, any bugs in your C/C++ extension can very easily corrupt the DHCP server memory, leading to a server failure. For this reason, use extreme care when writing and testing a C/C++ extension. Frequently, you should try the approach to an extension with a Tcl extension and then code the extension in C/C++ for increased performance.

### Related Topics

- Pointers into DHCP Server Memory in C++, page 30-11
- Init-Entry Entry Point in C++, page 30-12

### Pointers into DHCP Server Memory in C++

The C/C++ extension interface routines return pointers into DHCP server memory in two formats:

- **char** pointer to a series of bytes.
- Pointer to a structure called an abytes_t, which provides a pointer to a series of bytes with an associated length (defined in dex.h).
In both cases, the pointers into DHCP server memory are valid while the extension runs at that extension point. They are also valid for the rest of the extension points in the series processing this request. Thus, an abytes_t pointer returned in the post-packet-decode extension point is still valid in the post-send-packet extension point.

The pointers are valid for as long as the information placed in the environment dictionary is valid. However, there is one exception. One C/C++ routine, getType, returns a pointer to an abytes_t that references a type. These pointers are valid through the entire life of the extension. Typically, the server would call this routine in the init-entry extension point and save the pointers to the abytes_t structures that define the types in the static data of the shared library. Pointers to abytes_t structures returned by getType are valid from the init-entry call for initialization until the call for uninitialization.

Init-Entry Entry Point in C/C++

The DHCP server calls the init-entry extension point (see the “init-entry” section on page 30-29) once when configuring each extension and once when unconfiguring it. The dex.h file defines two extension point values passed as the extension points for the configure and unconfigure calls: DEX_INITIALIZE for configure and DEX_UNINITIALIZE for unconfigure. The environment dictionary value of the extension-point data item is initialize or uninitialize in each call.

When calling the init-entry extension point for initialize, if the environment dictionary data item persistent contains the value true, you can save and use the environment dictionary pointer any time before the return from the uninitialize call. In this way, background threads can use the environment dictionary pointer to log messages in the server log file. Note that you must interlock all access to the dictionary to ensure that at most one thread processes a call to the dictionary at a time. You can use the saved dictionary pointer up to when the extension returns from the uninitialize call. This way, the background threads can log messages during termination.

DHCP Request Processing Using Extensions

The Cisco Prime IP Express DHCP server has extension points to which you can attach your own extensions. They have descriptive names that indicate where in the processing flow of control to use them.

Because the extension points are tied to the processing of input requests from DHCP clients, it is helpful to understand how the DHCP server handles requests. Request processing comes in three general stages:

1. Initial request processing (see Table 30-1)
2. DHCPv4 or DHCPv6 processing (see Table 30-2 on page 30-13)
3. Final response processing (see Table 30-3 on page 30-13)

<table>
<thead>
<tr>
<th>Client Request Processing Stage</th>
<th>Extension Point Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Receive a packet from a DHCP client.</td>
<td>pre-packet-decode</td>
</tr>
<tr>
<td>2. Decode the packet.</td>
<td>post-packet-decode</td>
</tr>
<tr>
<td>3. Determines the client-classes.</td>
<td>post-class-lookup</td>
</tr>
<tr>
<td>4. Modifies the client-class.</td>
<td></td>
</tr>
</tbody>
</table>
### Chapter 30      Using Extension Points

**Cisco Prime IP Express 8.2 User Guide**

**Table 30-1  Initial Request Processing Using Extensions (continued)**

<table>
<thead>
<tr>
<th>Client Request Processing Stage</th>
<th>Extension Point Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. Processes the client-classes, looking up clients.</td>
<td>pre-client-lookup</td>
</tr>
<tr>
<td></td>
<td>post-client-lookup</td>
</tr>
<tr>
<td>6. Build a response container from the request.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 30-2  DHCPv4 or DHCPv6 Request Processing Using Extensions**

<table>
<thead>
<tr>
<th>Client Request Processing Stage</th>
<th>Extension Point Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. In DHCPv4, find a lease already associated with this client, if any, or locate a new lease for the client.</td>
<td></td>
</tr>
<tr>
<td>2. Serialize all requests associated with this client (processing continues when the request reaches the head of the serialization queue).</td>
<td></td>
</tr>
<tr>
<td>3. In DHCPv6, process the client request, generating leases if necessary. The server tries to provide the client with at least one preferred lease for each usable prefix per binding. You can generate leases and change lease states multiple times for a client request, but not for reserved leases.</td>
<td>generate-lease and lease-state-change (multiple calls are possible for both in DHCPv6)</td>
</tr>
<tr>
<td>4. Determine if the lease is (still) acceptable for this client (can occur multiple times in DHCPv6).</td>
<td>check-lease-acceptable</td>
</tr>
<tr>
<td>5. Initiate DNS Update operations as necessary (can occur multiple times in DHCPv6).</td>
<td></td>
</tr>
</tbody>
</table>

**Table 30-3  Final Response Processing Using Extensions**

<table>
<thead>
<tr>
<th>Client Response Processing Stage</th>
<th>Extension Point Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Gather all the data to include in the response packet.</td>
<td></td>
</tr>
<tr>
<td>2. Write to the lease database.</td>
<td></td>
</tr>
<tr>
<td>3. Prepare the response packet for encoding.</td>
<td>pre-packet-encode</td>
</tr>
<tr>
<td>4. Encode the response packet for transmission to the client.</td>
<td>post-packet-encode</td>
</tr>
<tr>
<td>5. Send the packet to the client.</td>
<td>post-send-packet</td>
</tr>
<tr>
<td>6. Release all context for the client and request.</td>
<td>environment-destructor</td>
</tr>
</tbody>
</table>

These steps and additional opportunities for using extensions are explained in the following sections. The extension points are indicated in **bold**.

**Related Topics**

- Enabling DHCPv6 Extensions, page 30-14
- Receiving Packets, page 30-14
- Decoding Packets, page 30-14
- Determining Client-Classes, page 30-15
- Modifying Client-Classes, page 30-15
- Processing Client-Classes, page 30-15
Enabling DHCPv6 Extensions

By default, extensions are assumed to support only DHCPv4. To write DHCPv6 extensions, you must implement an init-entry extension point that must:

1. Set the dhcp-support environment data item to v4 (for DHCPv4 only, the preset value), v6 (for DHCPv6 only), or v4,v6 (for DHCPv4 and DHCPv6). This data item indicates to the server what the extension is willing to support.

2. Set the extension-extension-api-version environment data item to 2. (The dhcp-support data item is ignored unless the extension-extension-api-version is set to 2.)

You might need to write separate extensions for DHCPv4 and DHCPv6, because of the differences in packet formats, DHCP protocol, and internal server data. However, the fundamentals of both kinds of extensions are very much the same.

The server calls these extension points at essentially the same places during processing, although it can call some DHCPv6 extension points multiple times due to the possibility of multiple lease requests per client.

Receiving Packets

The DHCP server receives DHCPv4 packets on port 67 and DHCPv6 packets on port 547 (the DHCP input ports) and queues them for processing. It attempts to empty the UDP input queue as quickly as possible and keeps all of the requests that it receives on an internal list for processing as soon as a free thread is available to process them. You can configure the length of this queue, and it will not grow beyond its maximum configured length.

Decoding Packets

When a free thread is available, the DHCP server allocates to it the task of processing an input request. The first action it takes is to decode the input packet to determine if it is a valid DHCP client packet. As part of this decoding process, the DHCP server checks all options to see if they are valid—if the lengths of the options make sense in the overall context of the request packet. It also checks all data in the DHCP request packet, but takes no action on any data in the packet at this stage.
Use the pre-packet-decode extension point to rewrite the input packet. After the DHCP server passes this extension point, it stores all information from the packet in several internal data structures to make subsequent processing more efficient.

**Determining Client-Classes**

If you configure an expression in the client-class-lookup-id, it is at this stage that the DHCP server evaluates the expression (see Chapter 26, “Using Expressions” for a description of expressions). The result of the expression is either <null>, or something converted to a string. The value of the string must be either a client-class name or <none>. In the case of <none>, the server continues to process the packet in the same way as if there were no client-class-lookup-id configured. In the case of a <null> response or an error evaluating the client-class-lookup-id, the server logs an error message and drops the packet (unless an extension configured at the post-class-lookup extension point specifically instructs the server not to drop the packet). As part of the process of setting the client-class, the DHCP server evaluates any limitation-id configured for that client-class and stores it with the request.

**Modifying Client-Classes**

After the DHCP server evaluates the client-class-lookup-id and sets the client-class, it calls any extension attached to the post-class-lookup extension point. You can use that extension to change any data that the client-class caused to become associated with the request, including the limitation-id. The extension also learns if the evaluation of the client-class-lookup-id dropped the packet. The extension not only finds out if it needs to drop the packet, it instructs the server not to drop the packet if it wants the server not to do so.

Also, an extension running at the post-class-lookup extension point can set a new client-class for the request, and uses the data from that client-class instead of the current one. This is the only extension point where setting the client-class actually uses that client-class for the request.

**Processing Client-Classes**

If you enabled client-class processing, the DHCP server performs it at this stage.

Use the pre-client-lookup extension point to affect the client to look up, possibly by preventing the lookup or supplying data that overrides the existing data. After the DHCP server passes the pre-client-lookup extension point, it looks up the client (unless the extension specifically prevents it) in the local database or in an LDAP database, if one was configured.

After the server looks up the client, it uses the data in the client entry to fill in additional internal data structures. The DHCP server uses data in the specified client-class entry to complete any data that the client entry does not specify. When the DHCP server retrieves all the data stored in the various places in the internal data structures for additional processing, it runs the next extension point.

Use the post-client-lookup extension point to review the operation of the client-class lookup process, such as examining the internal server data structures filled in from the client-class processing. You can also use the extension point to change any data before the DHCP server does additional processing.
Building Response Containers

At this stage, the DHCP server determines the request type and builds an appropriate response container based on the input. For example, if the request is a DHCPDISCOVER, the server creates a DHCP OFFER response to perform the processing. If the input request is a BOOTP request, the server creates a BOOTP response to perform the response processing.

For DHCPv6, a server creates an ADVERTISE or REPLY packet, depending on the request.

Determining Networks and Links

The DHCP server must determine the subnet from which every request originated and map that into a set of address pools, scopes, prefixes, or links that contain IP addresses.

For DHCPv4, internal to the DHCP server is the concept of a network, which, in this context, refers to a LAN segment or physical network. In the DHCP server, every scope or prefix belongs to a single network.

Some scopes or prefixes are grouped together on the same network because their network numbers and subnet masks are identical. Others are grouped because they are related through the primary-scope or prefix pointer.

The Cisco Prime IP Express DHCP server determines the network to use to process a DHCP client request in the following sequence:

1. Determining the source address, either the giaddr or, if the giaddr is zero, the address of the interface on which the request arrived.
2. Using this address to search for any scope or prefix that was configured in the server that is on the same subnet as this address. If the server does not find a scope or prefix, it drops the request.
3. After finding the scope or prefix, using its network in subsequent processing.

For DHCPv6 processing, see the “Determining Links and Prefixes” section on page 27-3.

Finding Leases

For DHCPv4, now that when the DHCP server establishes the network, it searches the hash table held at the network level to see if the network already knows the client-id. “Already knows,” in this context, means that this client previously received an offer or a lease on this network, and the lease was not offered to or leased by a different client since that time. Thus, a current lease or an available expired lease appears in the network level hash table. If the DHCP server finds a lease, it proceeds to the next step, which is to serialize all requests for the same IP address.

If the DHCP server does not find a lease, and if this is a BOOTP or DHCPDISCOVER request, the server looks for a reserved lease from a scope or prefix in the network.

If it finds a reserved lease, the server checks whether the scope or prefix and lease are both acceptable. The following must be true regarding the reserved lease and the scope or prefix that contains it:

- The lease must be available (not leased to another DHCP client).
- The scope or prefix must support the request type (BOOTP or DHCP).
- The scope or prefix must not be in a deactivated state.
- The lease must not be in a deactivated state.
The selection tags must contain all of the client selection-criteria and none of the client selection-criteria-excluded.

The scope or prefix must not be in a renew-only state.

If the reserved lease is acceptable, the server proceeds to the next step, which is to serialize all requests for the IP address. Having failed to find an existing or reserved lease for this client, the server now attempts to find any available IP addresses for this client.

The general process the DHCP server uses is to scan all of the scopes or prefixes associated with this network in round-robin order, looking for one that is acceptable for the client and also has available addresses. An acceptable scope or prefix has the following characteristics:

- If the client has selection-criteria associated with it, the selection tags must contain all of the client inclusion criteria.
- If the client has selection-criteria-excluded associated with it, the selection tags must contain none of the client exclusion criteria.
- The scope or prefix must support the client request type—If the client request is a DHCPREQUEST, you must enable the scope or prefix for DHCP. Likewise, if the request is a BOOTP request, you must enable the scope or prefix for BOOTP and dynamic BOOTP.
- It must not be in a renew-only state.
- It must not be in deactivated state.
- It must have an available address.

If the server does not find an acceptable scope or prefix, it logs a message and drops the packet.

For DHCPv6 processing, see the “Determining Links and Prefixes” section on page 27-3.

**Serializing Lease Requests**

Because multiple DHCP requests can be in process simultaneously for one client and lease, you must serialize DHCPv4 requests at the lease level. The server queues them on the lease and processes them in the order of queueing.

For DHCPv6, the server serializes on the client (per link) and not on the lease.

**Determining Lease Acceptability**

For DHCPv4, the DHCP server now determines if the lease is (still) acceptable for the client. In the case where this is a newly acquired lease for a first-time client, it will be acceptable. However, in the case where the server processes a renewal for an existing lease, the acceptability criteria might have changed since the server granted the lease, and you need to check its acceptability again.

If the client has a reservation that is different from the current lease, the server first determines if the reserved lease is acceptable. The criteria for release acceptability are:

- The reserved lease must be available.
- The reserved lease must not be in a deactivated state.
- The scope or prefix must not be in a deactivated state.
- If the request is BOOTP, the scope or prefix must support BOOTP.
- If the request is DHCP, the scope or prefix must support DHCP.
If the client has any selection-criteria, the selection tags must contain all of the client inclusion criteria.

If the client has any selection-criteria-excluded, the selection tags must contain none of the client exclusion criteria.

If the client previously associated with this lease is not the current client, the scope or prefix must not be in a renew-only state.

If the reserved lease meets all of these criteria, the DHCP server considers the current lease unacceptable. If there is no reserved lease for this client, or the reserved lease did not meet the criteria for acceptability, the DHCP server examines the current lease for acceptability.

The criteria for acceptability are:

- The lease must not be in a deactivated state.
- The scope or prefix must not be in a deactivated state.
- If the request is BOOTP, the scope or prefix must support BOOTP. If the request is DHCP, the scope or prefix must support DHCP.
- If the client does not have a reservation for this lease, and the request is BOOTP, the scope or prefix must support dynamic BOOTP.
- If the client does not have a reservation for this lease, no other client can either.
- If the client has any selection-criteria, the selection tags must contain all of the client inclusion criteria.
- If the client has any selection-criteria-excluded, the selection tags must contain none of the client exclusion criteria.
- If the client previously associated with this lease is not the current client, the scope or prefix must not be in a renew-only state.

At this point in the DHCP server processing, you can use the check-lease-acceptable extension point. You can use it to change the results of the acceptability test. Do this only with extreme care.

Upon determining that a lease is unacceptable, the DHCP server takes different actions, depending on the particular DHCP request currently being processed.

- **DHCPDISCOVER**—The DHCP server releases the current lease and attempts to acquire a different, acceptable lease for this client.
- **DHCPREQUEST SELECTING**—The DHCP server sends a NACK to the DHCP client because the lease is invalid. The client should then immediately issue a DISCOVER request to acquire a new DHCPOFFER.
- **DHCPRENEW, DHCPREBIND**—The DHCP server sends a NACK to the DHCP client to attempt to force the DHCP client into the INIT phase (attempt to force the DHCP client into issuing a DHCPDISCOVER request). The lease is still valid until the client actually issues the request.
- **BOOTP**—The DHCP server releases the current lease and attempts to acquire a different, acceptable lease for this client.

Take extreme care with the check-lease-acceptable extension point. If the answer the extension point returns does not match the acceptability checks in the search for an available lease performed in a DHCPDISCOVER or dynamic BOOTP request, an infinite server loop can result (either immediately or
on the next DHCPDISCOVER or Bootstrap request). In this case, the server would acquire a newly available lease, determine that it was not acceptable, try to acquire a newly available lease, and determine that it was not acceptable, in a continuous loop.

### DHCPv6 Leasing

The DHCP server processes IPv6 lease requests by scanning the client request for IA_NA, IA_TA, and IA_PD options (see the “DHCPv6 Bindings” section on page 27-8). For each of these options, the server considers any leases that the client explicitly requests. If the lease already exists for the client and binding (IA option and IAID), the server determines if the lease is still acceptable. For leases that the client requests that do not already exist for the client, the server tries to give that lease to the client if:

- Another client or binding is not already using the lease.
- The prefix for the lease has the client-request flag set in its `allocation-algorithms` attribute.
- The lease is usable and on a usable prefix (see the “DHCPv6 Prefix Usability” section on page 30-19).

Next, the server tries to assure that clients are using reservations and that a client has a usable lease with a nonzero preferred lifetime on each usable prefix on the link. Thus, the server processes each of these bindings as follows:

1. Adds any client reservations (not already in use) to the binding, provided the reservation flag is set in the prefix `allocation-algorithms` attribute. The server uses the first binding of the appropriate type for the reservation; that is, it uses address leases for IA_NA bindings and prefix leases for IA_PD bindings.

2. If the client has no lease with a nonzero preferred lifetime on each prefix that the client can use, the server tries to allocate a lease to the client. The prefix `allocation-algorithms` flags control how the server allocates the lease.

### Related Topics

- DHCPv6 Prefix Usability, page 30-19
- DHCPv6 Lease Usability, page 30-19
- DHCPv6 Lease Allocation, page 30-20

### DHCPv6 Prefix Usability

A usable prefix:

- Is not deactivated.
- Did not expire.
- Allows leases of the binding type.
- Matches the client selection criteria, if any.
- Does not match the client selection exclusion criteria, if any.

### DHCPv6 Lease Usability

A usable lease is:

- Not unavailable.
• Not revoked.
• Not deactivated.
• Not reserved for a different client.
• Not subject to inhibit-all-renews or inhibit-renews-at-reboot.
• Renewable if being renewed (IA_TA leases are not renewable).
• Leasable with a nonzero valid lifetime.

DHCPv6 Lease Allocation

When the server needs to allocate a new lease on a prefix, it calls any extensions registered at the generate-lease extension point if the prefix extension flag is set in the allocation-algorithms attribute. (See the “generate-lease” section on page 30-35.) The extensions can supply the address (IA_NA or IA_TA binding) or prefix (IA_PD binding) to be assigned, request that the server use its normal allocation algorithm (if enabled in allocation-algorithms), or request the server to skip assigning a lease on this prefix. The server might call the extension again if it supplied an address or prefix that is not valid or is already in use.

If extensions are not allowed, there are no extensions registered, or the extension requests the normal allocation algorithm of the server, the server allocates a randomly generated address or finds the first best-fit available prefix (as controlled by the prefix allocation-algorithms attribute) and creates the lease.

Once the server has a lease and does an acceptability check on it (see the “DHCPv6 Lease Usability” section on page 30-19), the server calls any extensions registered at the check-lease-acceptable extension point to allow the extension to alter the acceptability of the lease. (See the “check-lease-acceptable” section on page 30-37.) You would typically only use this extension point to change an acceptable result into an unacceptable one; however, the server allows an unacceptable result to be changed to an acceptable one, although this is strongly discouraged because of possibly adverse consequences. If the lease is not acceptable, the server likely tries to allocate another lease; thus, use care to avoid an infinite loop. In some cases, you might need the check-lease-acceptable and generate-lease extension points for full control of the leases a client gets: generate-lease can request the server to skip allocation of the lease.

The server calls the check-lease-acceptable extension point for each client request for each lease.

Gathering Response Packet Data

In this stage of processing, the DHCP server collects all the data to send back in the DHCP response and determines the address and port to which to send the response. You can use the pre-packet-encode extension point to change the data sent back to the DHCP client in the response, or to change the address to which to send the DHCP response. (See the “pre-packet-encode” section on page 30-38.)

Caution

Any packets dropped at the pre-packet-encode extension point, whether they be DHCP or BOOTP packets, still show the address to be leased in the Cisco Prime IP Express lease state database, for as long as the remaining lease time. Because of this, it is advisable to drop packets at an earlier point.
Encoding Response Packets

In this stage, the DHCP encodes the information in the response data structure into a network packet. If this DHCP client requires DNS activity, the DHCP server queues a DNS work request to the DNS processing subsystem in the DHCP server. That request runs whenever it can, but generally not before sending the packet to the client. (See the “Processing DNS Requests” section on page 30-21.)

Updating Stable Storage

At this stage, the DHCP server ensures that the on-disk copy of the information is up to date with respect to the IP address before proceeding. For DHCPv6, this can involve multiple leases.

Sending Packets

Use the post-send-packet extension point (see the “post-send-packet” section on page 30-39) for any processing that you want to perform outside of the serious time constraints of the DHCP request-response cycle. After the server sends the packet to the client, it calls this extension point.

Processing DNS Requests

Here is a simplified view of what the DHCP server does to add names to DNS:

1. **Builds up a name to use for the A record**—The DHCP server creates the name that it will use in the forward (A record) DNS request. For DHCPv6, these are AAAA records. The DNS name comes from a variety of sources including the client-requested-host-name and client-domain-name data items, which are usually populated from options in the DHCP request, and the DNS update configuration (including the host-name-generator/v6-host-name-generator expressions).

2. **Tries to add the name, asserting that none exists yet**—At this stage, the prerequisites for the DNS name update request indicate that the name should not exist. If it succeeds, the DHCP server proceeds to update the reverse record.

3. **Tries to add the name, asserting that the server should supply it**—The server tries to add the hostname, asserting that the host exists and has the same TXT record (or DHCID record for DHCPv6) as the one that was sent.
   - If this succeeds, the server proceeds to the next step.
   - If it fails, the server checks if it exhausted its naming retries, in which case it exits and logs an error.
   - If it did not exhaust its naming entries, it returns to the first step, which is to build up a name for the A record.

   For DHCPv6, the server uses DHCID records instead of TXT records. Also, DHCPv6 clients can have multiple leases and the forward zones can be the same or potentially different.

4. **Updates the reverse record**—Now that the DHCP server knows which name to associate with the reverse (PTR) record, it can update the reverse record with no prerequisites, because it can assume it is the owner of the record. If the update fails, the DHCP server logs an error.
Tracing Lease State Changes

The server calls the `lease-state-change` extension point whenever (and only when) a lease changes state. The existing state is in the response dictionary `lease-state` data item. The new state is in the environment dictionary under `new-state`. The `new-state` never equals the existing state (if it did, the server would not call the extension). You should consider this extension to be read-only and not make modifications to any dictionary items, because the server calls it in many different places. Use this extension point only for tracking changes to a lease state.

Controlling Active Leasequery Notifications

The server determines whether a lease is queued for active leasequery notifications based on the `leasequery-send-all` attribute of `dhcp-listener`. If this attribute is enabled, the DHCP server always sends a notification to an active leasequery client. If disabled or unset, the DHCP server only sends notifications which are necessary to maintain accurate state in the active leasequery client.

To allow customer written extensions to control the sending of a lease (such as only on specific state changes), a new data item, `active-leasequery-control`, has been added to both the request and response dictionaries. These data items have three values:

- 0—unspecified (the server determines whether to send the notification)
- 1—send (the server will send the notification)
- 2—do not send (the server will not send the notification)

The `active-leasequery-control` data item is initialized as 0, unspecified.

Note: These data items may be written and read, but the value that is read is only the value that might have been previously written.

These data items can force the DHCP server to take specific actions after being written, but reading them without previously writing them will always return 0, unspecified. These data items will not let you determine the choices that the DHCP server makes when it comes to deciding whether or not to send a message to an active leasequery client concerning the changes (if any) made to a lease that is being processed. Thus, these data items are technically read/write, but reading them will only allow you to determine what you may have previously written into them.

These data items are examined (the response dictionary is examined first, then the request) when the lease is written to the internal lease state database as that is when the lease is also queued for active leasequery notification. This occurs after the check-lease-acceptable and lease-state-change extensions points, but prior to the pre-packet-encode extension point. Therefore, any changes made to these attributes at or after the pre-packet-encode extension point will be ignored.

Whether a lease is queued for active leasequery notification is determined as follows:

<table>
<thead>
<tr>
<th>Response's active-leasequery-control</th>
<th>Request's active-leasequery-control</th>
<th>Leasequery-send-all</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0—unspecified</td>
<td>0—unspecified</td>
<td>False or unset</td>
<td>Conditional (see leasequery-send-all attribute description)</td>
</tr>
<tr>
<td>0—unspecified</td>
<td>0—unspecified</td>
<td>True</td>
<td>Sent</td>
</tr>
<tr>
<td>0—unspecified</td>
<td>1—send</td>
<td>Ignored</td>
<td>Sent</td>
</tr>
</tbody>
</table>
Note

The `active-leasequery-control` of response and request is examined prior to any examination of the `leasequery-send-all` attribute.

If either of these dictionary data items has a value other than unspecified, that value will override any value configured in the `leasequery-send-all` attribute of the dhcp listener.

Note

You cannot control the sending of active leasequery information by writing a single extension that runs only at the `lease-state-change` extension point, because that extension point is only called when there is a change in state of a lease.

Lease state changes may not occur when you might expect them to. For example, if a lease is leased, and that same client goes through a DISCOVER/OFFER/REQUEST/ACK cycle, the `lease-state-change` extension point is not called since the lease does not actually go through a state change internally and it remains leased throughout the cycle. Thus, to gain absolute control over the transmission of information to active leasequery clients, you have to initialize the `active-leasequery-control` attribute in request processing, and then possibly alter it or override it by operating on the response dictionary value at the `lease-state-change` extension point.

---

### Extension Dictionaries

Every extension is a routine with three arguments. These arguments represent the request dictionary, response dictionary, and environment dictionary. Not every dictionary is available to every extension. Table 30-4 shows the extensions points and the dictionaries that are available to them.

#### Table 30-4 Extension Points and Relevant Dictionaries

<table>
<thead>
<tr>
<th>Extension Point</th>
<th>Dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td>init-entry</td>
<td>Environment</td>
</tr>
<tr>
<td>pre-packet-decode</td>
<td>Request, Environment</td>
</tr>
<tr>
<td>post-packet-decode</td>
<td>Request, Environment</td>
</tr>
<tr>
<td>pre-client-lookup</td>
<td>Request, Environment</td>
</tr>
<tr>
<td>pre-dns-add-forward</td>
<td>Environment (Deprecated extension point)</td>
</tr>
<tr>
<td>post-client-lookup</td>
<td>Request, Environment</td>
</tr>
<tr>
<td>post-class-lookup</td>
<td>Request, Environment</td>
</tr>
<tr>
<td>generate-lease</td>
<td>Request, Response, Environment</td>
</tr>
<tr>
<td>lease-state-change</td>
<td>Response, Environment</td>
</tr>
<tr>
<td>check-lease-acceptable</td>
<td>Request, Response, Environment</td>
</tr>
</tbody>
</table>
Extension Dictionaries

Note

When the server sends DHCPv6 Reconfigure messages, it can call the pre-packet-encode, post-packet-encode, and post-send-packet extension points without a request.

For the request and response dictionaries, you can use the isValid method to probe if the dictionary is available for an extension point.

Each of the three dictionaries consists of name-value pairs. The environment dictionary, which is available to every extension point, is the simplest dictionary. The request and response dictionaries are more complex and their data is typed. Thus, when you set a value in one of these dictionaries, you need to match the data type to the value. You can use the dictionaries for getting, putting, and removing values.

Related Topics

Environment Dictionary, page 30-24
Request and Response Dictionaries, page 30-26

Environment Dictionary

The environment dictionary is available at all extension points. It is strictly a set of name-value pairs in which both the name and the value are strings.

The DHCP server uses the environment dictionary to communicate with extensions in different ways at different extension points. At some extension points, the server places information in the environment dictionary for the extension to modify. In others, the extension can place values in the environment dictionary to control the flow or data after the extension finishes its processing.

The environment dictionary is unique in that an extension can put any name-value pair in it. Although you do not get an error for using undocumented name-value pairs, the server does not recognize them. These name-value pairs can be useful for your extension points to communicate data among them.

The DHCP server creates the environment dictionary when a DHCP request arrives and the dictionary remains with that request through the processing. Thus, an extension that runs at the post-packet-decode extension point can put data into the environment dictionary, and then an extension run at the pre-packet-encode extension point might read that data from the dictionary.

Note

The init-entry extension point has a unique environment dictionary.

Related Topics

General Environment Dictionary Data Items, page 30-25
Initial Environment Dictionary, page 30-26
General Environment Dictionary Data Items

The data items in Table 30-5 are valid in the environment dictionary at all extension points. (See the individual extension point sections for environment dictionary data items specific to each one.)

The data items are input, output, or both:
- Input—The DHCP server sets the value and inputs it to the extension.
- Output—The value is output to the DHCP server, which reads it, and acts upon it.

Table 30-5   General Environment Dictionary Data Items

<table>
<thead>
<tr>
<th>Environment Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drop</td>
<td>If the drop value is equal to the string true when the extension exits, the DHCP server drops the input packet and logs a message in the log file. Initially set to false. Available at most extension points, but not all (such as generate-lease).</td>
</tr>
<tr>
<td>extension-name</td>
<td>Name with which the extension was configured. You can configure the same piece of code as several different extensions and at several different extension points. This allows one piece of code to do different things, depending on how you configure it. The code can also use this string to find itself in the extension-name-sequence string, for which it needs to know its own name.</td>
</tr>
<tr>
<td>extension-name-sequence</td>
<td>Provides a comma-separated string representing the configured extensions for this extension point. It allows an extension to determine which extensions can run before and after it. The extension-name data item provides the currently running extension. For example, if you configure tclfirst as the first extension and dexscript as the fifth, the extension-name-sequence would contain &quot;tclfirst,,,,dexscript&quot;.</td>
</tr>
<tr>
<td>extension-point</td>
<td>Name of the extension point. For example, post-packet-decode.</td>
</tr>
<tr>
<td>extension-sequence</td>
<td>String that is the sequence number of the extension at the extension point.</td>
</tr>
<tr>
<td>log-drop-message</td>
<td>If the drop value is equal to the string true, and the log-drop-message value is equal to the string false when the extension exits, then the DHCP server drops the input packet, but does not log a message in the log file. Does not apply to init-entry.</td>
</tr>
<tr>
<td>trace-level</td>
<td>Setting this to a number makes that number the current setting of the extension-trace-level server attribute for all extensions processing this request.</td>
</tr>
</tbody>
</table>
You can configure an extension with `init-args` and `init-entry`. Alternatively, you can specify configuration information for an extension to read out of the environment dictionary. You can set the DHCP property `initial-environment-dictionary` with a series of attribute-value pairs, and each pair is available in every environment dictionary. Using this capability, you can specify a variety of configuration and customizing information. Any extension can simply read this data directly out of the environment dictionary, without having to store it in some static data area, as is required with the `init-args` or `init-entry` approach.

You can read the values defined using the `initial-environment-dictionary` approach from any environment dictionary. You can also define new values for any attributes that appear in the `initial-environment-dictionary`. These new values are then available for the life of that environment dictionary (usually the life of the request packet being processed). However, this does not change the contents of any other environment dictionary. Any new environment dictionary (associated with a different request) sees the attribute-value pairs defined by the `initial-environment-dictionary` property of the DHCP server.

In addition, these `initial-environment-dictionary` attribute-value pairs do not appear in an enumeration of the values of the environment dictionary. They are only available if you request an attribute value not currently defined in the environment dictionary. The attribute-value pairs do not actually appear in the environment dictionary. Thus, if you define a new value for one of the attributes, that new value does appear in the environment dictionary. If you later delete the value, the original one is again available if you should request it.

### Request and Response Dictionaries

The request and response dictionaries have a fixed set of accessible names. However, you cannot access all names from every extension point. These dictionaries make internal server data structures available to the extension for read-write or, in some cases, read-only access. Each data item has a particular data type. If you omit the correct data type (for C/C++ extensions) on a put operation, or if the DHCP server cannot convert it to the correct data type (for Tcl extensions), the extension will fail.

The request dictionary is available at the beginning of the processing of a request. After the server creates a response, both the request and response dictionaries are available. It is an error to access a response dictionary before it is available.
In general, you cannot use an extension to change information data in the server. In some cases, however, you can use an extension to change configured data, but only for the duration of the processing for just that single request.

Appendix C contains details on the options and data items available for the received client request (the Request Dictionary) and for the response sent (the Response dictionary).

**Related Topics**

- Decoded DHCP Packet Data Items, page 30-27
- Using Parameter List Option, page 30-28

**Decoded DHCP Packet Data Items**

The DHCP protocol is a request-response UDP-based protocol and, thus, the stimulation for a DHCP server operation is usually a DHCP request from a client. The result is usually a DHCP response sent back to that client.

The DHCP extension facility makes the information input in the DHCP request available to extensions at most of the extension points, and the information to be sent as a response to a DHCP request available at the `pre-packet-encode` extension point (see the “pre-packet-encode” section on page 30-38).

In addition to this DHCP packet-based information, there is additional data that the DHCP server uses when processing DHCP requests. This data becomes associated with either the DHCP request or the DHCP response as part of the architecture of the server. Much of this data is also made available to extensions, and much of it can be both read and written—in many cases altering the processing algorithms of the DHCP server.

The request and response dictionaries, therefore, contain two classes of data in each dictionary. They contain decoded packet data items as well as other request or response associated data items. The decoded packet data items are those data items directly contained in or derived from the DHCP request or DHCP response. Access to the decoded packet data items allows you to read and, in some cases, rewrite the DHCP request and DHCP response packet. *Figure 30-1 on page 30-27* shows the relationship between the request and the response dictionaries.

![Figure 30-1 Extensions Request and Response Dictionaries](image)

You can access information from the DHCP request packet, such as the `giaddr`, `ciaddr`, and all the incoming DHCP options by using the decoded packet data items in the request dictionary. Similarly, you can set the `giaddr` and `ciaddr`, and add and remove DHCP options in the outgoing DHCP response by accessing the decoded packet data items in the response dictionary.
It is important to realize that access to the packet information provided by the decoded packet data items is not all available to you. The specific data items available to that extension point are listed in the description of each extension point. Because the decoded packet data items are always accessible as a group, they are listed as a group.

You access DHCP options by name. If the option is not present, the server returns no data for that option. If you place an option into the decoded request or decoded response, it replaces any option with the same name already in the decoded request or decoded response, unless, in the put operation, you want the data specifically appended to existing data.

Some DHCP options can have multiple values. For example, the routers option can have one or more IP addresses associated with it. Access to these multiple values is by indexed operations on the option name.

A clear operation on the request or response dictionary removes all the options in the decoded packet.

**Using Parameter List Option**

There is one option, dhcp-parameter-request-list, that the DHCP server specially handles in two ways, available as either a:

- Multiple-valued option of bytes under the name dhcp-parameter-request-list.
- Blob (sequence of bytes) option under the name dhcp-parameter-request-list-blob.

You can get or put the option using either name. The DHCP server handles the dhcp-parameter-request-list (and its -blob variant as well) differently in the response dictionary than in the request dictionary. When you access this option in the request dictionary, it is just another DHCP option in the request dictionary. In the response dictionary, however, special processing takes place.

You can use the dhcp-parameter-request-list option in the response dictionary to control the order of the options returned to the DHCP or BOOTP client. When you put the option in the response dictionary, the DHCP server reorders the existing options so that the ones listed in the option are first and in the order that they appear in the list. Then, the remaining options appear in their current order after the last ones that were in the list. The DHCP server retains the list, and uses it to order any future options that it puts into the response, until it replaces the list with a new one.

When an extension does a get operation for the dhcp-parameter-request-list in the response dictionary, it does not look in the decoded response packet to find an option. Instead, the DHCP server synthesizes one that contains the list of all options currently in the decoded response packet.

**Extension Point Descriptions**

The following sections describe each extension point, their actions, and data items. For all the extension points, you can read the extension-point and set the trace-level data item values in the environment dictionary. For most extension points, you can also tell the server to drop the packet.

**Related Topics**

init-entry, page 30-29  
pre-packet-decode, page 30-30  
post-packet-decode, page 30-31  
post-class-lookup, page 30-32  
pre-client-lookup, page 30-32
The **init-entry** extension point is an additional one that the DHCP server calls when it configures or unconfigures the extension, which occurs when starting, stopping, or reloading the server. This entry point has the same signature as the others for the extension, but you can use only the environment dictionary. You do not configure the **init-entry** extension with `dhcp attachExtension` in the CLI, but you do so implicitly by defining an **init-entry** on an already configured extension. See Table 30-6 for the environment dictionary data items specific to **init-entry**.

### Table 30-6  init-entry Environment Dictionary Data Items

<table>
<thead>
<tr>
<th>Environment Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dhcp-support</code> (input/output)</td>
<td>Version or versions of DHCP for which the server should call the registered extension points for the extension. Can be <code>v4</code>, <code>v6</code>, or <code>v4,v6</code>.</td>
</tr>
<tr>
<td><code>extension-extensionapi-version</code> (output)</td>
<td>Minimum version of the extension API required by the extension.</td>
</tr>
<tr>
<td><code>init-args</code> (input)</td>
<td>Configure arguments by setting <code>init-args</code> on an existing extension point. These arguments are present for both the configure and unconfigure calls of the <strong>init-entry</strong> entry point. The extension point name for the configure call is <code>initialize</code>, and for the unconfigure call is <code>uninitialize</code>.</td>
</tr>
</tbody>
</table>
| `server-dhcp-support` (input) | The server sets this data item to indicate what the server is configured to support. Can be `v4`, `v6`, or `v4,v6`, depending on the DHCP server `dhcp-support` attribute setting (which requires setting expert attribute visibility=3) and whether any prefixes are configured:  
  - If `dhcp-support=both` and prefixes are not configured, then `server-dhcp-support` is set to `v4`.  
  - If `dhcp-support=both` and one or more prefixes are configured, then `server-dhcp-support` is set to `v4,v6`.  
  - If `dhcp-support=v4`, then `server-dhcp-support` is set to `v4`.  
  - If `dhcp-support=v6` and one or more prefixes are configured, then `server-dhcp-support` is set to `v6`. |
| `server-extensionapi-version` (input) | Version of the server extension API. |
**Note**

You must supply an **init-entry** extension point to enable extension points for DHCPv6 (or disable them for DHCPv4).

In addition to configuring an **init-entry** with the name of the entry point, you can also configure a string of arguments that the DHCP server loads in the environment dictionary under the string **arguments** before calling the **init-entry** point. Using **arguments**, you can create a customized extension by giving it different initialization arguments and thus not require a change to the code to elicit different behavior.

**Note**

The order in which the server calls extensions at the **init-entry** extension point can be different from reload to reload, or release to release.

**Caution**

An extension, when called to uninitialize, must terminate any threads it creates and clean up after itself before returning. Once the extension returns, the DHCP server unloads the extension from memory, which could result in a server failure if a thread an extension created is left running.

---

**pre-packet-decode**

The dictionaries available for **pre-packet-decode** are request and environment.

This extension point is the first one the DHCP server encounters when a request arrives. The server calls it after receiving a packet but before it decodes the packet (at the **post-packet-decode** extension point). An extension can use this extension point to examine a packet and alter it before the server decodes it, or cause the server to drop it.

Two key data items in the request dictionary are for use with **pre-packet-decode** are **client-packet** and **packet**. These can be used to examine the received packet, modify the packet, and write it back.

**Caution**

The request dictionary **client-packet** and **packet** data items used for **pre-packet-decode** are available at any extension point that has a request dictionary. However, you should not directly alter or replace the packet at any extension point other than **pre-packet-decode**, because doing so can have unexpected side effects. For example, the server might never pick up the changes to the packet, or options data can change unexpectedly during processing.

An extension that uses **getBytes** with client-packet or packet directly alters the bytes of packet by writing into the returned buffer. However, an extension must use **put** or **putBytes** to adjust the length of the packet (and the operation can fail if the packet is too big). For DHCPv6, adjusting the length of the client portion of the packet, if relayed, requires updating the lengths in the Relay Message options in the packet.

It is up to an extension to handle parsing the packet to locate what it needs and properly alter the packet, if that is the intent.

Because the server has not yet decoded the received packet, most request dictionary data items are not available (as the server normally fills them in from the received packet). Thus, this extension point must extract data directly from the packet. The extension must also properly handle incorrectly formatted packets.
If you enable incoming-packet-detail logging, the server logs the received packet after calling the extensions registered at this extension point. If DHCP server debug tracing is configured with V is 3 or more, the server also logs the packet before calling the extensions registered for this extension point, if at least one extension is registered.

**post-packet-decode**

The dictionaries available for `post-packet-decode` are request and environment.

**Related Topics**

- Extension Description, page 30-31
- Overriding Client Identifiers, page 30-31

**Extension Description**

This extension point immediately follows the decoding of the input packet and precedes any processing on the data in the packet. The primary activity for an extension at this point is to read information from an input packet and do something with it. For example, you might use it to rewrite the input packet.

The `post-packet-decode` extension point is one of the easiest extension points to use. If you can express the change in server behavior as a rewrite of the input DHCP or BOOTP packet, you should use this extension point. Because the packet was decoded, but not processed in any way, the number of side effects are very limited.

The `post-packet-decode` extension point is the only one at which you can modify the decoded input packet and ensure that the server recognizes all the modifications. You can have the extension drop the packet and terminate further processing by using the `drop` data item in the environment dictionary.

**Overriding Client Identifiers**

To override client identifiers (IDs), you can set an expression value for the `override-client-id` attribute for a client-class or use the `override-client-id` data item at the `post-packet-decode` extension point. The extension method maps the client to a different identifier than the server.

There is a variant of the extension data item where you can get or put the override client ID as a string: `override-client-id-string`. You can also request the data type of the override client ID through the read-only `override-client-id-data-type` data item.

Different values are returned based on how you put and get the `override-client-id` or its `override-client-id-string` variant (see Table 30-7 for some examples).

**Table 30-7**  
**Puts and Gets of Client ID Overrides**

<table>
<thead>
<tr>
<th>Action</th>
<th>Data Item Used</th>
<th>Put Value</th>
<th>Get Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>put</td>
<td><code>override-client-id</code></td>
<td>01:02:03:04</td>
<td></td>
</tr>
<tr>
<td>putBytes</td>
<td><code>override-client-id</code></td>
<td>01 02 03 04</td>
<td></td>
</tr>
<tr>
<td>get</td>
<td><code>override-client-id</code></td>
<td></td>
<td>01:02:03:04 (blob)</td>
</tr>
<tr>
<td>getBytes</td>
<td><code>override-client-id</code></td>
<td></td>
<td>01 02 03 04 (raw bytes)</td>
</tr>
<tr>
<td>get[Bytes]</td>
<td><code>override-client-id-string</code></td>
<td></td>
<td>01:02:03:04 (blob-as-string)</td>
</tr>
<tr>
<td>get[Bytes]</td>
<td><code>override-client-id-data-type</code></td>
<td></td>
<td>blob</td>
</tr>
</tbody>
</table>
**Table 30-7 Puts and Gets of Client ID Overrides (continued)**

<table>
<thead>
<tr>
<th>Action</th>
<th>Data Item Used</th>
<th>Put Value</th>
<th>Get Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>put[Bytes]</td>
<td>override-client-id-string</td>
<td>01:02:03:04 test</td>
<td></td>
</tr>
<tr>
<td>get[Bytes]</td>
<td>override-client-id-string</td>
<td>01:02:03:04 (string) test (string)</td>
<td></td>
</tr>
<tr>
<td>get[Bytes]</td>
<td>override-client-id-data-type</td>
<td>nstr</td>
<td></td>
</tr>
</tbody>
</table>

The equivalent client-override-client-id data items (that you can use in later extension points where the response dictionary is valid) function the same way, although they are read-only.

When using [v6-]override-client-id expressions, leasequery by client-id requests may need to specify the override-client-id attribute to correctly retrieve the information on the lease(s) for the client.

---

**post-class-lookup**

The dictionaries available for **post-class-lookup** are request and environment.

The server calls this extension point only if there is a client-class-lookup-id; otherwise, it is similar to a **post-packet-decode**. The server calls the **post-class-lookup** extension point after evaluating the client-class-lookup-id and setting the client-class data for this client.

On input to this extension point, the environment dictionary has the drop data item set to true or false. You can change this setting by extension to drop the packet (or not drop it), and the server recognizes the change. The server also looks at the log-drop-message to decide whether to log the drop.

The extension point can also set the client-class-name in the environment dictionary, which sets the named client-class for this packet, regardless of the previous client-class. This setting has an effect only if the drop environment dictionary data item value is false on exiting the extension point.

---

**pre-client-lookup**

The dictionaries available for **pre-client-lookup** are request and environment.

You can use this extension point only if you enabled client-class processing for your DHCP server. This extension point allows an extension to perform any or all of these actions:

- Modify the client that the server looks up during client-class processing.
- Specify individual data items to override those found from the client entry or client-class it specifies.
- Instruct the server to skip the client lookup altogether. In this case, the only client data used is data that the extension supplied in the environment dictionary.

Although the request dictionary is available to make decisions about the operation of an extension running at this extension point, the environment dictionary controls all the operations.
Related Topics

Environment Dictionary for pre-client-lookup, page 30-33
post-client-lookup, page 30-34
Overriding Client Identifiers, page 30-31

Environment Dictionary for pre-client-lookup

The environment dictionary data items in Table 30-8 are the control data items available at pre-client-lookup for clients and client-classes.

If you set the environment dictionary data items in Table 30-9 on page 30-33, their values override those determined from the client lookup (either in the internal database or from LDAP). If you do not add anything to the dictionary, the server uses what is available in the client lookup.

