CHAPTER 14

Support Tools and Advanced Concepts

This chapter contains information on, and explains the use of, tools that help you maintain Cisco Broadband Access Center (Cisco BAC) as well as speed and improve the installation, deployment, and use of this product.

This chapter discusses:

Note This section contains several examples of tool use. In many cases, the tool filenames include a path specified as *BPR_HOME*. This indicates the default home directory location.

Cisco BAC Tools

Cisco BAC provides automated tools that you use to perform certain functions more efficiently. Table 14-1 lists the various tools that this Cisco BAC release supports.

Table 14-1 Cisco BAC Tools

Tool	Description	Refer...
Configuration File Utility	Used to test, validate, and view Cisco BAC template and configuration files.	Using the Configuration File Utility for Template, page 5-32
Cisco BAC Process Watchdog	Interacts with the Cisco BAC watchdog daemon to observe the status of the Cisco BAC system components, and stop or start servers.	Using the Cisco BAC Process Watchdog from the Command Line, page 9-2
RDU Log Level Tool	Sets the log level of the RDU, and enables or disables debugging log output.	Using the RDU Log Level Tool, page 10-4

Table 14-1 Cisco BAC Tools (continued)

Tool	Description	Refer...
PacketCable Certificates Tool	Installs, and manages, the KDC certificates that are required by the KDC for its operation.	Using the PKCert.sh Tool, page 14-2
KeyGen Tool	Generates PacketCable service keys.	Using the KeyGen Tool, page 14-9
Changing Network Registrar Properties Tool	Used to change key configuration properties used by Cisco BAC extensions that are incorporated into the Cisco Network Registrar DHCP server.	Using the changeNRProperties.sh Tool, page 14-11
SNMP Agent Configuration Tool	Manages the SNMP agent.	Using the snmpAgentCfgUtil.sh Tool, page 10-10
Diagnostics Tool	Collects server data related to system performance and troubleshooting.	Troubleshooting Using the Diagnostics Tool, page 16-5
BundleState.sh Tool	Bundles diagnostics data related to server state for support escalations.	Bundling Server State for Support, page 16-10
Disk Space Monitoring Tool	Sets threshold values for one or more file systems. When these thresholds are surpassed, an alert is generated until additional disk space is available.	Using the disk_monitor.sh Tool, page 14-13

Using the PKCert.sh Tool

The PKCert tool creates the KDC certificate and its corresponding private key. It also allows you to verify certificate chains and copy and rename a certificate chain to the names required by the KDC.

Note This tool is available only when the KDC component is installed.

Running the PKCert Tool

Run the PKCert tool by executing the PKCert.sh command, which resides by default in the *BPR_HOME/kdc* directory.

Syntax Description

PKCert.sh *function option*

- *function*—Identifies the function to be performed. You can choose:
  - **-c**—Creates a KDC certificate. See Creating a KDC Certificate, page 14-3.
  - **-v**—Verifies and normalizes the PacketCable certificate set. See Validating the KDC Certificates, page 14-4.
  - **-z**—Sets the log level for debug output that is stored in the *pkcert.log* file. See Setting the Log Level for Debug Output, page 14-5.

  Note If you have trouble using these options, specify **-?** to display available help information.
- *option*—Implements optional functions, depending on the function you selected.

Creating a KDC Certificate

To create the KDC certificate:

Step 1 Change directory to */opt/CSCObac/kdc*.

Step 2 Run the PKCert.sh tool using this syntax:

PKCert.sh -s *dir* **-d** *dir* **-c** *cert* **-e -r** *realm* **-a** *name* **-k** *keyFile* [**-n** *serial#*] [**-o**]

- **-s** *dir*—Specifies the source directory
- **-d** *dir*—Specifies the destination directory
- **-c** *cert*—Uses the service provider certificate (DER encoded)
- **-e**—Identifies the certificate as a Euro-PacketCable certificate
- **-r** *realm*—Specifies the Kerberos realm for the KDC certificate
- **-a** *name*—Specifies the DNS name of the KDC
- **-k** *keyFile*—Uses the service provider private key (DER encoded)
- **-n** *serial#*—Sets the certificate serial number
- **-o**—Overwrites existing files

When a new certificate is created and installed, the new certificate identifies the realm in the subject alternate name field. The new certificate is unique to its current environment in that it contains the:

- KDC realm.
- DNS name associated with this KDC that the Multimedia Terminal Adapter (MTA) will use.