**Table 30-8  pre-client-lookup Environment Dictionary Control Data Items**

<table>
<thead>
<tr>
<th>Environment Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-specifier</td>
<td>Name of the client the client-class processing code looks up, in CNRDB or LDAP. If you change the name at this extension point, the DHCP server looks up the client you specify.</td>
</tr>
<tr>
<td>default-client-class-name</td>
<td>Instructs the server to use the value associated with the default-client-class-name option as the class-name if:</td>
</tr>
<tr>
<td></td>
<td>• The client-specifier data item was not specified in the pre-client-lookup script.</td>
</tr>
<tr>
<td></td>
<td>• The server could not locate the specific client entry.</td>
</tr>
<tr>
<td></td>
<td>The default-client-class-name data item then assumes precedence over the class-name associated with the default client.</td>
</tr>
<tr>
<td>release-by-ip</td>
<td>Applies to DHCPRELEASE requests only. If set to true, instructs the server to release the lease by the IP address if it cannot retrieve the lease by the client-id as derived from the DHCPRELEASE request.</td>
</tr>
<tr>
<td>skip-client-lookup</td>
<td>The value is determined by the server configuration. If set to true, the DHCP server skips the normal client lookup that it would have performed immediately upon exit from this extension.</td>
</tr>
<tr>
<td></td>
<td>The only data items used to describe this client are those placed in the environment dictionary (see Table 30-9).</td>
</tr>
</tbody>
</table>

**Table 30-9  pre-client-lookup Environment Dictionary Override Data Items**

<table>
<thead>
<tr>
<th>Environment Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>Convert this string to a number and use the result as the action. The numbers you can use are 0 (for none) and 1 (for exclude).</td>
</tr>
<tr>
<td>authenticate-until</td>
<td>Absolute time, measured in seconds, from January 1, 1970. Use to indicate the time at which the client authentication expires.</td>
</tr>
<tr>
<td></td>
<td>When the client authentication expires, the DHCP server uses the values in the client unauthenticated-client-class option instead of its client-class to fill in missing data items in the client entry.</td>
</tr>
</tbody>
</table>
Table 30-9  pre-client-lookup Environment Dictionary Override Data Items (continued)

<table>
<thead>
<tr>
<th>Environment Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-class-name (input/output)</td>
<td>Use the client-class specified by this data item to fill in the missing information in the client entry. If there is no client-class corresponding to the name specified, the DHCP server logs a warning and continues processing. If you specify none, the DHCP server acts as if the client entry did not include the client-class name.</td>
</tr>
<tr>
<td>domain-name (output)</td>
<td>Use this domain name for the client DNS operations in preference to the one specified in the DNS update configuration. The DNS server shown as the primary server for the domain in the scope or prefix must also be the primary server for the domain you specified. If the domain name is not overridden in the client or client-class entry, the DHCP server uses the domain name from the scope or prefix. If the client entry or the extension contains the word none, the DHCP server uses the domain name from the scope or prefix.</td>
</tr>
<tr>
<td>host-name (output)</td>
<td>Use this for the client in preference to the host-name options specified in the input packet, or any data from the client or client-class entry. If you set this to none, the DHCP server does not use any information from the client or client-class entry, but uses the name from the client request.</td>
</tr>
<tr>
<td>policy-name (input/output)</td>
<td>Use this policy as the policy specified for the client entry, overriding any policy specified by that client entry.</td>
</tr>
<tr>
<td>selection-criteria (input/output)</td>
<td>List of comma-separated strings, each specifying (for this input packet) the selection criteria for the client. Any scope or prefix the client uses must have all of these selection tags. Use this data item to override any criteria specified in the client or client-class entry. If you do, the DHCP server does not use the client entry selection criteria, independent of whether they were stored in the local or LDAP database. If you set this data item to none, the DHCP server does not use selection tags for the packet. If you set this to a null string, the DHCP server treats it as if it were not set and uses the selection criteria from the client or client-class entry.</td>
</tr>
<tr>
<td>unauthenticated-client-class-name (input/output)</td>
<td>Name of the client-class to use if the server does not authenticate the client. If you want to indicate without specifying the unauthenticated-client-class-name, use an invalid client-class name as the value of this data item. You can use the value none or any name that is not a client-class name. The DHCP server logs an error that the client-class is not present.</td>
</tr>
</tbody>
</table>

**post-client-lookup**

The dictionaries available for post-client-lookup are request and environment.

You can use this extension point to examine the results of the entire client-class processing operation, and take an action based on those results. You might want to use it to rewrite some of the results, or to drop the packet. If you want to override the hostname in the packet returned from the client-class.
processing from an extension running at the **post-client-lookup** extension point, set the hostname to the **client-requested-host-name** data item in the request dictionary. This causes Cisco Prime IP Express to look to the server as though the packet came in with whatever string you specified in that data item.

You also can use this extension point to place some data items in the environment dictionary to affect the processing of an extension running at the **pre-packet-encode** extension point (see the “pre-packet-encode” section on page 30-38), where it might load different options into the response packet or take other actions.

### Environment Dictionary for post-client-lookup

The environment dictionary data items included in Table 30-10 are available at **post-client-lookup**.

#### Table 30-10  post-client-lookup Environment Dictionary Data Items

<table>
<thead>
<tr>
<th>Environment Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>client-specifier</strong> (input)</td>
<td>Name of the client that the client-class processing looked up.</td>
</tr>
</tbody>
</table>
| **cnr-ldap-query-failed** (input) | The DHCP server sets this attribute to ease recovery from LDAP server failures so that a post-client-lookup script can respond to an LDAP server failure. The DHCP server, after a client lookup, sets this flag to **true** if the LDAP query failed because of an LDAP server error. If the server received a response from the LDAP server, one of two conditions occurs:  
  - It sets the flag to **false**.  
  - The **cnr-ldap-query-failed** attribute does not appear in the environment dictionary. |

### generate-lease

The dictionaries available for **generate-lease** are request, response, and environment. This extension point is available for DHCPv6 only. See Table 30-11 for the environment dictionary data items that apply to this extension point.

#### Table 30-11  generate-lease Environment Dictionary Data Items

<table>
<thead>
<tr>
<th>Environment Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>attempts</strong> (input)</td>
<td>Number of times that the server calls this extension for a single lease.</td>
</tr>
<tr>
<td><strong>default-prefix-length</strong> (input)</td>
<td>Set to the default prefix length (from policies). The Expert mode longest-prefix-length and shortest-prefix-length data items, if not set, default to the default-prefix-length.</td>
</tr>
<tr>
<td><strong>extension-point</strong> (input)</td>
<td>Name of the extension point. For example, <strong>post-packet-decode</strong>.</td>
</tr>
<tr>
<td><strong>extension-sequence</strong> (input)</td>
<td>String that is the sequence number of this extension at this extension point.</td>
</tr>
<tr>
<td><strong>generated-address</strong> (output)</td>
<td>Address the extension wants the server to use for the lease.</td>
</tr>
</tbody>
</table>
You can use this extension point to generate a DHCPv6 address or prefix and allow the extension to control the address or prefix.

The server calls **generate-lease** only if the prefix is configured to allow extensions to be called during address allocation or prefix delegation—the extension flag must be set in the `allocation-algorithms` attribute for the prefix. When the server calls the **generate-lease** extension:

- The server sets the prefix context for the response dictionary to the prefix on which the lease is to be created. (Calling `setObject` with DEX_PREFIX and DEX_INITIAL will return to this context.)
- No lease context exists, because the server has not yet created a lease. However, lease-binding data items, in particular `lease-binding-type` and `lease-binding-iaid` are available. (Calling `setObject` with DEX_LEASE and DEX_INITIAL returns to this context and also sets the prefix, because a lease context sets three contexts: lease, binding, and prefix.)
- The server sets the `skip-lease` environment dictionary data item to false.
- The server sets the (read-only) `attempts` environment dictionary data item with the number of times (starting with 1) it called the extension to create this lease.
- For prefix delegation, the following environment dictionary data items are available:
  - `prefix-length`—Prefix length (requested or default prefix length).
  - `default-prefix-length`—Default prefix length (from policies).
  - `longest-prefix-length`—Longest allowable prefix (from policies).
  - `shortest-prefix-length`—Shortest allowable prefix (from policies).

When the extension returns, it can:

- Request an explicit address (for stateful address assignment) by setting the address on the `generated-address` environment dictionary data item. If the address is not available for the client (that is, if the address is already in use) or is not contained by the prefix, the server might call this extension again.
- Request an explicit prefix (for prefix delegation assignment) by setting the prefix on the `generated-prefix` environment dictionary data item. If the prefix is not available for the client or is not contained by the prefix, the server might call this extension again. The prefix is not available for the client under the following conditions:
  - if the prefix is already in use
  - if it is contained in a shorter prefix that has already been delegated
  - if a longer prefix contained in it has already been delegated by the server
  The server will not reject the prefix if it is shorter or longer than allowed by the policy.
- Cause the server not to assign a lease by setting the `skip-lease` environment dictionary data item to true.

<table>
<thead>
<tr>
<th>Environment Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>generated-prefix</code> (output)</td>
<td>Delegated DHCPv6 prefix the extension wants the server to use for the lease.</td>
</tr>
<tr>
<td><code>prefix-length</code> (input)</td>
<td>Set to the requested or default prefix length.</td>
</tr>
<tr>
<td><code>skip-lease</code> (output)</td>
<td>Set to TRUE if the extension does not want the server to generate the lease.</td>
</tr>
</tbody>
</table>
• Allow normal address assignment or prefix delegation to occur by not setting any of the above.

The server calls the extension point at most 500 times for each lease (this limit is the same one that currently applies when the server randomly generates leases). The server calls an extension multiple times only if the extension supplies an unusable address or delegated prefix (that is not in range for the prefix or already exists).

**Note**
You cannot request the server to drop the packet at this extension point.

### check-lease-acceptable

The dictionaries available for `check-lease-acceptable` are request, response, and environment.

This extension point comes immediately after the server determines whether the current lease is acceptable for this client. You can use this extension to examine the results of that operation, and to cause the routine to return different results. See the “Determining Lease Acceptability” section on page 30-17.

The `acceptable` data item is available in the environment dictionary at this extension point. This is a read/write data item that the DHCP server initializes depending on if the lease is acceptable for this client. You can read and change this result in an extension. Setting the acceptable data item to true indicates that it is acceptable; setting it to false indicates that it is unacceptable.

### lease-state-change

The dictionaries available for `lease-state-change` are response and environment.

The server calls this extension point when a lease state changes for either all state changes or when exiting the state specified in the `exiting-state` environment data item (see Table 30-12 on page 30-37). The existing state is in the `lease-state` response dictionary data item. The new state is in the environment dictionary data item `new-state`. The server never calls the extension point if the new state matches the existing one.

Use this extension point mainly for read-only purposes, although you can place data in the environment dictionary so that other extension points can get it later.

The `lease-state-change` can also have a different environment dictionary, such as for lease expirations.

<table>
<thead>
<tr>
<th>Environment Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>exiting-state</code> (output)</td>
<td>For an extension attached to the <code>lease-state-change</code> extension point, if specified, the <code>lease-state-change</code> extension point is called only if the current state of the lease is the state specified by <code>exiting-state</code>. The extension is only called when the specified state is exited. If not specified, and the extension is attached to the <code>lease-state-change</code> extension point, the extension will be called for all state changes. If specified, the <code>exiting-state</code> must specify a valid lease state: available, offered, leased, expired, unavailable, released, other-available, pending-available, revoked. There is no strict state transition table. In a failover environment, the server that receives a binding update message sets the state to whatever its partner informs it to be, without requiring specific state transitions.</td>
</tr>
</tbody>
</table>
Extension Point Descriptions

Chapter 30 Using Extension Points

Related Topics

Environment Dictionary for lease-state-change, page 30-38

Environment Dictionary for lease-state-change

The current state is in the lease-state lease information data item in the response dictionary, and the state being changed to is in the environment dictionary under the new-state data item. The response lease-state and environment new-state data items are read-only.

pre-packet-encode

The dictionaries available for pre-packet-encode are request, response, and environment.

Note

For DHCPv6 Reconfigure messages, there is no request dictionary (because Reconfigure is a server-initiated message). Thus, enabled extensions should check the response msg-type for ADVERTISE or REPLY or use isValid on the request to ensure that the Reconfigure message exists.

post-packet-encode

The dictionaries available for post-packet-encode are request, response, and environment.

Note

For DHCPv6 Reconfigure messages, there is no request dictionary (because Reconfigure is a server-initiated message). Thus, enabled extensions should check the response msg-type for ADVERTISE or REPLY or use isValid on the request to ensure that the request dictionary exists.

The server calls this extension point after encoding a packet, but before sending it to the client. The server can thereby examine and alter the packet before it sends the packet to the client, or the extension can cause the server to drop the packet (although the server might have made changes to its internal and on-disk data that will not be backed out if the packet is dropped).

The client-packet and packet data items were added to the response dictionary with similar behavior as described for the request dictionary in the “pre-packet-decode” section on page 30-30. Note that this extension point is the only one that can request the response client-packet or packet data items, because no packet exists at any other extension point. Also, the server does not process the changes made to the packet; the server simply sends the altered packet to the client.

If you enable outgoing-packet-detail logging, the server logs the packet after calling the extensions registered at this extension point. If DHCP server debug tracing is configured with X>=3, the server also logs the packet before calling the extensions registered for this extension point, but only if at least one extension is registered.

pre-dns-add-forward

Instead of using the pre-dns-add-forward extension point, you can use the host-name-generator (for DHCPv4) and v6-host-name-generator (for DHCPv6) expressions configurable on the DNS update configurations.
Note

The **pre-dns-add-forward** extension point is deprecated and documentation for it removed. A future Cisco Prime IP Express release may remove the extension point completely. Instead, use an earlier request extension point (such as **post-client-lookup**) to set the required options, such as the **client-fqdn** option.

---

**post-send-packet**

Use the **post-send-packet** extension point for any processing that you want to perform outside of the serious time constraints of the DHCP request-response cycle. After the server sends the packet to the client, it calls this extension point.

Note

For DHCPv6 Reconfigure messages, there is no request dictionary (because Reconfigure is a server-initiated message). Thus, enabled extensions should check the response **msg-type** for ADVERTISE or REPLY or use **isValid** on the request to ensure that the request dictionary exists.

---

**environment-destructor**

The **environment-destructor** extension point allows an extension to clean up any context that it might be holding. The only dictionary available for this extension point is environment.

The environment dictionary is available for all extension points called for a single client request. Because some extensions may need to maintain context information between the multiple extension points called for a single client request, and because the server might drop requests at several places during processing, an extension cannot reliably release context that it might have created for that request. The **environment-destructor** extension point now makes it possible to reliably remove this context when processing of a request has completed, for whatever reason.

Note

The server calls all extensions attached to the **environment-destructor** extension point, even if the server did not call each extension at any other attachment point.
Part 6

Virtual Appliance
Introduction to Cisco Prime IP Express Virtual Appliance

The Cisco Prime IP Express virtual appliance aims at eliminating the installation, configuration and maintenance costs associated with running Cisco Prime IP Express on a local system. It also guarantees portability and thus reduces the risk in moving Cisco Prime IP Express from one machine to another.

You must get a license of Cisco Prime IP Express virtual appliance, and download the virtual appliance from Cisco.com. Upon initializing the virtual appliance, you have to add the license file. Cisco Prime IP Express will then be up and running, available to be configured. This is applicable for both the local and regional appliance.

This is different from just downloading a copy of Cisco Prime IP Express and installing it on a server or virtual machine provided by the customer, in that the operating system on which Cisco Prime IP Express runs is also provided in the virtual appliance.

The Cisco Prime IP Express virtual appliance supports VMware ESXi 5.0 or later platforms.

To know about the difference between vApp and a virtual appliance, see the User’s Guide to Deploying vApps and Virtual Appliances.

How the Cisco Prime IP Express Virtual Appliance Works

The virtual appliance consists of a virtual machine in the OVF format, which contains a runnable guest OS (CentOS 6.0) and Cisco Prime IP Express installed on that OS. When the virtual appliance is installed, Cisco Prime IP Express is already installed and is started by the virtual machine power-up like in the case of any regular server power-up.

How to Download the Cisco Prime IP Express Virtual Appliance

Do the following to download the Cisco Prime IP Express virtual appliance:
Step 1 You have to point the ESXi systems at a Cisco web-site and download an OVF format virtual appliance containing the guest operating system and a running version of Cisco Prime IP Express and power up the virtual machine/appliance.

Step 2 Set a root password for the virtual appliance when you are prompted to. There is no default password. For more details on deployment the virtual appliance and configuration on first boot, see Cisco Prime IP Express Installation Guide.

Two virtual appliances will be created, one for a local Cisco Prime IP Express cluster and one for a regional cluster. Each of these will have the security kit installed.

Monitoring Disk Space Availability

To view the disk space availability, do the following:

Step 1 In the vSphere Client window, select the host/server on which the virtual Cisco Prime IP Express appliance resides.

Step 2 Click Storage Views to see the list of the machines hosted by the server and the details about the space currently used by each machine.

Also, you can go to the Virtual Machines tab to view both the Provisioned Space and the Used Space by machine.

Step 3 Click Summary.

The Resources area of the Summary tab, displays the capacity of the disk and the CPU and memory used.

Step 4 Select the virtual machine and click the Summary tab.

The Resources area of the Summary tab displays the disk space details for the machine.

To monitor the disk space availability using the console, do the following:

Step 1 Select the virtual machine in the vSphere Client window and either click the Console tab on the right pane or right-click the virtual machine name and choose Open Console.

Step 2 Log in as root and type df -k. The disk space details are displayed.

If the disk space on the disk mounted on /var/nwreg2 is not enough, then you should increase the size of the disk (see the “Increasing the Size of Disk” section on page 31-2).

Increasing the Size of Disk

The cnr_growfilesystem script causes the data partition, /dev/sdb1 to grow to be the size of the entire data disk. It is trivial to expand the size of the data disk that the operating system sees, using VMware. To ensure that the operating system recognizes the bigger disk, you have to restart the VM and run the cnr_growfilesystem after increasing the size of the disk.
Before running the cnr_growfilesystem script, ensure that you backup the entire /var/nwreg2/ data and that it is not stored anywhere in the file system under /var/nwreg2. Running the cnr_growfilesystem script after restarting the VM ensures that the filesystem uses all the space in the partition.

If you need a bigger disk, do the following:

**Step 1**  Stop the VM.
**Step 2**  Increase the size of the disk by changing the size in the Virtual Machine Properties window. To open the Virtual Machine Properties window, you have to select the host name, right-click, and choose Edit Settings.
**Step 3**  Restart the VM. This ensures that the partition for the Cisco Prime IP Express data covers the entire disk.
**Step 4**  After the VM is up and running, log in as the root user and run the cnr_growfilesystem script:

```
/opt/nwreg2/local/usrbin/cnr_growfilesystem
```

This script will:

- Stop Cisco Prime IP Express
- Unmount the data partition
- Increase the size of the partition to be the size of the disk
- Turn the ext3 filesystem into an ext2 filesystem
- Expand the filesystem to be the size of the partition (which is now the size of the disk)
- Turn the ext2 filesystem back into an ext3 filesystem
- Remount the disk

**Troubleshooting**

If you experience any issues while working with the Cisco Prime IP Express virtual appliance, we recommend you to do the following:

Examine the log files in /var/nwreg2/{local | regional}/logs. Look particularly for errors in the log files as these signal exceptional conditions. If you are unable to resolve the problem and you have purchased Cisco support, then submit a case to Cisco Technical Assistance Center (TAC) regarding the problem.
Managing the Cisco Prime IP Express Virtual Appliance

This chapter describes how to manage a virtual appliance that is already deployed on a server and how to access the Cisco Prime IP Express from the virtual appliance. For details on how to deploy the virtual appliance and configure on first boot, see Cisco Prime IP Express Installation Guide.

Invoking the Cisco Prime IP Express Virtual Appliance

You can invoke the Cisco Prime IP Express application directly or you can invoke the management console of the virtual appliance itself. To invoke the Cisco Prime IP Express application directly, you should use the URL http://hostname:8080. The secure https connection is also available via the URL https://hostname:8443.

Do the following to invoke the Cisco Prime IP Express virtual appliance management console:

Step 1
Open the web browser and access the server with virtual appliance. For example, if default ports were used during the installation, the URLs would be https://hostname:5480. The default port for the virtual appliance is 5480.

Note

Step 2
Choose I understand the risks when you get the warning 'This Connection is Untrusted'.

Step 3
Click Add Exception and Confirm Security Exception for this page.

The virtual appliance login page is displayed.

Step 4
Log in as root with the password you set during initial bootup. The Cisco Prime IP Express virtual appliance home page is displayed. This page displays the basic information about the Cisco Prime IP Express virtual appliance and also has the Reboot and Shut down buttons. To reboot the server, you can click Reboot and to shut down the server, click Shutdown.

Monitor the Web UI/browser console window and notice that the window remains grayed out during the reboot. It will refresh when the reboot is complete. If you clicked Shutdown, start the server after the shutdown. After the server is up, ensure that the browser console can connect to the Cisco Prime IP Express virtual appliance.
Chapter 32  Managing the Cisco Prime IP Express Virtual Appliance

Modifying Virtual Appliance Configuration

You can modify the virtual appliance configuration after you log in. The following are the list of modifications you can make in the Cisco Prime IP Express virtual appliance UI:

- Setting the Time Zone, page 32-2
- Viewing Network Status, page 32-2
- Modifying Network Address Settings, page 32-2
- Configuring Proxy Server, page 32-3

Setting the Time Zone

You can check whether the time zone is set to the zone chosen during the initial configuration in the Time Zone tab under System. To set a different time zone, do the following:

Step 1  Choose the desired time zone from the System Time Zone drop-down list.
Step 2  Click Save Settings to save the changes you made or click Cancel Settings to reset to the original value that was configured during the first boot.

Viewing Network Status

To view the network values entered during the initial setup, go to the Status tab under Network.

Modifying Network Address Settings

To modify the network address settings, do the following:

Step 1  To modify the default IP settings, choose the Use the following IP settings option in the Address tab under Network. You can modify the IP Address, Netmask, Gateway, Preferred DNS Server, Alternate DNS Server, or Hostname.
Step 2  Click Save Settings to save the changes or click Cancel Settings to reset to the original value.
Configuring Proxy Server

To configure a proxy server, do the following:

**Step 1** Check the **Use a proxy server** check box in the **Proxy** tab under **Network** to configure a proxy server. You can now add the Proxy server, Proxy Port, Proxy Username, and Proxy Password. The Proxy Username and Proxy Password values are optional.

**Step 2** Click **Save Settings** to save the changes or click **Cancel Settings** to reset to the original value.

Accessing Cisco Prime IP Express Application

To access Cisco Prime IP Express Web UI from the virtual appliance home page, click the Application Home link on top right corner of this page.

Configurations and Restrictions

Cisco supports the Cisco Prime IP Express virtual appliance on VMware ESXi 5.0 or later platforms. While the OVF distribution format that we use allows the Cisco Prime IP Express virtual appliance to run on a variety of virtualized environments (that is, other than VMware), we do not support any environments other than VMware in this release.
Part 7

Appendices
Resource Records

Resource records comprise the data within a DNS zone. There is no fixed limit to the number of resource records a zone can own. In general, there can be zero, one, or more resource records of a given type. However, there are constraints on the number of certain types of records a zone can have.

All resource records have these required entries:

- **Name**—Name (host) that owns the record, such as example.com.
- **Class (not required for all formats)**—DNS supports only the IN (Internet) class of record.
- **TTL (time to live)**—Amount of time to store the record in cache, in seconds. If you do not include a TTL, Cisco Prime IP Express uses the zone default TTL, defined in the SOA resource record.
- **Type**—Type of the record, such as A, NS, SOA, and MX. There are many types that various RFCs define, although ten or fewer are in common use.
- **Record data**—Data types whose format and meaning varies with record type.

Table A-1 lists all the resource record types Cisco Prime IP Express supports. It provides the field syntax and the field descriptions, as well as how the fields are represented in the Cisco Prime IP Express GUI.

<table>
<thead>
<tr>
<th>Record No.</th>
<th>Name</th>
<th>Syntax and Description</th>
<th>RFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 1</td>
<td>Host Address—Name-to-address mapping for the zone</td>
<td>name ttl class A address <strong>Web UI:</strong> Add or Edit Host for Zone page: Hostname, IP Address or Resource Records for Zone page: Name, TTL, Type, Data nrcmd&gt; zone example.com addRR host123 3600 IN A 192.168.40.123</td>
<td>1035</td>
</tr>
</tbody>
</table>
### Table A-1 Resource Records (continued)

<table>
<thead>
<tr>
<th>Record</th>
<th>No.</th>
<th>Name</th>
<th>Syntax and Description</th>
<th>RFC</th>
</tr>
</thead>
</table>
| A6     | 38  | IPv6 Address—  
(Obsolete; use AAAA records instead) | name ttl class A6 address  
In the data, the suffix address is an IPv6 address encoded in  
network order (high-order octet first). There must be  
exactly enough octets in this field to contain a number of  
bits equal to 128 minus prefix length, with 0 to 7 leading  
pad bits to make this field an integral number of octets. Pad  
bits, if present, must be set to zero when loading a zone file  
and ignored on reception. For example:  
2001:0:734c:c0::  
|      |     |      | Web UI: Resource Records for Zone page:  
Name, TTL, Type=A6, Data=prefixlength suffixaddr  
prefixname, with data in the form:  
0 2345:00c1:ca11:1:1234:5678:9abc:def0  
nrcmd> zone example.com addRR host456 A6 0  
1345:c1:ca11:1:1234:5678:9abc:def0 | 6563 |
| AAAA  | 28  | IPv6 Address—  
|      |     |                          | name ttl class AAAA address  
Data is the IPv6 address format of eight sets of four  
hexadecimal digits, separated by colons. The first set of four  
digits is the high-order 16 bits of the address. You can omit  
leading zeros in sets and omit a value in a set if the value of  
the set is zero.  
|      |     |                          | Web UI: Resource Records for Zone page:  
Name, TTL, Type=AAAA, Data=address  
nrcmd> zone example.com addRR host456 AAAA  
1345:c1:ca11:1:1234:5678:9abc:def0 | 3596 |
| AFSDB | 18  | Andrew File  
System (AFS)  
Data Base—  
|      |     |                          | name ttl class AFSDB subtype hostname  
Subtype is either 1—AFS cell database server, or 2—DCE  
authentication name server. Hostname is the domain name  
of host that has a server for the cell named by the owner.  
|      |     |                          | Web UI: Resource Records for Zone page:  
Name, TTL, Type=AFSDB, Data=subtype hostname  
nrcmd> zone example.com addRR host4 AFSDB 1  
AFSDBhost.example.com. | 1183 |
### Table A-1  Resource Records (continued)

<table>
<thead>
<tr>
<th>Record</th>
<th>No.</th>
<th>Name</th>
<th>Syntax and Description</th>
<th>RFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNAME</td>
<td>5</td>
<td>Canonical Name—Aliases or nicknames</td>
<td><em>alias ttl class CNAME canonicalname</em> You cannot have any other resource records associated with a CNAME. Aliases are useful when you want the outside world to know a single, easily remembered name. You can also use aliases when a host changes its name. In that case, ensure that you have a CNAME pointer so that when people use the original name, it can be resolved to the newer one. <strong>Web UI</strong>: Resource Records for Zone page: Name=alias, TTL, Type=CNAME, Type, Data=canonicalname nrcmd&gt; zone example.com addRR host456 CNAME host1234</td>
<td>1035</td>
</tr>
<tr>
<td>DHCID</td>
<td>49</td>
<td>Dynamic Host Configuration Identifier—(RFC 4701)</td>
<td><em>name ttl class DHCID data</em> The DNS server uses this RR to allow DHCP clients and servers to update DNS automatically. This RR is not user-configurable. The data is the result of a one-way hash computation of the client message and the domain name. Sample RR output for an IPv6 address: chi6.example.com IN DHCID (AAIBY2/AuCccgoJbxaxcQc9TUapptP6910jxfNuVAA2kJEA=)</td>
<td>4701</td>
</tr>
<tr>
<td>HINFO</td>
<td>13</td>
<td>Host Info—Hardware and software information for the host</td>
<td><em>name ttl class HINFO cpu os</em> Data is the hardware (CPU) and operating system. <strong>Web UI</strong>: Resource Records for Zone page: Name, TTL, Type=HINFO, Data=cpu os nrcmd&gt; zone example.com addRR host5 HINFO CPU1 OS2</td>
<td>1035</td>
</tr>
<tr>
<td>ISDN</td>
<td>20</td>
<td>Integrated Services Digital Network (ISDN) Address—</td>
<td><em>name ttl class ISDN ISDNnumber [subaddr]</em> Data is the ISDN number of the owner and Direct Dial In, if any, and an optional ISDN subaddress string <strong>Web UI</strong>: Resource Records for Zone page: Name, TTL, Type=ISDN, Data=ISDNnumber [subaddr] nrcmd&gt; zone example.com addRR host6 ISDN ISDN88888</td>
<td>1183</td>
</tr>
<tr>
<td>MB</td>
<td>7</td>
<td>Mailbox Domain Name—</td>
<td><em>name ttl class MB mbox</em> Data is the domain name of the host with the specified mailbox. <strong>Web UI</strong>: Resource Records for Zone page: Name, TTL, Type=MB, Data=mbox nrcmd&gt; zone example.com addRR host7 MB mailbox.example.com</td>
<td>1035</td>
</tr>
</tbody>
</table>
### Table A-1 Resource Records (continued)

<table>
<thead>
<tr>
<th>Record</th>
<th>No.</th>
<th>Name</th>
<th>Syntax and Description</th>
<th>RFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MG</td>
<td>8</td>
<td>Mail Group Member—</td>
<td><em>name ttl class MG mgroup</em></td>
<td>1035</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data is the domain name of the mailbox group (mailing list).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI</strong>: Resource Records for Zone page:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Name, TTL, Type=MG, Data=mgroup</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>nrcmd&gt; zone example.com addRR host7 MG mgroup.example.com.</code></td>
<td></td>
</tr>
<tr>
<td>MINFO</td>
<td>14</td>
<td>Mailbox Info—</td>
<td><em>name ttl class MINFO respmbox errormbox</em></td>
<td>1035</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data is the mailbox responsible for the mailing list, and the mailbox to receive error messages.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI</strong>: Resource Records for Zone page:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Name, TTL, Type=MINFO, Data=respmbox errormbox</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>nrcmd&gt; zone example.com addRR host7 MINFO resp.example.com. error.example.com.</code></td>
<td></td>
</tr>
<tr>
<td>MR</td>
<td>9</td>
<td>Mail Rename—</td>
<td><em>name ttl class MR newmbox</em></td>
<td>1035</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data is the mailbox name to rename the owner mailbox.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI</strong>: Resource Records for Zone page:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Name, TTL, Type=MR, Data=newmbox</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>nrcmd&gt; zone example.com addRR host7 MR renamemb.example.com.</code></td>
<td></td>
</tr>
<tr>
<td>MX</td>
<td>15</td>
<td>Mail Exchanger—</td>
<td><em>name ttl class MX pref mxname</em></td>
<td>1035</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Where to deliver the mail for a domain name</td>
<td>Data is the preference value (16-bit integer for the preference for the record, with lower values having preference), and the domain name of the mail exchanger for the owner.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI</strong>: Resource Records for Zone page:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Name, TTL, Type=MX, Data=pref mxname</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>nrcmd&gt; zone example.com addRR host8 MX 10 exchanger.example.com.</code></td>
<td></td>
</tr>
</tbody>
</table>
Table A-1  Resource Records (continued)

<table>
<thead>
<tr>
<th>Record</th>
<th>No.</th>
<th>Name</th>
<th>Syntax and Description</th>
<th>RFC</th>
</tr>
</thead>
</table>
| NAPTR  | 35  | Naming Authority Pointer—Produce a new domain label or Universal Resource Identifier (URI). You can then use DNS to look up services for many resource names that are not in domain name syntax. | name ttl class NAPTR order pref flags serv regexp replace  
   - order—16-bit integer for the order in which to process the NAPTR records to ensure the correct ordering of rules, with low numbers processed before high numbers.  
   - pref—16-bit unsigned integer for the order in which to process NAPTR records with equal order values, with low numbers processed before high numbers.  
   - flags—Character-string containing flags to control aspects of rewriting and interpreting fields, single characters from the set [A-Z0-9] (not case-sensitive); the S, A and U flags denote a terminal lookup, the P flag says that the remainder of the application-side algorithm should be carried out protocol-specific.  
   - serv—Valid protocols or services.  
   - regexp—String containing a substitution expression applied to the original string held by the client to construct the next domain name to look up. (For common regex usage, see Table 5-4 on page 5-30).  
   - replace—Next FQDN to query for NAPTR, SRV, or address records, depending on the value of the flags field. | 2915 |
| NS     | 2   | Name Server—Authoritative server for the zone                       | name ttl class NS nameserver  
Machines that provide name service must not reside in the owner domain. For each domain, you must have at least one NS record. NS records for a domain must exist in both the zone that delegates the domain and in the domain itself. NS record names must have an equivalent A record (they cannot point to an alias). | 1035 |

Web UI: Resource Records for Zone page: 
Name, State, TTL, Type=NAPTR, Data=order pref flags service regexp replace

```bash
nrcmd> zone 8.6.4.e164.arpa addRR 4.3.2.1.6.7.9 naptr 100 10 u sip+E2U /^.*$/sip:info@tele2.se/ .
```

Web UI: Add or Edit Zone page Nameservers: 
NS TTL, Add Nameserver

```bash
nrcmd> zone example.com addRR @ NS DNSserv2.example.com.
```
## Table A-1 Resource Records (continued)

<table>
<thead>
<tr>
<th>Record</th>
<th>No.</th>
<th>Name</th>
<th>Syntax and Description</th>
<th>RFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSAP</td>
<td>22</td>
<td>Network Service Access Point (NSAP) Address—</td>
<td>name ttl class NSAP NSAPaddr</td>
<td>1706</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data is the NSAPaddr—Octet values assigned by the assigning authority, a character string of the type used in TXT and HINFO records (see RFC 1706).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI:</strong> Resource Records for Zone page: Name, TTL, Type=NSAP, Data=NSAPaddr</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nrcmd&gt; zone example.com addRR host10 NSAP 39840f80005a0000000001e13708002010726e00</td>
<td></td>
</tr>
<tr>
<td>PTR</td>
<td>12</td>
<td>Pointer—Reverse mapping</td>
<td>name ttl class PTR dname</td>
<td>1035</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data is the domain name of host having the reverse record indicated by the owner. PTR records are used for reverse mapping, specifically in the in-addr.arpa zones for translation of addresses to names. PTRs use official names, not aliases. The name in a PTR record is the local IP address portion of the reverse name.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI:</strong> Resource Records for Zone page: Name, State, TTL, Type=PTR, Data=dname</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nrcmd&gt; zone example.com addRR 45.40.168.192.in-addr.arpa. PTR host1234</td>
<td></td>
</tr>
<tr>
<td>RP</td>
<td>17</td>
<td>Responsible Person—</td>
<td>name ttl class RP mbox txthost</td>
<td>1183</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data is the domain name of the mailbox for the responsible person, and the domain name of host where TXT records exist.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI:</strong> Resource Records for Zone page: Name, TTL, Type=RP, Data=mbox txthost</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nrcmd&gt; zone example.com addRR host7 RP resp.example.com. text.example.com.</td>
<td></td>
</tr>
<tr>
<td>RT</td>
<td>21</td>
<td>Route Through—</td>
<td>name ttl class RT pref intermediatehost</td>
<td>1183</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data is the pref—16-bit integer for preference to give to this record among others of the same owner, and intermediatehost—domain name of the host serving as intermediate to reach the owner.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI:</strong> Resource Records for Zone page: Name, TTL, Type=RT, Data=pref intermediatehost</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nrcmd&gt; zone example.com addRR host7 RT 10 routthru.example.com.</td>
<td></td>
</tr>
</tbody>
</table>
### Table A-1 Resource Records (continued)

<table>
<thead>
<tr>
<th>Record</th>
<th>No.</th>
<th>Name</th>
<th>Syntax and Description</th>
<th>RFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOA</td>
<td>6</td>
<td>Start of Authority — Every zone must have a single SOA record</td>
<td>name ttl class SOA primeserver hostmaster (serial refresh retry expire minimum)</td>
<td>1035</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI:</strong> Add or Edit Zone page SOA Attributes: Serial Number, SOA TTL, Nameserver, Contact E-Mail, Secondary Refresh, Secondary Retry, Secondary Expire, Minimum TTL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>nrcmd&gt; zone example.com addRR @ 172800 IN SOA ns hostmaster 1 10800 3600 604800 86400</code></td>
<td></td>
</tr>
<tr>
<td>SRV</td>
<td>33</td>
<td>Service Location —</td>
<td>name ttl class SRV priority weight port target</td>
<td>2782</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• priority — 16-bit priority to give the record among the owner SRV records.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• weight — 16-bit load to give the record at the same priority level.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• port — 16-bit port on which to run the service.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• target — Domain name of host running on the specified port.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Administrators can use several servers for a single domain, move services between hosts with little difficulty, and designate some hosts as primary servers for a service and others as backups. Clients ask for a specific service or protocol for a domain and receive the names of any available servers. See Chapter 29, “Configuring DNS Update” for how this record affects Windows servers.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI:</strong> Resource Records for Zone page: Name, TTL, Type=SRV, Data=priority weight port target</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>nrcmd&gt; zone example.com addRR host2 SRV 10 1 60 host7.example.com.</code></td>
<td></td>
</tr>
<tr>
<td>TXT</td>
<td>16</td>
<td>Text —</td>
<td>name ttl class TXT textstring</td>
<td>1035</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data is one or more text character strings that can contain any type of information.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Web UI:</strong> Resource Records for Zone page: Name, TTL, Type=TEXT, Data=textstring</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>nrcmd&gt; zone example.com addRR host2 TXT &quot;this message&quot;</code></td>
<td></td>
</tr>
<tr>
<td>Record</td>
<td>No.</td>
<td>Name</td>
<td>Syntax and Description</td>
<td>RFC</td>
</tr>
<tr>
<td>--------</td>
<td>-----</td>
<td>------</td>
<td>------------------------</td>
<td>-----</td>
</tr>
</tbody>
</table>
| WKS    | 11  | Well Known Services— | name ttl class WKS addr protocol servicelist  

*addr*—32-bit IP address.  
*protocol*—8-bit IP protocol number, which can be TCP or UDP.  
*servicelist*—Variable-length bit map in 8-bit multiples of services, which can be TIME, TELNET, FTP, or SMTP.  

**Web UI:** Resource Records for Zone page:  
Name, TTL, Type=WKS, Data=addr protocol servicelist  
nrcmd> zone example.com addRR host8 WKS 192.168.40.56 TCP TELNET  

| X25    | 19  | X.25 Address— | name ttl class X25 PSDNaddr  
Data is the character string of the Public Switch Data Network (PSDN) address in the X.121 numbering plan associated with the owner.  

**Web UI:** Resource Records for Zone page:  
Name, TTL, Type=X25, Data=PSDNaddr  
nrcmd> zone example.com addRR host9 IN X25 311061700956  

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>1035</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1183</td>
</tr>
</tbody>
</table>
DHCP Options

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The data items themselves are also called options.

This appendix contains DHCP options and BOOTP vendor extensions from RFC 2132, and includes the validation type for each option, as indicated in Table B-10 on page B-15.

This appendix also contains the standard Microsoft client options and several tables displaying the options sorted by categories.

Option Descriptions

The following sections describe the DHCP options in detail:

- RFC 1497 Vendor Extensions, page B-1
- IP Layer Parameters Per Host, page B-3
- IP Layer Parameters Per Interface, page B-4
- Link Layer Parameters Per Interface, page B-4
- TCP Parameters, page B-5
- Application and Service Parameters, page B-5
- DHCPv4 Extension Options, page B-8
- DHCPv6 Options, page B-11
- Microsoft Client Options, page B-10
- Options by Number, page B-15
- Options by Cisco Prime IP Express Name, page B-20
- Option Validation Types, page B-26

RFC 1497 Vendor Extensions

Table B-1 on page B-2 lists the vendor extensions as defined in RFC 1497.
## Table B-1  RFC 1497 Vendor Extension Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pad</td>
<td>0</td>
<td>1 octet</td>
<td>Causes the subsequent fields to align on word boundaries.</td>
</tr>
<tr>
<td>End</td>
<td>255</td>
<td>1 octet</td>
<td>End of valid information in the vendor field. Subsequent octets should be filled with the Pad options.</td>
</tr>
<tr>
<td>Subnet Mask</td>
<td>1</td>
<td>4 octets</td>
<td>Client subnet mask, as per RFC 950. If both the Subnet Mask and the Router option are specified in a DHCP reply, the Subnet Mask option must be first.</td>
</tr>
<tr>
<td>Time Offset</td>
<td>2</td>
<td>4 octets</td>
<td>Offset of the client subnet, in seconds, from Universal Time (UT). The offset is expressed as a two's-complement 32-bit integer. A positive offset indicates a location east of the zero meridian and a negative offset indicates a location west of the zero meridian.</td>
</tr>
<tr>
<td>Router</td>
<td>3</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of IP addresses for routers on the client subnet. Routers should be in order of preference.</td>
</tr>
<tr>
<td>Time Server</td>
<td>4</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of RFC 868 compliant time servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Name Server Option</td>
<td>5</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of IEN 116 name servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Domain Name Server</td>
<td>6</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of Domain Name System (STD 13, RFC 1035) name servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Log Server</td>
<td>7</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of MIT-LCS UDP log servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Cookie Server</td>
<td>8</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of RFC 865-compliant cookie servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>LPR Server</td>
<td>9</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of RFC 1179-compliant line printer servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Impress Server</td>
<td>10</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of Imagen Impress servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Resource Location Server</td>
<td>11</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of RFC 887-compliant resource location servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Host Name</td>
<td>12</td>
<td>1 octet minimum</td>
<td>Name of the client. The name may or may not be qualified with the local domain name. See RFC 1035 for the character set restrictions.</td>
</tr>
<tr>
<td>Boot File Size</td>
<td>13</td>
<td>2 octets</td>
<td>Number of 512-octet blocks in the default boot file.</td>
</tr>
<tr>
<td>Merit Dump File</td>
<td>14</td>
<td>1 octet minimum</td>
<td>Path name of a file to which the client core image should be placed in the event the client crashes. The path is formatted as a character string consisting of characters from the NVT ASCII character set.</td>
</tr>
<tr>
<td>Domain Name</td>
<td>15</td>
<td>1 octet minimum</td>
<td>Domain name that the client should use when resolving hostnames through the Domain Name System.</td>
</tr>
<tr>
<td>Swap Server</td>
<td>16</td>
<td>4 octets</td>
<td>IP address of the client swap server.</td>
</tr>
</tbody>
</table>
Table B-1  RFC 1497 Vendor Extension Options (continued)

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root Path</td>
<td>17</td>
<td>1 octet minimum</td>
<td>Path name that contains the client root disk. The path is formatted as a character string consisting of characters from the NVT ASCII character set.</td>
</tr>
<tr>
<td>Extensions Path</td>
<td>18</td>
<td>1 octet minimum</td>
<td>Uses a string to specify a file, retrievable through TFTP. The file contains information that can be interpreted in the same way as the 64-octet vendor-extension field within the BOOTP response, with these exceptions: the length of the file is unconstrained, and all references to instances of this option in the file are ignored.</td>
</tr>
</tbody>
</table>

IP Layer Parameters Per Host

Table B-2 lists the options that affect the operation of the IP layer on a per-host basis.

Table B-2  IP Layer Parameters Per Host Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Forwarding Enable/Disable</td>
<td>19</td>
<td>1 octet</td>
<td>Specifies whether the client should configure its IP layer for packet forwarding. Values: 0=disable; 1=enable</td>
</tr>
<tr>
<td>Non-Local Source Routing Enable/Disable</td>
<td>20</td>
<td>1 octet</td>
<td>Specifies whether the client should configure its IP layer to allow forwarding of datagrams with non-local source routes. Values: 0=disable; 1=enable</td>
</tr>
<tr>
<td>Policy Filter</td>
<td>21</td>
<td>8 octet minimum; multiples of 8</td>
<td>Policy filters for non-local source routing. The filters consist of a list of IP addresses and masks that specify destination/mask pairs with which to filter incoming source routes. Any source-routed datagram whose next-hop address does not match one of the filters should be discarded by the client.</td>
</tr>
<tr>
<td>Maximum Datagram Reassembly Size</td>
<td>22</td>
<td>2 octets</td>
<td>Maximum size datagram that the client should be prepared to reassemble. Value: 576 minimum</td>
</tr>
<tr>
<td>Default IP Time-to-Live</td>
<td>23</td>
<td>1 octet</td>
<td>Default TTL that the client should use on outgoing datagrams. Values: 1 to 255</td>
</tr>
<tr>
<td>Path MTU Aging Timeout</td>
<td>24</td>
<td>4 octets</td>
<td>Timeout (in seconds) to use when aging Path MTU values (defined in RFC 1191).</td>
</tr>
<tr>
<td>Path MTU Plateau Table</td>
<td>25</td>
<td>2 octets minimum; multiples of 2</td>
<td>Table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. Value: 68 minimum</td>
</tr>
</tbody>
</table>
Appendix B      DHCP Options

IP Layer Parameters Per Interface

Table B-3 lists the options that affect the operation of the IP layer on a per-interface basis. A client can issue multiple requests, one per interface, to configure interfaces with their specific parameters.

Table B-3     IP Layer Parameters Per Interface Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface MTU</td>
<td>26</td>
<td>2 octets</td>
<td>MTU to use on this interface. The minimum legal value for the MTU is 68.</td>
</tr>
<tr>
<td>All Subnets Are Local</td>
<td>27</td>
<td>1 octet</td>
<td>Specifies whether or not the client can assume that all subnets of the IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>network to which the client is connected use the same MTU as the subnet of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>that network to which the client is directly connected. Values: 1=all</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>subnets share same MTU; 0=some directly-connected subnets can have smaller</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MTUs.</td>
</tr>
<tr>
<td>Broadcast Address</td>
<td>28</td>
<td>4 octets</td>
<td>Broadcast address in use on the client subnet.</td>
</tr>
<tr>
<td>Perform Mask Discovery</td>
<td>29</td>
<td>1 octet</td>
<td>Specifies whether or not the client should perform subnet mask discovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>using ICMP. Values: 0=disable; 1=enable</td>
</tr>
<tr>
<td>Mask Supplier</td>
<td>30</td>
<td>1 octet</td>
<td>Specifies whether or not the client should respond to subnet mask requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>using ICMP. Values: 0=do not respond; 1=respond</td>
</tr>
<tr>
<td>Perform Router Discovery</td>
<td>31</td>
<td>1 octet</td>
<td>Specifies whether or not the client should solicit routers using the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Router Discovery mechanism defined in RFC 1256. Values: 0=disable; 1=enable</td>
</tr>
<tr>
<td>Router Solicitation Address</td>
<td>32</td>
<td>4 octets</td>
<td>Address to which the client should transmit router solicitation requests.</td>
</tr>
<tr>
<td>Static Route</td>
<td>33</td>
<td>8 octet minimum; multiples of 8</td>
<td>List of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are in descending order of priority. The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination. The default route (0.0.0.0) is an illegal destination for a static route.</td>
</tr>
</tbody>
</table>

Link Layer Parameters Per Interface

Table B-4 lists the options that affect the operation of the data link layer on a per-interface basis.
### TCP Parameters

Table B-5 lists the options that affect the operation of the TCP layer on a per-interface basis.

#### Table B-5  TCP Parameter Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP Default TTL</td>
<td>37</td>
<td>1 octet</td>
<td>Default TTL that the client should use when sending TCP segments. Value: minimum 1</td>
</tr>
<tr>
<td>TCP Keepalive Interval</td>
<td>38</td>
<td>4 octets</td>
<td>Interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. The time is specified as a 32-bit unsigned integer. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application. Value: 32-bit unsigned; 0=do not generate keepalive messages unless specifically requested.</td>
</tr>
<tr>
<td>TCP Keepalive Garbage</td>
<td>39</td>
<td>1 octet</td>
<td>Specifies the whether or not the client should send TCP keep-alive messages with an octet of garbage for compatibility with older implementations. Values: 0=do not send; 1=send</td>
</tr>
</tbody>
</table>

### Application and Service Parameters

Table B-6 lists some miscellaneous options used to configure miscellaneous applications and services.

#### Table B-6  Application and Service Parameter Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Information Service (NIS) Domain</td>
<td>40</td>
<td>1 octet minimum</td>
<td>Name of the client NIS domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set.</td>
</tr>
<tr>
<td>Network Information Service (NIS) Servers</td>
<td>41</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of IP addresses indicating NIS servers available to the client. Servers should be in order of preference.</td>
</tr>
</tbody>
</table>
### Appendix B  DHCP Options

#### Option Descriptions

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Time Protocol Servers</td>
<td>42</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of IP addresses indicating NTP servers that are available to the client. Servers should be in order of preference.</td>
</tr>
</tbody>
</table>
| Vendor-Specific Information           | 43  | 1 octet minimum             | This option is used by clients and servers to exchange vendor-specific information. The information is an opaque object of \( n \) octets, presumably interpreted by vendor-specific code on the clients and servers. The definition of this information is vendor specific. The vendor is indicated in the `dhcp-class-identifier` option. Servers not equipped to interpret the vendor-specific information sent by a client must ignore it (although it can be reported). Clients that do not receive desired vendor-specific information should make an attempt to operate without it, although they can do so (and announce they are doing so) in a degraded mode. If a vendor potentially encodes more than one item of information in this option, then the vendor should encode the option using encapsulated vendor-specific options as described here. The encapsulated vendor-specific options field should be encoded as a sequence of code, length, and value fields of identical syntax to the DHCP options field with these exceptions:  
  - There should not be a magic cookie field in the encapsulated vendor-specific extensions field.  
  - Codes other than 0 or 255 can be redefined by the vendor within the encapsulated vendor-specific extensions field, but should conform to the tag-length-value syntax defined in section 2. Code 255 (END), if present, signifies the end of the encapsulated vendor extensions, not the end of the vendor extensions field. If the code 255 is not present, then the end of the enclosing vendor-specific information field is taken as the end of the encapsulated vendor-specific extensions field. |
| NetBIOS over TCP/IP Name Server        | 44  | 4 octet minimum; multiples of 4 | List of RFC 1001/1002 NBNS name servers in order of preference.                                                                                                                                              |
| NetBIOS over TCP/IP Datagram Distribution Server | 45  | 4 octet minimum; multiples of 4 | List of RFC 1001/1002 NBDD servers in order of preference.                                                                                                                                                  |
### Table B-6 Application and Service Parameter Options (continued)

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetBIOS over TCP/IP Node Type</td>
<td>46</td>
<td>1 octet</td>
<td>Allows NetBIOS over TCP/IP client, which are configured as described in RFC 1001/1002. Values: Single hexadecimal octet that identifies the client type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 0x1=B-node (broadcast node)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 0x2=P-node (point-to-point node)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 0x4=M-node (mixed node)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 0x8=H-node</td>
</tr>
<tr>
<td>NetBIOS over TCP/IP Scope</td>
<td>47</td>
<td>1 octet minimum</td>
<td>NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002.</td>
</tr>
<tr>
<td>X Window System Font Server</td>
<td>48</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of X Window System Font servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>X Window System Display Manager</td>
<td>49</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of IP addresses of systems that are running the X Window System Display Manager and are available to the client. Addresses should be in order of preference.</td>
</tr>
<tr>
<td>Network Information Service (NIS+) Domain</td>
<td>64</td>
<td>1 octet minimum</td>
<td>Name of the client NIS+ domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set.</td>
</tr>
<tr>
<td>Network Information Service (NIS+) Servers</td>
<td>65</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of IP addresses indicating NIS+ servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Mobile IP Home Agent</td>
<td>68</td>
<td>0 octets minimum; multiples of 4; expected, 4 octets (single home agent address)</td>
<td>List of IP addresses indicating mobile IP home agents available to the client. Agents should be in order of preference. Value: 32-bit address; 0=no home agents available</td>
</tr>
<tr>
<td>Simple Mail Transport Protocol (SMTP) Server</td>
<td>69</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of SMTP servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Post Office Protocol (POP3) Server</td>
<td>70</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of POP3 servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Network News Transport Protocol (NNTP) Server</td>
<td>71</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of NNTP servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>World Wide Web (WWW) Server</td>
<td>72</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of World Wide Web (WWW) servers available to the client. Servers should be in order of preference.</td>
</tr>
</tbody>
</table>
### DHCPv4 Extension Options

Table B-6 lists the DHCPv4 extension options.

#### Table B-6  Application and Service Parameter Options (continued)

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finger Server</td>
<td>73</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of Finger servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>Internet Relay Chat Server</td>
<td>74</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of IRC servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>StreetTalk Server</td>
<td>75</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of StreetTalk servers available to the client. Servers should be in order of preference.</td>
</tr>
<tr>
<td>StreetTalk Directory Assistance (STDA) Server</td>
<td>76</td>
<td>4 octet minimum; multiples of 4</td>
<td>List of STDA servers available to the client. Servers should be in order of preference.</td>
</tr>
</tbody>
</table>

#### Table B-7  DHCPv4 Extensions

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requested IP Address</td>
<td>50</td>
<td>4 octets</td>
<td>Used in a client request (DHCPDISCOVER) to allow the client to request that a particular IP address be assigned.</td>
</tr>
<tr>
<td>IP Address Lease Time</td>
<td>51</td>
<td>4 octets</td>
<td>Used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address. In a server reply (DHCPOFFER), a DHCP server uses this option to specify the lease time it is willing to offer. Value: seconds, as 32-bit unsigned integer</td>
</tr>
<tr>
<td>Option Overload</td>
<td>52</td>
<td>1 octet</td>
<td>Indicates that the DHCP sname or file fields are being overloaded by using them to carry DHCP options. A DHCP server inserts this option if the returned parameters will exceed the usual space allotted for options. If this option is present, the client interprets the specified additional fields after it concludes interpretation of the standard option fields. Values: 1=file field is used to hold options; 2=sname field is used to hold options; 3=both fields are used to hold options</td>
</tr>
<tr>
<td>DHCP Message Type</td>
<td>53</td>
<td>1 octet</td>
<td>Used to convey the type of DHCP message. The preset value is 1 (DHCPDISCOVER). Values: 1=DHCPDISCOVER; 2=DHCPOFFER; 3=DHCPREQUEST; 4=DHCPDECLINE; 5=DHCPACK; 6=DHCPNAK; 7=DHCPRELEASE; 8=DHCPINFORM; 13=LEASEQUERY</td>
</tr>
</tbody>
</table>
### DHCP Options

**Server Identifier**
- **No.:** 54
- **Length:** 4 octets
- **Description:** Used in DHCP OFFER and DHCPREQUEST messages, and can optionally be included in the DHCPACK and DHCPNAK messages. DHCP servers include this option in the DHCP OFFER in order to allow the client to distinguish between lease offers. DHCP clients use the contents of the server identifier field as the destination address for any DHCP messages unicast to the DHCP server. DHCP clients also indicate which of several lease offers is being accepted by including this option in a DHCPREQUEST message. The identifier is the IP address of the selected server.

**Parameter Request List**
- **No.:** 55
- **Length:** 1 octet minimum
- **Description:** Used by a DHCP client to request values for specified configuration parameters. The list of requested parameters is specified as n octets, where each octet is a valid DHCP option code as defined in this document. The client can list the options in order of preference. The DHCP server does not have to return the options in the requested order, but must try to insert the options in the order that the client requested.

**Message**
- **No.:** 56
- **Length:** 1 octet minimum
- **Description:** Used by a DHCP server to provide an error message to a DHCP client in a DHCPNAK message in the event of a failure. A client can use this option in a DHCPDECLINE message to indicate why the client declined the offered parameters. The message consists of n octets of NVT ASCII text, which the client can display on an available output device.

**Maximum DHCP Message Size**
- **No.:** 57
- **Length:** 2 octets
- **Description:** Maximum-length DHCP message that a server is willing to accept. The length is specified as an unsigned 16-bit integer. A client can use the maximum DHCP message size option in DHCPDISCOVER or DHCPREQUEST messages, but should not use the option in DHCPDECLINE messages. Value: 576 minimum

**Renewal (T1) Time Value**
- **No.:** 58
- **Length:** 4 octets
- **Description:** Time interval from address assignment until the client transitions to RENEWING state. Value: seconds, as 32-bit unsigned integer

**Rebinding (T2) Time Value**
- **No.:** 59
- **Length:** 4 octets
- **Description:** Time interval from address assignment until the client transitions to REBINDING state. Value: seconds, as 32-bit unsigned integer

**Vendor Class Identifier**
- **No.:** 60
- **Length:** 1 octet minimum
- **Description:** Used by DHCP clients to optionally identify the vendor type and configuration of a DHCP client. The information is a string of n octets, interpreted by servers. Vendors can choose to define specific vendor class identifiers to convey particular configuration or other identification information about a client. For example, the identifier can encode the client hardware configuration. Servers not equipped to interpret the class-specific information sent by a client must ignore it (although it can be reported). Servers that respond should only use option 43 to return the vendor-specific information to the client.
### Microsoft Client Options

Table B-8 lists the standard Microsoft client options.

---

Table B-7 DHCPv4 Extensions (continued)

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client-Identifier</td>
<td>61</td>
<td>2 octet minimum</td>
<td>Used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain. DHCP servers should treat identifiers as opaque objects. The client identifier can consist of type-value pairs similar to the htype/chaddr fields. For instance, it can consist of a hardware type and hardware address. In this case, the type field should be one of the ARP hardware types defined in STD2. A hardware type of 0 (zero) should be used when the value field contains an identifier other than a hardware address (for example, a fully qualified domain name). For correct identification of clients, each client-identifier must be unique among the client-identifiers used on the subnet to which the client is attached. Vendors and system administrators are responsible for choosing client-identifiers that meet this requirement for uniqueness.</td>
</tr>
<tr>
<td>Bootfile Name</td>
<td>67</td>
<td>1 octet minimum</td>
<td>Identifies a bootfile when the file field is the DHCP header that has been used for DHCP options.</td>
</tr>
<tr>
<td>Relay Agent Information</td>
<td>82</td>
<td></td>
<td>Identifies the DHCP relay agent information.</td>
</tr>
<tr>
<td>iSNS</td>
<td>83</td>
<td>14 bytes minimum</td>
<td>Identifies the Internet Storage Name Service (see RFC 4174)</td>
</tr>
<tr>
<td>BCMS Controller Domain</td>
<td>88</td>
<td>Variable</td>
<td>List of Broadcast and Multicast Service (BCMS) controller domains (see RFC 4280)</td>
</tr>
<tr>
<td>BCMS Address</td>
<td>89</td>
<td>4 octets minimum</td>
<td>List of IP addresses for the BCMS controller (see RFC 4280)</td>
</tr>
<tr>
<td>Lease Query</td>
<td>91</td>
<td>4 octets</td>
<td>Time of the most recent access of the client sending a DHCPLEASEQUERY (see RFC 4388).</td>
</tr>
<tr>
<td>Lease Query</td>
<td>92</td>
<td>4 octets</td>
<td>All IP addresses associated with the client specified in a particular DHCPLEASEQUERY message (see RFC 4388).</td>
</tr>
</tbody>
</table>
Appendix B  DHCP Options

Table B-8  Microsoft DHCP Client Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dhcp-lease-time</td>
<td>51</td>
<td>14 days</td>
</tr>
<tr>
<td>domain-name</td>
<td>15</td>
<td>A domain name such as cisco.com</td>
</tr>
<tr>
<td>domain-name-servers</td>
<td>6</td>
<td>IP address of the name servers</td>
</tr>
<tr>
<td>netbios-name-servers</td>
<td>44</td>
<td>WINS server address</td>
</tr>
<tr>
<td>netbios-node-type</td>
<td>46</td>
<td>Identifies the NetBIOS client type; note that Cisco Prime IP Express displays a warning if it is not present</td>
</tr>
<tr>
<td>routers</td>
<td>3</td>
<td>IP address of the router for this subnet</td>
</tr>
</tbody>
</table>

DHCPv6 Options

Table B-9 on page B-11 lists the DHCPv6 options, along with their defined data types. All the option packets include at least an option length (option-len) and a variable length data field. There can also be additional parameter settings, as described in the table. Many of these options are described in RFC 3315.

Table B-9  DHCPv6 Options

<table>
<thead>
<tr>
<th>Cisco Prime IP Express Name (Type)</th>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-identifier AT_BLOB</td>
<td>1</td>
<td>DUID identifying a client between a client and a server.</td>
</tr>
<tr>
<td>server-identifier AT_BLOB</td>
<td>2</td>
<td>DUID identifying a server between a client and a server.</td>
</tr>
<tr>
<td>ia-na AT_BLOB</td>
<td>3</td>
<td>Nontemporary Addresses option with the associated parameters and addresses. Parameters are the unique ID, time the client contacts the addresses in the IA to extend the lifetime, and time the client contacts any available server to extend the lifetime of the addresses.</td>
</tr>
<tr>
<td>ia-ta AT_BLOB</td>
<td>4</td>
<td>Temporary Addresses option with the associated parameters and addresses.</td>
</tr>
<tr>
<td>iaaddr AT_BLOB</td>
<td>5</td>
<td>IPv6 addresses associated with an IA_NA or IA_TA. (The IAADDR must be encapsulated in the options field of an IA_NA or IA_TA option.) The IAADDR option includes preferred and valid lifetime fields, and the options field that encapsulates the options specific to this address.</td>
</tr>
<tr>
<td>oro AT_SHORT</td>
<td>6</td>
<td>Option Request option (ORO) that identifies a list of options in a message between a client and a server. A client can include this option in a Solicit, Request, Renew, Rebind, Confirm, or Information-request message to inform the server about options the client wants from the server. A server can include this option in a Reconfigure message to indicate which option updates the client should request.</td>
</tr>
</tbody>
</table>
## Table B-9 DHCPv6 Options (continued)

<table>
<thead>
<tr>
<th>Cisco Prime IP Express Name (Type)</th>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>preference AT_INT8</td>
<td>7</td>
<td>A server sends this option to a client to affect what server the client selects.</td>
</tr>
<tr>
<td>elapsed-time AT_SHORT</td>
<td>8</td>
<td>A client sends this option to a server to indicate how long the client has been trying to complete a message exchange.</td>
</tr>
<tr>
<td>relay-message AT_BLOB</td>
<td>9</td>
<td>DHCP message in a Relay-forward or Relay-reply message.</td>
</tr>
<tr>
<td>auth AT_BLOB</td>
<td>11</td>
<td>Authenticates the identity and contents of a DHCP message. The parameters are the authentication protocol, the authentication algorithm, the replay detection method (RDM), and the authentication information.</td>
</tr>
<tr>
<td>server-unicast AT_IP6ADDR</td>
<td>12</td>
<td>The server sends this option to a client to indicate that the client can unicast messages to the server.</td>
</tr>
<tr>
<td>status-code AT_BLOB</td>
<td>13</td>
<td>Returns a status indication related to the DHCP message or option in which it appears. The parameters are the status code and status message.</td>
</tr>
<tr>
<td>rapid-commit AT_ZEROSIZE</td>
<td>14</td>
<td>Signals use of the two-message exchange for address assignment.</td>
</tr>
<tr>
<td>user-class AT_TYPECNT</td>
<td>15</td>
<td>Clients use this option to identify the type or category of user or applications it represents. A zero type count value field followed by user data (as a blob).</td>
</tr>
<tr>
<td>vendor-class AT_VENDOR_CLASS</td>
<td>16</td>
<td>Clients use this option to identify the vendor that manufactured the hardware on which they are running.</td>
</tr>
<tr>
<td>vendor-opts AT_VENDOR_OPTS</td>
<td>17</td>
<td>Clients and servers use this option to exchange vendor-specific information. The enterprise ID for the CableLabs vendor is 4491; the suboptions for CableLabs are listed in Table C-4 on page C-7.</td>
</tr>
<tr>
<td>interface-id AT_BLOB</td>
<td>18</td>
<td>Relay agents use this option to identify the interface on which the client message is received.</td>
</tr>
<tr>
<td>reconfigure-message AT_INT8</td>
<td>19</td>
<td>The server includes this in a Reconfigure message to indicate whether the client should respond with a Renew or Information-request message.</td>
</tr>
<tr>
<td>reconfigure-accept AT_ZEROSIZE</td>
<td>20</td>
<td>Clients use this option to announce to the server whether the client is willing to accept Reconfigure messages.</td>
</tr>
<tr>
<td>sip-servers-name AT_DNSNAME</td>
<td>21</td>
<td>Domain names of the SIP outbound proxy servers for the client. See RFC 3319.</td>
</tr>
<tr>
<td>sip-servers-address AT_IP6ADDR</td>
<td>22</td>
<td>IPv6 addresses of the SIP outbound proxy servers for the client.</td>
</tr>
<tr>
<td>dns-servers AT_IP6ADDR</td>
<td>23</td>
<td>IPv6 addresses of DNS recursive name servers.</td>
</tr>
<tr>
<td>domain-list AT_DNSNAME</td>
<td>24</td>
<td>Domain names in the domain search list.</td>
</tr>
</tbody>
</table>
### Table B-9 DHCPv6 Options (continued)

<table>
<thead>
<tr>
<th>Cisco Prime IP Express Name (Type)</th>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ia-pd AT_BLOB</td>
<td>25</td>
<td>IPv6 prefix delegation identity association and its associated parameters and prefixes. Parameters are the unique ID, time the client contacts the addresses in the IA to extend the lifetime, and time the client contacts any available server to extend the lifetime of the addresses.</td>
</tr>
<tr>
<td>iaprefix AT_BLOB</td>
<td>26</td>
<td>IPv6 prefixes associated with an IA_PD. The prefix must be encapsulated in the options field of an IA_PD option. Parameters are the valid and preferred lifetimes, prefix length, and the prefix.</td>
</tr>
<tr>
<td>nis-servers AT_IP6ADDR</td>
<td>27</td>
<td>List of IPv6 addresses of Network Information Service (NIS) servers available to the client (see RFC 3898).</td>
</tr>
<tr>
<td>nisp-servers AT_IP6ADDR</td>
<td>28</td>
<td>List of IPv6 addresses of NIS+ servers available to the client.</td>
</tr>
<tr>
<td>nis-domain-name AT_DNSNAME</td>
<td>29</td>
<td>Conveys the NIS domain name to the client.</td>
</tr>
<tr>
<td>nisp-domain-name AT_DNSNAME</td>
<td>30</td>
<td>Conveys the NIS+ domain name to the client.</td>
</tr>
<tr>
<td>sntp-servers AT_IP6ADDR</td>
<td>31</td>
<td>List of Simple Network Time Protocol (SNTP) servers available to the client (see RFC 4075).</td>
</tr>
<tr>
<td>info-refresh-time AT_TIME</td>
<td>32</td>
<td>Sets an upper bound for how long a client should wait before refreshing DHCPv6 information (see RFC 4242).</td>
</tr>
<tr>
<td>bcms-server-d AT_DNSNAME</td>
<td>33</td>
<td>List of BCMS controller domains (see RFC 4280).</td>
</tr>
<tr>
<td>bcms-server-a AT_IP6ADDR</td>
<td>34</td>
<td>List of IPv6 addresses for the Broadcast and Multicast Service (BCMS) controller (see RFC 4280).</td>
</tr>
<tr>
<td>geoconf-civic AT_BLOB</td>
<td>36</td>
<td>DHCP civic addresses configuration.</td>
</tr>
<tr>
<td>remote-id AT_BLOB</td>
<td>37</td>
<td>Relay agents that terminate switched or permanent circuits can add this option to identify remote hosts (see RFC 4649).</td>
</tr>
<tr>
<td>relay-agent-subscriber-id AT_BLOB</td>
<td>38</td>
<td>Allows assignment and activation of subscriber-specific actions (see RFC 4580).</td>
</tr>
<tr>
<td>client-fqdn AT_BLOB</td>
<td>39</td>
<td>DHCP client FQDN.</td>
</tr>
<tr>
<td>new-posix-timezone AT_BLOB</td>
<td>41</td>
<td>POSIX time zone, for example, EST5EDT4, M3.2.0/02:00,M11.1.0/02:00.</td>
</tr>
<tr>
<td>new-tzdb-timezone AT_BLOB</td>
<td>42</td>
<td>POSIX time zone database name, for example, Europe/Zurich.</td>
</tr>
<tr>
<td>ero AT_BLOB</td>
<td>43</td>
<td>Relay agent Echo Request option to inform the server of the list of relay agent options to echo back.</td>
</tr>
<tr>
<td>lq-query AT_BLOB</td>
<td>44</td>
<td>Used only in a LEASEQUERY message; identifies the query being performed. The option includes the query type, link-address (or 0::0), and options to provide data needed for the query.</td>
</tr>
</tbody>
</table>
### Table B-9 DHCPv6 Options (continued)

<table>
<thead>
<tr>
<th>Cisco Prime IP Express Name (Type)</th>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-data AT_BLOB</td>
<td>45</td>
<td>Encapsulates the data for a single client on a single link in a LEASEQUERY-REPLY message.</td>
</tr>
<tr>
<td>clt-time AT_TIME</td>
<td>46</td>
<td>Client last transaction time encapsulated in the client-data option; identifies how long ago the server last communicated with the client (in seconds).</td>
</tr>
<tr>
<td>lq-relay-data AT_BLOB</td>
<td>47</td>
<td>Used only in a LEASEQUERY-REPLY message; provides the relay agent data used when the client last communicated with the server.</td>
</tr>
<tr>
<td>lq-client-link AT_IP6ADDR</td>
<td>48</td>
<td>Used only in a LEASEQUERY-REPLY message; identifies the links on which the client has one or more bindings. It is used in reply to a query when no link-address was specified and the client is found to be on more than one link.</td>
</tr>
<tr>
<td>lost-server AT_DNSNAME</td>
<td>51</td>
<td>A DHCPv6 client will request a LoST server domain name in an Options Request Option (ORO) (see RFC 3315). This option contains a single domain name and must contain precisely one root label.</td>
</tr>
<tr>
<td>capwap_ac_v6 AT_STRING</td>
<td>52</td>
<td>Carries a list of 128-bit (binary) IPv6 addresses indicating one or more Control and Provisioning of Wireless Access Point (CAPWAP) Access Controllers (ACs) available to the Wireless Termination Point (WTP) (see RFC 5417).</td>
</tr>
<tr>
<td>mos-address AT_IP6ADDR</td>
<td>54</td>
<td>Mobility Sever (MoS) IPv6 Address for DHCP v4.</td>
</tr>
<tr>
<td>mos-fqdn AT_BLOB</td>
<td>55</td>
<td>Mobility Sever (MoS) Domain Name List for DHCPv6.</td>
</tr>
<tr>
<td>ntp-server AT_BLOB</td>
<td>56</td>
<td>Serves as a container for server location information related to one Network Time Protocol (NTP) server or Simple Network Time Protocol (SNTP) server. This option can appear multiple times in a DHCPv6 message. Each instance of this option is to be considered by the NTP client or SNTP client as a server to include in its configuration. The option itself does not contain any value. Instead, it contains one or several suboptions that carry NTP server or SNTP server location.</td>
</tr>
<tr>
<td>access-domain AT_DNSNAME</td>
<td>57</td>
<td>Defines the domain name associated with the access network. This option contains a single domain name and, as such, must contain precisely one root label.</td>
</tr>
<tr>
<td>sip-ua-cs-domains AT_DNSNAME</td>
<td>58</td>
<td>Defines the list of domain names in the Session Initiation Protocol (SIP) User Agent Configuration Service Domains.</td>
</tr>
<tr>
<td>bootfile-url AT_NSTRING</td>
<td>59</td>
<td>Informs the client about a URL to a boot file.</td>
</tr>
<tr>
<td>bootfile-param AT_TYPECNT</td>
<td>60</td>
<td>Sent by the server to the client. It consists of multiple UTF-8 (see RFC3629) strings for specifying parameters for the boot file.</td>
</tr>
<tr>
<td>client-arch-type AT_SHORT</td>
<td>61</td>
<td>Provides parity with the Client System Architecture Type option (option 93) defined for DHCPv4.</td>
</tr>
</tbody>
</table>
The following tables display the DHCP options in various ways. They show the options sorted numerically, by Cisco Prime IP Express name, and by category.

DHCP options have a prescribed format and allowed values for their option parameters. Table B-10 lists each DHCP option and parameter type (in the Validation column). The parameter formats and allowed values come from the DHCP and Internet RFCs. All the DHCP options appear, but clients control only some, and the CLI only others.

### Options by Number

Table B-10 shows the DHCPv4 options sorted by option number, and includes the validation type. (See Table B-12 on page B-26 for details on the option validation types found in the Validation column.) A 0+ in the Comments column means a repeat count of zero or more occurrences, 1+ means one or more occurrences, 2n means multiple occurrences in multiples of 2. Comments also indicate whether the option includes suboptions, and, if so, how many.

Tip
---
For the syntax for adding more complex option data values for suboptions, see the “Adding Complex Values for Suboptions” section on page 22-7.

<table>
<thead>
<tr>
<th>No.</th>
<th>Cisco Prime IP Express Name</th>
<th>Protocol Name</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>pad</td>
<td>Pad</td>
<td>AT_NOLEN</td>
</tr>
<tr>
<td>1</td>
<td>subnet-mask</td>
<td>Subnet Mask</td>
<td>AT_IPADDR</td>
</tr>
<tr>
<td>2</td>
<td>time-offset</td>
<td>Time Offset</td>
<td>AT_STIME</td>
</tr>
<tr>
<td>3</td>
<td>routers</td>
<td>Router</td>
<td>AT_IPADDR</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Cisco Prime IP Express Name (Type)</th>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nii AT_BLOB</td>
<td>62</td>
<td>Provides parity with the Client Network Interface Identifier option (option 94) defined for DHCPv4.</td>
</tr>
<tr>
<td>geoloc AT_BLOB</td>
<td>63</td>
<td>Specifies the coordinate-based geographic location of the client, to be provided by the server.</td>
</tr>
<tr>
<td>afrt-name AT_DNSNAME</td>
<td>64</td>
<td>Defines a fully qualified domain name of the AFTR tunnel endpoint.</td>
</tr>
<tr>
<td>erp-local-domain-name AT_DNSNAME</td>
<td>65</td>
<td>Contains the name of the local ERP domain.</td>
</tr>
<tr>
<td>No.</td>
<td>Cisco Prime IP Express Name</td>
<td>Protocol Name</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>4</td>
<td>time-servers</td>
<td>Time Server</td>
</tr>
<tr>
<td>5</td>
<td>name-servers</td>
<td>Name Server</td>
</tr>
<tr>
<td>6</td>
<td>domain-name-servers</td>
<td>Domain Server</td>
</tr>
<tr>
<td>7</td>
<td>log-servers</td>
<td>Log Server</td>
</tr>
<tr>
<td>8</td>
<td>cookie-servers</td>
<td>Quotes Server</td>
</tr>
<tr>
<td>9</td>
<td>lpr-servers</td>
<td>LPR Server</td>
</tr>
<tr>
<td>10</td>
<td>impress-servers</td>
<td>Impress Server</td>
</tr>
<tr>
<td>11</td>
<td>resource-location-servers</td>
<td>RLP Server</td>
</tr>
<tr>
<td>12</td>
<td>host-name</td>
<td>Host Name</td>
</tr>
<tr>
<td>13</td>
<td>boot-size</td>
<td>Boot File Size</td>
</tr>
<tr>
<td>14</td>
<td>merit-dump</td>
<td>Merit Dump File</td>
</tr>
<tr>
<td>15</td>
<td>domain-name</td>
<td>Domain Name</td>
</tr>
<tr>
<td>16</td>
<td>swap-server</td>
<td>Swap Server</td>
</tr>
<tr>
<td>17</td>
<td>root-path</td>
<td>Root Path</td>
</tr>
<tr>
<td>18</td>
<td>extensions-path</td>
<td>Extension File</td>
</tr>
<tr>
<td>19</td>
<td>ip-forwarding</td>
<td>Forward On/Off</td>
</tr>
<tr>
<td>20</td>
<td>non-local-source-routing</td>
<td>SrcRte On/Off</td>
</tr>
<tr>
<td>21</td>
<td>policy-filters</td>
<td>Policy Filter</td>
</tr>
<tr>
<td>22</td>
<td>max-dgram-reassembly</td>
<td>Maximum DG Assembly</td>
</tr>
<tr>
<td>23</td>
<td>Default-ip-ttl</td>
<td>Default IP TTL</td>
</tr>
<tr>
<td>24</td>
<td>path-mtu-aging-timeout</td>
<td>MTU Timeout</td>
</tr>
<tr>
<td>25</td>
<td>path-mtu-plateau-tables</td>
<td>MTU Plateau</td>
</tr>
<tr>
<td>26</td>
<td>interface-mtu</td>
<td>MTU Interface</td>
</tr>
<tr>
<td>27</td>
<td>all-subnets-local</td>
<td>MTU Subnet</td>
</tr>
<tr>
<td>28</td>
<td>broadcast-address</td>
<td>Broadcast Address</td>
</tr>
<tr>
<td>29</td>
<td>perform-mask-discovery</td>
<td>Mask Discovery</td>
</tr>
<tr>
<td>30</td>
<td>mask-supplier</td>
<td>Mask Supplier</td>
</tr>
<tr>
<td>31</td>
<td>router-discovery</td>
<td>Router Discovery</td>
</tr>
<tr>
<td>32</td>
<td>router-solicitation-address</td>
<td>Router Request</td>
</tr>
<tr>
<td>33</td>
<td>static-routes</td>
<td>Static Route</td>
</tr>
<tr>
<td>34</td>
<td>trailer-encapsulation</td>
<td>Trailers</td>
</tr>
<tr>
<td>35</td>
<td>arp-cache-timeout</td>
<td>ARP Timeout</td>
</tr>
<tr>
<td>36</td>
<td>ieee802.3-encapsulation</td>
<td>Ethernet</td>
</tr>
<tr>
<td>37</td>
<td>default-tcp-ttl</td>
<td>Default TCP TTL</td>
</tr>
<tr>
<td>38</td>
<td>tcp-keepalive-interval</td>
<td>Keepalive Time</td>
</tr>
</tbody>
</table>
### Table B-10  DHCPv4 Options by Number (continued)

<table>
<thead>
<tr>
<th>No.</th>
<th>Cisco Prime IP Express Name</th>
<th>Protocol Name</th>
<th>Validation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>tcp-keepalive-garbage</td>
<td>Keepalive Data</td>
<td>AT_BOOL</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>nis-domain</td>
<td>NIS Domain</td>
<td>AT_STRING</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>nis-servers</td>
<td>NIS Servers</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>42</td>
<td>ntp-servers</td>
<td>NTP Servers</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>43</td>
<td>vendor-encapsulated-options</td>
<td>Vendor Specific</td>
<td>AT_BLOB</td>
<td>NM</td>
</tr>
<tr>
<td>44</td>
<td>netbios-name-servers</td>
<td>NetBIOS Name Server</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>45</td>
<td>netbios-dd-servers</td>
<td>NetBIOS Distribution Server</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>46</td>
<td>netbios-node-type</td>
<td>NetBIOS Node Type</td>
<td>AT_RANGBYTE</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>netbios-scope</td>
<td>NetBIOS Scope</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>font-servers</td>
<td>X Window Font</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>49</td>
<td>x-display-managers</td>
<td>X Window Manager</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>50</td>
<td>dhcp-requested-address</td>
<td>Address Request</td>
<td>AT_IPADDR</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>dhcp-lease-time</td>
<td>Address Time</td>
<td>AT_TIME</td>
<td>NM</td>
</tr>
<tr>
<td>52</td>
<td>dhcp-option-overload</td>
<td>Overload</td>
<td>AT_OVERLOAD</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>dhcp-message-type</td>
<td>DHCP Message Type</td>
<td>AT_MESSAGE</td>
<td>NM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(See the DHCP Message Type option in Table B-7 on page B-8)</td>
</tr>
<tr>
<td>54</td>
<td>dhcp-server-identifier</td>
<td>DHCP Server ID</td>
<td>AT_IPADDR</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>dhcp-parameter-request-list</td>
<td>Parameter List</td>
<td>AT_INT8</td>
<td>0+</td>
</tr>
<tr>
<td>56</td>
<td>dhcp-message</td>
<td>DHCP Message</td>
<td>AT_NSTRING</td>
<td>NM</td>
</tr>
<tr>
<td>57</td>
<td>dhcp-max-message-size</td>
<td>DHCP Maximum Message Size</td>
<td>AT_SHORT</td>
<td>NM</td>
</tr>
<tr>
<td>58</td>
<td>dhcp-renewal-time</td>
<td>Renewing Time</td>
<td>AT_TIME</td>
<td>NM</td>
</tr>
<tr>
<td>59</td>
<td>dhcp-rebinding-time</td>
<td>Rebinding Time</td>
<td>AT_TIME</td>
<td>NM</td>
</tr>
<tr>
<td>60</td>
<td>dhcp-class-identifier</td>
<td>Class Identifier</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>dhcp-client-identifier</td>
<td>Client Identifier</td>
<td>AT_BLOB</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>netwareip-domain</td>
<td>NetWare/IP Domain</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>netwareip-information</td>
<td>NetWare/IP Option</td>
<td>AT_BLOB</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>nis+-domain</td>
<td>NIS Domain Name</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>nis+-servers</td>
<td>NIS Server Address</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>66</td>
<td>tftp-server</td>
<td>TFTP Server Name</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>boot-file</td>
<td>Bootfile Name</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>mobile-ip-home-agents</td>
<td>Mobile IP Home Agent</td>
<td>AT_IPADDR</td>
<td>0+</td>
</tr>
<tr>
<td>No.</td>
<td>Cisco Prime IP Express Name</td>
<td>Protocol Name</td>
<td>Validation</td>
<td>Comments</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------</td>
<td>---------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>69</td>
<td>smtp-servers</td>
<td>SMTP Server</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>70</td>
<td>pop3-servers</td>
<td>POP3 Server</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>71</td>
<td>nntp-servers</td>
<td>NNTP Server</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>72</td>
<td>www-servers</td>
<td>WWW Server</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>73</td>
<td>finger-servers</td>
<td>Finger Server</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>74</td>
<td>irc-servers</td>
<td>IRC Server</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>75</td>
<td>streettalk-servers</td>
<td>StreetTalk Server</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>76</td>
<td>streettalk-directory-assistance-servers</td>
<td>STDA Server</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>77</td>
<td>dhcp-user-class-id</td>
<td>User Class ID</td>
<td>AT_TYPECNT</td>
<td>Suboptions (2)</td>
</tr>
<tr>
<td>78</td>
<td>slp-directory-agent</td>
<td>Service Location Protocol Directory Agent</td>
<td>AT_BLOB</td>
<td>Suboptions (2)</td>
</tr>
<tr>
<td>79</td>
<td>slp-service-scope</td>
<td>SLP Service Scope</td>
<td>AT_BLOB</td>
<td>Suboptions (2)</td>
</tr>
<tr>
<td>80</td>
<td>rapid-commit</td>
<td>Rapid Commit</td>
<td>AT_ZEROSIZE</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>client-fqdn</td>
<td>Client FQDN</td>
<td>AT_BLOB</td>
<td>Suboptions (4)</td>
</tr>
<tr>
<td>82</td>
<td>relay-agent-info</td>
<td>Relay Agent Information</td>
<td>AT_BLOB</td>
<td>For suboptions, see Table C-3 on page C-2</td>
</tr>
<tr>
<td>83</td>
<td>iSNS</td>
<td>Internet Storage Name Service (RFC4174)</td>
<td>AT_BLOB</td>
<td>Suboptions (7)</td>
</tr>
<tr>
<td>85</td>
<td>nds-servers</td>
<td>NDS Servers</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>86</td>
<td>nds-tree</td>
<td>NDS Tree Name</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>nds-context</td>
<td>NDS Context</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>bcms-servers-d</td>
<td>BCMS Controller Domain (RFC 4280)</td>
<td>AT_DNSNAME</td>
<td>1+</td>
</tr>
<tr>
<td>89</td>
<td>bcms-servers-a</td>
<td>BCMS Address</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>90</td>
<td>authentication</td>
<td>Authentication</td>
<td>AT_BLOB</td>
<td>Suboptions (5)</td>
</tr>
<tr>
<td>91</td>
<td>client-last-transaction-time</td>
<td>Lease Query Client Last Transaction Time</td>
<td>AT_TIME</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>associated-ip</td>
<td>Lease Query Associated IP Addresses</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>93</td>
<td>pxe-client-arch</td>
<td>AT_SHORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>pxe-client-network-id</td>
<td>AT_BLOB</td>
<td>Suboptions (2)</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>ldap-url</td>
<td>AT_NSTRING</td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>pxe-client-machine-id</td>
<td>AT_BLOB</td>
<td>Suboptions (2)</td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>user-auth</td>
<td>AT_NSTRING</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table B-10  DHCPv4 Options by Number (continued)

<table>
<thead>
<tr>
<th>No.</th>
<th>Cisco Prime IP Express Name</th>
<th>Protocol Name</th>
<th>Validation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>geoconf-civic</td>
<td>Civic Addresses Configuration</td>
<td>AT_BLOB</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>tz-posix</td>
<td>IEEE 1003.1 String</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>tz-database</td>
<td>Time Zone Database</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>netinfo-parent-server-addr</td>
<td></td>
<td>AT_IPADDR</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>netinfo-parent-server-tag</td>
<td></td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>initial-url</td>
<td></td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>auto-configure</td>
<td>Autoconfiguration</td>
<td>AT_RANGEBYTE</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>name-service-search</td>
<td>Name Service Search</td>
<td>AT_SHORT 1+</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>subnet-selection</td>
<td>Subnet Selection</td>
<td>AT_IPADDR</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>domain-search</td>
<td>Domain Search</td>
<td>AT_BLOB</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>sip-servers</td>
<td>SIP Servers</td>
<td>AT_BLOB</td>
<td>Suboptions (2)</td>
</tr>
<tr>
<td>121</td>
<td>classless-static-route</td>
<td>Classless Static Route</td>
<td>AT_BLOB</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>cablelabs-client-configuration)</td>
<td>CableLabs Client Configuration</td>
<td>AT_BLOB</td>
<td>Suboptions (10) (see the “cablelabs-client-configuration” section on page C-3)</td>
</tr>
<tr>
<td>123</td>
<td>geo-conf</td>
<td>GeoConf Option</td>
<td>AT_BLOB</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>v-i-vendor-class</td>
<td>Vendor-Identifying Vendor Class</td>
<td>AT_VENDOR_CLASS</td>
<td>NM</td>
</tr>
<tr>
<td>125</td>
<td>v-i-vendor-info</td>
<td>Vendor-Identifying Vendor-Specific Info</td>
<td>AT_VENDOR_OPTS</td>
<td>See also the cablelabs-125 suboptions in Table C-3 on page C-2</td>
</tr>
<tr>
<td>128</td>
<td>mcns-security-server</td>
<td>--</td>
<td>AT_IPADDR</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td>capwap-ac-v4</td>
<td></td>
<td>AT_IPADDR 1+</td>
<td></td>
</tr>
<tr>
<td>139</td>
<td>mos-address</td>
<td></td>
<td>AT_BLOB 0+</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>mos-fqdn</td>
<td></td>
<td>AT_BLOB 0+</td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>sip-ua-cs-domains</td>
<td></td>
<td>AT_DNSNAME 0+</td>
<td></td>
</tr>
<tr>
<td>142</td>
<td>andsf-v4</td>
<td>Access Network Discovery and Selection Function</td>
<td>AT_IPADDR</td>
<td></td>
</tr>
<tr>
<td>143</td>
<td>andsf-v6</td>
<td>Access Network Discovery and Selection Function</td>
<td>AT_IP6ADDR</td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>geoloc</td>
<td>Geospatial Location with Uncertainty</td>
<td>AT_BLOB</td>
<td></td>
</tr>
</tbody>
</table>
### Table B-10 DHCPv4 Options by Number (continued)

<table>
<thead>
<tr>
<th>No.</th>
<th>Cisco Prime IP Express Name</th>
<th>Protocol Name</th>
<th>Validation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>tftp-server-address</td>
<td>TFTP</td>
<td>AT_IPADDR</td>
<td>1+</td>
</tr>
<tr>
<td>161</td>
<td>cisco-leased-ip</td>
<td>Cisco</td>
<td>AT_IPADDR</td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>cisco-client-requested-host-name</td>
<td>Cisco</td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>cisco-client-last-transaction-time</td>
<td>Cisco</td>
<td>AT_INT</td>
<td></td>
</tr>
<tr>
<td>185</td>
<td>vpn-id</td>
<td>VPN Identifier</td>
<td>AT_BLOB</td>
<td>NM: Suboptions (2)</td>
</tr>
<tr>
<td>209</td>
<td>pxelinux-config-file</td>
<td></td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>pxelinux-path-prefix</td>
<td></td>
<td>AT_NSTRING</td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>prelinux-reboot-time</td>
<td></td>
<td>AT_TIME</td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>6rd</td>
<td></td>
<td>AT_BLOB</td>
<td></td>
</tr>
<tr>
<td>213</td>
<td>access-domain</td>
<td></td>
<td>AT_DNSNAME</td>
<td></td>
</tr>
<tr>
<td>220</td>
<td>subnet-alloc</td>
<td>Subnet Allocation</td>
<td>AT_BLOB</td>
<td>Suboptions (5)</td>
</tr>
<tr>
<td>221</td>
<td>cisco-vpn-id</td>
<td>Cisco VPN Identifier</td>
<td>AT_BLOB</td>
<td>Suboptions (2)</td>
</tr>
<tr>
<td>251</td>
<td>cisco-auto-configure</td>
<td>Cisco Autoconfiguration</td>
<td>AT_RANGEBYTE</td>
<td></td>
</tr>
<tr>
<td>255</td>
<td>end</td>
<td></td>
<td>AT_NOLEN</td>
<td>NM</td>
</tr>
</tbody>
</table>

### Options by Cisco Prime IP Express Name

Table B-11 lists the DHCP options by Cisco Prime IP Express name. (For each option validation type, cross-reference it by number to Table B-10 and check the Validation column.)