Examples

```
# ./PKCert.sh  -c "-s . -d /opt/CSCObac/kdc/<Operating System>/packetcable/certificates
-k CLCerts/Test_LSCA_privkey.der -c CLCerts/Test_LSCA.cer -r PCTEST.CISCO.COM -n 100
-a kdc.pctest.cisco.com -o"
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: .
Destination Directory: /opt/CSCObac/kdc/<Operating System>/packetcable/certificates
Private Key File: CLCerts/Test_LSCA_privkey.der
Certificate File: CLCerts/Test_LSCA.cer
Realm: PCTEST.CISCO.COM
Serial Number: 100
DNS Name of KDC: kdc.pctest.cisco.com
WARNING - Certificate File will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
File written: /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates/KDC_private_key.pkcs8
File written: /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates/KDC_private_key_proprietary.
File written: /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates/KDC_PublicKey.der
File written: /opt/CSCObac/kdc/<Operating System>/packetcable/certificates/KDC.cer
KDC Certificate Successfully Created at /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates/KDC.cer
```

This command creates the following files:

- */opt/CSCObac/kdc/<Operating System>/packetcable/certificates/KDC.cer*
- */opt/CSCObac/kdc/<Operating System>/packetcable/certificates/KDC_private_key.pkcs8*.

The KDC certificate will have a realm set to PCTEST.CISCO.COM, a serial number set to 100, and the fully qualified domain name (FQDN) of the KDC server set to kdc.pctest.cisco.com.

Validating the KDC Certificates

This command examines all files in the source directory specified and attempts to identify them as X.509 certificates. If legitimate X.509 certificates are found, the files are properly renamed and copied to the destination directory. An error is generated when more than one legitimate chain of certificates for a particular purpose (service provider or device) is identified. If this occurs, you must remove the extra certificate from the source directory and run the command again.

Note When you enter the **PKCert.sh -v -?** command, usage instructions for validating KDC certificates by using the PKCert tool appear.

To validate the KDC certificate:

Step 1 Change directory to */opt/CSCObac/kdc*.

Step 2 Run the PKCert.sh tool using this syntax:

PKCert.sh -v -s *dir* **-d** *dir* **-r** *dir* **-e**

- **-s** *dir*—Specifies the source directory
- **-d** *dir*—Specifies the destination directory
- **-o**—Overwrites any existing files

- **-r** *dir*—Specifies the reference certificate directory
- **-e**—Identifies the certificate as a Euro-PacketCable certificate

Verification is performed against reference certificates built into this package. If you specify the **-d** option, the certificates are installed in the target directory with name normalization.

Examples

```
# ./PKCert.sh  -v "-s /opt/CSCObac/kdc/TestCerts -d /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates -o"
Pkcert Version 1.0
Logging to pkcert.log
Output files will overwrite existing files in destination directory

Cert Chain(0)    Chain Type: Service Provider
[Local File]                          [Certificate Label]                    [PacketCable
Name]
CableLabs_Service_Provider_Root.cer   CableLabs_Service_Provider_Root.cer
Service_Provider.cer                  Service_Provider.cer
Local_System.cer                      Local_System.cer
KDC.cer                               KDC.cer

Cert Chain(1)    Chain Type: Device
[Local File]                          [Certificate Label]                    [PacketCable
Name]
MTA_Root.cer                          MTA_Root.cer
File written: /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates/CableLabs_Service_Provider_Root.cer
File written: /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates/Service_Provider.cer
File written: /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates/Local_System.cer
File written: /opt/CSCObac/kdc/<Operating System>/packetcable/certificates/KDC.cer

Service Provider Certificate Chain Written to Destination Directory
/opt/CSCObac/kdc/<Operating System>/packetcable/certificates


File written: /opt/CSCObac/kdc/<Operating System>/packetcable/certificates/MTA_Root.cer

Device Certificate Chain Written to Destination Directory /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates
```

Setting the Log Level for Debug Output

This command enables you to set the log level for debug output that is logged in *pkcert.log,* which resides in *BPR_HOME/kdc*. You can use the data in the log file to troubleshoot any problems that may have occurred while performing the requested tasks.

To set the log level for debug output:

Step 1 Change directory to */opt/CSCObac/kdc*.