### Table B-11 DHCP Options by Cisco Prime IP Express Name

<table>
<thead>
<tr>
<th>Cisco Prime IP Express Name</th>
<th>No.</th>
<th>Option Name</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>access-domain</td>
<td>213</td>
<td>Access Network Domain Name</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>access-domain</td>
<td>57</td>
<td>Access Network Domain Name</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>aftr-name</td>
<td>64</td>
<td>AFTR tunnel endpoint domain name</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>all-subnets-local</td>
<td>27</td>
<td>All Subnets Are Local</td>
<td>Interface</td>
</tr>
<tr>
<td>arp-cache-timeout</td>
<td>35</td>
<td>ARP Cache Timeout</td>
<td>Interface</td>
</tr>
<tr>
<td>associated-ip</td>
<td>92</td>
<td>Lease Query Associated IP</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>auth</td>
<td>11</td>
<td>Authentication</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>authentication</td>
<td>90</td>
<td>Authentication</td>
<td>--</td>
</tr>
<tr>
<td>auto-configuration</td>
<td>116</td>
<td>Auto-Configuration</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>bcms-server-a</td>
<td>34</td>
<td>BCMS Address v6</td>
<td>DHCPv6</td>
</tr>
</tbody>
</table>
### Table B-11  DHCP Options by Cisco Prime IP Express Name (continued)

<table>
<thead>
<tr>
<th>Cisco Prime IP Express Name</th>
<th>No.</th>
<th>Option Name</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>bcms-servers-a</td>
<td>89</td>
<td>BCMS Address</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>bcms-server-d</td>
<td>33</td>
<td>BCMS Controller Domain v6</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>bcms-servers-d</td>
<td>88</td>
<td>BCMS Controller Domain</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>bootfile-url</td>
<td>59</td>
<td>Boot File Uniform Resource Locator (URL)</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>bootfile-param</td>
<td>60</td>
<td>Boot File Parameters</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>boot-file</td>
<td>67</td>
<td>Bootfile Name</td>
<td>BOOTP</td>
</tr>
<tr>
<td>boot-size</td>
<td>13</td>
<td>Boot File Size</td>
<td>BOOTP</td>
</tr>
<tr>
<td>broadcast-address</td>
<td>28</td>
<td>Broadcast Address</td>
<td>Interface</td>
</tr>
<tr>
<td>cablelabs-client-configuration</td>
<td>122</td>
<td>CableLabs Client Configuration</td>
<td>Interface</td>
</tr>
<tr>
<td>capwap_ac_v4</td>
<td>138</td>
<td>CAPWAP AC</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>capwap_ac_v6</td>
<td>52</td>
<td>CAPWAP AC</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>cisco-autoconfigure</td>
<td>251</td>
<td>Cisco Autoconfiguration</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>cisco-client-last-transaction-time</td>
<td>163</td>
<td>Cisco Client Last Transaction Time</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>cisco-client-requested-host-name</td>
<td>162</td>
<td>Cisco Client Requested Host Name</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>cisco-leased-ip</td>
<td>161</td>
<td>Cisco Leased IP Address</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>cisco-vpn-id</td>
<td>221</td>
<td>Cisco VPN Identifier</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>classless-static-route</td>
<td>121</td>
<td>Classless Static Route</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>client-arch-type</td>
<td>61</td>
<td>Client System Architecture Type</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>client-data</td>
<td>45</td>
<td>Leasequery Reply Client Data</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>client-fqdn</td>
<td>81</td>
<td>DHCP Client FQDN</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>client-fqdn</td>
<td>39</td>
<td>DHCP Client FQDN</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>client-identifier</td>
<td>1</td>
<td>Client Identifier</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>client-last-transaction-time</td>
<td>91</td>
<td>Leasequery Client Last Transaction Time</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>clt-time</td>
<td>46</td>
<td>Leasequery Client Last Transaction Time</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>cookie-servers</td>
<td>8</td>
<td>Cookie Server</td>
<td>BOOTP</td>
</tr>
<tr>
<td>default-ip-ttl</td>
<td>23</td>
<td>Default IP Time-to-Live</td>
<td>Host IP</td>
</tr>
<tr>
<td>default-tcp-ttl</td>
<td>37</td>
<td>TCP Default TTL</td>
<td>Interface</td>
</tr>
<tr>
<td>dhcp-class-identifier</td>
<td>60</td>
<td>Vendor Class Identifier</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>dhcp-client-identifier</td>
<td>61</td>
<td>Client-Identifier</td>
<td>Basic</td>
</tr>
<tr>
<td>dhcp-lease-time</td>
<td>51</td>
<td>IP Address Lease Time</td>
<td>Lease Information, MS DHCP Client</td>
</tr>
<tr>
<td>dhcp-max-message-size</td>
<td>57</td>
<td>Maximum DHCP Message Size</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>dhcp-message-type</td>
<td>53</td>
<td>DHCP Message Type</td>
<td>DHCPv4</td>
</tr>
</tbody>
</table>
### Table B-11  DHCP Options by Cisco Prime IP Express Name (continued)

<table>
<thead>
<tr>
<th>Cisco Prime IP Express Name</th>
<th>No.</th>
<th>Option Name</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>dhcp-message</td>
<td>56</td>
<td>Message</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>dhcp-option-overload</td>
<td>52</td>
<td>Option Overload</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>dhcp-parameter-request-list</td>
<td>55</td>
<td>Parameter Request List</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>dhcp-rebinding-time</td>
<td>59</td>
<td>Rebinding (T2) Time Value</td>
<td>Lease Information, MS DHCP Client</td>
</tr>
<tr>
<td>dhcp-renewal-time</td>
<td>58</td>
<td>Renewing (T1) Time Value</td>
<td>Lease Information, MS DHCP Client</td>
</tr>
<tr>
<td>dhcp-requested-address</td>
<td>50</td>
<td>Requested IP Address</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>dhcp-server-identifier</td>
<td>54</td>
<td>Server Identifier</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>dhcp-user-class-id</td>
<td>77</td>
<td>User Class ID</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>dns-servers</td>
<td>23</td>
<td>DNS Recursive Name Server</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>domain-list</td>
<td>24</td>
<td>Domain Search List</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>domain-name</td>
<td>15</td>
<td>Domain Name</td>
<td>Basic, MS DHCP Client</td>
</tr>
<tr>
<td>domain-name-servers</td>
<td>6</td>
<td>Domain Name Server</td>
<td>Basic, MS DHCP Client</td>
</tr>
<tr>
<td>domain-search</td>
<td>119</td>
<td>Domain Search</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>elapsed-time</td>
<td>8</td>
<td>Elapsed Time</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>end</td>
<td>255</td>
<td>End</td>
<td>--</td>
</tr>
<tr>
<td>ero</td>
<td>43</td>
<td>Relay Agent Echo Request Option</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>erp-local-domain-name</td>
<td>65</td>
<td>Local ERP domain name</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>extensions-path</td>
<td>18</td>
<td>Extensions Path</td>
<td>BOOTP</td>
</tr>
<tr>
<td>finger-servers</td>
<td>73</td>
<td>Finger Server</td>
<td>Servers</td>
</tr>
<tr>
<td>font-servers</td>
<td>48</td>
<td>X Window System Font Server</td>
<td>Servers</td>
</tr>
<tr>
<td>geo-conf</td>
<td>123</td>
<td>GeoConf</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>geoconf-civic</td>
<td>99</td>
<td>Civic Addresses Configuration</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>geoconf-civic</td>
<td>36</td>
<td>Civic Addresses Configuration</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>geoloc</td>
<td>63</td>
<td>Geolocation</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>host-name</td>
<td>12</td>
<td>Host Name</td>
<td>Basic</td>
</tr>
<tr>
<td>ia-na</td>
<td>3</td>
<td>Identity Association for Nontemporary Addresses</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>ia-pd</td>
<td>25</td>
<td>Prefix Delegation</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>ia-ta</td>
<td>4</td>
<td>Identity Association for Temporary Addresses</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>iaaddr</td>
<td>5</td>
<td>IA Address</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>iaprefix</td>
<td>26</td>
<td>IA Prefix</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>ieee802.3-encapsulation</td>
<td>36</td>
<td>Ethernet Encapsulation</td>
<td>Interface</td>
</tr>
<tr>
<td>impress-servers</td>
<td>10</td>
<td>Impress Server</td>
<td>BOOTP</td>
</tr>
<tr>
<td>Cisco Prime IP Express Name</td>
<td>No.</td>
<td>Option Name</td>
<td>Category</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-----</td>
<td>-------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>info-refresh-time</td>
<td>32</td>
<td>Information Refresh Time</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>interface-id</td>
<td>18</td>
<td>Interface Identifier</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>interface-mtu</td>
<td>26</td>
<td>Interface MTU</td>
<td>Interface</td>
</tr>
<tr>
<td>ip-forwarding</td>
<td>19</td>
<td>IP Forwarding Enable/Disable</td>
<td>Host IP</td>
</tr>
<tr>
<td>irc-servers</td>
<td>74</td>
<td>IRC Server</td>
<td>Servers</td>
</tr>
<tr>
<td>isns</td>
<td>83</td>
<td>iSNS</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>log-servers</td>
<td>7</td>
<td>Log Server</td>
<td>Servers</td>
</tr>
<tr>
<td>lpr-servers</td>
<td>9</td>
<td>LPR Server</td>
<td>Servers</td>
</tr>
<tr>
<td>lq-client-link</td>
<td>48</td>
<td>Leasequery Client Link Reply</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>lq-query</td>
<td>44</td>
<td>Leasequery</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>lq-relay-data</td>
<td>47</td>
<td>Leasequery Relay Agent Reply</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>mask-supplier</td>
<td>30</td>
<td>Mask Supplier</td>
<td>Interface</td>
</tr>
<tr>
<td>max-dgram-reassembly</td>
<td>22</td>
<td>Maximum Datagram Reassembly Size</td>
<td>Host IP</td>
</tr>
<tr>
<td>mcns-security-server</td>
<td>128</td>
<td>--</td>
<td>Servers</td>
</tr>
<tr>
<td>merit-dump</td>
<td>14</td>
<td>Merit Dump File</td>
<td>BOOTP</td>
</tr>
<tr>
<td>mobile-ip-home-agents</td>
<td>68</td>
<td>Mobile IP Home Agent</td>
<td>Servers</td>
</tr>
<tr>
<td>mos-address</td>
<td>139</td>
<td>MoS IPv4 Address</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>mos-address</td>
<td>54</td>
<td>MoS IPv6 Address</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>mos-fqdn</td>
<td>140</td>
<td>MoS Domain Name List</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>mos-fqdn</td>
<td>55</td>
<td>MoS Domain Name List</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>name-servers</td>
<td>5</td>
<td>Name Server</td>
<td>BOOTP</td>
</tr>
<tr>
<td>name-service-search</td>
<td>117</td>
<td>Name Service Search</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>nds-context</td>
<td>87</td>
<td>NDS Context</td>
<td>NetWare Client</td>
</tr>
<tr>
<td>nds-servers</td>
<td>85</td>
<td>NDS Servers</td>
<td>NetWare Client</td>
</tr>
<tr>
<td>nds-tree</td>
<td>86</td>
<td>NDS Tree Name</td>
<td>NetWare Client</td>
</tr>
<tr>
<td>netbios-dd-servers</td>
<td>45</td>
<td>NetBIOS over TCP/IP Datagram Distribution Server</td>
<td>WINS/NetBIOS</td>
</tr>
<tr>
<td>netbios-name-servers</td>
<td>44</td>
<td>NetBIOS over TCP/IP Name Server</td>
<td>WINS/NetBIOS, MS DHCP Client</td>
</tr>
<tr>
<td>netbios-node-type</td>
<td>46</td>
<td>NetBIOS over TCP/IP Node Type</td>
<td>WINS/NetBIOS, MS DHCP Client</td>
</tr>
<tr>
<td>netbios-scope</td>
<td>47</td>
<td>NetBIOS over TCP/IP Scope</td>
<td>WINS/NetBIOS, MS DHCP Client</td>
</tr>
<tr>
<td>netwareip-domain</td>
<td>62</td>
<td>NetWare/IP Domain Name</td>
<td>NetWare Client</td>
</tr>
<tr>
<td>netwareip-information</td>
<td>63</td>
<td>NetWare/IP Information</td>
<td>NetWare Client</td>
</tr>
<tr>
<td>new-posix-timezone</td>
<td>41</td>
<td>POSIX time zone string</td>
<td>DHCPv6</td>
</tr>
</tbody>
</table>
Table B-11  DHCP Options by Cisco Prime IP Express Name (continued)

<table>
<thead>
<tr>
<th>Cisco Prime IP Express Name</th>
<th>No.</th>
<th>Option Name</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>new-tzdb-timezone</td>
<td>42</td>
<td>POSIX time zone database name</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>nii</td>
<td>62</td>
<td>Client Network Interface Identifier</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>nis+-domain</td>
<td>64</td>
<td>NIS+ Domain</td>
<td>Servers</td>
</tr>
<tr>
<td>nis+-servers</td>
<td>65</td>
<td>Network Information Service (NIS+) Servers</td>
<td>Servers</td>
</tr>
<tr>
<td>nis-domain</td>
<td>40</td>
<td>NIS Domain</td>
<td>Servers</td>
</tr>
<tr>
<td>nis-domain-name</td>
<td>29</td>
<td>NIS Domain Name</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>nis-servers</td>
<td>41</td>
<td>Network Information Service (NIS) Servers</td>
<td>Servers</td>
</tr>
<tr>
<td>nis-servers</td>
<td>27</td>
<td>NIS Servers</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>nisp-domain-name</td>
<td>30</td>
<td>NIS+ Domain Name</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>nisp-servers</td>
<td>28</td>
<td>NIS+ Servers</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>nntp-servers</td>
<td>71</td>
<td>NNTP Server</td>
<td>Servers</td>
</tr>
<tr>
<td>non-local-source-routing</td>
<td>20</td>
<td>Non-Local Source Routing</td>
<td>Host IP</td>
</tr>
<tr>
<td>ntp-server</td>
<td>56</td>
<td>Message</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>ntp-servers</td>
<td>42</td>
<td>NTP Servers</td>
<td>Servers</td>
</tr>
<tr>
<td>option-time</td>
<td>8</td>
<td>Option Time</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>oro</td>
<td>6</td>
<td>Option Request Option</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>pad</td>
<td>0</td>
<td>Pad</td>
<td>--</td>
</tr>
<tr>
<td>path-mtu-aging-timeout</td>
<td>24</td>
<td>Path MTU Aging Timeout</td>
<td>Host IP</td>
</tr>
<tr>
<td>path-mtu-plateau-tables</td>
<td>25</td>
<td>Path MTU Plateau Table</td>
<td>Host IP</td>
</tr>
<tr>
<td>perform-mask-discovery</td>
<td>29</td>
<td>Perform Mask Discovery</td>
<td>Interface</td>
</tr>
<tr>
<td>policy-filters</td>
<td>21</td>
<td>Policy Filter</td>
<td>Host IP</td>
</tr>
<tr>
<td>pop3-servers</td>
<td>70</td>
<td>POP3 Server</td>
<td>Servers</td>
</tr>
<tr>
<td>preference</td>
<td>7</td>
<td>Preference</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>pxelinux-config-file</td>
<td>209</td>
<td>Configuration File</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>pxelinux-path-prefix</td>
<td>210</td>
<td>Path Prefix</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>pxelinux-reboot-time</td>
<td>211</td>
<td>Reboot Time</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>rapid-commit</td>
<td>80</td>
<td>Rapid Commit</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>rapid-commit</td>
<td>14</td>
<td>Rapid Commit</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>reconfigure-accept</td>
<td>20</td>
<td>Reconfigure Accept</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>reconfigure-message</td>
<td>19</td>
<td>Reconfigure Message</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>relay-agent-info</td>
<td>82</td>
<td>DHCP Relay Agent Information</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>relay-agent-subscriber-id</td>
<td>38</td>
<td>Relay Agent Subscriber ID</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>relay-message</td>
<td>9</td>
<td>Relay Message</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>remote-id</td>
<td>37</td>
<td>Relay Agent Remote ID</td>
<td>DHCPv6</td>
</tr>
</tbody>
</table>
### Table B-11  DHCP Options by Cisco Prime IP Express Name (continued)

<table>
<thead>
<tr>
<th>Cisco Prime IP Express Name</th>
<th>No.</th>
<th>Option Name</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource-location-servers</td>
<td>11</td>
<td>Resource Location Server</td>
<td>BOOTP</td>
</tr>
<tr>
<td>root-path</td>
<td>17</td>
<td>Root Path</td>
<td>BOOTP</td>
</tr>
<tr>
<td>router-discovery</td>
<td>31</td>
<td>Perform Router Discovery</td>
<td>Interface</td>
</tr>
<tr>
<td>router-solicitation-address</td>
<td>32</td>
<td>Router Solicitation Address</td>
<td>Interface</td>
</tr>
<tr>
<td>routers</td>
<td>3</td>
<td>Router</td>
<td>Basic, MS DHCP Client</td>
</tr>
<tr>
<td>server-identifier</td>
<td>2</td>
<td>DHCPv6 Server Identifier</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>server-unicast</td>
<td>12</td>
<td>Server Unicast</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>sip-servers</td>
<td>120</td>
<td>SIP Servers</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>sip-servers-name</td>
<td>21</td>
<td>SIP Servers Domain Name List</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>sip-servers-address</td>
<td>22</td>
<td>SIP Servers IPv6 Address List</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>sip-ua-cs-domains</td>
<td>141</td>
<td>SIP UA Configuration Service Domains</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>sip-ua-cs-domains</td>
<td>58</td>
<td>SIP User Agent Configuration Service Domains</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>slp-directory agent</td>
<td>78</td>
<td>SLP Directory Agent</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>slp-service-scope</td>
<td>79</td>
<td>SLP Service Scope</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>smtp-servers</td>
<td>69</td>
<td>SMTP Server</td>
<td>Servers</td>
</tr>
<tr>
<td>sntp-servers</td>
<td>31</td>
<td>SNTP Configuration</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>static-route</td>
<td>33</td>
<td>Static Route</td>
<td>Interface</td>
</tr>
<tr>
<td>status-code</td>
<td>13</td>
<td>Status Code</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>streettalk-directory-assistance-servers</td>
<td>76</td>
<td>STDA Server</td>
<td>Servers</td>
</tr>
<tr>
<td>streettalk-servers</td>
<td>75</td>
<td>StreetTalk Server</td>
<td>Servers</td>
</tr>
<tr>
<td>subnet-mask</td>
<td>1</td>
<td>Subnet Mask</td>
<td>Basic</td>
</tr>
<tr>
<td>subnet-selection</td>
<td>118</td>
<td>Subnet Selection</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>swap-server</td>
<td>16</td>
<td>Swap Server</td>
<td>BOOTP</td>
</tr>
<tr>
<td>tcp-keepalive-garbage</td>
<td>39</td>
<td>TCP Keepalive Garbage</td>
<td>Interface</td>
</tr>
<tr>
<td>tcp-keepalive-interval</td>
<td>38</td>
<td>TCP Keepalive Interval</td>
<td>Interface</td>
</tr>
<tr>
<td>tftp-server</td>
<td>66</td>
<td>TFTP Server Name</td>
<td>Servers</td>
</tr>
<tr>
<td>time-offset</td>
<td>2</td>
<td>Time Offset</td>
<td>BOOTP</td>
</tr>
<tr>
<td>time-servers</td>
<td>4</td>
<td>Time Server</td>
<td>BOOTP</td>
</tr>
<tr>
<td>trailer-encapsulation</td>
<td>34</td>
<td>Trailer Encapsulation</td>
<td>Interface</td>
</tr>
<tr>
<td>tz-database</td>
<td>101</td>
<td>TZ Database String</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>tz-posix</td>
<td>100</td>
<td>IEEE 1003.1 String</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>user-auth</td>
<td>98</td>
<td>User Authentication</td>
<td>DHCPv4</td>
</tr>
<tr>
<td>user-class</td>
<td>15</td>
<td>User Class</td>
<td>DHCPv6</td>
</tr>
<tr>
<td>vendor-class</td>
<td>16</td>
<td>Vendor Class</td>
<td>DHCPv6</td>
</tr>
</tbody>
</table>
Option Validation Types

Table B-12 defines the DHCP option validation types. Note that you cannot use some of them to define custom options.

<table>
<thead>
<tr>
<th>Validation</th>
<th>Description—Web UI Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT_BLOB</td>
<td>List of binary bytes—binary</td>
</tr>
<tr>
<td>AT_BOOL</td>
<td>Boolean—boolean</td>
</tr>
<tr>
<td>AT_DATE</td>
<td>Bytes representing a date—date</td>
</tr>
<tr>
<td>AT_DNSNAME</td>
<td>DNS name—DNS name</td>
</tr>
<tr>
<td>AT_INT</td>
<td>Unsigned 32-bit integer—unsigned 32-bit</td>
</tr>
<tr>
<td>AT_INT8</td>
<td>8-bit integer—unsigned 8-bit</td>
</tr>
<tr>
<td>AT_INTI</td>
<td>Unsigned 32-bit integer (Intel)—unsigned 32-bit (Intel)</td>
</tr>
<tr>
<td>AT_IPADDR</td>
<td>32-bit IP address—IP address</td>
</tr>
<tr>
<td>AT_IP6ADDR</td>
<td>128-bit IPv6 address—IPv6 address</td>
</tr>
<tr>
<td>AT_MACADDR</td>
<td>Bytes representing a MAC address—MAC address</td>
</tr>
<tr>
<td>AT_MESSAGE</td>
<td>Unsigned 8-bit message (not usable for custom options)</td>
</tr>
<tr>
<td>AT_NOLEN</td>
<td>No length (used for PAD and END only)</td>
</tr>
<tr>
<td>AT_NSTRING</td>
<td>Sequence of ASCII characters—string</td>
</tr>
<tr>
<td>AT_OVERLOAD</td>
<td>Overload bytes (not usable for custom options)</td>
</tr>
<tr>
<td>AT_RANGEBYTE</td>
<td>Range of bytes (not usable for custom options)</td>
</tr>
<tr>
<td>AT_RANGEINT</td>
<td>Range of shorts (not usable for custom options)</td>
</tr>
<tr>
<td>AT_RDNSNAME</td>
<td>Relative DNS name—relative DNS name</td>
</tr>
<tr>
<td>AT_SHORT</td>
<td>Unsigned 16-bit integer—unsigned 16-bit</td>
</tr>
<tr>
<td>AT_SHRTI</td>
<td>Unsigned 16-bit integer (Intel)—unsigned 16-bit (Intel)</td>
</tr>
</tbody>
</table>
### Table B-12 Validation Types (continued)

<table>
<thead>
<tr>
<th>Validation</th>
<th>Description—Web UI Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT_SINT</td>
<td>Signed 32-bit integer—signed 32-bit</td>
</tr>
<tr>
<td>AT_SINT8</td>
<td>8-bit integer—signed 8-bit</td>
</tr>
<tr>
<td>AT_SINTI</td>
<td>Signed 32-bit integer (Intel)—signed 32-bit (Intel)</td>
</tr>
<tr>
<td>AT_SSHORT</td>
<td>Signed 16-bit integer—signed 16-bit</td>
</tr>
<tr>
<td>AT_SSHRTI</td>
<td>Signed 16-bit integer (Intel)—signed 16-bit (Intel)</td>
</tr>
<tr>
<td>AT_STIME</td>
<td>Signed 32-bit signed integer representing time—signed time</td>
</tr>
<tr>
<td>AT_STRING</td>
<td>Unrestricted sequence of ASCII characters—string</td>
</tr>
<tr>
<td>AT_TIME</td>
<td>Unsigned 32-bit integer representing time—unsigned time</td>
</tr>
<tr>
<td>AT_TYPECNT</td>
<td>Type requiring two child definition: size of the type field, and type of data—counted-type:</td>
</tr>
<tr>
<td></td>
<td>For the DHCPv4 dhcp-user-class-id option (77), the repeating pattern is:</td>
</tr>
<tr>
<td></td>
<td>[ len (1 byte) ] [ data, of single type ]</td>
</tr>
<tr>
<td></td>
<td>For the DHCPv6 user-class option (15), the repeating pattern is:</td>
</tr>
<tr>
<td></td>
<td>[ len (2 byte) ] [ data, of single type ]</td>
</tr>
<tr>
<td>AT_VENDOR_CLASS</td>
<td>Vendor-class option (enterprise ID followed by opaque data; if DHCPv4, enterprise ID is followed by EID length)—vendor-class</td>
</tr>
<tr>
<td>AT_VENDOR_OPTS</td>
<td>Vendor-specific options data (enterprise ID followed by TLVs of vendor-specific data; if DHCPv4, enterprise ID is followed by EID length)—vendor-opts</td>
</tr>
<tr>
<td>AT_ZEROSIZE</td>
<td>32 bits of zero size (no longer used for PAD and END)</td>
</tr>
</tbody>
</table>

### Note

AT_TIME takes the value entered in seconds, by default. For example, if you enter 60, it is taken as 60 seconds and if you enter 60s/60m/2h, it is taken as 60 seconds/60 minutes/2 hours and displayed as 60s/60m/2h.
DHCP Extension Dictionary

This appendix describes the DHCP extension dictionary entries and the application program interface (API) to the extension dictionary. It describes the data items available in the request and response dictionaries, and the calls to use when accessing dictionaries from Tcl extensions and shared libraries.

Extension Dictionary Entries

A dictionary is a data structure that contains key-value pairs. There are two types of dictionaries: the attribute dictionaries that the request and response dictionaries use, and the environment dictionary. This section describes the request and response dictionaries; the environment dictionary entries are described in the “Tcl Environment Dictionary Methods” section on page C-29.

Decoded DHCP Packet Data Items

The decoded DHCPv4 packet data items represent the information in the DHCP packet, and are available in both the request and response dictionaries. These dictionaries provide access to considerably more internal server data structures than just the decoded request and decoded response.

All of the options followed by an asterisk (*) are multiple, which means that there can be more than one value associated with each option. In the DHCP/BOOTP packet, all of these data items appear in the same option. However, in the extension interface, these multiple data items are accessible through indexing.

You can access options that do not have names in Table C-3 on page C-2 as option–n, where n is the option number. All fields are read/write. Table C-1 describes the field values for the DHCPv4 packets; Table C-2 on page C-2 describes the field values for the DHCPv6 messages.

Table C-1 DHCPv4 and BOOTP Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaddr</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>ciaddr</td>
<td>IP address</td>
</tr>
<tr>
<td>file</td>
<td>string</td>
</tr>
<tr>
<td>flags</td>
<td>16-bit unsigned integer</td>
</tr>
<tr>
<td>giaddr</td>
<td>IP address</td>
</tr>
<tr>
<td>hlen</td>
<td>8-bit unsigned integer</td>
</tr>
</tbody>
</table>
### Table C-1  DHCPv4 and BOOTP Fields (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>hops</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>htype</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>op</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>secs</td>
<td>16-bit unsigned integer</td>
</tr>
<tr>
<td>siaddr</td>
<td>IP address</td>
</tr>
<tr>
<td>sname</td>
<td>string</td>
</tr>
<tr>
<td>xid</td>
<td>32-bit unsigned integer</td>
</tr>
<tr>
<td>yiaddr</td>
<td>IP address</td>
</tr>
</tbody>
</table>

### Table C-2  DHCPv6 Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>hop-count</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>link-address</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>msg-type</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>peer-address</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>xid</td>
<td>32-bit unsigned integer</td>
</tr>
</tbody>
</table>

### Table C-3  DHCPv4 and BOOTP Options

<table>
<thead>
<tr>
<th>Name (*=multivalue)</th>
<th>Number</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>all-subnets-local</td>
<td>27</td>
<td>byte-valued boolean</td>
</tr>
<tr>
<td>authentication</td>
<td>90</td>
<td>blob (sequence of bytes); 5 fields</td>
</tr>
<tr>
<td>auto-configure</td>
<td>116</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>arp-cache-timeout</td>
<td>35</td>
<td>unsigned time</td>
</tr>
<tr>
<td>bcmcs-servers-a*</td>
<td>89</td>
<td>IP address</td>
</tr>
<tr>
<td>bcmcs-servers-d*</td>
<td>88</td>
<td>DNS name</td>
</tr>
<tr>
<td>boot-file</td>
<td>67</td>
<td>string</td>
</tr>
<tr>
<td>boot-size</td>
<td>13</td>
<td>16-bit unsigned integer</td>
</tr>
<tr>
<td>broadcast-address</td>
<td>28</td>
<td>IP address</td>
</tr>
<tr>
<td>cablelabs-125</td>
<td>125</td>
<td>binary</td>
</tr>
<tr>
<td>(v-i-vendor-info ID: 4491)</td>
<td></td>
<td>suboption:</td>
</tr>
<tr>
<td>oro</td>
<td>1</td>
<td>Option request, 8-bit unsigned integer (8-bit unsigned integers)</td>
</tr>
<tr>
<td>tftp-servers</td>
<td>2</td>
<td>IP addresses of TFTP servers</td>
</tr>
<tr>
<td>erouter-container</td>
<td>3</td>
<td>Erouter container options (binary; TLV encoded options)</td>
</tr>
</tbody>
</table>
### Table C-3  DHCPv4 and BOOTP Options (continued)

<table>
<thead>
<tr>
<th>Name (*=multivalue)</th>
<th>Number</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>packetcable-mib-env</td>
<td>4</td>
<td>MIB environment indicator (8-bit enumeration)</td>
</tr>
<tr>
<td>modem-capabilities</td>
<td>5</td>
<td>Modem capabilities encoding (binary; TLV5 encoded data)</td>
</tr>
<tr>
<td>dhcpv6-servers</td>
<td>123</td>
<td>DHCPv6 server suboptions (binary)</td>
</tr>
<tr>
<td>ip-pref</td>
<td>124</td>
<td>IPv4 or IPv6 preference (8-bit enumeration)</td>
</tr>
<tr>
<td>cablelabs-client-configuration</td>
<td>122</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>[ccc-]primary-dhcp-server</td>
<td>1</td>
<td>IP address</td>
</tr>
<tr>
<td>[ccc-]secondary-dhcp-server</td>
<td>2</td>
<td>IP address</td>
</tr>
<tr>
<td>[ccc-]provisioning-server</td>
<td>3</td>
<td>blob (the first byte must be the type byte, with 0 for RFC 1035 encoding, and 1 for IP address encoding, for which the address must be in network order)</td>
</tr>
<tr>
<td>[ccc-]as-backoff-retry-blob</td>
<td>4</td>
<td>12-byte blob (3 unsigned 4-byte integers, which must be in network order); configures the Kerberos AS-REQ/AS-REP timeout, back-off, and retry mechanism</td>
</tr>
<tr>
<td>[ccc-]ap-backoff-retry-blob</td>
<td>5</td>
<td>12-byte blob (3 unsigned 4-byte integers, which must be in network order); configures the Kerberos AP-REQ/AP-REP timeout, back-off, and retry mechanism</td>
</tr>
<tr>
<td>[ccc-]kerberos.realm</td>
<td>6</td>
<td>variable-length blob (an RFC 1035 style name); a Kerberos realm name is required</td>
</tr>
<tr>
<td>[ccc-]use-tgt</td>
<td>7</td>
<td>1-byte unsigned integer boolean; indicates whether to use a Ticket Granting Ticket (TGT) when obtaining a service ticket for one of the application servers</td>
</tr>
<tr>
<td>[ccc-]provisioning-timer</td>
<td>8</td>
<td>1-byte unsigned integer; defines the maximum time allowed for the provisioning process to finish</td>
</tr>
<tr>
<td>[ccc-]ticket-control-mask</td>
<td>9</td>
<td>2-byte unsigned integer, in host order</td>
</tr>
<tr>
<td>[ccc-]kdc-addresses-blob</td>
<td>10</td>
<td>variable-length (multiple of 4) IP address, in network order</td>
</tr>
<tr>
<td>cisco-autoconfigure</td>
<td>251</td>
<td>bounded byte</td>
</tr>
<tr>
<td>cisco-client-last-transaction-time</td>
<td>163</td>
<td>unsigned 32-bit integer</td>
</tr>
<tr>
<td>cisco-client-requested-host-name</td>
<td>162</td>
<td>string</td>
</tr>
<tr>
<td>cisco-leased-ip</td>
<td>161</td>
<td>IP address</td>
</tr>
<tr>
<td>cisco-vpn-id</td>
<td>221</td>
<td>blob (structured)</td>
</tr>
<tr>
<td>classless-static-route</td>
<td>121</td>
<td>blob (structured)</td>
</tr>
<tr>
<td>client-fqdn</td>
<td>81</td>
<td>blob (sequence of bytes); 4 fields: flags, rcode-1, rcode-2, and domain-name</td>
</tr>
<tr>
<td>cookie-servers*</td>
<td>8</td>
<td>IP address</td>
</tr>
<tr>
<td>Name (*=multivalue)</td>
<td>Number</td>
<td>Value</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>default-ip-ttl</td>
<td>23</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>default-tcp-ttl</td>
<td>37</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>dhcp-class-identifier</td>
<td>60</td>
<td>string</td>
</tr>
<tr>
<td>dhcp-client-identifier</td>
<td>61</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>dhcp-lease-time</td>
<td>51</td>
<td>unsigned time</td>
</tr>
<tr>
<td>dhcp-max-message-size</td>
<td>57</td>
<td>16-bit unsigned integer</td>
</tr>
<tr>
<td>dhcp-message</td>
<td>56</td>
<td>string</td>
</tr>
<tr>
<td>dhcp-message-type</td>
<td>53</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>dhcp-option-overload</td>
<td>52</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>dhcp-parameter-request-list*</td>
<td>55</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>dhcp-parameter-request-list-blob*</td>
<td>55</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>dhcp-rebinding-time</td>
<td>59</td>
<td>unsigned time</td>
</tr>
<tr>
<td>dhcp-renewal-time</td>
<td>58</td>
<td>unsigned time</td>
</tr>
<tr>
<td>dhcp-requested-address</td>
<td>50</td>
<td>IP address</td>
</tr>
<tr>
<td>dhcp-server-identifier</td>
<td>54</td>
<td>IP address</td>
</tr>
<tr>
<td>dhcp-user-class-id</td>
<td>77</td>
<td>set of counted len byte arrays; 2 fields: typcnt-size and user-data</td>
</tr>
<tr>
<td>domain-name</td>
<td>15</td>
<td>string</td>
</tr>
<tr>
<td>domain-name-servers*</td>
<td>6</td>
<td>IP address</td>
</tr>
<tr>
<td>domain-search</td>
<td>119</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>extensions-path</td>
<td>18</td>
<td>string</td>
</tr>
<tr>
<td>finger-servers*</td>
<td>73</td>
<td>IP address</td>
</tr>
<tr>
<td>font-servers*</td>
<td>48</td>
<td>IP address</td>
</tr>
<tr>
<td>geo-conf</td>
<td>123</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>geoconf-civic</td>
<td>99</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>host-name</td>
<td>12</td>
<td>string</td>
</tr>
<tr>
<td>ieee802.3-encapsulation</td>
<td>36</td>
<td>byte-valued boolean</td>
</tr>
<tr>
<td>impress-servers*</td>
<td>10</td>
<td>IP address</td>
</tr>
<tr>
<td>initial-url</td>
<td>114</td>
<td>string</td>
</tr>
<tr>
<td>interface-mtu</td>
<td>26</td>
<td>16-bit unsigned integer</td>
</tr>
<tr>
<td>ip-forwarding</td>
<td>19</td>
<td>byte-valued boolean</td>
</tr>
<tr>
<td>irc-servers*</td>
<td>74</td>
<td>IP address</td>
</tr>
<tr>
<td>iSNS</td>
<td>83</td>
<td>blob (sequence of bytes); 7 fields</td>
</tr>
<tr>
<td>ldap-url</td>
<td>95</td>
<td>string</td>
</tr>
<tr>
<td>log-servers*</td>
<td>7</td>
<td>IP address</td>
</tr>
<tr>
<td>lost-server</td>
<td>137</td>
<td>DNS Name (see RFC 5223)</td>
</tr>
<tr>
<td>Name (*=multivalue)</td>
<td>Number</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>lpr-servers*</td>
<td>9</td>
<td>IP address</td>
</tr>
<tr>
<td>lq-associated-ip*</td>
<td>92</td>
<td>IP address</td>
</tr>
<tr>
<td>lq-client-last-transaction-time</td>
<td>91</td>
<td>unsigned time</td>
</tr>
<tr>
<td>mask-supplier</td>
<td>30</td>
<td>byte-valued boolean</td>
</tr>
<tr>
<td>max-dgram-reassembly</td>
<td>22</td>
<td>16-bit unsigned integer</td>
</tr>
<tr>
<td>mcns-security-server</td>
<td>128</td>
<td>IP address</td>
</tr>
<tr>
<td>merit-dump</td>
<td>14</td>
<td>string</td>
</tr>
<tr>
<td>mobile-ip-home-agents*</td>
<td>68</td>
<td>IP address</td>
</tr>
<tr>
<td>name-servers*</td>
<td>5</td>
<td>IP address</td>
</tr>
<tr>
<td>name-service-search*</td>
<td>117</td>
<td>16-bit unsigned integer</td>
</tr>
<tr>
<td>nds-servers*</td>
<td>85</td>
<td>IP address</td>
</tr>
<tr>
<td>nds-tree</td>
<td>86</td>
<td>string</td>
</tr>
<tr>
<td>nds-context</td>
<td>87</td>
<td>string</td>
</tr>
<tr>
<td>netbios-dd-servers*</td>
<td>45</td>
<td>IP address</td>
</tr>
<tr>
<td>netbios-name-servers*</td>
<td>44</td>
<td>IP address</td>
</tr>
<tr>
<td>netbios-node-type</td>
<td>46</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>netbios-scope</td>
<td>47</td>
<td>string</td>
</tr>
<tr>
<td>netinfo-parent-server-addr</td>
<td>112</td>
<td>IP address</td>
</tr>
<tr>
<td>netinfo-parent-server-tag</td>
<td>113</td>
<td>string</td>
</tr>
<tr>
<td>netwareip-domain</td>
<td>62</td>
<td>string</td>
</tr>
<tr>
<td>netwareip-information</td>
<td>63</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>nis+-servers*</td>
<td>65</td>
<td>IP address</td>
</tr>
<tr>
<td>nis+domain</td>
<td>64</td>
<td>string</td>
</tr>
<tr>
<td>nis-domain</td>
<td>40</td>
<td>string</td>
</tr>
<tr>
<td>nis-servers*</td>
<td>41</td>
<td>IP address</td>
</tr>
<tr>
<td>nntp-servers*</td>
<td>71</td>
<td>IP address</td>
</tr>
<tr>
<td>non-local-source-routing</td>
<td>20</td>
<td>byte-valued boolean</td>
</tr>
<tr>
<td>ntp-servers*</td>
<td>42</td>
<td>IP address</td>
</tr>
<tr>
<td>pana agent</td>
<td>136</td>
<td>IP address(es) (see RFC 5192)</td>
</tr>
<tr>
<td>path-mtu-aging-timeout</td>
<td>24</td>
<td>unsigned time</td>
</tr>
<tr>
<td>path-mtu-plateau-tables*</td>
<td>25</td>
<td>16-bit unsigned integer</td>
</tr>
<tr>
<td>perform-mask-discovery</td>
<td>29</td>
<td>byte-valued boolean</td>
</tr>
<tr>
<td>policy-filters*</td>
<td>21</td>
<td>IP address (there can be two policy filters, each one having its own IP address)</td>
</tr>
<tr>
<td>pop3-servers*</td>
<td>70</td>
<td>IP address</td>
</tr>
<tr>
<td>posix-timezone</td>
<td>100</td>
<td>string (see RFC 4833)</td>
</tr>
</tbody>
</table>
### Table C-3  DHCPv4 and BOOTP Options (continued)

<table>
<thead>
<tr>
<th>Name (*=multivalue)</th>
<th>Number</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pxe-client-arch</td>
<td>93</td>
<td>16-bit unsigned integer</td>
</tr>
<tr>
<td>pxe-client-machine-id</td>
<td>97</td>
<td>blob (sequence of bytes); 2 fields: type-flag and uuid</td>
</tr>
<tr>
<td>pxe-client-network-id</td>
<td>94</td>
<td>blob (sequence of bytes); 2 fields: type-flag and version</td>
</tr>
<tr>
<td>rapid-commit</td>
<td>80</td>
<td>null-length</td>
</tr>
<tr>
<td>relay-agent-info suboptions:</td>
<td>82 suboption</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>relay-agent-circuit-id-data</td>
<td>1</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>relay-agent-remote-id-data</td>
<td>2</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>relay-agent-device-class-data</td>
<td>4</td>
<td>4-byte unsigned integer</td>
</tr>
<tr>
<td>relay-agent-subnet-selection-data</td>
<td>5</td>
<td>IP address</td>
</tr>
<tr>
<td>subscriber-id</td>
<td>6</td>
<td>string identifying the network client or subscriber</td>
</tr>
<tr>
<td>radius-attributes</td>
<td>7</td>
<td>supported attributes are user, class, and framed-pool</td>
</tr>
<tr>
<td>authentication</td>
<td>8</td>
<td>binary</td>
</tr>
<tr>
<td>v-i-vendor-opts</td>
<td>9</td>
<td>vendor options</td>
</tr>
<tr>
<td>cisco-subnet-selection</td>
<td>150</td>
<td>IP address</td>
</tr>
<tr>
<td>cisco-vpn-id</td>
<td>151</td>
<td>binary</td>
</tr>
<tr>
<td>cisco-server-id-override</td>
<td>152</td>
<td>IP address</td>
</tr>
<tr>
<td>relay-agent-vpn-id-data</td>
<td>181</td>
<td>string</td>
</tr>
<tr>
<td>relay-agent-server-id-override-data</td>
<td>182</td>
<td>IP address</td>
</tr>
</tbody>
</table>

**Note**  The relay-agent-circuit-id, relay-agent-remote-id, and relay-agent-device-class suboptions, which returned the two bytes (suboption code and data length) preceding the suboption data, are deprecated, but still available.

<table>
<thead>
<tr>
<th>Name (*=multivalue)</th>
<th>Number</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource-location-servers*</td>
<td>11</td>
<td>IP address</td>
</tr>
<tr>
<td>root-path</td>
<td>17</td>
<td>string</td>
</tr>
<tr>
<td>router-discovery</td>
<td>31</td>
<td>byte-valued boolean</td>
</tr>
<tr>
<td>router-solicitation-address</td>
<td>32</td>
<td>IP address</td>
</tr>
<tr>
<td>routers*</td>
<td>3</td>
<td>IP address</td>
</tr>
<tr>
<td>sip-servers</td>
<td>120</td>
<td>blob (sequence of bytes); 2 fields: flag and sip-server-list</td>
</tr>
<tr>
<td>slp-directory-agent*</td>
<td>78</td>
<td>blob (sequence of bytes); 2 fields: mandatory and agent-ip-list</td>
</tr>
<tr>
<td>slp-service-scope*</td>
<td>79</td>
<td>blob (sequence of bytes); 2 fields: mandatory and slp-scope-list</td>
</tr>
<tr>
<td>smtp-servers*</td>
<td>69</td>
<td>IP address</td>
</tr>
<tr>
<td>static-routes*</td>
<td>33</td>
<td>IP address</td>
</tr>
</tbody>
</table>
### Table C-3 DHCPv4 and BOOTP Options (continued)

<table>
<thead>
<tr>
<th>Name (*=multivalue)</th>
<th>Number</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>streettalk-directory-assistance-servers*</td>
<td>76</td>
<td>IP address</td>
</tr>
<tr>
<td>streettalk-servers*</td>
<td>75</td>
<td>IP address</td>
</tr>
<tr>
<td>subnet-alloc</td>
<td>220</td>
<td>blob (sequence of bytes); 5 fields: flags, subnet-request, subnet-info, subnet-name, and subnet-suggested-lease-time</td>
</tr>
<tr>
<td>subnet-mask</td>
<td>1</td>
<td>IP address</td>
</tr>
<tr>
<td>subnet-selection</td>
<td>118</td>
<td>IP address</td>
</tr>
<tr>
<td>swap-server</td>
<td>16</td>
<td>IP address</td>
</tr>
<tr>
<td>tcp-keepalive-internal</td>
<td>38</td>
<td>unsigned time</td>
</tr>
<tr>
<td>tcp-keepalive-garbage</td>
<td>39</td>
<td>byte-valued boolean</td>
</tr>
<tr>
<td>tftp-server</td>
<td>66</td>
<td>string</td>
</tr>
<tr>
<td>time-offset</td>
<td>2</td>
<td>signed time</td>
</tr>
<tr>
<td>time-servers*</td>
<td>4</td>
<td>IP address</td>
</tr>
<tr>
<td>trailer-encapsulation</td>
<td>34</td>
<td>byte-valued boolean</td>
</tr>
<tr>
<td>tzdb-timezone</td>
<td>101</td>
<td>string (see RFC 4833)</td>
</tr>
<tr>
<td>user-auth</td>
<td>98</td>
<td>string</td>
</tr>
<tr>
<td>vendor-encapsulated-options</td>
<td>43</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>v-i-vendor-class</td>
<td>124</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>v-i-vendor-info</td>
<td>125</td>
<td>blob (sequence of bytes)</td>
</tr>
<tr>
<td>vpn-id</td>
<td>185</td>
<td>blob (structured); 2 fields: flag and vpn-id</td>
</tr>
<tr>
<td>www-servers*</td>
<td>72</td>
<td>IP address</td>
</tr>
<tr>
<td>x-display-managers*</td>
<td>49</td>
<td>IP address</td>
</tr>
</tbody>
</table>

Table C-4 lists the DHCPv6 options.

**Note**

Access to these options is available using the `putOption`, `getOption`, and `removeOption` methods only.

### Table C-4 DHCPv6 Options

<table>
<thead>
<tr>
<th>Name (*=multivalue)</th>
<th>Number</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth</td>
<td>11</td>
<td>binary; 5 fields: protocol, algorithm, replay-detection-method, replay-detection, and auth-info</td>
</tr>
<tr>
<td>bcmcs-server-a*</td>
<td>34</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>bcmcs-server-d*</td>
<td>33</td>
<td>DNS name</td>
</tr>
<tr>
<td>cablelabs-17 (vendor-opts ID: 4491)</td>
<td>17</td>
<td>vendor-opts; 20 suboptions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>suboption:</td>
</tr>
</tbody>
</table>
### Table C-4  DHCPv6 Options (continued)

<table>
<thead>
<tr>
<th>Name (*=multivalue)</th>
<th>Number</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>oro</td>
<td>1</td>
<td>16-bit unsigned integer</td>
</tr>
<tr>
<td>device-type</td>
<td>2</td>
<td>string</td>
</tr>
<tr>
<td>embedded-components-list</td>
<td>3</td>
<td>string</td>
</tr>
<tr>
<td>device-serial-number</td>
<td>4</td>
<td>string</td>
</tr>
<tr>
<td>hardware-version-number</td>
<td>5</td>
<td>string</td>
</tr>
<tr>
<td>software-version-number</td>
<td>6</td>
<td>string</td>
</tr>
<tr>
<td>boot-rom-version</td>
<td>7</td>
<td>string</td>
</tr>
<tr>
<td>vendor-oui</td>
<td>8</td>
<td>string</td>
</tr>
<tr>
<td>model-number</td>
<td>9</td>
<td>string</td>
</tr>
<tr>
<td>vendor-name</td>
<td>10</td>
<td>string</td>
</tr>
<tr>
<td>ecm-cfg-encaps</td>
<td>15</td>
<td>string</td>
</tr>
<tr>
<td>tftp-servers</td>
<td>32</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>config-file-name</td>
<td>33</td>
<td>string</td>
</tr>
<tr>
<td>syslog-servers</td>
<td>34</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>modem-capabilities</td>
<td>35</td>
<td>binary</td>
</tr>
<tr>
<td>device-id</td>
<td>36</td>
<td>binary</td>
</tr>
<tr>
<td>rfc868-servers</td>
<td>37</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>time-offset</td>
<td>38</td>
<td>unsigned time</td>
</tr>
<tr>
<td>ip-pref</td>
<td>39</td>
<td>8-bit unsigned integer</td>
</tr>
<tr>
<td>cmts-capabilities</td>
<td>1025</td>
<td>binary; 1 suboption: docsis-version</td>
</tr>
<tr>
<td>cm-mac-address</td>
<td>1026</td>
<td>binary</td>
</tr>
<tr>
<td>erouter-container</td>
<td>1027</td>
<td>binary</td>
</tr>
<tr>
<td>cablelabs-client-configuration</td>
<td>2170</td>
<td>IPv6 address; 2 suboptions (various data types)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>suboption:</td>
</tr>
<tr>
<td>primary-dhcp-server</td>
<td>1</td>
<td>IP address</td>
</tr>
<tr>
<td>secondary-dhcp-server</td>
<td>2</td>
<td>IP address</td>
</tr>
<tr>
<td>cablelabs-client-configuration-v6</td>
<td>2171</td>
<td>IPv6 address; 9 suboptions (various data types)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>suboption:</td>
</tr>
<tr>
<td>primary-dhcpv6-server-selector-id</td>
<td>1</td>
<td>binary</td>
</tr>
<tr>
<td>secondary-dhcpv6-server-selector-id</td>
<td>2</td>
<td>binary</td>
</tr>
<tr>
<td>provisioning-server</td>
<td>3</td>
<td>binary</td>
</tr>
<tr>
<td>as-backoff-retry</td>
<td>4</td>
<td>binary</td>
</tr>
<tr>
<td>ap-backoff-retry</td>
<td>5</td>
<td>binary</td>
</tr>
<tr>
<td>kerberos-realm</td>
<td>6</td>
<td>DNS name</td>
</tr>
</tbody>
</table>
### Table C-4  DHCPv6 Options (continued)

<table>
<thead>
<tr>
<th>Name (*=multivalue)</th>
<th>Number</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>use-tgt</td>
<td>7</td>
<td>unsigned 8-bit</td>
</tr>
<tr>
<td>provisioning-timer</td>
<td>8</td>
<td>unsigned 8-bit</td>
</tr>
<tr>
<td>ticket-control-mask</td>
<td>9</td>
<td>unsigned 16-bit</td>
</tr>
<tr>
<td>client-data</td>
<td>45</td>
<td>binary (options) (see RFC 5007)</td>
</tr>
<tr>
<td>client-fqdn</td>
<td>39</td>
<td>binary; 2 fields: flags and domain-name</td>
</tr>
<tr>
<td>client-identifier</td>
<td>1</td>
<td>binary</td>
</tr>
<tr>
<td>clt-time</td>
<td>46</td>
<td>32-bit unsigned time (see RFC 5007)</td>
</tr>
<tr>
<td>dns-servers*</td>
<td>23</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>domain-list*</td>
<td>24</td>
<td>DNS name</td>
</tr>
<tr>
<td>elapsed-time</td>
<td>8</td>
<td>unsigned 16-bit</td>
</tr>
<tr>
<td>ero</td>
<td>43</td>
<td>unsigned 16-bit (see RFC 4994)</td>
</tr>
<tr>
<td>geocfg-civic</td>
<td>36</td>
<td>binary</td>
</tr>
<tr>
<td>ia-na</td>
<td>3</td>
<td>binary; 3 fields: iaid, t1, and t2</td>
</tr>
<tr>
<td>ia-pd</td>
<td>25</td>
<td>binary; 3 fields: iaid, t1, and t2</td>
</tr>
<tr>
<td>ia-prefix</td>
<td>26</td>
<td>binary; 4 fields: preferred-lifetime, valid-lifetime, prefix-length, and prefix</td>
</tr>
<tr>
<td>ia-ta</td>
<td>4</td>
<td>binary; 1 suboption: iaid</td>
</tr>
<tr>
<td>iaaddr</td>
<td>5</td>
<td>binary; 3 fields: address, preferred-lifetime, and valid-lifetime</td>
</tr>
<tr>
<td>info-refresh-time</td>
<td>32</td>
<td>unsigned time</td>
</tr>
<tr>
<td>interface-id</td>
<td>18</td>
<td>binary</td>
</tr>
<tr>
<td>lost-server</td>
<td>51</td>
<td>DNS Name (see RFC 5223)</td>
</tr>
<tr>
<td>lq-client-links</td>
<td>48</td>
<td>IPv6 address(es) (see RFC 5007)</td>
</tr>
<tr>
<td>lq-query</td>
<td>44</td>
<td>binary structured (see RFC 5007)</td>
</tr>
<tr>
<td>lq-relay-data</td>
<td>47</td>
<td>binary (DHCPv6 message) (see RFC 5007)</td>
</tr>
<tr>
<td>new-posix-timezone</td>
<td>41</td>
<td>string (RFC 4833)</td>
</tr>
<tr>
<td>new-tzdb-timezone</td>
<td>42</td>
<td>string (RFC 4833)</td>
</tr>
<tr>
<td>nis-domain-name*</td>
<td>29</td>
<td>DNS name</td>
</tr>
<tr>
<td>nis-servers*</td>
<td>27</td>
<td>IP address</td>
</tr>
<tr>
<td>nisp-domain-name*</td>
<td>30</td>
<td>DNS name</td>
</tr>
<tr>
<td>nisp-servers*</td>
<td>28</td>
<td>IP address</td>
</tr>
<tr>
<td>oro*</td>
<td>6</td>
<td>unsigned 16-bit</td>
</tr>
<tr>
<td>pana agent</td>
<td>40</td>
<td>IPv6 address(es) (see RFC 5192)</td>
</tr>
<tr>
<td>preference</td>
<td>7</td>
<td>unsigned 8-bit</td>
</tr>
<tr>
<td>rapid-commit</td>
<td>14</td>
<td>zero size</td>
</tr>
<tr>
<td>reconfigure-accept</td>
<td>20</td>
<td>zero size</td>
</tr>
<tr>
<td>reconfigure-message</td>
<td>19</td>
<td>unsigned 8-bit</td>
</tr>
</tbody>
</table>
Appendix C  DHCP Extension Dictionary

Table C-4  DHCPv6 Options (continued)

<table>
<thead>
<tr>
<th>Name (*=multivalue)</th>
<th>Number</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>relay-agent-subscriber-id</td>
<td>38</td>
<td>binary</td>
</tr>
<tr>
<td>relay-message</td>
<td>9</td>
<td>binary</td>
</tr>
<tr>
<td>remote-id</td>
<td>37</td>
<td>binary; 2 fields: enterprise-id and remote-id</td>
</tr>
<tr>
<td>server-identifier</td>
<td>2</td>
<td>binary (AT_BLOB)</td>
</tr>
<tr>
<td>server-unicast</td>
<td>12</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>sip-servers-address*</td>
<td>22</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>sip-servers-name*</td>
<td>21</td>
<td>DNS name</td>
</tr>
<tr>
<td>sntp-servers*</td>
<td>31</td>
<td>IP address</td>
</tr>
<tr>
<td>status-code</td>
<td>13</td>
<td>binary; 2 fields: status-code and status-message</td>
</tr>
<tr>
<td>user-class*</td>
<td>15</td>
<td>counted-type; 2 fields: typecnt-size and user-data</td>
</tr>
<tr>
<td>vendor-class</td>
<td>16</td>
<td>vendor-class</td>
</tr>
<tr>
<td>vendor-opts</td>
<td>17</td>
<td>vendor-opts (see also cablelabs-17)</td>
</tr>
</tbody>
</table>

Table C-5  Request Dictionary Specific Data Items

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>active-leasequery-control</td>
<td>int (v4)</td>
</tr>
<tr>
<td>allow-bootp</td>
<td>int (v4)</td>
</tr>
<tr>
<td>allow-dhcp</td>
<td>int (v4)</td>
</tr>
<tr>
<td>allow-dynamic-bootp</td>
<td>int (v4)</td>
</tr>
<tr>
<td>bootp-reply-options</td>
<td>blob (v4)</td>
</tr>
<tr>
<td>client-class-name</td>
<td>string (v4, v6)</td>
</tr>
</tbody>
</table>

Request Dictionary

Table C-5 lists the data items that you can set in the request dictionary at any time. The DHCP server reads them at various times. Unless indicated otherwise, all operations are read/write.