Step 2 Run the PKCert.sh tool using this syntax:

PKCert.sh -s *dir* **-d** *dir* **-k** *keyFile* **-c** *cert* **-r** *realm* **-a** *name* **-n** *serial#* **-o** {**-z error | info | debug}**

- **-s** *dir*—Specifies the source directory

- **-d** *dir*—Specifies the destination directory
- **-k** *keyFile*—Uses the service provider private key (DER encoded)
- **-c** *cert*—Uses the service provider certificate (DER encoded)
- **-r** *realm*—Specifies the Kerberos realm for the KDC certificate
- **-a** *name*—Specifies the DNS name of the KDC
- **-n** *serial#*—Sets the certificate serial number
- **-o**—Overwrites existing files
- **-z**—Sets the log level for debug output that is stored in the *pkcert.log* file. The values you can choose are:
  - **error**—Specifies the logging of error messages.
  - **info**—Specifies the logging of informational messages.
  - **debug**—Specifies the logging of debug messages. This is the default setting.

Examples

Example 1

In this example, the log level is set for collecting error messages.

```
# ./PKCert.sh  -c "-s /var/certsInput -d /var/certsOutput -k
/var/certsInput/Local_System.der -c /var/certsInput/Local_System.cer -r PCTEST.CISCO.COM
-n 100 -a kdc.pctest.cisco.com -o -z error"
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: PCTEST.CISCO.COM
Serial Number: 100
DNS Name of KDC: kdc.pctest.cisco.com
Setting debug to error
WARNING - Certificate File will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

Copy KDC.cer to the KDC certificate directory (i.e. /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObac/kdc/solaris)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e.
/opt/CSCObac/kdc/solaris)
```

Example 2

In this example, the log level is set for collecting information messages.

```
# ./PKCert.sh -c "-s /var/certsInput
> -d /var/certsOutput
> -k /var/certsInput/Local_System.der
> -c /var/certsInput/Local_System.cer
> -r PCTEST.CISCO.COM
```

```
> -n 100
> -a kdc.pctest.cisco.com
> -o -z info"
 INFO [main] 2007-05-02 06:32:26,280 (PKCert.java:97) - Pkcert Version 1.0
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: PCTEST.CISCO.COM
Serial Number: 100
DNS Name of KDC: kdc.pctest.cisco.com
Setting debug to info
 INFO [main] 2007-05-02 06:32:26,289 (PKCCreate.java:69) - PKCCreate startup
WARNING - Certificate File will be overwritten
 INFO [main] 2007-05-02 06:32:26,291 (PKCCreate.java:341) - WARNING - Certificate File
will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

Copy KDC.cer to the KDC certificate directory (i.e. /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObac/kdc/solaris)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e.
/opt/CSCObac/kdc/solaris)
```

Example 3

In this example, the log level is set for debugging.