**Table C-5  Request Dictionary Specific Data Items**

- **active-leasequery-control**: Controls the sending of a lease (such as only on specific state changes). Values are: 0—unspecified (the server determines whether to send the notification), 1—send (the server will send the notification), and 2—do not send (the server will not send the notification). The `active-leasequery-control` is initialized as 0, that is, unspecified.

- **allow-bootp**: If set to 1, allows BOOTP for any scope for this request. Read during scope selection and while checking for lease acceptability.

- **allow-dhcp**: If set to 1, allows DHCP for any scope for this request. Read during scope selection and while checking for lease acceptability.

- **allow-dynamic-bootp**: If set to 1, allows dynamic BOOTP for any scope for this request. Read during scope selection and while checking for lease acceptability.

- **bootp-reply-options**: Overrides any `v4-bootp-reply-options` in any policy; read when gathering data for the output packet. (There are no IPv6 bootp-reply-options.)

- **client-class-name**: Name of the client-class used to complete the client information (if any). Read-only.
### Table C-5  Request Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-class-policy</td>
<td>string (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Name of the policy that is associated with the client-class. If set, it must be with the name of a policy that was already configured in the server.</td>
</tr>
<tr>
<td>client-domain-name</td>
<td>string (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Domain name that the client wants to use. If it does not exist, in which case the DHCP server uses the domain name specified in the scope. Read when queuing the request for DNS update just prior to the update of stable storage. For DHCPv6, overrides the client-fqdn value and used for DNS updates.</td>
</tr>
<tr>
<td>client-host-name</td>
<td>string (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Hostname for the client in DNS; read when queuing in the request for a DNS update just before updating stable storage. Places the actual name in DNS when that operation finishes. For DHCPv6, overrides the client-fqdn value and used for DNS updates.</td>
</tr>
<tr>
<td>client-id</td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Client identification that the server uses to track the client. Can be the client-id sent with a request or internally generated from the MAC address. See client-id-created-from-mac-address. For DHCPv6, usually the client DUID.</td>
</tr>
<tr>
<td>client-id-created-from-mac-address</td>
<td>int (v4)</td>
</tr>
<tr>
<td></td>
<td>If set to 1, the client-id must be created for internal use from the client-supplied MAC address and should not be used in reporting.</td>
</tr>
<tr>
<td>client-ipaddress</td>
<td>IP address (v4)</td>
</tr>
<tr>
<td></td>
<td>IP address from which the client sent its packet. Note that it could be zero if the client does not yet have an IP address.</td>
</tr>
<tr>
<td>client-limitation-id</td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Limitation ID for the client.</td>
</tr>
<tr>
<td>client-lookup-id</td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Client lookup ID calculated by the client-lookup-id expression of the client-class.</td>
</tr>
<tr>
<td>client-mac-address</td>
<td>blob (v4)</td>
</tr>
<tr>
<td></td>
<td>MAC address stored in the client object associated with the request dictionary. Has the same format (and was created from) the mac-address.</td>
</tr>
<tr>
<td>client-os-type</td>
<td>int (v4)</td>
</tr>
<tr>
<td></td>
<td>Change the client entry of the request packet by setting this at the pre-client-lookup or post-client-lookup extension points. Can also be read at check-lease-acceptable, but cannot be set there. To set the value, you must first set the os-type in the post-packet-decode request dictionary.</td>
</tr>
<tr>
<td>client-packet</td>
<td>blob (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>The client portion of the received packet. For DHCPv4, this is the complete packet. For DHCPv6, this is the client message. (See packet to obtain the full packet.)</td>
</tr>
<tr>
<td>client-policy</td>
<td>string (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Name of the policy that is associated with the client entry. If set, must be the name of a preconfigured policy in the DHCP server.</td>
</tr>
<tr>
<td>client-port</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Port from which the client sent its request.</td>
</tr>
</tbody>
</table>
### DHCP Extension Dictionary

#### Extension Dictionary Entries

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>client-requested-host-name</strong></td>
<td>string (v4)</td>
</tr>
<tr>
<td>Hostname that the client requested be used for the DNS update. The DHCP server saves this information so that a change can be detected.</td>
<td></td>
</tr>
<tr>
<td><strong>client-unicast</strong></td>
<td>boolean (v6, read-only)</td>
</tr>
<tr>
<td>True if the received packet was unicast by the client to the server.</td>
<td></td>
</tr>
<tr>
<td><strong>client-wants-nulls-in-strings</strong></td>
<td>int (v4)</td>
</tr>
<tr>
<td>Determines whether the DHCP server returns strings to the client terminated with a null. If set to 1, the server terminates strings with a null. If set to 0, it does not terminate strings with a null. Set before <strong>post-packet-decode</strong> and read when encoding the response packet after <strong>pre-packet-encode</strong>.</td>
<td></td>
</tr>
<tr>
<td><strong>derived-vpn-id</strong></td>
<td>int (v4, v6, read-only)</td>
</tr>
<tr>
<td>VPN identifier. See <strong>vpn-name</strong> for details.</td>
<td></td>
</tr>
<tr>
<td><strong>destination-ipaddress</strong></td>
<td>IP address (v6, read-only)</td>
</tr>
<tr>
<td>Destination IPv6 address of the packet.</td>
<td></td>
</tr>
<tr>
<td><strong>dhcp-reply-options</strong></td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td>Overrides any v4-reply-options or v6-reply-options specified in a policy; read when gathering data for the output packet.</td>
<td></td>
</tr>
<tr>
<td><strong>dump-packet</strong></td>
<td>int (v4, v6, write-only)</td>
</tr>
<tr>
<td>When set to 1, Cisco Prime IP Express dumps the current decoded DHCP/BOOTP packet to the log file. An extension can put the value 1 into this data item at multiple points in its execution. This might be useful when debugging extensions.</td>
<td></td>
</tr>
<tr>
<td><strong>failover-role</strong></td>
<td>int (v4, v6, read only)</td>
</tr>
<tr>
<td>Determines the failover server role. The failover server role can be one of three values:</td>
<td></td>
</tr>
<tr>
<td>- <strong>None</strong>—Failover is not configured.</td>
<td></td>
</tr>
<tr>
<td>- <strong>Main/Backup</strong>—Failover is configured and the role of the failover server</td>
<td></td>
</tr>
<tr>
<td><strong>failover-state</strong></td>
<td>int (v4, v6, read only)</td>
</tr>
<tr>
<td>Determines failover server state. The failover state can be normal, partner-down, communications-interrupted, recover, potential-conflict, recover-done, startup, shutdown, or paused. If failover is not configured the value is none.</td>
<td></td>
</tr>
<tr>
<td><strong>import-packet</strong></td>
<td>int (v4)</td>
</tr>
<tr>
<td>Determines whether the server treats the packet as if it came from an import client. If set to 1, the server treats the client as an import client and performs all DNS operations on it before sending an ACK. Read when checking the server import mode (right after <strong>post-packet-decode</strong>), getting ready for DNS processing, and when setting the reply address.</td>
<td></td>
</tr>
<tr>
<td><strong>limitation-count</strong></td>
<td>int (v4)</td>
</tr>
<tr>
<td>Number of simultaneous users allowed with the same <strong>limitation-id</strong>.</td>
<td></td>
</tr>
<tr>
<td><strong>limitation-id</strong></td>
<td>blob (v4)</td>
</tr>
<tr>
<td>Calculated by the <strong>limitation-id</strong> expression (if any) for the client-class in which this request falls.</td>
<td></td>
</tr>
<tr>
<td><strong>limitation-id-null</strong></td>
<td>int (v4)</td>
</tr>
<tr>
<td>Set to 1(TRUE) if the <strong>limitation-id</strong> is null, 0 (FALSE) if another value.</td>
<td></td>
</tr>
</tbody>
</table>
Table C-5  Request Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>log-client-criteria-processing</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, logs the criteria processing for the client for this request. Read when trying to acquire a new lease for a client that does not have one, and when checking for lease acceptability.</td>
<td></td>
</tr>
<tr>
<td>log-client-detail</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, logs the client-class processing for this request. Read at the end of client-class processing, after <strong>post-client-lookup</strong>.</td>
<td></td>
</tr>
<tr>
<td>log-dns-update-detail</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, logs DNS update details for this request.</td>
<td></td>
</tr>
<tr>
<td>log-dropped-bootp-packets</td>
<td>int (v4)</td>
</tr>
<tr>
<td>If set to a 1, logs dropped BOOTP packets for this request.</td>
<td></td>
</tr>
<tr>
<td>log-dropped-dhcp-packets</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, logs dropped DHCP packets for this request.</td>
<td></td>
</tr>
<tr>
<td>log-dropped-waiting-packets</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, logs dropped waiting packets for this request.</td>
<td></td>
</tr>
<tr>
<td>log-failover-detail</td>
<td>int (v4)</td>
</tr>
<tr>
<td>If set to a 1, logs a more detailed level of failover activity, such as all failover state changes.</td>
<td></td>
</tr>
<tr>
<td>log-incoming-packet-detail</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, checks whether detailed incoming packet tracing occurred for this request, so that you do not need to put a separate trace on it. Read before packet decoding and the first extension point.</td>
<td></td>
</tr>
<tr>
<td>log-incoming-packets</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, logs the incoming packets for this request. Read after <strong>post-decode-packet</strong>.</td>
<td></td>
</tr>
<tr>
<td>log-ldap-create-detail</td>
<td>int (v4)</td>
</tr>
<tr>
<td>If set to a 1, logs messages whenever the DHCP server initiates a lease state entry creation to, receives a response from, or retrieves a result or error message from an LDAP server.</td>
<td></td>
</tr>
<tr>
<td>log-ldap-query-detail</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, logs messages whenever the DHCP server initiates a query to, receives a response from, or retrieves a query result or an error message from an LDAP server.</td>
<td></td>
</tr>
<tr>
<td>log-ldap-update-detail</td>
<td>int (v4)</td>
</tr>
<tr>
<td>If set to a 1, logs messages whenever the DHCP server initiates an update lease state to, receives a response from, or retrieves a result or error message from an LDAP server.</td>
<td></td>
</tr>
<tr>
<td>log-leasequery</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, logs messages when leasequery packets are processed without internal errors and result in an ACK or a NAK.</td>
<td></td>
</tr>
<tr>
<td>log-missing-options</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, logs missing options (those a client requests but the DHCP server cannot return). Read while gathering data for the response.</td>
<td></td>
</tr>
<tr>
<td>log-outgoing-packet-detail</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If set to a 1, logs a detailed dump of the outgoing packet for this request. Read after <strong>pre-packet-encode</strong> and just before sending the packet to the DHCP client.</td>
<td></td>
</tr>
</tbody>
</table>
### Table C-5 Request Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>log-success-messages</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>If set to a 1, logs the success messages.</td>
</tr>
<tr>
<td>log-unknown-criteria</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>If set to a 1, logs any unknown criteria specified in</td>
</tr>
<tr>
<td></td>
<td>the client inclusion or exclusion criteria for this</td>
</tr>
<tr>
<td></td>
<td>request. Read when acquiring a new client lease or</td>
</tr>
<tr>
<td></td>
<td>checking lease acceptability for an existing client.</td>
</tr>
<tr>
<td>log-v6-lease-detail</td>
<td>int (v6)</td>
</tr>
<tr>
<td></td>
<td>If set to 1, logs individual messages about DHCPv6</td>
</tr>
<tr>
<td></td>
<td>leasing activity.</td>
</tr>
<tr>
<td>mac-address</td>
<td>blob (v4)</td>
</tr>
<tr>
<td></td>
<td>MAC address that came in the client packet. The first</td>
</tr>
<tr>
<td></td>
<td>byte is the hardware type, the second is the</td>
</tr>
<tr>
<td></td>
<td>hardware length, and the remaining (up to 16) is the</td>
</tr>
<tr>
<td></td>
<td>information from the chaddr read just after</td>
</tr>
<tr>
<td></td>
<td>post-packet-decode.</td>
</tr>
<tr>
<td></td>
<td>This is a useful aggregation of the htype, hlen, and</td>
</tr>
<tr>
<td></td>
<td>chaddr fields of the DHCP packet. When read it is</td>
</tr>
<tr>
<td></td>
<td>constructed from these fields; when written it is</td>
</tr>
<tr>
<td></td>
<td>placed into these fields.</td>
</tr>
<tr>
<td>max-client-lookups</td>
<td>integer (v4, v6))</td>
</tr>
<tr>
<td></td>
<td>Maximum number of client database lookups allowed.</td>
</tr>
<tr>
<td></td>
<td>Usually a small integer such as 2; the preset value</td>
</tr>
<tr>
<td></td>
<td>is 1.</td>
</tr>
<tr>
<td>override-client-id</td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Blob used for the current client-id value. Replaces</td>
</tr>
<tr>
<td></td>
<td>any client-id from the incoming packet (although both</td>
</tr>
<tr>
<td></td>
<td>values are kept in the lease state database).</td>
</tr>
<tr>
<td>override-client-id-data-type</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Returns the data type of the override-client-id, either</td>
</tr>
<tr>
<td></td>
<td>“nstr” for string or “blob” for blob.</td>
</tr>
<tr>
<td>override-client-id-string</td>
<td>string (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Current client-id value in string format that replaces</td>
</tr>
<tr>
<td></td>
<td>any client-id from the incoming packet (although both</td>
</tr>
<tr>
<td></td>
<td>values are kept in the lease state database). For a</td>
</tr>
<tr>
<td></td>
<td>get, if the override-client-id is not a string, the</td>
</tr>
<tr>
<td></td>
<td>binary data is formatted as blob data, which is then</td>
</tr>
<tr>
<td></td>
<td>returned as the “string.”</td>
</tr>
<tr>
<td>packet</td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>The received packet. For DHCPv4, this is the same as</td>
</tr>
<tr>
<td></td>
<td>client-packet. For DHCPv6, this is the full packet</td>
</tr>
<tr>
<td></td>
<td>if relayed or the same as client-packet if not</td>
</tr>
<tr>
<td></td>
<td>relayed. It should only be written from the</td>
</tr>
<tr>
<td></td>
<td>pre-packet-decode extension point; the server then</td>
</tr>
<tr>
<td></td>
<td>decodes this new packet instead of the packet</td>
</tr>
<tr>
<td></td>
<td>received from the client.</td>
</tr>
<tr>
<td>ping-clients</td>
<td>int (v4)</td>
</tr>
<tr>
<td></td>
<td>If set to a 1, performs a ping before offering a lease</td>
</tr>
<tr>
<td></td>
<td>for this request. Read just before determining if a</td>
</tr>
<tr>
<td></td>
<td>lease is acceptable for a client.</td>
</tr>
<tr>
<td>relay-agent-circuit-id</td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Contents of the circuit-id suboption of option 82.</td>
</tr>
<tr>
<td>relay-agent-circuit-id-data</td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Contents of just the data part of the circuit-id</td>
</tr>
<tr>
<td></td>
<td>suboption of option 82.</td>
</tr>
<tr>
<td>relay-agent-device-class-data</td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Contents of the device-class suboption of option 82.</td>
</tr>
<tr>
<td>relay-agent-radius-attributes</td>
<td>blob (v4)</td>
</tr>
</tbody>
</table>
### Table C-5 Request Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents of the radius suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>relay-agent-radius-class</td>
<td>string (v4)</td>
</tr>
<tr>
<td>Encapsulated class attribute of the radius suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>relay-agent-radius-pool-name</td>
<td>string (v4)</td>
</tr>
<tr>
<td>Encapsulated framed-pool attribute of the radius suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>relay-agent-radius-user</td>
<td>string (v4)</td>
</tr>
<tr>
<td>Encapsulated user attribute of the radius suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>relay-agent-remote-id</td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td>Contents of the remote-id suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>relay-agent-remote-id</td>
<td>blob (v4, v6)</td>
</tr>
<tr>
<td>Contents of just the data part of the remote-id suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>relay-agent-server-id-override-data</td>
<td>IPv6 address (v4, v6)</td>
</tr>
<tr>
<td>Contents of the server-id suboption of option 82. If the IANA suboption 182 is in the packet, that value appears; otherwise, the Cisco suboption 152 value appears.</td>
<td></td>
</tr>
<tr>
<td>relay-agent-subscriber-id</td>
<td>string (v4)</td>
</tr>
<tr>
<td>Contents of the subscriber-id suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>relay-count</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>Number of DHCPv6 relay hops.</td>
<td></td>
</tr>
<tr>
<td>reply-options</td>
<td>blob</td>
</tr>
<tr>
<td>Overrides any DHCPv4 reply options specified in any policy. Read when gathering data for the output packet.</td>
<td></td>
</tr>
<tr>
<td>reply-to-client-address</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>For v4, if set to 1, the server sends the response packet to the client-ipaddress and the client-port. For v6, if set to 1, the server sends the response packet back to the address and port of the sender (client or relay agent). If 0, the server sends the response using the RFC mandated algorithm.</td>
<td></td>
</tr>
<tr>
<td>reserved-addresses</td>
<td>IP address (v4, read/write)</td>
</tr>
<tr>
<td>List of addresses reserved for the client. The first available address to match a usable Scope (which must have restrict-to-reservations enabled) will be assigned to the client.</td>
<td></td>
</tr>
<tr>
<td>reserved-ip6addresses</td>
<td>IP address (v6, read/write)</td>
</tr>
<tr>
<td>List of addresses reserved for the client. All available addresses to match a usable Prefix (which must have restrict-to-reservations enabled) will be assigned to the client.</td>
<td></td>
</tr>
<tr>
<td>reserved-prefixes</td>
<td>IP address (v6, read/write)</td>
</tr>
<tr>
<td>List of prefixes reserved for the client. All available prefixes to match a usable Prefix (which must have restrict-to-reservations enabled) will be assigned to the client.</td>
<td></td>
</tr>
<tr>
<td>selection-criteria</td>
<td>string (v4, v6)</td>
</tr>
<tr>
<td>Comma-separated string that contains the scope selection criteria.</td>
<td></td>
</tr>
<tr>
<td>selection-criteria-excluded</td>
<td>string (v4, v6)</td>
</tr>
<tr>
<td>Comma-separated string that contains the scope exclusion criteria.</td>
<td></td>
</tr>
</tbody>
</table>
Table C-5  Request Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>send-ack-first</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>If set to a 1, updates DNS after the ACK for DHCP requests. Read just before initiating the DNS operation.</td>
</tr>
<tr>
<td>source-ipaddress</td>
<td>IPv6 address (v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>IPv6 source address of the packet.</td>
</tr>
<tr>
<td>trace-id</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>ID used by the system to trace the packet.</td>
</tr>
<tr>
<td>transaction-time</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Time, in seconds since 1970, that the input packet was decoded.</td>
</tr>
<tr>
<td>update-dns</td>
<td>string (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>Requests partial, full, or no dynamic DNS updates on a per-request packet basis. Input and output values are: 1=update-all, 2=update-fwd-only, 3=update-rev-only, and 0=update-none.</td>
</tr>
<tr>
<td>update-dns-for-bootp</td>
<td>int (v4)</td>
</tr>
<tr>
<td></td>
<td>If set to a 1, updates DNS for BOOTP requests for this request. Read just before initializing the DNS operation for BOOTP.</td>
</tr>
<tr>
<td>verbose-logging</td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td></td>
<td>If set to a 1, logs verbose messages for this request. Read at various times during processing.</td>
</tr>
<tr>
<td>vpn-description</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Description for the VPN. See vpn-name for details.</td>
</tr>
<tr>
<td>vpn-name</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Name of the VPN. The request dictionary does not have valid values for these items at post-packet-decode, but does at all other extension points, because the VPN has not yet been determined. This is so that a script can change the derived-vpn-id option or suboption at post-packet-decode and thereby affect the VPN used for a lease.</td>
</tr>
<tr>
<td>vpn-vpn-id</td>
<td>blob, typically 7 bytes (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Virtual private network identifier. See vpn-name for details.</td>
</tr>
<tr>
<td>vpn-vrf-name</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Virtual routing and forwarding table identifier for the VPN. See vpn-name for details.</td>
</tr>
</tbody>
</table>

Response Dictionary

Table C-6 lists the data items you can set in the response dictionary at any time. The DHCP server reads them at various times. Unless indicated otherwise, the operation is read/write.
Table C-6  Response Dictionary Specific Data Items

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>active-leasequery-control</td>
<td>int (v4)</td>
</tr>
<tr>
<td>Controls the sending of a lease (such as only on specific state changes). Values are: 0—unspecified (the server determines whether to send the notification), 1—send (the server will send the notification), and 2—do not send (the server will not send the notification). The active-leasequery-control is initialized as 0, that is, unspecified.</td>
<td></td>
</tr>
<tr>
<td>client-active-lease-count</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>Number of active leases on the DHCPv6 client.</td>
<td></td>
</tr>
<tr>
<td>client-creation-time</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>Creation time of the IPv6 client.</td>
<td></td>
</tr>
<tr>
<td>client-domain-name</td>
<td>string (v4, read-only)</td>
</tr>
<tr>
<td>From the client information in the lease, the domain name that the client wants to use. It might not exist, in which case the DHCP server uses the domain name specified in the scope. Read when queuing the request for DNS update just prior to the update of stable storage.</td>
<td></td>
</tr>
<tr>
<td>client-expiration-time</td>
<td>int (v4, v6, read-only)</td>
</tr>
<tr>
<td>The highest lease expiration time given to the client by this server (in seconds, since 1970).</td>
<td></td>
</tr>
<tr>
<td>client-host-name</td>
<td>string (v4, read-only)</td>
</tr>
<tr>
<td>From the client information in the lease, the hostname that the DHCP server puts into DNS. Read when queueing the request for a DNS update just before updating stable storage.</td>
<td></td>
</tr>
<tr>
<td>client-id</td>
<td>blob (v4, v6, read-only)</td>
</tr>
<tr>
<td>From the client information in the lease, the client identification that the server used to keep track of the client. This might be the client-id sent with a request or internally generated from the MAC address. For DHCPv6, usually the client DUID.</td>
<td></td>
</tr>
<tr>
<td>client-id-created-from-mac-address</td>
<td>int (v4, read-only)</td>
</tr>
<tr>
<td>From the client information in the lease. If set to 1, the client-id must be created from the MAC address and should not be used in reporting.</td>
<td></td>
</tr>
<tr>
<td>client-last-transaction-time</td>
<td>int (v4, v6, read-only)</td>
</tr>
<tr>
<td>Time, in seconds, since 1970, that the DHCP server last heard from this client.</td>
<td></td>
</tr>
<tr>
<td>client-limitation-id</td>
<td>blob (v4, read-only)</td>
</tr>
<tr>
<td>Limitation identifier of the client associated with the current lease.</td>
<td></td>
</tr>
<tr>
<td>client-mac-address</td>
<td>blob (v4, read-only)</td>
</tr>
<tr>
<td>From the client information in the lease, the MAC address stored in the client object associated with the request dictionary. Has the same format as (and was created from) the mac-address.</td>
<td></td>
</tr>
<tr>
<td>client-os-type</td>
<td>int (v4)</td>
</tr>
<tr>
<td>Change the client entry of the request packet by setting this at the pre-client-lookup or post-client-lookup extension points. Can also be read at check-lease-acceptable, but cannot be set there. To set the value, you must first set the os-type in the post-packet-decode request dictionary.</td>
<td></td>
</tr>
<tr>
<td>client-override-client-id</td>
<td>blob (v4, v6, read-only)</td>
</tr>
<tr>
<td>Blob used for the current client-id value. Replaces any client-id from the incoming packet (although both values are kept in the lease state database).</td>
<td></td>
</tr>
</tbody>
</table>
### Table C-6  Response Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-override-client-id-data-type</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td>Returns the data type of the client-override-client-id, either str for string or blob for blob.</td>
<td></td>
</tr>
<tr>
<td>client-override-client-id-string</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td>Current client-id value in string format that replaces any client-id from the incoming packet (although both values are kept in the lease state database). For a get, if the client-override-client-id is not a string, the binary data is formatted as blob data, which is then returned as the “string.”</td>
<td></td>
</tr>
<tr>
<td>client-packet</td>
<td>blob (v4, v6, read-only)</td>
</tr>
<tr>
<td>The client portion of the response packet. For DHCPv4, this is the complete packet. For DHCPv6, this is the client message. (See packet to obtain the full packet.) Only available from the post-packet-encode extension point.</td>
<td></td>
</tr>
<tr>
<td>client-reconfigure-key</td>
<td>string (v6)</td>
</tr>
<tr>
<td>Returns the client-reconfigure-key attribute value of the DHCPv6 lease.</td>
<td></td>
</tr>
<tr>
<td>client-reconfigure-key-generation-time</td>
<td>string (v6)</td>
</tr>
<tr>
<td>Returns the client-reconfigure-key-generation-time attribute value of the DHCPv6 lease.</td>
<td></td>
</tr>
<tr>
<td>client-relay-address</td>
<td>IPv6 address (v6, read-only)</td>
</tr>
<tr>
<td>Source IPv6 address for the (last) relay.</td>
<td></td>
</tr>
<tr>
<td>client-relay-message</td>
<td>string (v6, read-only)</td>
</tr>
<tr>
<td>Last relayed DHCPv6 message, excluding the client message.</td>
<td></td>
</tr>
<tr>
<td>client-requested-host-name</td>
<td>string (v4)</td>
</tr>
<tr>
<td>From the client information in the lease, the hostname that the client requested for the DNS update.</td>
<td></td>
</tr>
<tr>
<td>client-user-defined-data</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td>Returns the value previously or currently associated with the client, as derived from the user-defined-data environment dictionary data item. It returns the previously associated value if requested in a check-lease-acceptable or lease-state-change extension point. It returns the current value if requested in a pre-packet-encode or post-send-packet extension point.</td>
<td></td>
</tr>
<tr>
<td>client-vendor-class</td>
<td>string (v4, v6)</td>
</tr>
<tr>
<td>Returns the client-vendor-class attribute value of the DHCPv4 or DHCPv6 lease.</td>
<td></td>
</tr>
<tr>
<td>client-vendor-info</td>
<td>string (v4, v6)</td>
</tr>
<tr>
<td>Returns the client-vendor-info attribute value of the DHCPv4 or DHCPv6 lease.</td>
<td></td>
</tr>
<tr>
<td>client-write-sequence</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>Write sequence of the client IPv6 request.</td>
<td></td>
</tr>
<tr>
<td>client-write-time</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>Time of the client IPv6 write request.</td>
<td></td>
</tr>
<tr>
<td>derived-vpn-id</td>
<td>int (v4, v6, read-only)</td>
</tr>
<tr>
<td>VPN identifier.</td>
<td></td>
</tr>
<tr>
<td>domain-name-changed</td>
<td>int (v4)</td>
</tr>
<tr>
<td>If set to 1, the domain name in the current packet differs from the domain name used in the DNS update. Read after check-lease-acceptable and before pre-packet-encode.</td>
<td></td>
</tr>
</tbody>
</table>
### Table C-6  Response Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dump-packet</td>
<td>int (v4, v6, write-only)</td>
</tr>
</tbody>
</table>

When set to 1, Cisco Prime IP Express dumps the current decoded DHCP/BOOTP packet to the log file. An extension can put the value 1 into this data item at multiple points in its execution. This might be useful when debugging extensions.

| failover-role                     | int (v4, v6, read-only)                 |

Determines the failover server role. The failover server role can be one of three values:

- **None**—Failover is not configured.
- **Main/Backup**—Failover is configured and the role of the failover server

| failover-state                    | int (v4, v6, read-only)                 |

Determines failover server state. The failover state can be normal, partner-down, communications-interrupted, recover, potential-conflict, recover-done, startup, shutdown, or paused. If failover is not configured the value is none.

| host-name-changed                 | int (v4)                                |

If set to 1, the hostname in the current packet differs from that used in the DNS update. Read after **check-lease-acceptable** and before **pre-packet-encode**.

| host-name-in-dns                  | int (v4, v6)                            |

If set to 1, the hostname is in DNS. Read after **check-lease-acceptable** and before **pre-packet-encode**. Written after the hostname goes into DNS.

| lease-binding-iaid                | int (v6, read-only)                     |

IPv6 lease binding IAID.

| lease-binding-rebinding-time      | int (v6, read-only)                     |

IPv6 lease binding rebinding time.

| lease-binding-renewal-time        | int (v6, read-only)                     |

IPv6 lease binding renewal time.

| lease-binding-type                | string (v6, read-only)                 |

IPv6 lease binding type: "IA_NA", "IA_TA", or "IA_PD".

| lease-client-reserved             | int (v4, v6, read-only)                 |

Returns 1 if the lease is client reserved and 0 if not.

| lease-creation-time               | string (v6, read-only)                 |

IPv6 lease creation time.

| lease-deactivated                 | int (v4, v6, read-only)                 |

If set to 1, reports that the lease is deactivated.

| lease-dns-forward-backup-server-address | IP address (v4, v6, read-only) |

Address of the backup DNS server that receives DNS updates for the DHCPv4 and DHCPv6 lease, if the server specified in **lease-dns-forward-server-address** is down.

| lease-dns-forward-server-address  | IP address (v4, v6, read-only)         |

Address of the DNS server that receives dynamic DNS updates for the DHCPv4 and DHCPv6 lease.

| lease-dns-forward-update         | string (v4, v6, read-only)             |


### Appendix C  DHCP Extension Dictionary

<table>
<thead>
<tr>
<th>Name of the update configuration that determines the forward zones to be included in DNS updates for the DHCPv4 and DHCPv6 lease. Returns TRUE if update-all or update-fwd-only is set.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lease-dns-forward-zone-name</strong></td>
</tr>
<tr>
<td>Name of an optional forward zone for DNS updates.</td>
</tr>
<tr>
<td><strong>lease-dns-forward-backup-server-address</strong></td>
</tr>
<tr>
<td>Address of the backup DNS server that receives DNS updates for a DHCPv4 and DHCPv6 lease, if the server specified in lease-dns-forward-server-address is down.</td>
</tr>
<tr>
<td><strong>lease-dns-forward-host-bytes</strong></td>
</tr>
<tr>
<td>The number of bytes in a lease IP address to use for a reverse zone.</td>
</tr>
<tr>
<td><strong>lease-dns-forward-prefix-length</strong></td>
</tr>
<tr>
<td>Prefix length of the reverse zone for ip6.arpa updates.</td>
</tr>
<tr>
<td><strong>lease-dns-forward-server-address</strong></td>
</tr>
<tr>
<td>Address of the DNS server address that receives dynamic DNS updates for the DHCPv4 and DHCPv6 lease.</td>
</tr>
<tr>
<td><strong>lease-dns-forward-update</strong></td>
</tr>
<tr>
<td>Name of the update configuration that determines which reverse zones to include in a DNS update for the DHCPv4 and DHCPv6 lease. Returns TRUE if update-all or update-fwd-only is set.</td>
</tr>
<tr>
<td><strong>lease-dns-forward-zone-name</strong></td>
</tr>
<tr>
<td>DNS reverse (in-addr.arpa and ip6.arpa) zone that is updated with PTR records.</td>
</tr>
<tr>
<td><strong>lease-fqdn</strong></td>
</tr>
<tr>
<td>Fully qualified domain name assigned to the DHCPv6 lease by the server (and possibly successfully entered into DNS).</td>
</tr>
<tr>
<td>The lease-fqdn may be the name that is expected to be added to DNS for the lease or the actual name added. If host-name-in-dns is equal to true, the actual lease-fqdn is in DNS.</td>
</tr>
<tr>
<td><strong>lease-requested-fqdn</strong></td>
</tr>
<tr>
<td>Partial or fully qualified domain name most recently requested by the client for the DHCPv6 lease.</td>
</tr>
<tr>
<td><strong>lease-giaddr</strong></td>
</tr>
<tr>
<td>Lease giaddr.</td>
</tr>
<tr>
<td><strong>lease-ipaddress</strong></td>
</tr>
<tr>
<td>For DHCPv4, the address of the lease associated with the client. For DHCPv6, the IPv6 address or IPv6 prefix (address and prefix-length) of the lease for the current context (See setObject method).</td>
</tr>
<tr>
<td><strong>lease-preferred-lifetime</strong></td>
</tr>
<tr>
<td>Preferred lifetime of the IPv6 lease.</td>
</tr>
<tr>
<td><strong>lease-prefix-name</strong></td>
</tr>
<tr>
<td>Prefix name of the IPv6 lease.</td>
</tr>
<tr>
<td><strong>lease-relay-agent-info</strong></td>
</tr>
<tr>
<td>Entire contents of option 82.</td>
</tr>
</tbody>
</table>
### Table C-6  Response Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>lease-relay-agent-circuit-id</td>
<td>blob (v4)</td>
</tr>
<tr>
<td>Accesses and manipulates the relay agent circuit ID as stored with the lease of a response. Requires the suboption number 1 as the first byte. Deprecated in favor of the lease-relay-agent-circuit-id-data item.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-circuit-id-data</td>
<td>blob (v4, use instead of deprecated lease-relay-agent-circuit-id)</td>
</tr>
<tr>
<td>Accesses and manipulates the relay-agent-circuit-id-data as stored with the lease of a response.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-device-class-data</td>
<td>blob (v4)</td>
</tr>
<tr>
<td>Contents of the device-class suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-radius-attributes</td>
<td>blob (v4)</td>
</tr>
<tr>
<td>Contents of the radius suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-radius-class</td>
<td>string (v4)</td>
</tr>
<tr>
<td>Encapsulated class attribute of the radius suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-radius-pool-name</td>
<td>string (v4)</td>
</tr>
<tr>
<td>Encapsulated framed-pool attribute of the radius suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-radius-user</td>
<td>string (v4)</td>
</tr>
<tr>
<td>Encapsulated user attribute of the radius suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-remote-id</td>
<td>blob (v4)</td>
</tr>
<tr>
<td>Accesses and manipulates the relay-agent-remote-id data as stored with the lease of a response. Requires suboption number 2 as the first byte. Deprecated in favor of the lease-relay-agent-remote-id-data item.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-remote-id-data</td>
<td>blob (v4, use instead of lease-relay-agent-remote-id item)</td>
</tr>
<tr>
<td>Accesses and manipulates the relay-agent-remote-id-data as stored with the lease of a response.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-server-id-override-data</td>
<td>IP address (v4)</td>
</tr>
<tr>
<td>Accesses and manipulates the relay-agent-server-id-override-data as stored with the lease of a response.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-subnet-selection-data</td>
<td>IP address (v4)</td>
</tr>
<tr>
<td>Accesses and manipulates the relay-agent-subnet-selection-data as stored with the lease of a response.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-subscriber-id</td>
<td>string (v4)</td>
</tr>
<tr>
<td>Contents of the subscriber-id suboption of option 82.</td>
<td></td>
</tr>
<tr>
<td>lease-relay-agent-vpn-id-data</td>
<td>blob (v4)</td>
</tr>
<tr>
<td>Accesses and manipulates the relay-agent-vpn-id data as stored with the lease of a response.</td>
<td></td>
</tr>
<tr>
<td>lease-reserved</td>
<td>int (v4, v6, read-only)</td>
</tr>
<tr>
<td>Returns 1 if the lease is lease reserved and 0 if not.</td>
<td></td>
</tr>
<tr>
<td>lease-start-time-of-state</td>
<td>int (v4, v6, read-only)</td>
</tr>
<tr>
<td>Time, in seconds since 1970, that this lease was first placed into its current state.</td>
<td></td>
</tr>
</tbody>
</table>
### Table C-6  Response Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>lease-state</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>State of the lease, which can be available, offered, leased, expired, unavailable, released, other-available (DHCPv4 only), pending-available (DHCPv4 only), or revoked (DHCPv6 only).</td>
</tr>
<tr>
<td>lease-state-expiration-time</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Expiration time of the IPv6 lease state.</td>
</tr>
<tr>
<td>lease-status</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Returns “nonexistent,” “owned-by-client,” or “exists.” Used to determine if a lease exists and if the current client owns it. If “exists” is returned, the lease exists but the current owner does not own it (limited information on the lease is available).</td>
</tr>
<tr>
<td>lease-valid-lifetime</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Valid lifetime of the IPv6 lease.</td>
</tr>
<tr>
<td>lease-vpn-description</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Description for the VPN stored with the lease of a response.</td>
</tr>
<tr>
<td>lease-vpn-id</td>
<td>int (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Identifier for the VPN stored with the lease of a response.</td>
</tr>
<tr>
<td>lease-vpn-name</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Name of the VPN stored with the lease of a response.</td>
</tr>
<tr>
<td>lease-vpn-vpn-id</td>
<td>blob, typically 7 bytes (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Virtual private network (VPN) identifier stored with the lease of a response.</td>
</tr>
<tr>
<td>lease-vpn-vrf-name</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Virtual routing and forwarding table identifier for the VPN stored with the lease of a response.</td>
</tr>
<tr>
<td>mac-address</td>
<td>blob (v4)</td>
</tr>
<tr>
<td></td>
<td>MAC address that came in the client packet. The first byte is the hardware type, the second is the hardware length, and the remaining (up to 16) is the information from the chaddr. This is a useful aggregation of the htype, hlen, and chaddr fields of the DHCP packet. When read it is constructed from these fields; when written it is placed into these fields.</td>
</tr>
<tr>
<td>override-client-id</td>
<td>blob (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Blob used for the current client-id value. Replaces any client-id from the incoming packet (although both values are kept in the lease state database).</td>
</tr>
<tr>
<td>override-client-id-data-type</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Returns the data type of the override-client-id, either “nstr” for string or “blob” for blob.</td>
</tr>
<tr>
<td>override-client-id-string</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td></td>
<td>Current client-ID value in string format that replaces any client-id from the incoming packet (although both values are kept in the lease state database).</td>
</tr>
<tr>
<td></td>
<td>For a get, if the override-client-id is not a string, the binary data is formatted as blob data, which is then returned as the “string.”</td>
</tr>
</tbody>
</table>
The response packet. For DHCPv4, this is the same as client-packet. For DHCPv6, this is the full packet if relayed or the same as client-packet if not relayed. It should only be read or written from the post-packet-encode extension point; if written, the server will then send the new packet to the client.

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>packet</td>
<td>blob (v4, v6, use only at post-packet-decode)</td>
</tr>
<tr>
<td>ping-clients</td>
<td>int (v4)</td>
</tr>
<tr>
<td>prefix-address</td>
<td>IPv6 prefix (v6, read-only)</td>
</tr>
<tr>
<td>prefix-allocate-random</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>prefix-allocate-via-best-fit</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>prefix-allocate-via-client-request</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>prefix-allocate-via-extension</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>prefix-allocate-via-reservation</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>prefix-allocate-via-interface-identifier</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>prefix-allocation-group</td>
<td>string (v6, read-only)</td>
</tr>
<tr>
<td>prefix-allocation-group-priority</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>prefix-deactivated</td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>prefix-dhcp-type</td>
<td>string (v6, read-only)</td>
</tr>
<tr>
<td>prefix-expiration-time</td>
<td>string (v6, read-only)</td>
</tr>
<tr>
<td>prefix-link-group-name</td>
<td>string (v6, read-only)</td>
</tr>
<tr>
<td>prefix-link-name</td>
<td>string (v6, read-only)</td>
</tr>
</tbody>
</table>
### Table C-6  Response Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>prefix-link-type</code></td>
<td>string (v6, read-only)</td>
</tr>
<tr>
<td>Link type (topological, location-independent, or universal).</td>
<td></td>
</tr>
<tr>
<td><code>prefix-name</code></td>
<td>string (v6, read-only)</td>
</tr>
<tr>
<td>Name of the prefix.</td>
<td></td>
</tr>
<tr>
<td><code>prefix-range</code></td>
<td>IPv6 address (v6, read-only)</td>
</tr>
<tr>
<td>IPv6 address range of the prefix.</td>
<td></td>
</tr>
<tr>
<td><code>prefix-restrict-to-reservations</code></td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>If set to 1, the prefix has restrict-to-reservations enabled.</td>
<td></td>
</tr>
<tr>
<td><code>prefix-selection-tags</code></td>
<td>string (v6, read-only)</td>
</tr>
<tr>
<td>Selection tags of the prefix.</td>
<td></td>
</tr>
<tr>
<td><code>relay-count</code></td>
<td>int (v6, read-only)</td>
</tr>
<tr>
<td>Number of DHCPv6 relay hops.</td>
<td></td>
</tr>
<tr>
<td><code>reply-ipaddress</code></td>
<td>IPv4 or IPv6 address (v4, v6)</td>
</tr>
<tr>
<td>IP address to use when replying to the DHCP client. Read just after <code>pre-packet-encode</code>. If you change its value in a <code>pre-packet-encode</code>, the IP address you place in it should be for a system that can respond to ARP queries (unless it is a broadcast address). Even if unicast is enabled and the broadcast flag is not set in the DHCP request, the local ARP cache is not set with a mapping from a new <code>reply-ipaddress</code> in the <code>pre-packet-encode</code> to the MAC address in the DHCP request.</td>
<td></td>
</tr>
<tr>
<td><code>reply-port</code></td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>Port to use when replying to the DHCP client. Read just after <code>pre-packet encode</code>.</td>
<td></td>
</tr>
<tr>
<td><code>response-source</code></td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td>The source of the response (the major activity that invoked the extension). Output values are: client (Received client packet), failover (Received binding update from the failover partner), timeout (Lease expiration or grace period end), operator (Request from a user interface), one-lease-per-client (One lease per client removing a client from an old lease because of a new one), unknown (None of the above). This data item helps an extension to determine what processing it should do whether a request dictionary is present or not. (The <code>isValid</code> method can also be used to determine whether a dictionary is valid.)</td>
<td></td>
</tr>
<tr>
<td><code>reverse-name-in-dns</code></td>
<td>int (v4, v6)</td>
</tr>
<tr>
<td>If equal to 1, the reverse name is in DNS. Read before initializing a DNS operation.</td>
<td></td>
</tr>
<tr>
<td><code>scope-allow-bootp</code></td>
<td>int (v4, read-only)</td>
</tr>
<tr>
<td>If set to 1, the scope allows BOOTP. Written after a DNS operation finishes.</td>
<td></td>
</tr>
<tr>
<td><code>scope-allow-dhcp</code></td>
<td>int (v4, read-only)</td>
</tr>
<tr>
<td>If set to 1, the scope allows DHCP.</td>
<td></td>
</tr>
<tr>
<td><code>scope-allow-dynamic-bootp</code></td>
<td>int (v4, read-only)</td>
</tr>
<tr>
<td>If set to 1, the scope allows dynamic BOOTP.</td>
<td></td>
</tr>
<tr>
<td><code>scope-available-leases</code></td>
<td>int (v4, read-only)</td>
</tr>
<tr>
<td>Number of available leases on the current scope.</td>
<td></td>
</tr>
</tbody>
</table>
### Table C-6  Response Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: (v4=\text{DHCPv4}, v6=\text{DHCPv6}))</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>scope-deactivated</code></td>
<td><code>int (v4, read-only)</code></td>
</tr>
<tr>
<td>If set to 1, the scope is deactivated.</td>
<td></td>
</tr>
<tr>
<td><code>scope-dns-forward-server-address</code></td>
<td><code>IP address (v4, read-only)</code></td>
</tr>
<tr>
<td>DNS server to use for the DNS forward address.</td>
<td></td>
</tr>
<tr>
<td><code>scope-dns-forward-zone-name</code></td>
<td><code>string (v4, read-only)</code></td>
</tr>
<tr>
<td>Forward zone name configured in the scope.</td>
<td></td>
</tr>
<tr>
<td><code>scope-dns-number-of-host-bytes</code></td>
<td><code>int (v4, read-only)</code></td>
</tr>
<tr>
<td>Number of host bytes used by the DHCP server code that handles DNS updates.</td>
<td></td>
</tr>
<tr>
<td><code>scope-dns-reverse-server-address</code></td>
<td><code>IP address (v4, read-only)</code></td>
</tr>
<tr>
<td>DNS server to use for the DNS reverse address.</td>
<td></td>
</tr>
<tr>
<td><code>scope-dns-reverse-zone-name</code></td>
<td><code>string (v4, read-only)</code></td>
</tr>
<tr>
<td>Reverse zone name configured in the scope.</td>
<td></td>
</tr>
<tr>
<td><code>scope-network-number</code></td>
<td><code>IP address (v4, read-only)</code></td>
</tr>
<tr>
<td>Network number of the scope that contains the lease the DHCP server is processing.</td>
<td></td>
</tr>
<tr>
<td><code>scope-ping-clients</code></td>
<td><code>int (v4, read-only)</code></td>
</tr>
<tr>
<td>If set to 1, the scope associated with the current lease was configured to support a ping operation prior to offering a lease.</td>
<td></td>
</tr>
<tr>
<td><code>scope-primary-network-number</code></td>
<td><code>IP address (v4, read-only)</code></td>
</tr>
<tr>
<td>Network number of this primary scope.</td>
<td></td>
</tr>
<tr>
<td><code>scope-primary-subnet-mask</code></td>
<td><code>IP address (v4, read-only)</code></td>
</tr>
<tr>
<td>Subnet mask of this primary scope.</td>
<td></td>
</tr>
<tr>
<td><code>scope-renew-only</code></td>
<td><code>int (v4, read-only)</code></td>
</tr>
<tr>
<td>If set to 1, the scope is renew-only.</td>
<td></td>
</tr>
<tr>
<td><code>scope-renew-only-expire-time</code></td>
<td><code>int (v4, read-only)</code></td>
</tr>
<tr>
<td>Absolute time, in seconds since January 1, 1970, at which a renew-only scope should cease to be renew-only.</td>
<td></td>
</tr>
<tr>
<td><code>scope-restrict-to-reservations</code></td>
<td><code>int (v4, read-only)</code></td>
</tr>
<tr>
<td>If set to 1, the scope has restrict-to-reservations enabled.</td>
<td></td>
</tr>
<tr>
<td><code>scope-selection-tags</code></td>
<td><code>string (v4, read-only)</code></td>
</tr>
<tr>
<td>Comma-separated string that contains the scope selection criteria. Use this data item for decisions based on scopes.</td>
<td></td>
</tr>
<tr>
<td><code>scope-send-ack-first</code></td>
<td><code>int (v4, read-only)</code></td>
</tr>
<tr>
<td>If set to 1, the scope sends an ACK before performing the rest of the processing.</td>
<td></td>
</tr>
<tr>
<td><code>scope-subnet-mask</code></td>
<td><code>IP address (v4, read-only)</code></td>
</tr>
<tr>
<td>Subnet mask of the scope that contains the lease the DHCP server is processing.</td>
<td></td>
</tr>
<tr>
<td><code>scope-update-dns</code></td>
<td><code>string (v4, read-only)</code></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cisco Prime IP Express 8.2 User Guide
### Extension Dictionary API

This section contains the dictionary method calls to use when accessing dictionaries from Tcl extensions and shared libraries.