Note The sample output has been trimmed for demonstration purposes.

```
# ./PKCert.sh -c "-s /var/certsInput -d /var/certsOutput -k
/var/certsInput/Local_System.der -c /var/certsInput/Local_System.cer -r PCTEST.CISCO.COM
-n 100 -a kdc.pctest.cisco.com -o -z debug"
 INFO [main] 2007-05-02 06:32:06,029 (PKCert.java:97) - Pkcert Version 1.0
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: IPFONIX.COM
Serial Number: 100
DNS Name of KDC: bacdev3-dpe-4.cisco.com
Setting debug to debug
 INFO [main] 2007-05-02 06:32:06,038 (PKCCreate.java:69) - PKCCreate startup
WARNING - Certificate File will be overwritten
 INFO [main] 2007-05-02 06:32:06,039 (PKCCreate.java:341) - WARNING - Certificate File
will be overwritten
DEBUG [main] 2007-05-02 06:32:06,054 (PKCert.java:553) - Characters Read: 1218
DEBUG [main] 2007-05-02 06:32:06,056 (PKCert.java:583) - Binary File:
/var/certsInput/Local_System.der Read. Length: 1218
DEBUG [main] 2007-05-02 06:32:06,062 (PKCert.java:553) - Characters Read: 943
DEBUG [main] 2007-05-02 06:32:06,063 (PKCert.java:583) - Binary File:
/var/certsInput/Local_System.cer Read. Length: 943
```

```
DEBUG [main] 2007-05-02 06:32:06,064 (PKCert.java:455) - Jar File Path:
/opt/CSCObac/lib/pkcerts.jar
DEBUG [main] 2007-05-02 06:32:06,065 (PKCert.java:456) - Opened jar file:
/opt/CSCObac/lib/pkcerts.jar
DEBUG [main] 2007-05-02 06:32:06,067 (PKCert.java:460) - Jar entry unfiltered:
Tag_Packetcable_Tag/
DEBUG [main] 2007-05-02 06:32:06,068 (PKCert.java:460) - Jar entry unfiltered:
Tag_Packetcable_Tag/CableLabs_Service_Provider_Root.cer
...
DEBUG [main] 2007-05-02 06:32:06,115 (PKCert.java:472) - File:
Tag_Packetcable_Tag/Manu.cer
DEBUG [main] 2007-05-02 06:32:06,116 (PKCert.java:472) - File:
Tag_Packetcable_Tag/Service_Provider.cer
DEBUG [main] 2007-05-02 06:32:06,121 (PKCCreate.java:91) - Found 7 files in jar.
DEBUG [main] 2007-05-02 06:32:06,827 (KDCCert.java:98) - SP Cert subject name:
C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs Local System CA
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
DEBUG [main] 2007-05-02 06:32:07,687 (KDCCert.java:293) - Setting issuer to:
C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs Local System CA
DEBUG [main] 2007-05-02 06:32:07,699 (KDCCert.java:231) - DERVisibleToGeneral
org.bouncycastle.asn1.DERGeneralString@bd0b4ea6

DEBUG [main] 2007-05-02 06:32:07,700 (KDCCert.java:231) - DERVisibleToGeneral
org.bouncycastle.asn1.DERGeneralString@5035bc0

DEBUG [main] 2007-05-02 06:32:07,701 (KDCCert.java:231) - DERVisibleToGeneral
org.bouncycastle.asn1.DERGeneralString@5035bc0

DEBUG [main] 2007-05-02 06:32:07,703 (KDCCert.java:210) - DERCombineTagged [0] IMPLICIT
    DER ConstructedSequence
        ObjectIdentifier(1.3.6.1.5.2.2)
        Tagged [0]
            DER ConstructedSequence
                Tagged [0]
                    org.bouncycastle.asn1.DERGeneralString@5035bc0
                Tagged [1]
                    DER ConstructedSequence
                        Tagged [0]
                            Integer(2)
                        Tagged [1]
                            DER ConstructedSequence
                                org.bouncycastle.asn1.DERGeneralString@bd0b4ea6
                                org.bouncycastle.asn1.DERGeneralString@5035bc0

File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

Copy KDC.cer to the KDC certificate directory (i.e. /opt/CSCObac/kdc/<Operating
System>/packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObac/kdc/solaris)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e.
/opt/CSCObac/kdc/solaris)
```

Using the KeyGen Tool

The KeyGen tool is used to generate PacketCable service keys. The service keys are symmetric triple data encryption standard (triple DES or 3DES) keys (shared secret) required for KDC communication. The KDC server requires service keys for each of the provisioning FQDNs of the DPE. Any changes made to the DPE provisioning FQDN from the DPE command-line interface (CLI) requires a corresponding change to the KDC service key filename. This change is necessary because the KDC service key uses the DPE provisioning FQDN as part of its filename.

The KeyGen tool, which resides in the *BPR_HOME/kdc* directory, uses command-line arguments for the DPE provisioning FQDN, realm name, and a password, and generates the service key files.

Note When running this tool, remember to enter the same password that you used to generate the service key on the DPE (by using the **service packetCable** *1..1* **registration kdc-service-key** command from the DPE CLI). For information on setting this password, see the *Cisco Broadband Access Center DPE CLI Reference 4.2*.

The KDC server reads the service keys on startup. Any modification to the service keys requires that you restart the KDC server.

Syntax Description

keygen *options fqdn realm password*

- *options* are:
  - **-?**—Displays this usage message and exits the command.
  - **-v** or **-version**—Displays the version of this tool and exits the command.
  - **-q** or **-quiet**—Implements a quiet mode whereby no output is created.
  - **-c** or **-cms**—Creates a service key for the CMS system.
- *fqdn*—Identifies the FQDN of the DPE and is a required entry.
- *realm*—Identifies the Kerberos realm and is a required entry.
- *password*—Specifies the password to be used. This is also a required field. The password must be from 6 to 20 characters.

Three service key files are written in the KDC keys directory using this filename syntax:

mtafqdnmap,*fqdn*@*REALM*

mtaprovsrvr,*fqdn*@*REALM*

krbtgt,*REALM*@*REALM*

- *fqdn*—Identifies the FQDN of the DPE.
- *REALM*—Identifies the Kerberos realm.