### Tcl Attribute Dictionary API

In an attribute dictionary, the keys are constrained to be the names of attributes as defined in the Cisco Prime IP Express DHCP server configuration. The values are the string representation of the legal values for that particular attribute. For example, IP addresses are specified by the dotted-decimal string representation of the address, and enumerated values are specified by the name of the enumeration. This means that numbers are specified by the string representation of the number.

Attribute dictionaries are unusual in that they can contain more than one instance of a key. These instances are ordered, with the first instance at index zero. Some of the attribute dictionary methods allow an index to indicate a particular instance or position in the list of instances to be referenced.

---

### Table C-6  Response Dictionary Specific Data Items (continued)

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Value (Protocol: v4=DHCPv4, v6=DHCPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS updates for forward or reverse zones. Output values are: 1=update-all, 2=update-fwd-only, 3=update-rev-only, and 0=update-none.</td>
<td></td>
</tr>
<tr>
<td>scope-update-dns-enabled</td>
<td>boolean (v4, read-only)</td>
</tr>
<tr>
<td>If set to 1, the scope has update DNS enabled for forward and reverse zones. Deprecated in favor of scope-update-dns.</td>
<td></td>
</tr>
<tr>
<td>scope-update-dns-for-bootp</td>
<td>int (v4, read-only)</td>
</tr>
<tr>
<td>If set to 1, the scope has update DNS enabled for BOOTP.</td>
<td></td>
</tr>
<tr>
<td>trace-id</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td>ID used by the system to trace the packet.</td>
<td></td>
</tr>
<tr>
<td>transaction-time</td>
<td>int (v4, v6, read-only)</td>
</tr>
<tr>
<td>Time, in seconds since 1970, that the request was decoded.</td>
<td></td>
</tr>
<tr>
<td>vpn-description</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td>Description for the VPN.</td>
<td></td>
</tr>
<tr>
<td>vpn-name</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td>Name of the VPN.</td>
<td></td>
</tr>
<tr>
<td>vpn-vpn-id</td>
<td>blob, typically 7 bytes (v4, v6, read-only)</td>
</tr>
<tr>
<td>Virtual private network (VPN) identifier.</td>
<td></td>
</tr>
<tr>
<td>vpn-vrf-name</td>
<td>string (v4, v6, read-only)</td>
</tr>
<tr>
<td>Virtual routing and forwarding table (VRF) identifier for the VPN.</td>
<td></td>
</tr>
</tbody>
</table>
Attribute dictionaries use commands with which you can change and access the values in the dictionaries. Table C-7 lists the commands to use with the request and response dictionaries. In this case, you can define the dict variable as request or response.

See the install-path/examples/dhcp/tcl/tclextension.tcl file for examples.

### Table C-7 Tcl Request and Response Dictionary Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>$dict get attribute [index [bMore]]</td>
</tr>
<tr>
<td>getOption</td>
<td>$dict getOption arg-type [arg-data]</td>
</tr>
<tr>
<td>isValid</td>
<td>$dict isValid</td>
</tr>
<tr>
<td>isV4</td>
<td>$dict isV4</td>
</tr>
<tr>
<td>isV6</td>
<td>$dict isV6</td>
</tr>
<tr>
<td>log</td>
<td>$dict log level message …</td>
</tr>
<tr>
<td>moveToOption</td>
<td>$dict moveToOption arg-type [arg-data]</td>
</tr>
</tbody>
</table>

**get**

Returns the value of the attribute from the dictionary, represented as a string. If the dictionary does not contain the attribute, the empty string is returned instead. If you include the index value, this returns the index\textsuperscript{th} instance of the attribute. Some attributes can appear more than once in the request or response packet. The index selects which instance to return.

If you include the \textit{bMore}, the \textit{get} method sets \textit{bMore} to TRUE if there are more attributes after the one returned, otherwise to FALSE. Use this to determine whether to make another call to \textit{get} to retrieve other instances of the attribute.

**getOption**

Gets the data for an option as a string. See Table C-8 on page C-28 for the \textit{arg-type} values. If the next argument is a numeric value, it is assumed to be a number, otherwise a name. Note that this function always returns a pointer to a string, which can be zero length if the option does not exist or has length zero. For sample usage, see the “Handling Vendor Class Option Data” section on page C-42.

**isValid**

The \textit{isValid} method returns TRUE if there is a request or response (depending on the dictionary passed in); FALSE otherwise. Extensions such as \textit{lease-state-change} can use this method to determine whether a dictionary is available.

The \textit{isV4} method returns TRUE if this extension is being called for a DHCPv4 packet; FALSE otherwise. Calling this method from an \textit{init-entry} routine returns FALSE.

The \textit{isV6} method returns TRUE if this extension is being called for a DHCPv6 packet; FALSE otherwise. Calling this method from an \textit{init-entry} routine returns FALSE.

**log**

Puts a message into the DHCP server logging system. The level should be LOG\_ERROR, LOG\_WARNING, or LOG\_INFO. The remaining arguments are concatenated and sent to the logging system at the specified level.

**Note**

Use the LOG\_ERROR and LOG\_WARNING levels sparingly, because the server flushes its log file with messages logged at these levels. Using these levels for messages that are likely to occur frequently (such as client requests) can have severe impact on disk I/O performance.

**moveToOption**

Sets the context for subsequent \textit{get}, \textit{put}, and \textit{remove} option operations. See Table C-8 on page C-28 for the \textit{arg-type} values. Note that the context can become invalid if the option is removed (such as by \textit{removeOption}).
Appendix C      DHCP Extension Dictionary

**Table C-7**  Tcl Request and Response Dictionary Methods (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>put</td>
<td><code>$dict put attribute value [index]</code></td>
</tr>
<tr>
<td></td>
<td>Associates a value with the attribute in the dictionary. If you omit the index or set it to the special value REPLACE, this replaces any existing instances of the attribute with the single value. If you include the index value as the special value APPEND, this appends a new instance of the attribute to the end of the list of instances of the attribute. If you include the index value as a number, this inserts a new instance of the attribute at the position indicated. If you set the index value to the special value AUGMENT, this puts the attribute only if there is not one already.</td>
</tr>
<tr>
<td>putOption</td>
<td><code>$dict putOption data arg-type [arg-data] ...</code></td>
</tr>
<tr>
<td></td>
<td>Adds an option and its data or modifies the data for an option. See Table C-8 on page C-28 for the arg-type values. For sample usage, see the “Handling Vendor Class Option Data” section on page C-42.</td>
</tr>
<tr>
<td>remove</td>
<td><code>$dict remove attribute [index]</code></td>
</tr>
<tr>
<td></td>
<td>Removes the attribute from the dictionary. If you omit the index or set it to the special value REMOVE_ALL, this removes any existing instances of the attribute. If you include the index as a number, this removes the instance of the attribute at the position indicated. This method always returns 1, even if the dictionary does not contain that attribute at that index.</td>
</tr>
<tr>
<td>removeOption</td>
<td><code>$dict removeOption arg-type [arg-data] ...</code></td>
</tr>
<tr>
<td></td>
<td>Removes an option. See Table C-8 on page C-28 for the arg-type values. For sample usage, see the “Handling Vendor Class Option Data” section on page C-42.</td>
</tr>
<tr>
<td>setObject</td>
<td><code>$dict setObject obj-type [data]</code></td>
</tr>
<tr>
<td></td>
<td>(DHCPv6 only.) Sets the object for get, put, and remove methods, and alters the message on which the new option methods operate. See Table C-8 on page C-28 for the obj-type values. DHCPv6 extensions primarily use this method to access the leases and prefixes available for the client and link, or to get message header fields or options from relay packets. Unlike in DHCPv4, where one lease and scope are associated with a response, a DHCPv6 response can involve several leases and prefixes. Returns TRUE if the object exists; FALSE otherwise. For sample usage, see the “Handling Object Data” section on page C-42.</td>
</tr>
<tr>
<td>trace</td>
<td><code>$dict trace level message …</code></td>
</tr>
<tr>
<td></td>
<td>Returns a message in the DHCP server packet tracing system. At level 0, no tracing occurs. At level 1, it traces only that the server received the packet and sent a reply. At level 4, it traces everything. The remaining arguments are concatenated and sent to the tracing system at the specified level. The default tracing is set using the DHCP server extension-trace-level attribute.</td>
</tr>
</tbody>
</table>

**Note** For leases not associated with the current client, only minimal information is available.

**Table C-8**  Tcl arg-type and obj-type Values

<table>
<thead>
<tr>
<th>arg-type or obj-type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enterprise-id number/name</td>
<td>Enterprise-id number or name for the option definition set for the option or suboption.</td>
</tr>
<tr>
<td>home</td>
<td>Requests that the context is reset to the “top” of the current client or relay message.</td>
</tr>
<tr>
<td>index number/keyword</td>
<td>Number or keyword (replace, append, augment, raw, or remove_all) for the array index on which to operate.</td>
</tr>
</tbody>
</table>
Tcl Environment Dictionary Methods

Table C-9 describes the commands to use with the environment dictionary. In this case, you can define the dict variable as environ, as in the following procedure example:

```tcl
proc tclhelloworld2 { request response environ } {
    $environ put trace-level 4
    $environ log LOG_INFO "Environment hello world"
}
```
When writing DEX extensions for C/C++, you can specify keys as the attribute name string representation or by type (a byte sequence defining the attribute). This means that some of these access methods have four different variations that are the combinations of string or type for the key or value.

**Table C-9  Tcl Environment Dictionary Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear</td>
<td>$dict clear</td>
<td>Removes all entries from the dictionary.</td>
</tr>
<tr>
<td>containsKey</td>
<td>$dict containsKey key</td>
<td>Returns 1 if the dictionary contains the key, otherwise 0.</td>
</tr>
<tr>
<td>firstKey</td>
<td>$dict firstKey</td>
<td>Returns the name of the first key in the dictionary. Note that the keys are not stored sorted by name. If a key does not exist, returns the empty string.</td>
</tr>
<tr>
<td>get</td>
<td>$dict get key</td>
<td>Returns the value of the key from the dictionary. If a key does not exist, returns the empty string.</td>
</tr>
<tr>
<td>isEmpty</td>
<td>$dict isEmpty</td>
<td>Returns 1 if the dictionary has no entries, otherwise 0.</td>
</tr>
<tr>
<td>log</td>
<td>$dict log level message</td>
<td>Returns a message in the DHCP server logging system. The level should be one of LOG_ERROR, LOG_WARNING, or LOG_INFO. The remaining arguments are concatenated and sent to the logging system at the specified level. <strong>Note</strong> Use the LOG_ERROR and LOG_WARNING levels sparingly, because the server flushes its log file with messages logged at these levels. Using these levels for messages that are likely to occur frequently (such as client requests) can have a severe impact on disk I/O performance.</td>
</tr>
<tr>
<td>nextKey</td>
<td>$dict nextKey</td>
<td>Returns the name of the next key in the dictionary that follows the key returned in the last call to firstKey or nextKey. If a key does not exist, returns the empty string.</td>
</tr>
<tr>
<td>put</td>
<td>$dict put key value</td>
<td>Associates a value with the key, replacing an existing instance of the key with the new value.</td>
</tr>
<tr>
<td>remove</td>
<td>$dict remove key</td>
<td>Removes the key from the dictionary. Always returns 1, even if the dictionary did not contain the key.</td>
</tr>
<tr>
<td>size</td>
<td>$dict size</td>
<td>Returns the number of entries in the dictionary.</td>
</tr>
<tr>
<td>trace</td>
<td>$dict trace level message</td>
<td>Returns a message in the DHCP server packet tracing system. At level 0, no tracing occurs. At level 1, it traces only that the server received the packet and sent a reply. At level 4, it traces everything. The remaining arguments are concatenated and sent to the tracing system at the specified level. The default tracing is set using the DHCP server extension-trace-level attribute.</td>
</tr>
</tbody>
</table>

**DEX Attribute Dictionary API**

When writing DEX extensions for C/C++, you can specify keys as the attribute name string representation or by type (a byte sequence defining the attribute). This means that some of these access methods have four different variations that are the combinations of string or type for the key or value.
A basic DEX extension example might be:

```c
int DEXAPI dexhelloworld( int iExtensionPoint,
    dex_AttributeDictionary_t *pRequest,
    dex_AttributeDictionary_t *pResponse,
    dex_EnvironmentDictionary_t *pEnviron )
{
    pEnviron->log( pEnviron, DEX_LOG_INFO, "hello world" );
    return DEX_OK;
}
```

See the `install-path/examples/dhcp/dex/dexextension.c` file or other files in that directory for examples.

**DEX Request and Response Dictionary Methods**

DEX attribute dictionaries use active commands, called methods, with which you can change and access values. Table C-10 on page C-31 lists the methods to use with the request and response dictionaries. In this case, you can define the `pDict` variable as `pRequest` or `pResponse`, as in:

```c
pRequest->get( pRequest, "host-name", 0, 0 );
```

The `pszAttribute` is the `const char *` pointer to the attribute name that the application wants to access. The `pszValue` is the pointer to the `const char *` string that represents the data (returned for a `get` method, and stored in a `put` method). See Table C-11 on page C-35, Table C-12 on page C-36, and Table C-13 on page C-37 for the valid `iObjectType`, `iObjArgType`, and `iArgType` values, respectively.

### Table C-10  DEX Request and Response Dictionary Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocateMemory</td>
<td><code>void *pDict-&gt;allocateMemory( dex_AttributeDictionary_t *pDict, unsigned int iSize )</code></td>
</tr>
<tr>
<td>allocateMemory</td>
<td>Allocates memory in extensions that persists only for the lifetime of this request.</td>
</tr>
<tr>
<td>get</td>
<td><code>const char *pDict-&gt;get( dex_AttributeDictionary_t *pDict, const char *pszAttribute, int iIndex, abool_t *pbMore )</code></td>
</tr>
<tr>
<td>get</td>
<td>Returns the value of the <code>iIndex</code>ed instance of the attribute from the dictionary, represented as a string. If the dictionary does not contain the attribute (or that many instances of it), the empty string is returned instead. If <code>pbMore</code> is nonzero, the <code>get</code> method sets <code>pbMore</code> to TRUE if there are more instances of the attribute after the one returned, otherwise to FALSE. Use this to determine whether to make another call to <code>get</code> to retrieve other instances of the attribute.</td>
</tr>
<tr>
<td>getBytes</td>
<td><code>const abytes_t *pDict-&gt;getBytes( dex_AttributeDictionary_t *pDict, const char *pszAttribute, int iIndex, abool_t *pbMore )</code></td>
</tr>
<tr>
<td>getBytes</td>
<td>Returns the value of the <code>iIndex</code>ed instance of the attribute as a sequence of bytes. If the dictionary does not contain the attribute (or that many instances of it), returns 0 instead. If <code>pbMore</code> is nonzero, the <code>getBytes</code> method sets it to TRUE if there are more instances of the attribute after the one returned, otherwise to FALSE. Use this to determine whether to make another call to <code>getBytes</code> to retrieve other instances of the attribute.</td>
</tr>
</tbody>
</table>
Table C-10  DEX Request and Response Dictionary Methods (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>getBytesByType</td>
<td><code>const abytes_t *pDict-&gt;getBytesByType( dex_AttributeDictionary_t *pDict, const abytes_t *pszAttribute, int iIndex, abool_t *pbMore )</code></td>
</tr>
<tr>
<td></td>
<td>Returns the value of the indexed instance of the attribute from the dictionary as a sequence of bytes. If the dictionary does not contain the attribute (or that many instances of it), 0 is returned instead. If pbMore is nonzero, sets the variable pointed to TRUE if there are more instances of the attribute after the one returned, otherwise to FALSE. Use this to determine whether to make another call to get to retrieve other instances of the attribute.</td>
</tr>
<tr>
<td>getByType</td>
<td><code>const char *pDict-&gt;getByType( dex_AttributeDictionary_t *pDict, const abytes_t *pszAttribute, int iIndex, abool_t *pbMore )</code></td>
</tr>
<tr>
<td></td>
<td>Returns the value of the indexed instance of the attribute from the dictionary, represented as a string. If the dictionary does not contain the attribute (or that many instances of it), returns the empty string instead. If pbMore is nonzero, the getByType method sets pbMore to TRUE if there are more instances of the attribute after the one returned, otherwise to FALSE. Use this to determine whether to make another call to getByType to retrieve other instances.</td>
</tr>
<tr>
<td>getOption</td>
<td><code>const char *getOption( dex_AttributeDictionary_t *pDict, int iArgType, ... )</code></td>
</tr>
<tr>
<td></td>
<td>Gets the data for an option as a string. Note that this function always returns a pointer to a string, which can be zero length if the option does not exist or has length zero. To find out if the option exists, use getOptionBytes or specify DEX_INSTANCE_COUNT.</td>
</tr>
<tr>
<td>getOptionBytes</td>
<td><code>const abytes_t *getOptionBytes( dex_AttributeDictionary_t *pDict, int iArgType, ... )</code></td>
</tr>
<tr>
<td></td>
<td>Gets the data for an option as a sequence of bytes. Note that this function returns a null pointer if the option does not exist, and an abytes_t with a zero-length buffer if the option exists but is zero bytes long.</td>
</tr>
<tr>
<td>getType</td>
<td><code>const abytes_t *pDict-&gt;getType( dex_AttributeDictionary_t *pDict, const char *pszAttribute )</code></td>
</tr>
<tr>
<td></td>
<td>Returns a pointer to the byte sequence defining the attribute, if the attribute name matches a configured attribute, otherwise 0.</td>
</tr>
<tr>
<td>isValid</td>
<td><code>abool_t isValid( dex_AttributeDictionary_t *pDict )</code></td>
</tr>
<tr>
<td>isV4</td>
<td><code>abool_t isV4( dex_AttributeDictionary_t *pDict )</code></td>
</tr>
<tr>
<td>isV6</td>
<td><code>abool_t isV6( dex_AttributeDictionary_t *pDict )</code></td>
</tr>
<tr>
<td></td>
<td>The isValid method returns TRUE if there is a request or response (depending on the dictionary passed in); FALSE otherwise. Extensions such as lease-state-change can use this method to determine whether a dictionary is available.</td>
</tr>
<tr>
<td></td>
<td>The isV4 method returns TRUE if this extension is being called for a DHCPv4 packet; FALSE otherwise. Calling this method from an init-entry routine returns FALSE.</td>
</tr>
<tr>
<td></td>
<td>The isV6 method returns TRUE if this extension is being called for a DHCPv6 packet; FALSE otherwise. Calling this method from an init-entry routine returns FALSE.</td>
</tr>
</tbody>
</table>
**Table C-10  DEX Request and Response Dictionary Methods (continued)**

<table>
<thead>
<tr>
<th>Method</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>log</td>
<td>`abool_t pDict-&gt;log(dex_AttributeDictionary_t *pDict, int eLevel,</td>
</tr>
<tr>
<td></td>
<td>const char *pszFormat, ...)`</td>
</tr>
<tr>
<td></td>
<td>Returns a message in the DHCP server logging system. The <code>eLevel</code></td>
</tr>
<tr>
<td></td>
<td>should be one of <code>DEX_LOG_ERROR</code>, <code>DEX_LOG_WARNING</code>, or <code>DEX_LOG_INFO</code>.</td>
</tr>
<tr>
<td></td>
<td>The <code>pszFormat</code> is treated as a printf style format string, and it,</td>
</tr>
<tr>
<td></td>
<td>along with the remaining arguments, are formatted and sent to the</td>
</tr>
<tr>
<td></td>
<td>logging system at the specified level.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Use the <code>DEX_LOG_ERROR</code> and <code>DEX_LOG_WARNING</code> levels</td>
</tr>
<tr>
<td></td>
<td>sparingly, because the server flushes its log file with messages</td>
</tr>
<tr>
<td></td>
<td>logged at these levels. Using these levels for messages that</td>
</tr>
<tr>
<td></td>
<td>are likely to occur frequently (such as client requests) can have</td>
</tr>
<tr>
<td></td>
<td>severe impact on disk I/O performance.</td>
</tr>
<tr>
<td>moveToOption</td>
<td>`abool_t moveToOption(dex_AttributeDictionary_t *pDict, int iArgType,</td>
</tr>
<tr>
<td></td>
<td>...)`</td>
</tr>
<tr>
<td></td>
<td>Sets the context for subsequent <code>get</code>, <code>put</code>, and <code>remove</code> option</td>
</tr>
<tr>
<td></td>
<td>operations. Note that the context can become invalid if the option</td>
</tr>
<tr>
<td></td>
<td>is removed (such as with <code>removeOption</code>).</td>
</tr>
<tr>
<td>put</td>
<td>`abool_t pDict-&gt;put(dex_AttributeDictionary_t *pDict, const char *</td>
</tr>
<tr>
<td></td>
<td>pszAttribute, const char *pszValue, int iIndex)`</td>
</tr>
<tr>
<td></td>
<td>Converts <code>pszValue</code> to a sequence of bytes, according to the definition</td>
</tr>
<tr>
<td></td>
<td>of <code>pszAttribute</code> in the server configuration. Associates that</td>
</tr>
<tr>
<td></td>
<td>sequence of bytes with the attribute in the dictionary. If <code>iIndex</code></td>
</tr>
<tr>
<td></td>
<td>is the special value <code>DEX_REPLACE</code>, replaces any existing instances of</td>
</tr>
<tr>
<td></td>
<td>the attribute with a single value. If the special value <code>DEX_APPEND</code>,</td>
</tr>
<tr>
<td></td>
<td>appends a new instance of the attribute to its list. If the special</td>
</tr>
<tr>
<td></td>
<td>value <code>DEX_AUGMENT</code>, puts the attribute only if there is not one</td>
</tr>
<tr>
<td></td>
<td>already. Otherwise, inserts a new instance at the position indicated.</td>
</tr>
<tr>
<td></td>
<td>Returns TRUE unless the attribute name does not match any configured</td>
</tr>
<tr>
<td></td>
<td>attributes or the value could not be converted to a legal value.</td>
</tr>
<tr>
<td>putBytes</td>
<td>`abool_t pDict-&gt;putBytes(dex_AttributeDictionary_t *pDict, const char *</td>
</tr>
<tr>
<td></td>
<td>pszAttribute, const abytes_t *pszValue, int iIndex)`</td>
</tr>
<tr>
<td></td>
<td>Associates <code>pszValue</code> with the <code>pszAttribute</code> in the dictionary. If</td>
</tr>
<tr>
<td></td>
<td><code>iIndex</code> is the special value <code>DEX_REPLACE</code>, replaces any existing</td>
</tr>
<tr>
<td></td>
<td>instances of the attribute with a single new value. If the special</td>
</tr>
<tr>
<td></td>
<td>value <code>DEX_APPEND</code>, appends a new instance of the attribute to its</td>
</tr>
<tr>
<td></td>
<td>list. If the special value <code>DEX_AUGMENT</code>, puts the attribute only if</td>
</tr>
<tr>
<td></td>
<td>there is not one already. Otherwise, inserts a new instance at the</td>
</tr>
<tr>
<td></td>
<td>position indicated. Returns TRUE unless the attribute name does not</td>
</tr>
<tr>
<td></td>
<td>match a configured one.</td>
</tr>
<tr>
<td>putBytesByType</td>
<td>`abool_t pDict-&gt;putBytesByType(dex_AttributeDictionary_t *pDict,</td>
</tr>
<tr>
<td></td>
<td>const abytes_t *pszAttribute, const abytes_t *pszValue, int iIndex)`</td>
</tr>
<tr>
<td></td>
<td>Associates <code>pszValue</code> with the <code>pszAttribute</code> in the dictionary. If</td>
</tr>
<tr>
<td></td>
<td><code>iIndex</code> is the special value <code>DEX_REPLACE</code>, replaces any existing</td>
</tr>
<tr>
<td></td>
<td>instances of the attribute with the new value. If the special value</td>
</tr>
<tr>
<td></td>
<td><code>DEX_APPEND</code>, appends a new instance of the attribute to its list. If</td>
</tr>
<tr>
<td></td>
<td>the special value <code>DEX_AUGMENT</code>, puts the attribute only if there is</td>
</tr>
<tr>
<td></td>
<td>not one already. Otherwise, inserts a new instance of the attribute</td>
</tr>
<tr>
<td></td>
<td>at the position indicated.</td>
</tr>
<tr>
<td>putByType</td>
<td>`abool_t pDict-&gt;putByType(dex_AttributeDictionary_t *pDict, const abytes_t *</td>
</tr>
<tr>
<td></td>
<td>pszAttribute, const char *pszValue, int iIndex)`</td>
</tr>
<tr>
<td></td>
<td>Converts <code>pszValue</code> to a sequence of bytes, according to the definition</td>
</tr>
<tr>
<td></td>
<td>of <code>pszAttribute</code> in the server configuration. Associates that</td>
</tr>
<tr>
<td></td>
<td>sequence of bytes with the attribute in the dictionary. If <code>iIndex</code></td>
</tr>
<tr>
<td></td>
<td>is the special value <code>DEX_REPLACE</code>, replaces any existing instances of</td>
</tr>
<tr>
<td></td>
<td>the attribute with a single new value. If the special value <code>DEX_APPEND</code>,</td>
</tr>
<tr>
<td></td>
<td>appends a new instance of the attribute to its list. If the special</td>
</tr>
<tr>
<td></td>
<td>value <code>DEX_AUGMENT</code>, puts the attribute only if there is not one</td>
</tr>
<tr>
<td></td>
<td>already. Otherwise, inserts a new instance at the position</td>
</tr>
<tr>
<td></td>
<td>indicated.</td>
</tr>
</tbody>
</table>
Table C-10 DEX Request and Response Dictionary Methods (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>putOption</td>
<td>abool_t putOption( dex_AttributeDictionary_t *pDict, const char *pszValue, int iArgType, ... )</td>
</tr>
<tr>
<td>putOptionBytes</td>
<td>abool_t putOptionBytes( dex_AttributeDictionary_t *pDict, const abytes_t *pValue, int iArgType, ... )</td>
</tr>
<tr>
<td>remove</td>
<td>abool_t pDict-&gt;remove( dex_AttributeDictionary_t *pDict, const char *pszAttribute, int iIndex )</td>
</tr>
<tr>
<td>removeByType</td>
<td>abool_t pDict-&gt;removeByType( dex_AttributeDictionary_t *pDict, const abytes_t *pszAttribute, int iIndex )</td>
</tr>
<tr>
<td>removeOption</td>
<td>abool_t removeOption( dex_AttributeDictionary *pDict, int iArgType, ... )</td>
</tr>
<tr>
<td>setObject</td>
<td>abool_t setObject( dex_AttributeDictionary_t *pDict, int iObjectType, int iObjArgType, ... )</td>
</tr>
<tr>
<td>trace</td>
<td>abool_t pDict-&gt;trace( dex_AttributeDictionary_t *pDict, int iLevel, const char *pszFormat, ... )</td>
</tr>
</tbody>
</table>

**Differences in get, put, Option, Bytes, and OptionBytes Methods**

There are differences among the following DEX extension methods:

- **get** and **put**
- **getOption** and **putOption**
Appendix C      DHCP Extension Dictionary

The `getBytes` and `putBytes` methods return the requested information formatted as a string. The server converts the data to the string depending on the expected data type for the dictionary item. If the data type is unknown, the server returns the data in blob string format.

The `getBytes` and `getOptionBytes` methods return the requested information as the raw bytes (a pointer to a buffer and the size of that buffer). The server should have to read this buffer only, and it contains only the data from the option (no null terminator has been added, for example).

The `put` and `putOption` methods expect the data to be written as a formatted string. The server converts the data from the string depending on the expected data type for the dictionary item. If the data type is unknown, it is expected to be in blob string format.

The server passes raw bytes to the `putBytes` and `putOptionBytes` methods (a pointer to a buffer and the size of that buffer). The server only reads these bytes.

### Differences in get, put, remove, and ByType Methods

There are differences among the following DEX extension methods:

- `get`, `put`, and `remove`
- `getByType`, `putByType`, and `removeByType`

The server passes the `get`, `put`, and `remove` methods the name of the desired data item as a string. This requires that the server map the string to its internal data tables.

The server passes the `getByType`, `putByType`, and `removeByType` methods an internal data table reference, which the server must have previously obtained (such as in the extension init-entry) by calling the `getType` method on the string. This speeds processing for extensions, which can be important in applications requiring high performance.

---

**Note**

The internal data table that the `getType` method references is the same whether requested for the Request or Response dictionary. There is no need for separate `getType` calls on each dictionary for the same data item name.

### Table C-11   DEX iObjectType Values

<table>
<thead>
<tr>
<th>iObjectType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General definition: Object for which the context is to be changed. (See also Table C-12 on page C-36.)</td>
<td></td>
</tr>
<tr>
<td>DEXLEASE</td>
<td>Changes the lease (and prefix) context. Response dictionary only. Allows iObjectTypeArg: DEX_BY_IPV6ADDRESS DEX_BY_IPV6PREFIX DEX_BY_INSTANCE DEX_INITIAL</td>
</tr>
</tbody>
</table>
### Table C-11  DEX iObjectType Values (continued)

<table>
<thead>
<tr>
<th>iObjectType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEX_MESSAGE</td>
<td>Changes the message context to a relay message or the client message. Request and response dictionaries. Allows iObjArgType:</td>
</tr>
<tr>
<td></td>
<td>DEX_INITIAL</td>
</tr>
<tr>
<td></td>
<td>DEX_RELAY</td>
</tr>
<tr>
<td></td>
<td>DEX_BY_NUMBER</td>
</tr>
<tr>
<td>DEX_PREFIX</td>
<td>Changes the prefix context, but does not change the lease context. Response dictionary only. Allows iObjTypeArg:</td>
</tr>
<tr>
<td></td>
<td>DEX_BY_IPV6ADDRESS</td>
</tr>
<tr>
<td></td>
<td>DEX_BY_IPV6PREFIX</td>
</tr>
<tr>
<td></td>
<td>DEX_BY_INSTANCE</td>
</tr>
<tr>
<td></td>
<td>DEX_BY_NAME</td>
</tr>
<tr>
<td></td>
<td>DEX_INITIAL</td>
</tr>
</tbody>
</table>

### Table C-12  DEX iObjArgType Values

<table>
<thead>
<tr>
<th>iObjArgType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEX_BY_INSTANCE</td>
<td>Used with DEX_LEASE or DEX_PREFIX iObjectType. Requires that int follows to specify the instance number (starting with 0). Used to walk through the list of all available objects, but only through the list of objects applicable to the current request or response: for DEX_LEASE, the leases for that client (if any); for DEX_PREFIX, the prefixes on the current link (if any). Used with DEX_MESSAGE, a synonym for DEX_RELAY.</td>
</tr>
<tr>
<td>DEX_BY_IPV6ADDRESS</td>
<td>Used with DEX_LEASE and DEX_PREFIX iObjectType only. Requires that const unsigned char * follows to specify the 16-byte address.</td>
</tr>
<tr>
<td>DEX_BY_IPV6PREFIX</td>
<td>Used with DEX_LEASE or DEX_PREFIX iObjectType. Requires that const unsigned char * follows to specify a 17-byte prefix buffer (16-byte address followed by a 1-byte prefix length).</td>
</tr>
<tr>
<td>DEX_BY_NAME</td>
<td>Used with the DEX_PREFIX iObjectType only. Requires that a const char * follows to specify the name of the desired object.</td>
</tr>
<tr>
<td>DEX_INITIAL</td>
<td>Resets the context back to the original for the request or response, and has no additional argument. Sets the lease and prefix (DEX_LEASE), prefix (DEX_PREFIX), or message (DEX_MESSAGE) to what it was when the extension was originally called (which can be none).</td>
</tr>
<tr>
<td>DEX_RELAY</td>
<td>Used with DEX_MESSAGE iObjectType only. Requires that int follows to specify the relay (0 specifies the relay closest to the client). To set the message context back to the client, use setObject( pDict, DEX_MESSAGE, DEX_INITIAL ).</td>
</tr>
</tbody>
</table>
### Table C-13  DEX iArgType Values

<table>
<thead>
<tr>
<th>iArgType</th>
<th>Description</th>
</tr>
</thead>
</table>
| DEX_ARG_ARRAY             | Requires that a pointer to an array of `dex_OptionsArgs_t` follow, and is an alternative to specifying the argument list. Each `dex_OptionsArgs_t` structure has two fields:  
  - `iArgType`—One of the `iArgType` DEX values in this table.  
  - `pData`—Data (integer), pointer to the data (for strings and other data types), or ignored (if the `iArgType` takes no arguments).  
  Note that once the server encounters the DEX_ARG_ARRAY (in an argument list or in an array of `dex_OptionsArgs_t`), it ignores any subsequent arguments in the original list. |
| DEX_END                   | Note Required, has no additional argument, and marks the end of the argument list.                                                          |
| DEX_ENTERPRISE_NAME       | Requires that `const char *` follow to specify the option definition set name, from which the server extracts the enterprise-id to get the vendor option data. Valid only for vendor-identifying options. Requires that the vendor option definition set exists. |
| DEX_ENTERPRISE_ID         | Requires that `int` follow to specify the enterprise-id for the vendor.                                                                       |
| DEX_HOME                  | Moves the context back to the client or relay message options. Has no additional argument. Always returns success. If used, must be the first `iArgType`. Valid only for `getOption`, `getOptionBytes`, and `moveToOption` methods. |
| DEX_INDEX                 | Requires that `int` follow with the index of the option data (if any array of data is to be acted on). If omitted, index 0 is assumed, except for `removeOption`, in which case DEX_REMOVE_ALL is assumed. Use the special value DEX_RAW to get, put, or remove the entire option data. However, for the DHCPv4 Vendor-Identifying Vendor Options (RFC 3925 and RFC 4243), DEX_RAW returns the data for only one vendor (based on the instance or enterprise-id) and not that for the entire option. The DEX_RAW special value accesses the entire option (or suboption) data. It provides consistent access to the data, regardless of what the option definitions might specify in terms of the data type and repeat counts of the data type. It is recommended for general-purpose extensions that decode the data. Use the special values DEX_REPLACE (replace a value), DEX_APPEND (add to end), and DEX_AUGMENT (add if no value currently exists) with `putOption` and `putOptionBytes` methods, which operate the same as the `put`, `putByType`, `putBytes`, and `putBytesByType` methods. Use DEX_REMOVE_ALL for `removeOption` to remove the option completely. |
## Table C-13       DEX iArgType Values (continued)

<table>
<thead>
<tr>
<th>iArgType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEX_INDEX_COUNT</td>
<td>Results in an <strong>int</strong> value returned with the count of the number of indexed entries of the option, rather than the option data. Has no additional argument, and cannot be used with DEX_INDEX or DEX_INSTANCE_COUNT. DEX_END must follow. Valid only for <code>getOption</code> and <code>getOptionBytes</code>.</td>
</tr>
<tr>
<td>DEX_INSTANCE</td>
<td>Requires that <strong>int</strong> follow to specify the instance of the option (valid only for DHCPv6 options, which can have more than one instance). 0 specifies the first instance.</td>
</tr>
<tr>
<td>DEX_INSTANCE_COUNT</td>
<td>Results in an <strong>int</strong> value returned with the count of the number of instances of the option, rather than the option data. Has no additional argument and cannot be used with DEX_INSTANCE. DEX_END must follow. Valid only for <code>getOption</code> and <code>getOptionBytes</code>.</td>
</tr>
<tr>
<td>DEX_MORE</td>
<td>Requires that <strong>bool_t</strong> * follow to specify the location at which a <strong>more</strong> flag is to be written. This location is set to TRUE if more array items exist beyond the index that DEX_INDEX specified. Valid only for <code>getOption</code> and <code>getOptionBytes</code> methods.</td>
</tr>
<tr>
<td>DEX_MOVE_TO</td>
<td>Leaves the context at the option or suboption immediately preceding DEX_MOVE_TO. Has no additional argument. If omitted, the context does not change. Use <code>moveToOption</code> to move the context without getting any data. Valid only for <code>getOption</code> and <code>getOptionBytes</code> methods.</td>
</tr>
<tr>
<td>Note</td>
<td>An attempt to move to an option or suboption that does not exist logs an error. Use <code>moveToOption</code> if your extension did not previously confirm that the option exists.</td>
</tr>
<tr>
<td>DEX_OPTION_NAME</td>
<td>Requires that <strong>const char</strong> * follow to specify the desired option name. Option names should be in the <code>dhcpv4-config</code> or <code>dhcpv6-config</code> option definition set.</td>
</tr>
<tr>
<td>DEX_OPTION_NUMBER</td>
<td>Requires that <strong>int</strong> follow to specify the desired option number. Option numbers should be in the <code>dhcpv4-config</code> or <code>dhcpv6-config</code> option definition set, although there is no requirement that a definition exists. However, if the option does not exist, it is assumed to be a byte blob of data.</td>
</tr>
<tr>
<td>DEX_PARENT</td>
<td>Moves the context to the parent option. Has no additional argument. It does not move beyond the client or relay message and returns FALSE if the context does not change. If used, must be the first <code>iArgType</code>. Valid only for <code>getOption</code>, <code>getOptionBytes</code>, and <code>moveToOption</code> methods.</td>
</tr>
<tr>
<td>DEX_SUBOPTION_NAME</td>
<td>Requires that <strong>const char</strong> * follow to specify the name of the desired suboption. Suboptions must be in the current option definition.</td>
</tr>
<tr>
<td>DEX_SUBOPTION_NUMBER</td>
<td>Requires that <strong>int</strong> follow to specify the desired suboption number. Suboption numbers should be in the current option definition, although there is no requirement that a definition exists. However, if the suboption does not exist, it is assumed to be a byte blob of data.</td>
</tr>
<tr>
<td>DEX_VENDOR_NAME</td>
<td>Requires that <strong>const char</strong> * follow to specify the vendor string. The string serves only to find the appropriate option definition set.</td>
</tr>
</tbody>
</table>
DEX Environment Dictionary Methods

The environment dictionary uses active commands, called methods, with which you can change and access the dictionary values. Table C-14 lists the methods to use with the environment dictionary. In this case, you can define the \texttt{pDict} variable as \texttt{pEnviron}, as in:

\begin{verbatim}
  pEnviron->log( pEnviron, DEX_LOG_INFO, "Environment hello world" );
\end{verbatim}

\begin{table}[h]
\centering
\caption{DEX Environment Dictionary Methods}
\begin{tabular}{|l|l|}
\hline
Method         & Syntax                                                                 \\
\hline
allocateMemory & void *\texttt{pDict}->allocateMemory( dex_EnvironmentDictionary_t *\texttt{pDict}, \noindent\quad unsigned int \texttt{iSize} ) \\
      & Allocates memory for extensions that persists only for the lifetime of this request. \hline
clear          & void \texttt{pDict}->clear( dex_EnvironmentDictionary_t *\texttt{pDict} ) \\
      & Removes all entries from the dictionary. \hline
containsKey    & abool_t \texttt{pDict}->containsKey( dex_EnvironmentDictionary_t *\texttt{pDict}, \noindent\quad const char *\texttt{pszKey} ) \\
      & Returns TRUE if the dictionary contains the key, otherwise FALSE. \hline
firstKey       & const char *\texttt{pDict}->firstKey( dex_EnvironmentDictionary_t *\texttt{pDict} ) \\
      & Returns the name of the first key in the dictionary. Note that the keys are not stored sorted by name. If a key does not exist, returns zero. \hline
get            & const char *\texttt{pDict}->get( dex_EnvironmentDictionary_t *\texttt{pDict}, \noindent\quad const char *\texttt{pszKey} ) \\
      & Returns the value of the key from the dictionary. If a key does not exist, returns the empty string. \hline
isEmpty       & abool_t \texttt{pDict}->isEmpty( dex_EnvironmentDictionary_t *\texttt{pDict} ) \\
      & Returns TRUE if the dictionary has 0 entries, otherwise FALSE. \hline
log           & abool_t \texttt{pDict}->log( dex_EnvironmentDictionary_t *\texttt{pDict}, int \texttt{eLevel}, \noindent\quad const char *\texttt{pszFormat}, ... ) \\
      & Returns a message in the DHCP server logging system. The \texttt{eLevel} should be one of DEX_LOG_ERROR, DEX_LOG_WARNING, or DEX_LOG_INFO. The \texttt{pszFormat} is treated as a printf style format string, and it, along with the remaining arguments, are formatted and sent to the logging system at the specified level. \hline
nextKey       & const char *\texttt{pDict}->nextKey( dex_EnvironmentDictionary_t *\texttt{pDict} ) \\
      & Returns the name of the next key in the dictionary that follows the key returned in the last call to \texttt{firstKey} or \texttt{nextKey}. If a key does not exist, returns zero. \hline
put           & abool_t \texttt{pDict}->put( dex_EnvironmentDictionary_t *\texttt{pDict}, const char *\texttt{pszKey}, \noindent\quad const char* \texttt{pszValue} ) \\
      & Associates a value with the key, replacing an existing instance of the key with the new value. \hline
\end{tabular}
\end{table}
Handling Objects and Options

The following sections describe specialized ways of handling DHCP objects and options in extensions.

Using Object and Option Handling Methods

Extensions can call methods to set DHCP objects, and get, move to, put, and remove DHCP options. The methods are `setObject`, `getOption`, `moveToOption`, `putOption`, and `removeOption` methods in Tcl and C/C++.

These new callback methods were introduced primarily to provide support for DHCPv6. However, you can use the option-related functions for DHCPv4. In fact, it is recommended to use these methods for DHCPv4, because they provide richer access to options than the original `get[Bytes]`, `get[Bytes]ByType`, `put[Bytes]`, `put[Bytes]ByType`, and `remove[ByType]` methods.

See the “DEX Request and Response Dictionary Methods” section on page C-31 for the different usages of some of these methods in C/C++.

For DHCPv6, you must use the `setObject`, `getOption`, `moveToOption`, `putOption`, and `removeOption` methods to access options. The `setObject` method was introduced for DHCPv6, because there can be many leases, prefixes, and messages (client or multiple relay) that an extension might want to access. So, `setObject` serves to set the context for subsequent calls to get request and response dictionary data items and options. When the server calls an extension, the context is set to the current lease (if applicable), prefix (if applicable), and client message. For example, when the server calls the `pre-packet-encode` extension point, only the request and response dictionary message context is valid, and set to the corresponding client message, because there is no lease or prefix associated with this extension point. However, when the server calls the `lease-state-change` extension point, it sets the response dictionary lease context to the lease on which the state has changed, sets the response dictionary prefix context to the prefix for the lease, and sets the request and response dictionary message context to the corresponding client message.

<table>
<thead>
<tr>
<th>Method</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>remove</td>
<td>`abool_t pDict-&gt;remove( dex_EnvironmentDictionary_t *pDict,</td>
</tr>
<tr>
<td></td>
<td>const char *pszKey )</td>
</tr>
<tr>
<td></td>
<td>Removes the key and the associated value from the dictionary. Always</td>
</tr>
<tr>
<td></td>
<td>returns TRUE, even if the dictionary did not contain the key.</td>
</tr>
<tr>
<td>size</td>
<td><code>int pDict-&gt;size( dex_EnvironmentDictionary_t *pDict )</code></td>
</tr>
<tr>
<td></td>
<td>Returns the number of entries in the dictionary.</td>
</tr>
<tr>
<td>trace</td>
<td>`abool_t pDict-&gt;trace( dex_EnvironmentDictionary_t *pDict, int iLevel,</td>
</tr>
<tr>
<td></td>
<td>const char *pszFormat, ... )</td>
</tr>
<tr>
<td></td>
<td>Returns a message in the DHCP server packet tracing system. At level</td>
</tr>
<tr>
<td></td>
<td>0, no tracing occurs. At level 1, it traces only that the server</td>
</tr>
<tr>
<td></td>
<td>received the packet and sent a reply. At level 4, it traces everything.</td>
</tr>
<tr>
<td></td>
<td>The remaining arguments are concatenated and sent to the tracing</td>
</tr>
<tr>
<td></td>
<td>system at the specified level. The default tracing is set using the</td>
</tr>
<tr>
<td></td>
<td>DHCP server <code>extension-trace-level</code> attribute.</td>
</tr>
</tbody>
</table>

Tip

See the “DEX Request and Response Dictionary Methods” section on page C-31 for the different usages of some of these methods in C/C++.
Options and Suboptions in C/C++

Some C/C++ extensions provide specialized argument type values to handle DHCP options and suboptions. The DEX_OPTION_* argument type specifies to use the standard DHCPv4 or DHCPv6 option definition set and not the definitions under an option (or suboption). So, DEX_OPTION_* means that the server looks up the option name or number in the standard DHCPv4 or DHCPv6 option definition set, whereas DEX_SUBOPTION_* means that the server looks up the suboption name or number of the current option definition (if any).

Thus, when you access options in DHCPv6, you often use DEX_OPTION_* followed by DEX_OPTION_* when options are encapsulated. You would use DEX_SUBOPTION when looking at vendor options. For DHCPv4, you would use DEX_OPTION at the client packet level, and then DEX_SUBOPTION perhaps one or more times, depending on the nesting level. Generally, only options have enterprise numbers or vendor names, but there is no prohibition on this. The option definition sets determine what is valid (although one can walk off definitions, at which point everything is treated as binary bytes and thus it limits what is possible, and you cannot use the option or suboption names, but must use numbers).

The option ordering rules for the getOption, moveToOption, putOption, and removeOption methods are similar to the request expression syntax (see Table 26-1 on page 26-6). The ordering generally consists of:

- Preamble clause ([parent | home])
- Option clause (option [vendor | enterprise] [instance])
- Suboption clause (suboption [vendor | enterprise] [instance])
- End clause ([instance-count | index-count | [index] [more] end])

You can construct calls by using a preamble clause, followed by zero or more option clauses, followed by zero or more suboption clauses (which may themselves be followed by option and suboption clauses), followed by an end clause. Note that some things are possible only through a get method (such as instance-count, index-count, and more), and move-to can appear anywhere to move the context to the current option or suboption.