The service key file always contains a version field of 0x0000.

Examples

```
# keygen dpe.cisco.com CISCO.COM changeme
```

When this command is implemented, these KDC service keys are written to the *BPR_HOME/kdc/<Operating System>/keys* directory:

```
mtafqdnmap,dpe.cisco.com@CISCO.COM
```

```
mtaprovsrvr,dpe.cisco.com@CISCO.COM
krbtgt,CISCO.COM@CISCO.COM
```

Restart the KDC, so that the new keys are recognized. Use this Cisco BAC process watchdog command to restart the KDC:

```
# /etc/init.d/bprAgent restart kdc
```

This example illustrates the generation of a CMS service key:

```
# keygen -c cms-fqdn.com CMS-REALM-NAME changeme
```

When this command is implemented, this CMS service key is written to the *BPR_HOME/kdc/<Operating System>/keys* directory.

```
cms,cms-fqdn.com@CMS-REALM-NAME
```

Verifying the KDC Service Keys

Once you generate the service keys on the KDC and the DPE, verify if the service keys match on both components.

The KeyGen tool requires you to enter the same password that you used to generate the service key on the DPE using the **service packetCable** *1..1* **registration kdc-service-key** command. Once you set this password on the DPE, you can view the service key from the *dpe.properties* file, which resides in the *BPR_HOME/dpe/conf* directory. Look for the value against the */pktcbl/regsvr/KDCServiceKey=* property.

For example:

```
# more dpe.properties
/pktcbl/regsvr/KDCServiceKey=2e:d5:ef:e9:5a:4e:d7:06:67:dc:65:ac:bb:89:e3:2c:bb:
71:5f:22:bf:94:cf:2c
```

Note The output of this example has been trimmed for demonstration purposes.

To view the service key generated on the KDC, run the following command from the *BPR_HOME/kdc/<Operating System>/keys* directory:

od -Ax -tx1 mtaprovsrvr,*fqdn@REALM*

- *fqdn*—Identifies the FQDN of the DPE.
- *REALM*—Identifies the Kerberos realm.

The output that this command generates should match the value of the */pktcbl/regsvr/KDCServiceKey=* property in the *dpe.properties* file.

For example:

```
# od -Ax -tx1 mtaprovsrvr,dpe.cisco.com@CISCO.COM
0000000 00 00 2e d5 ef e9 5a 4e d7 06 67 dc 65 ac bb 89
0000010 e3 2c bb 71 5f 22 bf 94 cf 2c
000001a
```

In the examples shown here, note that the service key generated at the KDC matches the service key on the DPE.

Using the changeNRProperties.sh Tool

The Cisco BAC installation program establishes values for configuration properties used by Cisco BAC extensions that are incorporated into the Network Registrar DHCP server. You use the **changeNRProperties.sh** command, which is found in the *BPR_HOME/cnr_ep/bin* directory, to change key configuration properties.

Invoking the script without any parameters displays a help message listing the properties that can be set.

To run this command:

Step 1 Change directory to *BPR_HOME/cnr_ep/bin*.

Step 2 Run the **changeNRProperties.sh** command using this syntax:

changeNRProperties.sh *options*

Where *options* are:

- **-help**—Displays this help message. The **-help** option must be used exclusively. Do not use this with any other option.
- **-ep enabled | disabled**—Enables or disables the PacketCable property. Enter **-ep enabled** to enable the property, and **-ep disabled** to disable it.
- **-ec enabled | disabled**—Enables or disables the CableHome property. Enter **-ec enabled** to enable the property, and **-ec disabled** to disable it.
- **-d**—Displays the current properties. The **-d** option must be used exclusively. Do not use this with any other option.
- **-s** *secret*—Identifies the Cisco BAC shared secret. For example, if the shared secret is the word *secret*, enter **-s secret**.
- **-f** *fqdn*—Identifies the RDU FQDN. For example, if you use rdu.example.com as the fully qualified domain name, enter **-f rdu.example.com**.
- **-p** *port*—Identifies the RDU port you want to use. For example, if you want to use port number 49187, enter **-p 49187**.
- **-r** *realm*—Identifies the PacketCable realm. For example, if your PacketCable realm is EXAMPLE.COM, enter **-r EXAMPLE.COM**.

  Note You must enter the realm in uppercase letters.