The option definition determines its data format, which can differ from what the older functions return for a specific option. To handle specific options:

- For the vendor class options (v-i-vendor-class [124] for DHCPv4 and vendor-class [16] for DHCPv6), if you ask for a specific instance of the option (instead of by enterprise-id or name), the only way to get the enterprise-id is to ask for the raw data (DEX_INDEX with DEX_RAW).
- For the DHCPv4 vendor options (v-i-vendor-class [124] and v-i-vendor-opts [125]), operating on the raw data (DEX_INDEX with DEX_RAW) only applies to an instance (preset value 0) of that option, not the entire option. There is no way to get the entire data for this option, which means that you cannot use putOption for the entire data. This is not an issue with the DHCPv6 vendor options, because these are separate options.
- If one of the DHCPv4 vendor options (124 or 125) is not formatted properly, the entire data is returned as a blob (if you asked for instance 0 and did not specify a particular enterprise-id). However, if an extension tries to use putOption, depending on the operation, that data might be appended to the existing data, and the result will be formatted incorrectly.
- For the vendor options, if there is no option, putOption( pDict, "01:02", DEX_OPTION_NUMBER, 124, DEX_END ) fails because no enterprise-id is available. However, putOption( pDict, "00:00:00:09:04:03:65:66:67", DEX_OPTION_NUMBER, 124, DEX_END ) will work because it is assumed that 00:00:00:09 is the enterprise-id and the bytes following it starting with 04 are the length of the option data of that enterprise-id. Note that the length byte is
Examples of Option and Object Method Calls

These sections include some examples of how to use methods to handle DHCP option and object data.

Handling Vendor Class Option Data

For DHCPv4, to include the Vendor-Identifying Vendor Class option (124) data for two enterprise-ids in the response to the client, here is some sample Tcl code that uses the putOption method:

```tcl
$response putOption 65:66:67 option 124 enterprise 999998 #adds "abc" (65:66:67) under enterprise-id 999998
$response putOption 68:69:6a:6b option v-i-vendor-class enterprise 999998 index append
#appends "defg" (68:69:6a:6b) under the same enterprise-id
$response putOption 01:02:03:04 option 124 enterprise 999999 #adds 01:02:03:04 under enterprise-id 999999
```

To get the options, use the getOption method:

```tcl
$response getOption option v-i-vendor-class instance-count #returns 2 because there were two instances added (enterprise id 999998 and enterprise id 999999)
$response getOption option 124 #returns index 0 of instance 0, which is 65:66:67
$response getOption option 124 index-count #returns 2 because there were two vendor classes added for the first enterprise id (999999)
$response getOption option 124 index 1 #returns 68:69:6a:6b
$response getOption option 124 instance 1 index-count #returns 1 because there is only one vendor class
$response getOption option 124 instance 1 index raw #returns 00:0f:42:3f:05:04:01:02:03:04 for the complete encoding of the enterprise-id 999999 data (see RFC 3925)
$response getOption option 124 enterprise 999999 #returns 01:02:03:04
```

To remove the data, two removeOption calls are necessary because of the two separate enterprise-ids:

```tcl
$response removeOption option 124
$response removeOption option 124
```

Handling Object Data

Suppose that at the pre-packet-encode extension point you want to extract data for all of the leases for the client. Here is sample Tcl code that uses the setObject method:

```tcl
proc logleasesinit { request response environ } {
    if { [ $environ get "extension-point" ] == "initialize" } {
        # Set up for DHCPv6 only
        $environ put dhcp-support "v6"
        $environ put extension-extension-api-version 2
    }
}
proc logleases { request response environ } {
    for { set i 0 } { 1 } { incr i } {
    
```
# Set context to next lease
if ( ![response setObject lease $i] ) {
    # Lease does not exist, so done
    break
}

# Log the lease address, prefix name, and prefix address
$environ log LOG_INFO "Lease [response get lease-ipaddress], Prefix\n    [response get lease-prefix-name] - [response get prefix-address]"

# Restore the lease context to where we started
$response setObject lease initial
# Do other things...

The C++ equivalent code for this might be:

// Print the current leases for the client
for( int i=0; ; i++ ) {
    if( !pRes->setObject( pRes, DEX_LEASE, DEX_BY_INSTANCE, i ) )
        break;
    const char *pszLeaseAddress =
        pRes->get( pRes, "lease-ipaddress", 0, 0 );
    if( pszLeaseAddress == 0 )
        pszLeaseAddress = "<error>";
    const char *pszPrefixName =
        pRes->get( pRes, "prefix-name", 0, 0 );
    if( pszPrefixName == 0 )
        pszPrefixName = "<error>";
    pEnv->log(pEnv, DEX_LOG_INFO,
        "Lease %s, Prefix %s",
        pszLeaseAddress, pszPrefixName );
}
<table>
<thead>
<tr>
<th>A</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A record</td>
<td>DNS Address resource record (RR). Maps a hostname to its address and specifies the Internet Protocol address (in dotted decimal form) of the host. There should be one A record for each host address.</td>
</tr>
<tr>
<td>access control list (ACL)</td>
<td>DHCP mechanism whereby the server can allow or disallow the request or action defined in a packet. See also transaction signature (TSIG).</td>
</tr>
<tr>
<td>address block</td>
<td>Block of IP addresses to use with DHCP subnet allocation that uses on-demand address pools.</td>
</tr>
<tr>
<td>admin</td>
<td>Default name of the superuser or global administrator.</td>
</tr>
<tr>
<td>administrator</td>
<td>User account to adopt certain functionality, be it defined by role, constrained role, or group.</td>
</tr>
<tr>
<td>alias</td>
<td>Pointer from one domain name to the official (canonical) domain name.</td>
</tr>
<tr>
<td>allocation priority</td>
<td>An alternate method of control over allocating addresses among scopes other than the default round-robin method.</td>
</tr>
<tr>
<td>ARIN</td>
<td>American Registry of Internet Numbers, one of several regional Internet Registries (IRs), manages IP resources in North America, parts of the Caribbean, and subequatorial Africa. Cisco Prime IP Express provides an address space report for this registry.</td>
</tr>
<tr>
<td>Asynchronous Transfer Mode (ATM)</td>
<td>International standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells.</td>
</tr>
<tr>
<td>authoritative name server</td>
<td>DNS name server that possesses complete information about a zone.</td>
</tr>
<tr>
<td>AXFR</td>
<td>Full DNS zone transfer. See also zone transfer and IXFR.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berkeley Internet Name Domain (BIND)</td>
<td>Implementation of the Domain Name System (DNS) protocols. See also DNS.</td>
</tr>
</tbody>
</table>
binding  Collection of DHCP client options and lease information, managed by the main and backup DHCP servers. A binding database is a collection of configuration parameters associated with all DHCP clients. This database holds configuration information about all the datasets.

BOOTP  Bootstrap Protocol. Used by a network node to determine the IP address of its Ethernet interfaces, so that it can affect network booting.

C  
cable modem termination system (CMTS)  Cable modem termination system. Either a router or bridge, typically at the cable head end.

cache  Data stored in indexed disk files to reduce the amount of physical memory.

caching name server  Type of DNS server that caches information learned from other name servers so that it can answer requests quickly, without having to query other servers for each transaction.

canonical name  Another name for an alias DNS host, inherent in a CNAME resource record (RR).

case sensitivity  Values in Cisco Prime IP Express are not case-sensitive, with the exception of passwords.

Central Configuration Management (CCM) database  

chaddr  DHCP client hardware (MAC) address. Sent in an RFC 2131 packet between the client and server.

change logs, changesets  A change log is a group of changesets made to the Cisco Prime IP Express databases due to additions, modifications or deletions in the web UI. A changeset is a set of changes made to a single object in the database.

ciaddr  DHCP client IP address. Sent in an RFC 2131 packet between the client and server.

class of address  Category of an IP address that determines the location of the boundary between network prefix and host suffix. Internet addresses can be A, B, C, D, or E level addresses. Class D addresses are used for multicasting and are not used on hosts. Class E addresses are for experimental use only.

client-class  Cisco Prime IP Express feature that provides differentiated services to users that are connected to a common network. You can thereby group your user community based on administrative criteria, and then ensure that each user receives the appropriate class of service.

cluster  In Cisco Prime IP Express, a group of DNS, and DHCP servers that share the same database.

CNAME record  DNS Canonical Name resource record (RR). Used for nicknames or aliases. The name associated with the resource record is the nickname. The data portion is the official or canonical name.

CNRDB  Name of one of the Cisco Prime IP Express internal databases. The other is changeset database.

constraint  Assigned limitation on the role or allowable functionality of an administrator.
### D

**Data Over Cable Service Interface Specification (DOCSIS)**

Data Over Cable Service Interface Specification. Standard created by cable companies in 1995 to work toward an open cable system standard and that resulted in specifications for connection points, called interfaces.

**delegation**

Act of assigning responsibility for managing a DNS subzone to another server, or of assigning DHCP address blocks to local clusters.

**DHCP**

Dynamic Host Configuration Protocol. Designed by the Internet Engineering Task Force (IETF) to reduce the amount of configuration that is required when using TCP/IP. DHCP allocates IP addresses to hosts. It also provides all the parameters that hosts require to operate and exchange information on the Internet network to which they are attached.

**Digital Subscriber Line (DSL)**

Public network technology that delivers high bandwidth over conventional copper wiring at limited distances.

**DNS**

Domain Name System. Handles the growing number of Internet users. DNS translates names, such as www.cisco.com, into Internet Protocol (IP) addresses, such as 192.168.40.0, so that computers can communicate with each other.

**DNS update**

Protocol (RFC 2136) that integrates DNS with DHCP.

**domain**

Portion of the DNS naming hierarchy tree that refers to general groupings of networks based on organization type or geography. The hierarchy is root, top- or first-level, and second-level domain.

**domain name**

DNS name that can be either absolute or relative. An absolute name is the fully qualified domain name (FQDN) and is terminated with a period. A relative name is relative to the current domain and does not end with a period.

**dotted decimal notation**

Syntactic representation of a 32-bit integer that consists of four eight-bit numbers written in base 10 with dots separating them for a representation of IP addresses. Many TCP/IP application programs accept dotted decimal notation in place of destination machine names.

### E

**expression**

Construct commonly used in the Cisco Prime IP Express DHCP implementation to create client identities or look up clients. For example, an expression can be used to construct a scope from a template.

**extension and extension point**

In Cisco Prime IP Express, element of a script written in TCP, C, or C++ that customizes handling DHCP packets as the server processes them, and which supports additional levels of customizing DHCP clients.
failover  
Cisco Prime IP Express feature (as described in RFC 2131) that provides for multiple, redundant DHCP servers, whereby one server can take over in case of a failure. DHCP clients can continue to keep and renew their leases without needing to know or care which server is responding to their requests.

forwarder  
DNS server designated to handle all offsite queries. Using forwarders relieves other DNS servers from having to send packets offsite.

forwarding, DHCP  
Mechanism of forwarding DHCP packets to another DHCP server on a per-client basis. You can achieve this in Cisco Prime IP Express by using extension scripting.

FQDN  
Fully qualified domain name. Absolute domain name that unambiguously specifies a host location in the DNS hierarchy.

giaddr  
DHCP gateway (relay agent) IP address. Sent in an RFC 2131 packet between the client and server.

glue record  
DNS Address resource record that specifies the address of a subdomain authoritative name server. You only need glue records in the server delegating a domain, not in the domain itself.

group  
Associative entity that combines administrators so that they can be assigned roles and constrained roles.

High-Availability (HA) DNS  
DNS configuration in which a second primary server can be made available as a hot standby that shadows the main primary server.

HINFO record  
DNS Host Information resource record (RR). Provides information about the hardware and software of the host machine.

hint server  
See root hint server.

host  
Any network device with a TCP/IP network address.

IEEE  
Institute of Electrical and Electronics Engineers. Professional organization whose activities include developing communications and network standards.

in-addr.arpa  
DNS address mapping domain with which you can index host addresses and names. The Internet can thereby convert IP addresses back to hostnames. See also reverse zone.

IP address  
Internet Protocol address. For example, 192.168.40.123.

IP history  
Cisco Prime IP Express tool that records the lease history of IP addresses in a database.
IPv6  New IP standard involving 128-bit addresses. Cisco Prime IP Express provides a DHCPv6 implementation.

ISP  Internet Service Provider. Company that provides leased line, dialup, and DSL (Point-to-Point over Ethernet and DHCP) access to customers.

iterative query  Type of DNS query whereby the name server returns the closest answer to the querying server.

IXFR  Incremental zone transfer. Standard that allows Cisco Prime IP Express to update a slave (secondary) server by transferring only the changed data from the primary server.

lame delegation  Condition when DNS servers listed in a zone are not configured to be authoritative for the zone.


lease  IP address assignment to a DHCP client that also specifies how long the client can use the address. When the lease expires, the client must negotiate a new one with the DHCP server.

lease grace period  Length of time the lease is retained in the DHCP server database after it expires. This protects a client lease in case the client and server are in different time zones, their clocks are not synchronized, or the client is not on the network when the lease expires.

link group  Groups the links to accommodate CMTS Prefix Stability. The group-name attribute is used to specify the name of the group to which the link should belong.

lease history  A report that can be generated to provide a historical view of when a client was issued a lease, for how long, when the client or server released the lease before it expired, and if and when the server renewed the lease and for how long.

lease query  Process by which a relay agent can request lease (and reservation) data directly from a DHCP server in addition to gleaning it from client/server transactions.

link type  There are three different link types: topological, location-independent, and universal. Topological links means a client is allocated leases based on the network segment it is connected to. While the location-independent link type lets a subscriber, that is moved from one CMTS to another within a central office, to retain a delegated prefix, the universal link type lets the subscriber moving from one central office to another to retain the delegated prefix.

local cluster  Location of the local Cisco Prime IP Express servers. See also regional cluster.

localhost  Distinguished name referring to the name of the current machine. Localhost is useful for applications requiring a hostname.

loopback zone  DNS zone that enables the server to direct traffic to itself. The host number is almost always 127.0.0.1.
### Glossary

#### M

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MAC address</strong></td>
<td>Standardized data link layer address. Required for every port or device that connects to a LAN. Other devices in the network use these addresses to locate specific ports on the network and to create and update routing tables and data structures. MAC addresses are six bytes long and are controlled by the IEEE. Also known as a hardware address, MAC layer address, and physical address. A typical MAC address is 1,6,00:d0:ba:d3:bd:3b.</td>
</tr>
<tr>
<td><strong>mail exchanger</strong></td>
<td>Host that accepts electronic mail, some of which act as mail forwarders. See also MX record.</td>
</tr>
<tr>
<td><strong>master name server</strong></td>
<td>Authoritative DNS name server that transfers zone data to secondary servers through zone transfers.</td>
</tr>
<tr>
<td><strong>maximum client lead time (MCLT)</strong></td>
<td>In DHCP failover, a type of lease insurance that controls how much ahead of the backup server lease expiration the client lease expiration should be.</td>
</tr>
<tr>
<td><strong>multinetting</strong></td>
<td>State of having multiple DHCP scopes on one subnet or several LAN segments.</td>
</tr>
<tr>
<td><strong>Multiple Service Operator (MSO)</strong></td>
<td>Provides subscribers Internet access using cable or wireless technologies.</td>
</tr>
<tr>
<td><strong>multithreading</strong></td>
<td>Process of performing multiple server tasks.</td>
</tr>
<tr>
<td><strong>MX record</strong></td>
<td>DNS Mail Exchanger resource record (RR). Specifies where mail for a domain name should be delivered. You can have multiple MX records for a single domain name, ranked in preference order.</td>
</tr>
</tbody>
</table>

#### N

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nameserver</strong></td>
<td>DNS host that stores data and RRs for a domain.</td>
</tr>
<tr>
<td><strong>NAPTR</strong></td>
<td>DNS Naming Authority Pointer resource record (RR). Helps with name resolution in a particular namespace and is processed to get to a resolution service. Based on proposed standard RFC 2915.</td>
</tr>
<tr>
<td><strong>negative cache time</strong></td>
<td>Memory cache the DNS server maintains for a quick response to repeated requests for negative information, such as “no such name” or “no such data.” Cisco Prime IP Express discards this information at intervals.</td>
</tr>
<tr>
<td><strong>network ID</strong></td>
<td>Portion of the 32-bit IP address that identifies which network a particular system is on, determined by performing an AND operation of the subnet mask and the IP address.</td>
</tr>
<tr>
<td><strong>NOTIFY</strong></td>
<td>Standard (RFC 1996) whereby DNS master servers can inform their slaves that changes were made to their zones, and which initiates a zone transfer.</td>
</tr>
<tr>
<td><strong>nrcmd</strong></td>
<td>Cisco Prime IP Express command line interface (CLI).</td>
</tr>
</tbody>
</table>
## Glossary

### O

**on-demand address pool**  Wholesale IP address pool issued to a client (usually a VPN router or other provisioning device), from which it can draw for lease assignments. Also known as DHCP subnet allocation.

**option, DHCP**  DHCP configuration parameter and other control information stored in the options field of a DHCP message. DHCP clients determine what options get requested and sent in a DHCP packet. Cisco Prime IP Express allows for creating option definitions as well as the option sets to which they belong.

**Organization report**  One of the reports to be submitted to ARIN, POC being the other report. See also ARIN and POC report.

**Organizationally Unique Identifier (OUI)**  Assigned by the IEEE to identify the owner or ISP of a VPN. See also IEEE and virtual private network (VPN).

**owner**  Owners can be created as distinguishing factors for address blocks, subnets, and zones. In the context or DNS RRs, an owner is the name of the RR.

### P

**ping**  Packet Internetwork Groper. A common method for troubleshooting device accessibility that uses a series of Internet Control Message Protocol (ICMP) Echo messages to determine if a remote host is active or inactive, and the round-trip delay in communicating with the host.

**POC report**  Point of Contact report. One of the reports to be submitted to ARIN, Organization being the other report. See also ARIN and Organization report.

**policy**  Group of DHCP attributes or options applied to a single scope or group of scopes. Embedded policies can be created for scopes and other DHCP objects.

**polling**  Collection of subnet utilization or lease history data over a certain regular period.

**prefix allocation groups**  Groups prefixes in order to facilitate the prioritization of prefix allocation.

**prefix stability**  Clients can retain the delegated prefix when they change their location, that is even when they move from one CMTS to another (CMTS Prefix Stability) or move within an address space (Universal Prefix Stability).

**primary master**  DNS server from which a secondary server receive data through a zone transfer request.

**provisional address**  Address allocated by the DHCP server to unknown clients for a short time, one-shot basis.

**PTR record**  DNS Pointer resource record. Used to enable special names to point to some other location in the domain tree. Should refer to official (canonical) names and not aliases. See also in-addr.arpa.

**pulling and pushing objects**  The Cisco Prime IP Express regional cluster provides functions to pull network objects from the replica database of local cluster data, and push objects directly to the local clusters.
**recursive query**
DNS query where the name server asks other DNS server for any nonauthoritative data not in its own cache. Recursive queries continue to query all name servers until receiving an answer or an error.

**refresh interval**
Time interval in which a secondary DNS server checks the accuracy of its data by sending an AXFR packet to the primary server.

**region**
Regions can be created as distinguishing factors for address blocks, subnets, and zones. A region is distinct from the regional cluster.

**regional cluster**
Location of the regional Cisco Prime IP Express CCM server. See also local cluster.

**relay agent**
Device that connects two or more networks or network systems. In DHCP, a router on a virtual private network that is the IP helper for the DHCP server.

**replica database**
CCM database that captures copies of local cluster configurations at the regional cluster. These configurations can be pulled to the regional cluster so that they can be pushed to other local clusters.

**Request for Comments (RFC)**
TCP/IP set of standards.

**reservation**
IP address or lease that is reserved for a specific DHCP client.

**resolution exception**
Selectively forwarding DNS queries for specified domains to internal servers rather than recursively querying Internet root name and external servers.

**resolver**
Client part of the DNS client/server mechanism. A resolver creates queries sent across a network to a name server, interprets responses, and returns information to the requesting programs.

**resource record (RR)**
DNS configuration record, such as SOA, NS, A, CNAME, HINFO, WKS, MX, and PTR that comprises the data within a DNS zone. Mostly abbreviated as RR. See Appendix A, “Resource Records.”

**reverse zone**
DNS zone that uses names as addresses to support address queries. See also in-addr.arpa.

**role, constrained role**
Administrators can be assigned one or more roles to determine what functionality they have in the application. A constrained role is a role constrained by further limitations. There are general roles for DNS, host, address block, DHCP, and CCM database administration. You can further constrain roles for specific hosts and zones. Some roles have distinguishing subroles, such as the database subrole.

**root hint server**
DNS name server at the top of the hierarchy for all root name queries. A root name server knows the addresses of the authoritative name servers for all the top-level domains. Resolution of nonauthoritative or uncached data must start at the root servers. Sometimes called a hint server.

**round-robin**
Action when a DNS server rearranges the order of its multiple same-type records each time it is queried.

**routed bridge encapsulation (RBE)**
Process by which a stub-bridged segment is terminated on a point-to-point routed interface. Specifically, the router is routing on an IEEE 802.3 or Ethernet header carried over a point-to-point protocol, such as PPP, RFC 1483 ATM, or RFC 1490 Frame Relay.
scavenging  Action of periodically scanning dynamic updates to the DNS server for stale resource records and purging these records.

scope  Administrative grouping of TCP/IP addresses on a DHCP server. Required for lease assignments.

secondary master  DNS name server that gets its zone data from another name server authoritative for the zone. When a secondary master server starts up, it contacts the primary master, from which it receives updates.

secondary subnet  A single LAN might have more than one subnet number applicable to the same LAN or network segment in a router. Typically, one subnet is designated as primary, the others as secondary. A site might support addresses on more than one subnet number associated with a single interface. You must configure the DHCP server with the necessary information about your secondary subnets.

selection tags  Mechanisms that help select DHCPv4 scopes and DHCPv6 prefixes for clients and client-classes.

siaddr  IP address of the server to use in the next step of the DHCP boot process. Sent in an RFC 2131 packet between the client and server.

slave forwarder  DNS server that behaves like a stub resolver and passes most queries on to another name server for resolution. See also stub resolver.

slave servers  DNS server that always forwards queries it cannot answer from its cache to a fixed list of forwarding servers instead of querying the root name servers for answers.

SNMP notification  Simple Network Management Protocol messages that warn of server error conditions and problems. See also trap.

SOA record  DNS Start of Authority resource record (RR). Designates the start of a zone.

SRV record  Type of DNS resource record (RR) that allows administrators to use several servers for a single host domain, to move services from host to host with little difficulty, and to designate some hosts as primary servers for a service and others as backups.

staged edit mode  dhcp or dns edit mode in which the data is stored on the CCM server, but not live on the protocol server. See also synchronous edit mode.

stub resolver  DNS server that hands off queries to another server instead of performing the full resolution itself.

subnet allocation, DHCP  Cisco Prime IP Express use of on-demand address pools for entire subnet allocation of IP addresses to provisioning devices.

subnet mask  Separate IP address, or part of a host IP address, that determines the host address subnet. For example, 192.168.40.0 255.255.255.0 (or 192.168.40.0/24) indicates that the first 24 bits of the IP address are its subnet, 192.168.40. In this way, addresses do not need to be divided strictly along network class lines.

subnet pool  Set of IP addresses associated with a network number and subnet mask, including secondary subnets.

subnet sorting  Attribute of the Cisco Prime IP Express DNS server. By enabling it, the server checks the network address of the client before responding to a query.

subnet utilization  A report that can be generated to determine how many addresses in the subnet were allocated and what the free address space is.
**Glossary**

**subnetting**  
Action of dividing any network class into multiple subnetworks.

**subscriber limitation**  
Limitation to the number of addresses service providers can determine for the DHCP server to give out to devices on customer premises, handled in Cisco Prime IP Express by DHCP option 82 definitions.

**subzone**  
Partition of a delegated domain, represented as a child of the parent node. A subzone always ends with the name of its parent. For example, boston.example.com. can be a subzone of example.com.

**subzone delegation**  
Dividing a zone into subzones. You can delegate administrative authority for these subzones, and have them managed by people within those zones or served by separate servers.

**supernet**  
Aggregation of IP network addresses advertised as a single classless network address.

**synchronization**  
Synchronization can occur between the regional cluster and local clusters, the CCM and other protocol servers, failover servers, HA DNS servers, and routers.

**synchronous edit mode**  
dhcp or dns edit mode in which the data is live on the protocol server. See also staged edit mode.

**T**

**TAC**  
Cisco Technical Assistance Center. Cisco Prime IP Express provide a cnr_tactool utility to use in reporting issues to the TAC.

**TCP/IP**  
Suite of data communication protocols. Its name comes from two of the more important protocols in the suite: the Transmission Control Protocol (TCP) and the Internet Protocol (IP). It forms the basis of Internet traffic.

**template**  
DNS zones and DHCP scopes can have templates to create multiple objects with similar properties.

**transaction signature (TSIG)**  
DHCP mechanism that ensures that DNS messages come from a trusted source and are not tampered with. See also access control list (ACL).

**trap**  
Criteria set to detect certain SNMP events, such as to determine free addresses on the network. See also SNMP notification.

**trimming and compacting**  
Trimming is periodic elimination of old historical data to regulate the size of log and other files. Compacting is reducing data older than a certain age to subsets of the records.

**U**

**Universal Time (UT)**  
International standard time reference that was formerly called Greenwich Mean Time (GMT), also called Universal Coordinated Time (UCT).

**update configuration, DNS**  
Defines the relationship of a zone with its main and backup DNS servers for DNS update purposes.

**update map, DNS**  
Defines an update relationship between a DHCP policy and a list of DNS zones.
U

update policy, DNS  Provide a mechanism in DHCP for managing update authorization at the DNS RR level.

User Datagram Protocol (UDP)  Connectionless TCP/IP transport layer protocol.

V

virtual channel identifier (VCI) and virtual path identifier (VPI)  16-bit field in the header of an ATM cell. The VCI, together with the VPI, identifies the next destination of a cell as it passes through a series of ATM switches on its way to its destination. ATM switches use the VPI/VCI fields to identify the next network VCL that a cell needs to transit on its way to its final destination. The function of the VCI is similar to that of the DLCI in Frame Relay.

virtual private network (VPN)  Protocol over which IP traffic of private address space can travel securely over a public TCP/IP network. A VPN uses tunneling to encrypt all information at the IP level. See also VRF.

VRF  VPN Routing and Forwarding instance. Routing table and forwarding information base table, populated by routing protocol contexts. See also virtual private network (VPN).

W

well-known port  Any set of IP protocol port numbers preassigned for specific uses by transport level protocols, for example, TCP and UDP. Each server listens at a well-known port so clients can locate it.

WKS record  DNS Well Known Service resource record (RR). Used to list the services provided by the hosts in a zone. Common protocols are TCP and UDP.

Y

yiaddr  “Your” client IP address, or address that the DHCP server offers (and ultimately assigns) the client. Sent in an RFC 2131 packet between the client and server.

Z

zone  Delegation point in the DNS tree hierarchy that contains all the names from a certain point downward, except for those names that were delegated to other zones. A zone defines the contents of a contiguous section of the domain space, usually bounded by administrative boundaries. Each zone has configuration data composed of entries called resource records. A zone can map exactly to a single domain, but can also include only part of a domain, with the remainder delegated to another subzone.

zone distribution  Configuration that simplifies creating multiple zones that share the same secondary zone attributes. The zone distribution requires adding one or more predefined secondary servers.
zone of authority  Group of DNS domains for which a given name server is an authority.

zone transfer  Action that occurs when a secondary DNS server starts up and updates itself from the primary server. A secondary DNS server queries a primary name server with a specific packet type called AXFR (transfer all) or IXFR (incrementally transfer) and initiates a transfer of a copy of the database.
A

A6 records  A-2
AAAA records  A-2
  DNS update  29-2
About EDNS0  13-8
absolute domain names
  See FQDN
access control lists (ACLs)
  creating  29-9
acl command (CLI)
  create  29-10
ACLs
  See access control lists (ACLs)
addrblock-admin role  5-3
  core functionality  5-3
  ipv6-management subrole  5-3
address allocation
  attributes  21-14
  in scopes  21-15
  priority  21-12
  round-robin  21-11
address-block-policy command (CLI)
  delete  24-23
  get  24-23
  getOption  24-23
  listOptions  24-23
  listVendorOptions  24-23
  setVendorOption  24-23
  show  24-23
  unset  24-23
  unsetOption  24-23
  unsetVendorOption  24-23
address blocks  9-3
  adding  9-5
  administrator role  9-1
  children  9-9
  delegating  9-6
  embedded policies  24-23
  viewing  9-4
  when to add  9-5
address blocks, DHCP  20-5
  creating  24-22
  default subnet size  24-23
  orphaned leases  24-24
  policies, associating  24-22
addresses
  dynamic  9-1
  IP format  A-4
  IPv6  27-2
  provisional  25-12
  static  9-1
  usage, displaying  7-17
address infrastructure, creating  5-27
address pools, on-demand
  See subnet allocation, DHCP
address ranges
  adding  5-27
  scopes  21-11
  subnets  9-10
Address records
  See A records
address restrictions, zones  5-30
address space
  delegated  9-3, 9-4
  local, pulling from  5-36
managing 9-1
unified 9-2
address usage reports
displaying 7-17
running 23-21
addr-trap command (CLI)
create 1-8, 21-22
set
high-threshold 21-22
low-threshold 21-22
admin command (CLI)
enterPassword 5-6
set password 5-6
administrators 5-1, A-1
adding 5-5
centrally managing 5-17
local cluster 5-26
passwords
adding 5-5
changing 5-6
managing 5-6
pulling replica 5-19
pushing to local 5-18
regional 5-33
relationship to groups 5-2
types 5-2
admin role A-1
AFSDB records A-2
agent_server_log file 7-6
allocation priority, scopes 21-12
algorithm 21-12
limitation-id and 21-13
all-subnets-local, DHCP option B-4
and, DHCP expression 26-7
A records A-1
adding 16-2
arithmetic functions, DHCP expressions 26-6
arp-cache-timeout, DHCP option B-5
as-blob, DHCP expression 26-7
ash, DHCP expression 26-8
as-sint, DHCP expression 26-7
associated-ip, DHCP option B-18
as-string, DHCP expression 26-7
as-uint, DHCP expression 26-7
Asynchronous Transfer Mode (ATM) 26-24, A-1
AT_BLOB option validation B-26
AT_BOOL option validation B-26
AT_DATE option validation B-26
AT_DNSNAME option validation B-26
AT_INT8 option validation B-26, B-27
AT_INTI option validation B-26
AT_INT option validation B-26
AT_IP6ADDR option validation B-26
AT_IPADDR option validation B-26
AT_MACADDR option validation B-26
AT_MESSAGE option validation B-26
AT_NOLEN option validation B-26
AT_NSTRING option validation B-26
AT_OVERLOAD option validation B-26
AT_RANGEBYTE option validation B-26
AT_RANGESHORT option validation B-26
AT_RDNSNAME option validation B-26
AT_SHORT option validation B-26
AT_SHRTI option validation B-26
AT_SINTI option validation B-27
AT_SINT option validation B-27
AT_SSHORT option validation B-27
AT_SSHRTI option validation B-27
AT_STIME option validation B-27
AT_STRING option validation B-27
AT_TIME option validation B-27
AT_TYPECNT option validation B-27
AT_VENDOR_CLASS option validation B-27
AT_VENDOR_OPTS option validation B-27
AT_ZEROSIZE option validation B-27
attributes
displaying 2-5
Help window 2-6
modifying 2-5
visibility 2-5
auth, DHCPv6 option B-12
authentication, DHCP option B-18
authentication subrole (ccm-admin) 5-3
authentication subrole (regional-admin) 5-4
authoritative name servers 13-4, A-1
  adding 14-6
authorization subrole (ccm-admin) 5-3
authorization subrole (regional-admin) 5-4
auto-configure, DHCP option B-19

B
base roles 5-1
bcms-server-a, DHCPv6 option B-13
bcms-server-d, DHCPv6 option B-13
bcms-servers-a, DHCP option B-10
bcms-servers-d, DHCP option B-10
BIND files
  format 14-9
  importing 14-9
bit-and, DHCP expression 26-8
bit-andc1, DHCP expression 26-8
bit-andc2, DHCP expression 26-8
bit-eqv, DHCP expression 26-8
bit-not, DHCP expression 26-8
bit-or, DHCP expression 26-8
bit-orc1, DHCP expression 26-8
bit-orc2, DHCP expression 26-8
bit-xor, DHCP expression 26-8
boot-file, DHCP option B-10
BOOTP
  clients, moving/decommissioning 21-20
  configuring 24-1
    requirements 24-2
dynamic 28-31
  enabling 24-3
    PARTNER-DOWN state 28-32
  scopes, enabling for 21-20
  enabling, disabling 24-2
failover 28-31
file reply packet field 24-2
relay 24-3
  failover 28-8
routers 24-3
scopes, enabling for 21-20
siaddr reply packet field 24-2
sname reply packet field 24-2
static 28-31
boot-size, DHCP option B-2
broadcast-address, DHCP option B-4
byte, DHCP expression 26-8

C
C/C++
  API 30-9
    extensions 30-4, 30-9
cablelabs-125, DHCP option C-2
cablelabs-17, DHCPv6 option C-7
cablelabs-client-configuration, DHCP option B-19
  cache
    flushing DNS 18-10
  cache, refreshing session 2-4
  caching-only servers 13-5
  case-sensitivity of values A-2
catalina.date.log file 7-6
ccm_startup_log file 7-6
ccm_upgrade_status_log file 7-6
ccm-admin role 5-3
  authentication subrole 5-3
  authorization subrole 5-3
  core functionality 5-3
database subrole 5-3
  owner-region subrole 5-3
  server-management subrole 5-4
ccm command (CLI)
Index

polling attribute, setting 6-14
set 7-3
CCM database A-2
files 7-6
logging 7-6
ccm-management subrole (cfg-admin) 5-3
CCM server 1-1, 4-1
polling attributes 6-14
cdns_log file 7-6
cdns_startup_log file 7-6
cdns64 command (CLI)
create 18-5
enable 18-5
cdns command (CLI)
addException 18-4
addRootHint 18-8
listExceptions 18-4
removeException 18-4
resetStats 7-12
set
cache-max-ttl 18-8
cache-min-ttl 18-8
msg-cache-size 18-9
rrset-cache-size 18-9
CDNS domain redirect
add 18-13
overview 18-12
reorder 18-14
CDNS domain redirect command (CLI)
create 18-13
delete 18-14
list 18-14
cdns-management subrole (cfg-admin) 5-3
cdnssec command (CLI)
create 18-5
enable 18-5
central-cfg-admin role 5-4
core functionality 5-4
dhcp-management subrole 5-4
central configuration 6-1
Central Configuration Management (CCM) server
See CCM server
central-dns-admin role 5-4
core functionality 5-4
security-management subrole 5-4
server-management subrole 5-4
central-host-admin role 5-4
core functionality 5-4
cfg-admin role 5-3
ccm-management subrole 5-3
core functionality 5-3
dhcp-management subrole 5-3
dns-management subrole 5-3
snmp-management subrole 5-3
chaddr, DHCP field A-2
check-lease-acceptable extension point, DHCP 24-12, 30-37
checkports_log file 7-6
child
address blocks 9-9
subnets 9-9
ciaddr, DHCP field 24-19, A-2
cisco-auto-configure, DHCP option B-20
cisco-client-last-transaction-time, DHCP option B-20
cisco-client-requested-host-name, DHCP option B-20
cisco-leased-ip, DHCP option B-20
Cisco routers 24-3
cisco-vpn-id, DHCP option B-20
classless-static-route, DHCP option B-19
class of service 20-9
CLI 1-1, 2-10
command syntax 2-10
client-class command (CLI)
create 25-4
delete 25-4
list 25-4
listnames 25-4
set 25-4
add-to-environment-dictionary 23-17

default-vpn 24-21

host-name 25-5

limitation-id 25-15

limitation-key 26-1

over-limit-client-class-name 25-15

override-client-id 23-18

override-vpn 24-21

selection-criteria 25-4

show 25-4

client-classes

client entries, skipping 25-13

configuring 25-1

defining 25-2

DHCPv6 27-33

editing 25-5

embedded policies 25-5

enabling 20-11

failover synchronization effect 28-7

host-name setting 25-5

local, pulling 6-18

local, pushing 6-18

lookup ID 23-18

process 25-2

processing order to determine 25-7

quality of service 20-9

RADIUS pool name, mapping 25-7

regional 6-17

troubleshooting 25-8

user class identifier, mapping 25-7

client-class-policy command (CLI)

set 25-6

setLeaseTime 25-6

setOption 25-6

setV6Option 25-6

setV6VendorOption 25-6

setVendorOption 25-6

show 25-6

create 25-10

delete 25-10

list 25-10

listnames 25-10

set 25-10

action=one-shot 25-12

authenticate-until 25-13

client-class-name 25-10

client-lookup-id 26-1

default-vpn 24-21

limitation-id 25-15

limitation-key 26-1

over-limit-client-class-name 25-15

override-vpn 24-21

policy-name 25-12

selection-criteria 25-10

show 25-10

client-data, DHCPv6 option B-14

client-fqdn, DHCP option B-18

client-fqdn, DHCPv6 option B-13

client ID

overriding 23-18

reservations, using for lease 23-18

client-identifier, DHCP option B-10

client-identifier, DHCPv6 option B-11

client-last-transaction-time, DHCP option B-18

client-policy command (CLI)

set 25-11

setLeaseTime 25-11

setOption 25-11

setV6Option 25-11

setV6VendorOption 25-11

setVendorOption 25-11

show 25-11

client reservations

using 23-10

clients

authentication, limiting 25-13

caching parameters 25-13
configuring 25-9
default 25-9
DHCPv6 27-7, 27-34
editing 25-10
embedded policies 25-10
failover synchronization effect 28-6
hardware address A-2
IP address 24-19, A-2
lease request name 20-10
listing 25-10
properties 25-10
provisioning 25-14
subscribers, set by limitation-id 25-15
unknown 25-12
Windows 2000 25-11
your IP address 24-19, A-11
clt-time, DHCPv6 option B-14
cluster command (CLI)
create 6-3
polling attributes, setting 6-13
set 6-3
clusters
activating 6-12
data, recovering 6-12
deactivating 6-12
local 4-1
polling attributes 6-13
poll-replica-interval 6-11
poll-replica-offset 6-11
regional 4-1
secure connections 6-2
time skew effects 6-13
CNAME records A-3, A-2
cnr_exim utility 8-10
cnr_keygen utility 29-11
cnr_shadow_backup utility 8-3
using third party programs 8-3
cnr_tactool utility 7-26
cnrdb_checkpoint utility 8-13
cnrdb_recover utility 8-11
cnrdb_verify utility 8-12
CNRDB database A-12
backing up 8-5
damaged 8-6
files 8-5
log files 8-5
recovering 8-7
cnrsnmp_log file 7-6
cnrwebui_access_log.date.txt file 7-6
cnrwebui_log file 7-6
.com domain 13-2
comment, DHCP expression 26-8
COMMUNICATIONS-INTERRUPTED, failover 28-16
concat, DHCP expression 25-9
concepts and protocols 1-1
config_ccm_log file 7-6
configuration
file, lease-notification 23-28
guidelines 4-5
special cases 4-6
configuring client-classes
See client-classes
configuring DHCP servers
scopes, See scopes
configuring DHCP servers, See DHCP
configuring DNS servers
loopback zones, See loopback, zones
NOTIFY, See NOTIFY
secondary, See secondary, name servers
configuring DNS update
See DNS update
configuring DNS update maps
See DNS update, maps
configuring failover
See failover, DHCP
configuring policies
See policies
configuring virtual private networks
See VPNs

consistency rules
  adding 7-21
  listing 7-21
  viewing 7-21
constrained roles 5-2, A-8
cookie-servers, DHCP option B-2
create-prefix, link template expression 27-17
create-prefix-addr
  link template expression 27-17
  prefix template expression 27-24
create-prefix-range
  link template expression 27-17
  prefix template expression 27-24
create-v6-option
  link template expression 27-18, 27-25
cron task (UNIX) 23-27

dashboard 3-1
  Caching DNS general indicators chart 3-22
  DHCP address utilization chart 3-14
  DHCP buffer capacity chart 3-12
  DHCP DNS update activity chart 3-13
  DHCP failover status chart 3-15
  DHCP general indicators chart 3-15
  DHCP response latency chart 3-13
  DHCP server request activity 3-11
  DHCP server response activity 3-11
  DNS caching chart 3-22
  DNS general indicators chart 3-19
  DNS inbound zone transfers chart 3-17
  DNS incoming queries chart 3-21
  DNS network errors chart 3-18
  DNS outbound zone transfers chart 3-16
  DNS queries responses chart 3-20
  DNS queries type chart 3-20
  DNS related servers errors chart 3-18
  system metrics chart 3-9

databases
  backup 8-1
    strategies 8-2
  third party programs 8-3
  binding A-2
  CNRDB 8-2, A-2
  exporting 8-10
  importing 8-10
  log files 7-6
  replica 6-11
  startup, loading on 7-1
  database subrole (ccm-admin) 5-3
  database subrole (regional-admin) 5-4
  data directory, changing 8-2
  datatype, DHCP expression 26-9
  default-ip-ttl, DHCP option B-3
  default-tcp-ttl, DHCP option B-5
  delegated address space 9-3, 9-4
  delegation-only zones 17-3
  deployment cases
    large enterprise network 4-3, 29-2
    small to medium size LANs 4-3
  dex.h file 30-10
  dex extensions 30-9
  DHCID records A-3
  DNS update 29-2
  DHCP
    administration 20-2
    buffers, allocating 24-5
    clients
      client lease request name 20-10
      IP addresses 24-19, A-2
      MAC addresses 20-10, A-2
      your IP address 24-19, A-11
    client-server model 20-1
    configuration guidelines 4-5
    custom options 22-10
    custom options, adding 22-10
dashboard
- address utilization chart 3-14
- buffer capacity chart 3-12
- DHCP failover status chart 3-15
- DNS update activity chart 3-13
- general indicators chart 3-15
- response latency chart 3-13
- server request activity chart 3-11
- server response activity chart 3-11

defer lease extensions option 24-6

equal-priority-most-available ethernet addresses 21-13, 21-15

forwarding 24-24

last transaction time granularity 24-7

lease history collection 6-14

leasequery, See leasequery

lease-state updates to LDAP 25-25

lease times 21-2

library path 24-10

log settings 24-15

option 82 25-14

options 22-5

  RFC 2132 B-1
    syntax for more complex validation B-15

ping packets 24-6

policies, See policies

priority-address-allocation 21-15

related servers, displaying 7-18

related servers, getting 6-10

remote WAN configuration request

  buffers 28-13
    processing 20-10

requests, number of 24-5

requests for other servers, ignoring 23-20

responses, number of 24-6

reverse zones, synthesizing 29-4

sample users 20-2

scopes disabling for servers 21-20

configuring 24-1

forwarding, switching 24-24

interface, removing address 21-2

interface, setting 21-2

IP address of next DHCP A-9

logging 7-6, 29-31

troubleshooting 24-15

SMS network discovery records 24-10

switching 24-24

unicast, enabling 24-6

v6, See DHCPv6
dhcp_startup_log file 7-6
dhcp-address-block command (CLI)

  set
default-subnet-size 24-23

  vpn 24-21

  vpn-id 24-21

  unset 24-23
dhcp-admin role 5-4

core functionality 5-4

ipv6-management subrole 5-4
dhcp-client-identifier, DHCP option B-10
dhcp command (CLI)

  attachExtension 24-24, 30-2

  detachExtension 24-24, 30-2

  disable

    vpn-communication 24-24

  enable

    client-class 25-4

    collect-sample-counters 7-12

    defer-lease-extensions 24-6

    delete-orphaned-leases 24-23

    delete-orphaned-subnets 24-24

    get-subnet-mask-from-policy 22-5

    hardware-unicast 24-6

    ignore-requests-for-other-servers 23-20

ip-history 23-22
Index

GET

return-client-fqdn-if-asked 29-24
save-lease-renewal-time 25-24
skip-client-lookup 25-13
update-dns-for-bootp 24-3
use-client-fqdn 29-24
use-client-fqdn-first 29-24
use-ldap-client-data 25-21
get 24-8
getPrefixCount 27-29
getRelatedServers 6-10, 7-20
getScopeCount 21-11
getStats 7-16
lease history collection attributes 6-14
limitationList 7-3, 25-18
resetStats 7-12
set 24-8
  activity-summary-interval 7-12, 24-16
  client-cache-count 25-14
  client-cache-ttl 25-14
  client-class-lookup-id 23-18, 25-15, 26-1
  default-free-address-config 1-8, 3-14
  failover-recover 28-19
  last-transaction-time-granularity 24-7
  ldap-mode 25-22
  log-settings 7-6, 24-15, 29-21
  map-radius-class 25-7
  max-dhcp-requests 24-5, 28-13
  max-dhcp-responses 24-6
  max-ping-packets 24-6
  max-waiting-packets 24-16
  sms-lease-interval 24-10
  sms-library-path 24-10
  sms-network-discovery 24-10
  sms-site-code 24-10
  synthesize-reverse-zone 29-6
  traps-enabled 1-8
  username 25-20
  v6-default-free-address-config 1-8, 3-14
  vpn-communication 24-22
  setPartnerDown 28-12
  show 24-8
  start 7-3
  stop 7-3
  unset 24-8
  updateSms 24-9, 24-10

DHCPDISCOVER packets 25-14
dhcp-dns-update command (CLI) 29-7
  create 29-7
  enable
    update-dns-first 29-7
    update-dns-for-bootp 29-7
  set
    backup-server-addr 29-6
    backup-server-key 29-12
    dynamic-dns 29-7
    forward-zone-name 29-6
    reverse-zone-name 29-6
    reverse-zone-prefix-length 29-4
    server-addr 29-6
    server-key 29-12
    synthesize-name 29-5
    synthetic-name-stem 29-5
    v6-synthetic-name-generator 29-3
DHCP forwarding, enabling 24-24
dhcp-interface command (CLI) 21-2
DHCPPLEASEQUERY packets
  See leasequery
dhcp-lease-time, DHCP option B-8
dhcp-management subrole (central-cfg-admin) 5-4
dhcp-management subrole (cfg-admin) 5-3
dhcp-management subrole (regional-addr-admin) 5-5
dhcp-max-message-size, DHCP option B-9
dhcp-message, DHCP option B-9
dhcp-message-type, DHCP option B-8
dhcp-option-overload, DHCP option B-8
dhcp-parameter-request-list, DHCP option B-9
dhcp-rebinding-time, DHCP option B-9
dhcp-renewal-time, DHCP option B-9
DHCPRENEW packets 25-17
dhcp-requested-address, DHCP option B-8
DHCPREQUEST packets 21-2, B-8
dhcp-server-identifier, DHCP option B-9
dhcp-user-class-id, DHCP option B-18
DHCPv4 DNS update
   DHCID RR 29-21
   regress to TXT RR 29-22
   TXT RR 29-21
DHCPv6
   AAAA records. 29-2
   address generation 27-4
   bindings 27-8
   client-classes 27-33
   client FQDN, using 29-4
   clients 27-34
   concepts 27-2
   DHCID records. 29-2
   DNS update
   DNS update, upgrading 29-3
   lease affinity 27-8
   lease reservations 27-10
   leases 27-7
   links 27-2
   managing 27-1
   options 27-34
   policies 27-32
   policy hierarchy 27-12
   prefixes 27-2
   PTR records. 29-2
   reconfigure support 27-35
   server attributes
      max-client-leases 27-32
      v6-client-class-lookup-id 27-32
   synthetic names, generating 29-3
DHCPv6 DNS update
   DHCID RR 29-21
DHCPv6 failover 28-3
Digital Subscriber Line (DSL) 26-24, A-3
dig tool, troubleshooting DNS update 29-21
Discriminating Rate Limiter 24-7
distributions, zone 14-22
DNS
   address format 13-2
   authoritative server, See authoritative name servers
cache, flushing 18-10
caching-only servers, See caching-only servers
client/server model 13-5
configuration guidelines 4-5
dashboard
   general indicators chart 3-19
   inbound zone transfers chart 3-17
   network errors chart 3-18
   outbound zone transfers chart 3-16
   related servers errors chart 3-18
distributed database 14-1
domains, See domains
dynamic DNS update, See DNS update
exception handling 18-3
glue records 14-20, A-4
   invalid 14-20
   removing 14-21
lame delegation 14-20
log settings 17-10
maximum
   cache TTL property 18-7
   memory cache size 18-9
name servers 13-5
name-to-address resolution 13-5, A-1
negative cache time  A-6
options
   flushing DNS cache 18-10
   local and external ports 17-8
   maximum memory cache size 18-9
   ports 17-8
primary servers, See primary name servers
reverse name servers 13-6
root name servers, defining 18-8
secondary servers, See secondary, name servers

servers

commands 17-1, 18-1
logging 7-6, 17-10, 29-30
managing 17-1, 18-1
name 17-2, 18-5
network interfaces, configuring 17-2, 18-2
secondary, See secondary, name servers
simulating top-of-zone A records 29-30
slave servers, creating 18-2
troubleshooting 17-9
dns_startup_log file 7-6
dns_upgrade_status_log file 7-6

DNS64
Managing 18-4
dns-admin role 5-4
core functionality 5-4
ipv6-management subrole 5-4
security-management subrole 5-4
server-management subrole 5-4
dns command (CLI)
addForwarder 18-3
disable
subnet-sorting 17-4
enable
ixfr-enable 17-5
notify 17-6
round-robin 17-4
simulate-zone-top-dynupdate 29-30
subnet-sorting 17-4
findRR 16-6
forceXfer 7-2
get
round-robin 17-4
getRRCount 16-5
getStats 7-12
ha 19-8
getZoneCount 14-13
listExceptions 18-4

listForwarders 18-3
removeForwarder 18-3
resetStats 7-12
scavenge 7-2, 29-21
set
activity-summary-interval 17-10
log-settings 7-6, 17-10
mem-cache-size 17-9
notify-min-interval 17-9
notify-send-stagger 17-9
notify-wait 17-9
setPartnerDown 19-4
show 17-3, 18-6
dns-servers, DHCPv6 option B-12
dnscommand 13-7
dns ENUM
manage defaults 14-15
overview 14-15
DNS ENUM command (CLI)
add 14-16
remove 14-16
set default 14-15
DNS ENUM domain
add 14-16
pull 14-18
push 14-17
DNS ENUM domain command (CLI)
create 14-16
delete 14-16
DNS ENUM number
add 14-17
pull 14-19
push 14-18
DNS ENUM number command (CLI)
add 14-17
dns-management subrole (cfg-admin) 5-3
DNSSEC
Managing 18-5
dns-servers, DHCPv6 option B-12
DNS update
backup-server-key  29-12
benefits  20-6
configurations
  DHCPv6 synthetic name generator  26-18, 29-3
  failover synchronization effect  28-7
  policies, setting for  22-4
  synthetic name stem  26-18, 29-3
confirming records  29-19
creating  29-7
DHCPv6  29-2
definition  29-1
DHCPv6
  AAAA records  29-2
  DHCPID records  29-2
  PTR records  29-2
  synthetic names  29-3
dynamic-dns  29-7
enabling  14-22, 29-5
failover synchronization effect  28-6
forcing  29-32
large deployments  29-2
logging  29-21
maps
  creating  29-7, 29-8
  dhcp-named-policy  29-8
  dhcp-policy-selector  29-8
operation  20-7
policies
  configuring  29-15
  creating  29-16
  deleting  29-18
  editing  29-17
  interaction with previous releases  29-15
  rules  29-16
  zones, applying to  29-18
policies, See DNS update, policies
prerequisites  29-26
server-key  29-12
simulating top-of-zone A records  29-30
transition  29-21
troubleshooting  29-21
TSIG security  29-10
Windows 2000 clients  29-22
dns-update-map command (CLI)
  create  29-8
  set
    dhcp-named-policy  29-8
    dhcp-policy-selector  29-8
DNS views
  configure  15-1
  key points  15-1
  manage  15-2
  overview  13-7
  pull  15-4
  push  15-3
  reorder  15-2
  synchronize  15-3
DNS views command (CLI)
  reorder  15-3
DOCSIS  A-3
documentation
  conventions  xxxii
domain-list, DHCPv6 option  B-12
domain-name, DHCP option  B-2
domain names
  dot-separated  13-3
  space  13-2
tree structure  13-2
domain-name-servers, DHCP option  B-2
domains
difference from zones  13-3
  names, See domain names
  registering  13-3
domain-search, DHCP option  B-19
do-times, DHCP expression  26-9
DRL  24-7
dual zone updates  29-25
dynamic
addresses 9-1
BOOTP, See BOOTP, dynamic
DNS, See DNS update

edit mode
   staged A-9
   synchronous A-10
.edu domain 13-2
elapsed-time, DHCPv6 option B-12
end, DHCP option B-2
enterprise users 4-1
environment-destructor extension point, DHCP 24-12, 30-39
environment-dictionary, DHCP expression 26-10
environment dictionary, extension point 30-24
data items 30-25
   initial-environment-dictionary property 30-26
equal, DHCP expression 26-10
equali, DHCP expression 26-10
error, DHCP expression 26-11
event logging 7-5
Event Viewer, Windows 7-5
exception
   See resolution exception
export command (CLI)
   addresses, VPN 24-22
   leases 23-4
      -vpn 24-22
   option-set 22-17
expressions 26-1
   address range for scope templates 21-7
   creating 26-3
debugging 26-26
   embedded policy for scope templates 21-7
   link templates 27-16
   prefix templates 27-23
   scope templates 21-4
   scope name 21-7
   extension command (CLI)
      create 30-2
      list 30-2
      set
         init-entry 30-2
   extension points, DHCP 24-10
      listing 30-2
   extensions 24-10
      API C-26
      C/C++ 30-4, 30-9
      check-lease-acceptable 30-37
      client-classes
         modifying 30-15
         processing 30-15
      configuration errors 30-5
      creating 30-2
deciding approaches 30-3
decided packet data items C-1
definition 30-1
determining tasks 30-3
DEX 30-9
   environment methods C-39
dex.h file 30-10
DEX attributes
   dictionary C-30
      methods C-31
      dhcp-parameter-request-list option 30-28
      DHCPv6, enabling 30-14
dictionaries 30-5, 30-23
      entries C-1
DNS requests, processing 30-21
environment-destructor 30-39
environment dictionary 30-5, 30-24
failover synchronization effect 28-7
init-entry 30-29
   C/C++ 30-12
      Tcl 30-9
initial-environment-dictionary property 30-26
leases
  acceptability, determining  30-17
  finding  30-16
  requests, serializing  30-17
  state changes, tracing  30-22
lease-state-change  30-37
networks and links, determining  30-16
packet data items, decoded  30-27
packets
  decoding  30-14
  receiving  30-14
post-class-lookup  30-32
post-client-lookup  30-34
post-packet-decode  30-31
post-packet-encode  30-38
post-send-packet  30-39
pre-client-lookup  30-32
pre-packet-decode  30-30
pre-packet-encode  30-38
recognizing  30-6
request dictionary  30-5, 30-26, C-10
request processing  30-12
response containers, building  30-16
response dictionary  30-5, 30-26, C-16
response packets
  encoding  30-21
  gathering data  30-20
  sending  30-21
routine signature  30-4
stable storage, updating  30-21
Tcl  30-4, 30-7
  environment methods  C-29
Tcl attributes
  dictionary  C-26
  methods  C-27
  thread-safe  30-10
extensions-path, DHCP option  B-3
external authentication servers  5-8
  adding AD  5-11
adding RADIUS  5-9
configuring AD  5-10
configuring RADIUS  5-8
deleting AD  5-12
deleting RADIUS  5-9
pulling Active Directory  5-22
pulling RADIUS  5-20
pushing Active Directory  5-21
pushing RADIUS  5-20
external port  17-8

F
failover, DHCP
  address ranges, ensuring  28-8
  attributes
    related failover  6-4
    related servers  6-4, 6-9
  backup
    allocation boundary  21-15
    percentage  28-8
    server, removing  28-29
  backup percentage  28-8
  benefits  20-8
  BOOTP
    clients  28-31
    relay  28-8
  changing roles  28-27
  checklist  28-8
  configuring  28-1
  confirming  28-14
  creating server pairs  5-38
  dynamic BOOTP backup percentage  28-32
  lazy updates  28-10
  lease period factor  28-10
  lease query  28-20
  load balancing  28-13
    configuring  28-14
  local server synchronization  28-4
logging 28-7, 28-30
main server, adding new 28-29
maximum client lead time 28-10
monitoring 28-30
multiple interfaces 28-29
network failures 28-30
operation 28-1
  halting 28-29
pairs
  backup percentage 28-8
  creating 28-5
  synchronizing 28-7
partner down state 28-11
regional cluster synchronization 28-5, 28-14
regional pairs 6-20
related servers, getting 6-10
removing backup server 28-29
replacing servers with defective storage 28-28
request/response buffer settings 28-13
restrictions, communications interrupted 28-16
safe period 28-11
server pairs, creating 28-3
simple scenarios 28-2
states 28-15
state transitions 28-17
synchronize function 28-5
synchronizing 28-6
synchronizing pairs 5-38
troubleshooting 28-29
types 28-1
failover-pair command (CLI)
  create 28-5
  enable
    load-balancing 28-9
    use-safe-period 28-12
  set
    backup-pct 28-8
    dynamic-bootp-backup-pct 28-31
    load-balancing 28-14
  safe-period 28-12
  sync 28-7
filter, lease 23-8
finger-servers, DHCP option B-8
font-servers, DHCP option B-7
forwarding DNS servers 18-2
  listing 18-3
forwarding servers
  DHCP 24-24
FQDN 13-2, 16-2, A-4
  DHCP processing 20-10
  option, DHCP 29-23
free-address-low-threshold event, SNMP 1-4

G
gateway address 20-12, 24-19, A-4
generate-lease extension point, DHCP 24-11
geo-conf, DHCP option B-19
geoconf-civic, DHCP option B-19
geoconf-civic, DHCPv6 option B-13
giaddr, DHCP field 20-12, 24-19, A-4
glue records 14-20, 14-21
  .gov domain 13-2
  .gov domain 13-2
  grace period, leases 22-4
Granular Administration 5-12
grep tool (UNIX) 7-25
groups 5-1, 5-5
  adding 5-6
  interaction with roles 5-2
  pulling 5-23
  pushing 5-22

H
HA DNS
backup server, setting 19-3
configuring 19-1
enabling 19-3
main server, setting 19-3
partner down, setting 19-4
server pairs, creating 19-4
synchronizing server pairs 19-4
ha-dns-pair command (CLI)
create 19-4
set
  ha-dns-backup-server 19-3
  ha-dns-main-server 19-3
sync 19-4
Handling Malicious DNS Clients and Unresponsive Nameservers 17-8
Help
  attributes 2-6
  pages 2-6
HINFO records A-3, A-4
host-admin role 5-4
  core functionality 5-4
Host Info records
  See HINFO records
hostmaster, setting for zone 14-6
host-name, DHCP option B-2
host-regexpr syntax 5-30
hosts
  adding to zones 10-1, 16-9
  BOOTP 24-1
creating 5-29
dynamic 14-22, 29-1
editing 10-2
multiple interface 28-29
pinging 23-6
removing 10-3
testing address ranges 5-32
zone restrictions 5-30
HTTPS login 2-3
iaaddr, DHCPv6 option B-11
ia-na, DHCPv6 option B-11
ia-pd, DHCPv6 option B-13
iaprefix, DHCPv6 option B-13
ia-ta, DHCPv6 option B-11
ICMP
echo, See PING
  errors, ignoring 23-6
ieee802.3-encapsulation, DHCP option B-5
IETF 20-1, 21-22, A-3
if, DHCP expression 26-11
ifconfig tool (UNIX) 7-25
import command (CLI) 14-9
  keys 29-12
  leases 23-4, 24-22
  option-set 22-17
impress-servers, DHCP option B-2
in-addr.arpa domain 13-6, A-4
incremental zone transfers A-5
  enabling 4-5, 17-5
info-refresh-time, DHCPv6 option B-13
init-entry extension point
  C/C++ 30-12
  Tcl 30-9
init-entry extension point, DHCP 24-11, 30-29
initial-url, DHCP option B-19
install_cnr_log file 7-6
interface cards 21-2, 29-31
interface-id, DHCPv6 option B-12
interface-mtu, DHCP option B-4
Internet Control Message Protocol
  See ICMP
Internet Engineering Task Force 20-1, A-3
Internet Operating System
  See IOS
Intranet Builder 18-3
IOS, VPN support 24-17
Index

ip6-string, DHCP expression 26-11
IP addresses
  See addresses, IP
ipconfig utility 21-23, 25-11
ip-forwarding, DHCP option B-3
ip-helper 28-32
IP helper address 24-3
IP history
  See lease history
iphist utility 23-23
ip-string, DHCP expression 26-11
IPv6 addresses 27-1
IPv6 leases
  states 27-8
ipv6-management subrole (addrblock-admin) 5-3
ipv6-management subrole (dhcp-admin) 5-4
ipv6-management subrole (dns-admin) 5-4
irc-servers, DHCP option B-8
ISDN records A-3
iSNS, DHCP option B-10
is-string, DHCP expression 26-11

LDAP 16-8
  client
    configuration 25-19
data use, enabling 25-21
  configuring 25-19
connections 25-30
DHCP
  client queries 25-20
  mapping 25-20
directory support 25-20
distinguished name (dn) 25-20
embedded policies 25-22
entry creation 25-28
  enabling 25-29
event service, failover synchronization effect 28-6
failover configuration 28-8
filtering searches 25-23
lease-state attributes 25-25
passwords 25-20
protocol definition 25-19
queries, enabling 25-20
query-timeout 25-30
schema checking, disabling 25-19
state updates 25-26
storing lease data 25-21
threadwaittime 25-30
timeout 25-30
troubleshooting 25-29
typical attribute settings 25-30
unprovisioning client entries 25-22
updates, enabling 25-27
ldap command (CLI)
  create 25-20
delete 25-20
  enable
can-query 25-8, 25-21
can-update 25-27
getEntry 25-21
list 25-22

J

jsui_log.date.txt file 7-6

K

key command (CLI)
  create 29-11
keys, generating secrets 29-11

L

lame delegation 14-20, A-5
LAN segments 20-10, A-9
large enterprise deployments 4-3
lazy updates, failover 28-10

Cisco Prime IP Express 8.2 User Guide
OL-31070-01 IN-17
listnames 25-22
set
create-object-classes 25-29
dn-attribute 25-28
dn-create-format 25-28
dn-format 25-28
password 25-20
preference 25-22
search-filter 25-21
search-path 25-21
search-scope 25-21
update-search-attribute 25-27
update-search-filter 25-27
update-search-path 25-27
update-search-scope 25-27
username 25-27
setEntry 25-21
create-dictionary givenname 25-28
create-dictionary localityname 25-28
create-dictionary sn 25-28
update-dictionary carlicense 25-27
update-dictionary uid 25-27
show 25-22
unsetEntry 25-21
ldap-url, DHCP option 8-18
lease6 command (CLI)
activate 27-29
deactivate 27-29
force-available 27-29
list 23-10
lease command (CLI)
activate 23-7
deactivate 23-7
force-available 23-19, 24-3
list
-macaddr 23-10
set
address 25-24
client-dns-name 25-24
client-domain-name 25-24
client-host-name 25-24
client-id 25-24
client-mac-addr 25-24
expiration 25-24
flags 25-24
lease-renewal-time 25-24
start-time-of-state 25-24
state 25-24
vendor-class-identifier 25-25
show 23-2, 27-29
lease extensions, deferring 24-8
lease history 23-20
automatic trimming
age 23-26
interval 23-26
collecting 23-22
collection maximum age 6-14
database directory 23-22
enabling 6-14, 23-22
iphist utility 23-23
maximum age for trimming 23-26
polling
data 6-13
interval 6-14
offset 6-14
retry interval 6-14
querying 23-23
recording 23-22
reports 23-21
trimming 23-26
lease-history subrole (regional-addr-admin) 5-5
lease-notification command (CLI) 23-27
available 23-21
mail-host 23-27
recipients 23-27
scopes 23-27
specifying config file 23-28
lease period factor, failover 28-10
leasequery 23-28
  DHCPv4 pre-RFC implementation 23-29
  DHCPv4 RFC 4388 implementation 23-30
  DHCPv6 implementation 23-31
logging 24-16
  reservations and statistics 23-29
leases 20-3
  activity 7-20
  address usage reports 23-21
  affinity 27-8
  benefits 20-3
  database 7-6
  deactivating 24-3
    in scopes 23-7
  deferring extensions 24-6
  defined 20-2
  DHCPv6 27-7
  DHCPv6 life cycle 27-9
  displaying 7-20
  excluding addresses from ranges 23-7
  expired state 20-2
  exporting 23-4
  file 23-4
    time format 23-5
  forcing available 23-18
  grace period 22-4
  history reports 23-20
  importing 23-4
  inhibiting renewals 23-19
  LDAP attributes 25-26
  notification, receiving 23-27
  obtaining 20-7
  one-shot 25-12
  orphaned
    by VPN 24-23
  permanent 22-3, 23-3
  pinging before allocating 23-6
  provisional 25-12
  querying, See leasequery
  reacquiring 20-8
  reactivating 23-7
  recommended renewal times 4-6
  releasing 20-7
  renewal time, saving as state 25-24
  renewing 20-2
  reusing 24-3
  scopes 20-2, 23-1
    listing 23-2
    viewing 23-2, 27-29
  searching, filtering 23-8
  states 23-2, 25-24
  state updates in LDAP 25-25
  timeouts for unavailable times
    guidelines 23-3
    in import files 23-5
    overrides, allowing 23-4
  types 20-3
  unavailable
    clearing 23-19
    handling 23-20
  lease-state-change extension point, DHCP 24-12, 30-37
  length, DHCP expression 26-11
  let, DHCP expression 26-12
  license 5-16
  license command (CLI)
    create 5-16
    list 5-16
    listnames 5-16
    show 5-16
  licenses 5-15
    adding 2-2, 2-3
  link command (CLI)
    applyTemplate 27-20
    create 27-20
Index

template, with 27-20
  template-root-prefix, with 27-20
listPrefixes 27-20
listPrefixNames 27-20

links
  configuring 27-14
DHCPv6 27-2
failover
  synchronization effect 28-7
link-template command (CLI)
  apply-to (link) 27-16
  create 27-15
  clone 27-15
link templates
  expressions 27-16
Linux
  CLI location 2-10
list
  link template expression 27-17
  prefix template expression 27-24
local clusters 1-1, 4-1
  administration 5-1
  connecting to 6-10
  editing 6-3
  replicating data 6-11
  synchronizing with 6-10
  tutorial 5-25
  view of tree 6-2
localhost 14-4, 17-9
local port 17-8
lock files 8-3
log, DHCP expression 26-12
log.xxx files, CNRDB 8-5
logging
  NOTIFY, See NOTIFY
logging out 2-6
login, Web UI 2-2
log-servers, DHCP option B-2
loopback

addresses A-5
  zones 14-4, A-5
lpr-servers, DHCP option B-2
lq-client-links, DHCPv6 option B-14
lq-query, DHCPv6 option B-13
lq-relay-data, DHCPv6 option B-14
lshift, DHCP expression 26-8

M

MAC addresses, clients 20-10
management components 1-1
mask-blob, DHCP expression 26-12
mask-int, DHCP expression 26-12
mask-supplier, DHCP option B-4
max-dgram-reassembly, DHCP option B-3
maximum client lead time, failover 28-10, 28-16
MB records A-3
MCLT
  See maximum client lead time
mcns-security-server, DHCP option B-19
merit-dump, DHCP option B-2
MG records A-4
Microsoft DHCP stack 21-2
.mil domain 13-2
MINF records A-4
Miscellaneous Options and Settings 17-8, 18-12
mobile-ip-home-agents, DHCP option B-17
MR records A-4
multinetting 20-10, A-6
multiple interfaces, failover 28-29
multiple users 2-3
MX records A-4, A-6

N

name_dhcp_1_log file 7-6
name_dns_1_log file 7-6
Name Server records
   See NS records
name servers
   authoritative, See authoritative name servers
domain 13-5
   primary, See primary name servers
reverse 13-6
   secondary, See secondary, name servers
types 13-5
name-servers, DHCP option  B-2
name-service-search, DHCP option  B-19
name-to-address resolution 13-5, A-1
Naming Authority Pointer records
   See NAPTR records
NAPTR records 16-8, A-5
navigation bar labels xxxiii
nds-context, DHCP option  B-18
nds-servers, DHCP option  B-18
nds-tree, DHCP option  B-18
negative cache time  A-6
netbios-dd-servers, DHCP option  B-6
netbios-name-servers, DHCP option  B-6
netbios-node-type, DHCP option  B-7
netbios-scope, DHCP option  B-7
netinfo-parent-server-addr, DHCP option  B-19
netinfo-parent-server-tag, DHCP option  B-19
netwareip-domain, DHCP option  B-17
netwareip-information, DHCP option  B-17
Network Information Service
   See nis entries
network interfaces
   DNS server 17-2, 18-2
Network News Transport Protocol  B-7
network number 13-3
networks
   editing names 21-24
   listing 21-23
   managing 21-23
Network Time Protocol  B-6
   network time service, use of 6-13
nis+-domain, DHCP option  B-7
nis+-servers, DHCP option  B-7
nis-domain, DHCP option  B-5
nis-domain-name, DHCPv6 option  B-13
nisp-domain-name, DHCPv6 option  B-13
nis-p-servers, DHCPv6 option  B-13
nis-servers, DHCP option  B-5
nis-servers, DHCPv6 option  B-13
ntp-servers, DHCP option  B-7
non-local-source-routing, DHCP option  B-3
nonsecure login 2-2
NORMAL state, failover 28-15
not, DHCP expression 26-15
notification, lease 23-27
NOTIFY A-6
   enabling 17-5
   logging transactions 17-10
NSAP records A-6
nslookup utility 17-11
   troubleshooting DNS update 29-21
NS records A-5
ntp-servers, DHCP option  B-6
null, DHCP expression 26-13

O
on-demand address pools
   See subnet allocation, DHCP
option command (CLI)
   get 22-9
   listtypes 22-16
   show 22-9
   unset 22-9
options
   custom DHCP 22-10
   data types, listing 22-16
   DHCPv6 27-13
   setting 27-34
failover synchronization effect 28-7
policy hierarchy 22-3
option-set command (CLI)
  show 22-9
OptionSetJumpStart.txt file 22-17
OptionSetPXE.txt file 22-17
option sets
  exporting 22-17
  importing 22-17
  local
    pulling 22-18
    pushing 22-17
or, DHCP expression 26-13
organization, registering 12-4
Organizationally Unique Identifier
  See OUI
oro, DHCPv6 option B-11
OUI, for VPNs 24-20, A-7
owner command (CLI)
  create 11-1
owner-region subrole (ccm-admin) 5-3
owner-region subrole (regional-admin) 5-4
owners
  configuring 11-1
  managing 11-1
  pulling 11-3
  pushing 11-3

P

pad, DHCP option B-2
PARTNER-DOWN state, failover
  enabling 28-11
passwords 5-6
  administrator 5-5
    changing 5-6
  changing 2-4
  nondisplaying 5-6
path-mtu-aging-timeout, DHCP option B-3
path-mtu-plateau-tables, DHCP option B-3
PAUSED state, failover 28-17
PDU, SNMP 1-5
performance guidelines 4-5
perform-mask-discovery, DHCP option B-4
pick-first-value, DHCP expression 26-13
pinging hosts before offering 23-6
ping utility 23-6
Pointer (Reverse Mapping) record
  See PTR record
point of contact, registering 12-3
policies
  cloning 22-5
  compared with scopes 20-3
  configuring 22-1
  creating regional 6-16
  defined 20-3, 22-1, A-7
  DHCPv6 27-32
dual zone updates, allowing 29-25
  embedded 22-2
    editing 22-7
    scopes 21-18
    vendor options 21-19
  failover synchronization effect 28-6
guidelines 21-2
hierarchy 22-3
lease time overrides, allowing 23-4
local
  pulling 6-17
  pushing 6-16
named 22-2
options 20-10
  adding 22-5
  scopes 20-10
  scopes, See scopes
  suboptions
    adding 22-7
  system-default 22-1
  policies, DHCP
address block 24-22
pushing to local clusters 5-37
policy command (CLI)
create 22-5, 24-2
   clone 22-5
disable
   allow-client-a-record-update 29-23
enable
   allow-client-a-record-update 29-23, 29-24
   allow-dual-zone-dns-update 29-25
   allow-lease-time-override 23-4
   permanent-leases 22-5
   use-client-id-for-reservations 23-18
getOption 22-6
   dhcp-lease-time 22-5
listOptions 22-5
set 22-5, 24-2
   grace-period 25-12
   limitation-count 26-2
setLeaseTime 22-5
setOption 22-6, 24-3
   subnet-mask 22-5
unsetOption 22-6
policy-filters, DHCP option B-3
polling A-7
   interval 6-14
   lease history data 6-13
   offset 6-14
   retry interval 6-14
   time skew effects 6-13
pop3-servers, DHCP option B-7
post-class-lookup extension point, DHCP 24-11, 30-32
post-client-lookup extension point, DHCP 24-11, 30-34
Post Office Protocol B-7
post-packet-decode extension point, DHCP 24-11, 30-31
post-packet-encode extension point, DHCP 24-12, 30-38
post-send-packet extension point, DHCP 24-12, 30-39
POTENTIAL-CONFLICT state, failover 28-16
pre-client-lookup extension point, DHCP 24-11, 30-32
pre-dns-add-forward extension point, DHCP 24-12
preference, DHCPv6 option B-12
preferences, user 2-9
prefix allocation groups 27-6
prefix command (CLI)
   addReservation 27-29
   applyTemplate 27-29
   create 27-29
   template, with 27-29
   createReverseZone 29-4
   deleteReverseZone 29-4
   listLeases 27-29
   set
      selection-tags 25-5
prefixes
   configuring 27-20
   count on server, getting 27-29
   DHCPv6 27-2
      reverse zones, creating 29-4
failover
   synchronization effect 28-7
   interface-identifier, assigning 27-4
prefix stability 27-5
   CMTS 27-5
   universal 27-6
prefix-template command (CLI)
   apply-to (prefix) 27-23
   create 27-23
   clone 27-23
   set
      selection-tags 25-5
prefix templates
   expressions 27-23
pre-packet-decode extension point, DHCP 24-11, 30-30
pre-packet-encode extension point, DHCP 24-12, 30-38
primary name servers 13-4
   configuring 14-4
   defined 13-5
   SOA property A-7
zone, setting for 14-6
progn, DHCP expression 26-13
Protocol Data Unit, SNMP
See PDU
PTR records A-6, A-7
DNS update 28-2
Pushing Administrators Automatically to Local Clusters 5-18
pxe-client-arch, DHCP option B-18
pxe-client-machine-id, DHCP option B-18
pxe-client-network-id, DHCP option B-18
PXE clients 22-17
PXE clients, importing option sets for 22-17

dhcp-management subrole 5-5
database subrole 5-4
owner-region subrole 5-4

R

Rapid-commit, DHCPv6 option B-12
reconfigure-accept, DHCPv6 option B-12
reconfigure-message, DHCPv6 option B-12
reconfigure support, DHCPv6 27-35
RECOVER-DONE state, failover 28-16
RECOVER state, failover 28-16
recursive queries A-8
regional-addr-admin role 5-5
core functionality 5-5
dhcp-management subrole 5-5
lease-history subrole 5-5
regional-admin role 5-4
authentication subrole 5-4
authorization subrole 5-4
core functionality 5-4
database subrole 5-4
owner-region subrole 5-4
regional clusters 1-1, 4-1
adding
local clusters 5-34, 6-2
server clusters 6-2
administration 5-1
client-classes

Pulling 6-18
pushing 6-18
failover pairs 6-20
policies 6-16

Pushing 6-16

option definition sets 22-17
reservations
pushing 6-21
scope templates 6-15
pushing 6-15
subnets to local clusters
pushing 9-7
tutorial 5-32
VPNs 6-19

region command (CLI)
create 11-2
regions
configuring 11-1
managing 11-2
pulling 11-3
pushing 11-3
regular expressions (regex) 5-30
related servers
DHCP attributes 6-4
states 6-4
relative domain names 16-2
relay-agent-info, DHCP option B-18
relay-agent-subscriber-id, DHCPv6 option B-13
relay-message, DHCPv6 option B-12
remote-dns command (CLI)
create 17-9
remote-id, DHCPv6 option B-13
replica data 6-11
replica data, viewing 6-11
report command (CLI) 7-18, 23-21
reports
address usage 7-17, 23-21
Index

allocation 12-1

ARIN 12-1

IPv4 utilization 12-5

lease history 23-20, 23-21

managing 12-1

organization
  creating 12-4
  editing 12-5

point of contact 12-2
  editing 12-3

WHOIS/SWIP 12-6

request dictionary, DHCP expression 26-16

request dictionary, extension points 30-26

request dump, DHCP expression 26-16

request option, DHCP expression 26-13

request packetfield, DHCP expression 26-15

reservation command (CLI)
  delete 23-16

reservations, lease
  creating 23-13
  LEASEQUERY and 23-29
  pushing to local clusters 6-21
  removing 23-16
  using client ID 23-18

resolution exception 18-3

reverse

resource-LOCATION-servers, DHCP option B-2

resource records
  A A-1
  A6 A-2
  AAAA 29-2, A-2
  adding 16-2
  adding in CLI 16-2
  AFSDB A-2
  CNAME A-3, A-2
  configuring 16-1
  count on server, getting 16-5
  DHCID 29-2, A-3
  editing 16-1
    subzone information 14-22
  filtering 16-7
  HINFO A-3, A-4
  ISDN A-3
  listing 16-5
  MB A-3
  MG A-4
  MINFO A-4
  MR A-4
  MX A-4, A-6
  NAPTR A-5
  NS A-5
  NSAP A-6
  protecting 16-3
  PTR A-6, A-7
  removing 16-4
  RP A-6
  RT A-6
  SOA A-7, A-9
  SRV A-7
  TXT A-7
  types A-1
  WKS A-8, A-11

response dictionary, DHCP expression 26-16

response dictionary, extension points 30-26

response dump, DHCP expression 26-16

response option, DHCP expression 26-16

response packetfield, DHCP expression 26-16

return-last, DHCP expression 26-13

reverse
  domains 13-6
  mapping records A-6
  name servers 13-6
  zones 13-6, A-8
    configuring 14-11
    prefixes, creating from 29-4

RFCs
  1001, 1002 B-6
  1035 14-9
  1042 B-5
1179  B-2
1191  B-3
1256  B-4
1497  B-1
1995  4-5, 17-5
1996  4-5, 17-5
2131  24-2
2132  B-1
2136  14-22
2316  A-3
2685  24-20
2782  16-7, 29-27, A-7
2915  A-5
2916  16-8
3041  27-7
3046  25-14, 26-1
3074  28-13
3263  16-8
3315  27-1, B-11, B-14
3319  B-12
3403  16-8
3633  27-1, 27-7
3736  27-1
3898  B-13
4075  B-13
4174  B-10
4242  B-13
4280  B-10, B-13
4291  27-4
4388  23-30, B-10
4580  B-13
4649  B-13
4701  29-2, A-3
4702  29-2
4704  29-2
4833  B-15, C-5, C-7, C-9
5192  C-5
865   B-2
868   B-2
887   B-2
893   B-5
894   B-5
950   B-2
roles  5-1
adding  5-7
address block administrator  9-1
base  5-1
constrained  5-2, A-8
creating  5-30
groups  5-5
interaction with groups  5-2
pulling  5-24
pushing  5-24
subroles  5-2
root name servers  13-3, A-8
adding  18-8
slave servers  A-9
root-path, DHCP option  B-3
round-robin  A-8
enabling  17-4
scope selection  20-10, 21-11
routed bridge encapsulation (RBE)  A-8
router-discovery, DHCP option  B-4
routers
BOOTP relay  24-3
Cisco  24-3
gateway addresses  20-12, 24-19, A-4
policies for  20-3
subnet  22-5
routers, DHCP option  B-2
router-solicitation-address, DHCP option  B-4
RP records  A-6
RT records  A-6
S
safe period, failover  28-11, 28-16
scope command (CLI)
addRange 21-11
clearUnavailable 23-19
create 21-11
    template= 21-9
delete 21-23
disable 21-17
dehcp 21-20, 24-3, 28-31
ping-clients 23-20
enable 21-17
    dhcp 21-20
    deactivated 21-21
dehcp 21-20
dynamic-bootp 21-20, 24-3, 28-31
ignore-declines 23-20
renew-only 21-21
    update-dns-for-bootp 24-3
get 21-17
list 21-17
listLeases 23-2
listnames 21-17
listRanges 23-8
removeRange 23-8
removeReservation 23-16, 24-3
report-staged-edits 21-18
set 21-17
    backup-pct 28-8
    free-address-config 21-22
    policy 21-11
    primary-subnet 21-20
    selection-tag-list 25-5
show 21-17
unset
    primary-subnet 21-20
scopes
    address
        allocation 21-11
        ranges 21-9
    allocate-first-available 21-12, 21-14
    allocation-priority 21-12, 21-14
attributes
    bootp 21-20
    disabling 21-17
dynamic-bootp 21-20
    enabling 21-17
    getting 21-17
    listing 21-17
    primary-subnet 21-19
    setting 21-17
    showing 21-17
BOOTP, enabling 21-20
compared to policies 20-3
count on server, getting 21-11
creating 21-9
deactivating 21-21
defined 20-3
defining 21-2
DHCP, disabling 21-20
editing 21-17
edit modes 21-18
failover
    backup percentage 28-8
    synchronization effect 28-7
failover-backup-allocation-boundary 21-15, 21-16
forcing lease availability 23-18
free address SNMP traps 21-21
inhibiting lease renewals 23-19
internal address allocation 21-15
leases, See leases
moving/decommissioning BOOTP clients 21-20
Index

multiple 21-11
named policies 22-2
names 21-9
network addresses 21-9
pinging clients 23-6
policies, See policies
primary subnets 21-20
ranges
  adding 21-11
  listing 23-8
  removing 23-8
removing 21-22
  if not reusing addresses 21-22
  if reusing addresses 21-23
ranges 23-8
reusing addresses 21-23
renew-only 21-21
secondary 21-19
staged edit mode A-9
staged edits, reporting 21-18
synchronous edit mode A-10
traps, SNMP, free address 21-21
unreserving leases 23-16
VPNs 24-20
scope-template command (CLI)
  apply-to scope 21-9
  create 21-3
    clone 21-9
  set 21-8
    options-exp 21-6
    ranges-exp 21-6
    scope-name 21-6
    selection-tag-list 25-5
scope templates
  address range expressions 21-7
  cloning 21-9
  creating 21-3
  creating on regional cluster 5-37
  editing 21-8
  attributes 21-8
embedded policy expressions 5-38, 21-7
expressions 21-4
managing 21-3
name expression 5-38
pulling from local clusters 6-16
pushing to local clusters 6-15
range expressions 5-38
regional 6-15
scope name expressions 21-7
SCP
  See System Configuration Protocol
screen displays xxxiii
scripts
  See extensions
search, DHCP expression 26-17
search, lease 23-8
secondary
  expiration time 17-8
  master A-9
  name servers, DNS 13-4, A-9
    adding 14-13
    defined 13-5
  separating DHCP servers from 21-1
refresh time 14-14, 17-7
retry time 17-7
scopes 21-19
subnets 21-19, A-9
zones 14-13
  adding 14-13
secondary name servers 14-13
secure
  cluster connections 6-2
security-management subrole (central-dns-admin) 5-4
security-management subrole (dns-admin) 5-4
selection tags
  appending dhcp-user-class-id 25-8
  appending RADIUS class 25-8
  appending RADIUS pool 25-8
mapping RADIUS class 25-8
mapping RADIUS pool name 25-8
mapping user class identifier 25-8
server clusters, adding 6-2
server command (CLI)
  enable/disable start-on-reboot 7-1
getHealth 7-11
getStats 7-12
reload 7-3
serverLogs
  set logsize 7-7
  show 7-7
set 7-3
start 7-3
stop 7-3
server-identifier, DHCPv6 option B-11
server-management subrole (ccm-admin) 5-4
server-management subrole (central-dns-admin) 5-4
server-management subrole (dns-admin) 5-4
servers
  controlling 7-1
  events, logging 7-4
  failures, troubleshooting 7-25
  health, displaying 7-10
  IP address A-9
  related, states 6-4
  state, displaying 7-9
  statistics, showing 7-11
server-unicast, DHCPv6 option B-12
session command (CLI)
  cache refresh 2-4
  get
    dhcp-edit-mode 21-18
  set
    current-vpn 21-11, 24-20
    dhcp-edit-mode 21-18
    dns-edit-mode 14-2
Session Initiation Protocol (SIP) proxies 16-8
setq, DHCP expression 26-17
setting
  custom DHCP options 22-10
  negative cache time A-6
shadow backups 8-1
  manual 8-3
  third party backup programs and 8-3
time, setting 8-3
SHUTDOWN state, failover 28-17
siaddr, DHCP field 24-2, A-9
Simple Mail Transport Protocol
  See SNMP
  simulating top-of-zone A records 29-30
  single sign-on 2-4, 6-2
sip-servers, DHCP option B-19
sip-servers-address, DHCPv6 option B-12
slave servers 18-2, A-9
slp-directory-agent, DHCP option B-18
slp-service-scope, DHCP option B-18
SMS
  integrating with DHCP server 24-9
  network discovery (dhcp command) 24-10
smtp-servers, DHCP option B-7
SNMP
  free-address-low-threshold 1-4
  logging and tracing 7-6
  notification 1-2
  notification events 1-5
  PDU 1-5
  traps 1-4
    failover mismatch 28-31
    free address on scopes 21-21
    PDUs 1-2
    server not responding 28-30
  v2c standard 1-2
snmp command (CLI)
  disable server-active 1-4
  enable server-active 1-4
  set
    cache-ttl 1-4
community 1-4
trap-source-addr 1-4
snmp-interface command (CLI) 1-4
snmp-management subrole (cfg-admin) 5-3
snntp-servers, DHCPv6 option B-13
SOA records A-7, A-9
  defined 14-6
  secondary
    expiration time 17-8
    refresh time 17-7
    retry time 17-7
  TTL property 17-6
  Windows 2000 29-27
SRV records A-7
  enabling viewing 29-29
  Windows 2000 29-26
SSL
  cluster connections 6-2
SSO login 2-4
staged
  dhcp edit mode 21-18
  dns edit mode 14-1
  edit mode A-9
starts-with, DHCP expression 26-17
STARTUP state, failover 28-15
static
  addresses 21-2
  BOOTP 28-31
  routes B-4
static-routes, DHCP option B-16
statistics
  leasequery 23-32
  server 7-11
status-code, DHCPv6 option B-12
streettalk-directory-assistance-servers, DHCP option B-8
streettalk-servers, DHCP option B-8
subnet-alloc, DHCP option B-20
subnet allocation, DHCP 24-17, A-10
  configuring 24-22
subnet-mask, DHCP option B-2
subnets 9-3
  adding 5-27
  address ranges 9-10
  child 9-9
  client-accessible 20-11
  creating 5-36
  defined 20-6
  joining 21-19
  local B-4
  multiple logical 21-19
  pushing to local clusters 9-7
  reclaiming 9-9
  sorting, enabling 17-4
  viewing 9-4
subnets, DHCP
  address block 24-22
  allocation request 24-22
  increment 24-22
  initial 24-22
  subnet-selection, DHCP option B-19
subroles 5-2
  central administration management 5-17, 11-2
subscriber limitation, using option 82 25-14
  troubleshooting 25-18
subscribers, set by limitation-id 25-15
substring, DHCP expression 26-17
subzones A-9
  adding 14-19
  delegating 14-20, A-3, A-10
  editing 14-22
  lame delegation 14-20
  name server 14-20
  naming 14-20
  removing 14-21
swap-server, DHCP option B-2
synchronization, failover
  Complete operation 28-5
  Exact operation 28-5
operations 28-6
Update operation 28-5
synchronous
dhcp edit mode 21-18
dns edit mode 14-1
edit mode A-10
synthesize-host-name, DHCP expression 26-18
System Configuration Protocol (SCP) 4-2
System Management Server
See SMS
system metrics chart, dashboard 3-9

T
TAC tool 7-26
tasks, scheduling 7-3
Tcl
API 30-7
extensions 24-10, 24-11, 30-4, 30-7
TCP Default TTL, DHCP option B-5
tcp-keepalive-garbage, DHCP option B-5
tcp-keepalive-interval, DHCP option B-5
/temp directory 8-3
templates
scope 21-3
zone 14-2
Text records
See TXT records
time-offset, DHCP option B-2
timeouts for unavailable leases 23-20
time-servers, DHCP option B-2
time service, use of 6-13
Time to Live property
See TTL property
to-blob, DHCP expression 26-18
to-ip, DHCP expression 26-18
to-ip6, DHCP expression 26-18
to-lower, DHCP expression 26-19
database log files 7-6
server 4-1
top tool (UNIX) 7-25
to-sint, DHCP expression 26-19
to-string, DHCP expression 26-19
to-uint, DHCP expression 26-19
trailer-encapsulation, DHCP option B-5
translate, DHCP expression 26-20
trap command (CLI)
set
free-address-low-threshold 1-4
trap-recipient command (CLI)
create 1-4
traps, SNMP 1-2, 28-30
creating 21-22
failover synchronization effect 28-7
free-address-high 1-5
free-address-low 1-5
low and high address thresholds 21-22
recipients, creating 1-4
try, DHCP expression 26-20
TSIG keys 29-10
creating 29-11
DNS update configuration attributes 29-12
failover synchronization effect 28-7
importing 29-12
pulling 29-11
pushing 29-11
rules for secrets 29-12
secrets, generating 29-11
TTL property A-1
default 17-7, 22-2
responses 17-6
maximum
cache 18-7
memory cache size 18-9
negative cache A-6
zone 17-6
tutorial
Tomcat
Index

local cluster  5-25
regional cluster  5-32
TXT records  A-7
tz-database, DHCP option  B-19
tz-posix, DHCP option  B-19

U
Ultra-SCSI disk  4-3
unicast, enabling  24-6
unified address space  9-2
Universal Resource Identifiers (URIs)  16-8
UNIX, troubleshooting tools  7-25
update-policy (CLI command)
  create  29-16
  delete  29-18
  rules
    add  29-17
    remove  29-18
user-auth, DHCP option  B-18
user-class, DHCPv6 option  B-12
user interfaces  2-1
user preferences, setting  2-9
users
  differentiated services  20-9
  event warnings  7-5
  lease availability  20-7
  types of  4-1
utility programs
  ipconfig  21-23
  ping  23-6
  third party backup  8-3

V
validate-host-name, DHCP expression  26-21
vendor-class, DHCPv6 option  B-12
vendor-encapsulated-options, DHCP option  B-17
vendor-opts, DHCPv6 option  B-12
virtual channel identifier  26-24
virtual path identifier  26-24, A-11
virtual private networks
  See VPNs
virtual routing and forwarding table ID
  See VRFs
virus scanning, excluding directories  8-9
visibility of attributes  2-5
v-i-vendor-class, DHCP option  B-19
v-i-vendor-info, DHCP option  B-19
vmstat tool (UNIX)  7-25
vpn command (CLI)
  create  24-20
  set
  vrf-name  24-20
vpn-id, DHCP option  B-20
VPNs  20-5, 24-17
  configuring  24-17
  creating  24-19
  current  24-21
  failover synchronization effect  28-7
  identifier  24-20
  IOS support  24-17
  leases in, importing  24-22
  local
    pulling  6-20
    pushing  6-19
  orphaned leases  24-23
  regional  6-19
  scopes  24-20
  setting current  21-11
  VRFs  24-20
VRFs  24-18, 24-20

W
Web UI  1-1
attributes
displaying 2-5
modifying 2-5
visibility 2-5
changes, committing 2-5
deployment scenarios 4-2
help
attributes 2-6
topics 2-6
logging 7-6
logging in 2-2
navigation 2-4
Well Known Services record
See WKS records
Windows
CLI location 2-10
Event Viewer 7-5
ipconfig utility 25-11
logging 7-4
mouse buttons xxxiii
Windows 2000
client properties 25-11
DNS update 29-22, 29-29
domain controllers 29-29
environments 29-28
SRV records 29-26, A-7
Windows SMS
See SMS
winipcfg utility 25-11
WKS records A-8, A-11
www-servers, DHCP option B-7

X
x-display-managers, DHCP option B-7

Y
yiaddr, DHCP field 24-19, A-11

Z
zone
edit mode, setting 14-2
templates
cloning 14-4
creating using CLI 14-4
zone (CLI command)
listRR
dns 29-19
set
update-policy-list 29-19
zone command (CLI)
addDNSRR 10-2, 16-2
addHost 10-2
addRR 10-2, 16-2
A 14-7
PTR 14-12
-staged or -sync 16-2
applyTemplate 14-7
create
primary 14-7
template, using 14-7
enable
notify 17-6
findRR 16-7
forceXfer 14-15
get
serial 14-6, 14-7
getScavengeStartTime 29-21
listHosts 10-2, 14-7
listRR 14-8, 16-5
removeDNSRR 10-3, 16-4
removeHost 10-3
removeRR 10-3, 16-4
set
defttl 17-7
dist-map 14-25
expire 17-8
log-settings 29-20
nameservers 14-7
notify-set 17-6
refresh 17-7
retry 17-8
scvg-ignore-restart-interval 29-20
scvg-interval 29-20
scvg-no-refresh-interval 29-20
scvg-refresh-interval 29-20
show 14-7
zone-dist command (CLI)
  addSecondary 14-25
  create 7-20, 14-25
  list 7-20
  set
    master-servers 14-25
  sync 14-26
zone distributions
  creating 7-20, 14-25
  listing 7-20
  managing 14-22
  master servers, adding 14-25
  secondaries, adding 14-25
  synchronizing 14-26
zone-regexpr syntax 5-30
zones
  address restrictions 5-30
  applying update policies 29-18
  authoritative name servers 14-6
  configuring 10-1, 16-1, 17-1, 18-1
  count on server, getting 14-13
  defined 13-3
  delegated subzone records, editing 14-22
  delegated subzones, removing 14-21
  delegation-only 17-3
  difference from domains 13-3
  displaying resource records 16-5
  DNS update, enabling 14-22
  domains, adding 16-2
dual zone updates 29-25
  edits
    staged and synchronous modes 14-1
  hostmaster 14-6
  hosts
    adding 10-1, 16-9
    removing 10-3
  host tables, editing 10-2
  importing 14-9
  infrastructure 5-28
  listing 5-28
  loopback, See loopback, zones
  names, creating 14-5
  point of delegation 13-3
  primary server, setting 14-6
  resource records
    adding 16-2
    filtering 16-7
    protecting 16-3
    removing 16-4
  restricting hosts 5-30
  reverse, See reverse zones
  scavenging 7-2, 29-19
    start time, getting 29-21
  secondary
    expiration time 17-8
    refresh time 17-7
    retry time 17-7
  secondary name servers, adding 14-13
  serial numbers 14-6
  staged edit mode A-9
  subzones, delegating A-10
  synchronous edit mode A-10
  template, adding from 14-7
  transfers, See zone transfers
  TTL property, setting 17-6
  zone distributions, associating 14-25
  zone transfers, enabling 14-14
  zone-template command (CLI)
apply-to 14-4
create 14-4
   clone 14-4
set
   dist-map 14-25
zone templates
   applying to zones 14-4
   creating 14-2
   zone distributions, associating 14-25
zone transfers
   defined 13-5
   enabling 14-14
   forcing 7-2, 14-14
   forcing all 14-15
zone tree, viewing 5-28