- **-g** *prov_group*—Identifies the provisioning group. For example, if you are using provisioning group called group1, enter **-g group1**.
- **-t 00 | 01**—Identifies whether or not the PacketCable TGT is set to off or on. For example, to set the TGT to off, enter **-t 00;** to set this to on, enter **-t 01**.
- **-a** *ip*—Identifies the PacketCable primary DHCP server address. For example, if the IP address of your primary DHCP server is 10.10.10.2, enter **-a 10.10.10.2**.
- **-b** *ip*—Identifies the PacketCable secondary DHCP server address. For example, if the IP address of your secondary DHCP server is 10.10.10.4, enter **-b 10.10.10.4**. You can also enter **-b null** to set a null value, if appropriate.
- **-y** *ip*—Identifies the PacketCable primary DNS server address. For example, if the IP address of the PacketCable primary DNS server is 10.10.10.6, enter **-y 10.10.10.6**.

- **-z** *ip*—Identifies the PacketCable secondary DNS server address. For example, if the IP address of your secondary DNS server is 10.10.10.8, enter **-z 10.10.10.8**. You can also enter **-z null** to set a null value, if appropriate.
- **-o** *prov_ip man_ip*—Sets the management address to use for communication with the DPE identified by the given provisioning address. For example, if the IP address of your provisioning group is 10.10.10.7, enter **-o 10.10.10.7 10.14.0.4**. You can also enter a null value, if appropriate; for example, **-o 10.10.10.7 null**.

Step 3 Restart the DHCP server.

Examples

This is an example of changing the Network Registrar extensions by using the NR Extensions Properties tool:

```
# /opt/CSCObac/cnr_ep_bin/changeNRProperties.sh -g primary1
Current NR Properties:
RDU Port: 49187
RDU FQDN: rdu.example.com
Provisioning Group: primary1
Shared Secret: fggTaLg0XwKRs
PacketCable Enable: enabled
CableLabs client TGT: 01
CableLabs client Realm: EXAMPLE.COM
CableLabs client Primary DHCP Server: 10.10.1.2
CableLabs client Secondary DHCP Server: NOT SET
CableLabs client Primary DNS Server: 10.10.1.2
CableLabs client Secondary DNS Server: NOT SET
```

Note You must restart your NR DHCP server for the changes to take effect.

This is an example of viewing the current properties:

```
# /opt/CSCObac/cnr_ep_bin/changeNRProperties.sh -d
Current NR Properties:
RDU Port: 49187
RDU FQDN: rdu.example.com
Provisioning Group: primary1
Shared Secret: fggTaLg0XwKRs
PacketCable Enable: enabled
CableLabs client TGT: 01
CableLabs client Realm: EXAMPLE.COM
CableLabs client Primary DHCP Server: 10.10.1.2
CableLabs client Secondary DHCP Server: NOT SET
CableLabs client Primary DNS Server: 10.10.1.2
CableLabs client Secondary DNS Server: NOT SET
```

Using the disk_monitor.sh Tool

Monitoring available disk space is an important system administration task. You can use a number of custom written scripts or commercially available tools to do so.

The **disk_monitor.sh** command, which resides in the *BPR_HOME/rdu/samples/tools* directory, sets threshold values for one or more file systems. When these thresholds are surpassed, an alert is generated through the Solaris syslog facility, at 60-second intervals, until additional disk space is available.

Note We recommend that, at a minimum, you use the **disk_monitor.sh** script to monitor the *BPR_DATA* and *BPR_DBLOG* directories.

Syntax Description

disk_monitor.sh *filesystem-directory x [filesystem-directory* x*]*

- *filesystem-directory*—Identifies any directory in a file system to monitor.
- *x*—Identifies the percentage threshold applied to the specified file system.
- *filesystem-directory**—Identifies multiple file systems.
- *x**—Specifies percentage thresholds to be applied to multiple file systems.

Examples

Example 1

This example specifies that a notification be sent out when the */var/CSCObac* file system reaches 80 percent of its capacity.

```
# ./disk_monitor.sh /var/CSCObac 80
```

When the database logs disk space reaches 80-percent capacity, an alert similar to the following one is sent to the syslog file:

```
Dec 7 8:16:06 perf-u80-1 BPR: [ID 702911 local6.warning] File system /var/bpr usage is 81%
(threshold is 80%)
```

Example 2

This example describes how you can run the disk_monitor.sh tool as a background process. Specifying an ampersand (&) at the end of the command immediately returns output while running the process in the background.

```
# ./disk_monitor.sh /var/CSCObac 80 &
1020
```