Configuration and Administration Guide for Cisco Unified Customer Voice Portal 9.0(1)

First Published: July 06, 2012

Last Modified: June 22, 2016

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

Preface xv

Conventions xviii

Support xix

Feedback xix

PART I

Introduction to Cisco Unified Customer Voice Portal (Unified CVP) Configuration 1

CHAPTER 1

Configuration Overview 3

Call Flow Model Prerequisites 3

Required Skills 3

Design Prerequisites 3

Additional Information 4

Before You Begin 4

Developer Services 4

Network Information 5

H.323 Deprecation 5

SIP Proxy 6

Ring No Answer Settings with SIP 6

Unified CVP Installation 6

Routing Calls Through the Network to the VRU 7

Values for Unified CVP 7

Apply Contact Center Gateway Debug Settings 7

Network VRU Types 8

SIP DN Pattern Matching Algorithm 9

Additional Information 10

Operations Console 13

CHAPTER 2 High-Level Configuration Instructions 15

Best Practices for Order of Device Operations 16

Standalone Call Flow Models 16

VXML Server Standalone Call Flow Model 17

Unified CVP Standalone with ICM Lookup Call Flow Model 18

Set Up Unified CVP VXML Server Standalone Call Flow Models 20

Example Gateway Settings for Standalone Call Flow Model 23

Example Dial-Peer for Standalone Call Flow Model 24

Unified CVP Call Director Call Flow Models 25

Unified CVP Call Director Call Flow Model Unified ICME 25

Unified CVP Call Director Call Flow Model Unified ICMH 26

Set Up Call Director Call Flow Model 27

DNS Zone File Configuration for Call Director Call Flow Model 36

DNS Zone File Linux Configuration Example 36

DNS Zone File MS DNS Configuration Example 37

Unified CVP Comprehensive Call Flow Models 37

Unified CVP Comprehensive Call Flow Model Unified ICME 38

Unified CVP Comprehensive Call Flow Model Unified ICMH 40

Set Up Comprehensive Call Flow Model Using SIP for Both ICME and ICMH 42

DNS Zone File Configuration for Comprehensive Call Flow Model 62

DNS Zone File Linux NAMED Configuration Example 62

DNS Zone File MS DNS Configuration Example 62

Unified CVP VRU Call Flow Models with NIC Routing 63

Type 8 Unified CVP VRU-Only Call Flow Model Unified ICME 64

Type 8 Unified CVP VRU-Only Call Flow Model Unified ICMH 65

Set Up Type 8 Unified CVP VRU-Only Unified ICME and Unified ICMH 66

Type 3 or 7 Unified CVP VRU-Only Call Flow Model Network VRU Unified ICMH 73

Set Up Type 3 or 7 VRU-Only Call Flow Model Network VRU Unified ICMH 75

VoiceXML Gateway Configuration: Example Gateway Settings for Type 3 or Type 7 78

VoiceXML Gateway Configuration: Example of Dial-Peer for ICM VRU Label for Type 3 or Type 7 79

Unified CVP Comprehensive Call Flows for Pre-Routed Calls 80

Calls Which Arrive at Unified ICME Through a Pre-Route-Only NIC 80
Calls Which Are Originated by Unified CM 81
Calls Which Are Originated by an ACD or Call Routing Interface 83
Common Unified ICMH Configuration for Unified CVP Switch Leg 84
Set Up Common Unified ICMH Unified CVP Switch Leg 84
Define Unified CVP ECC Variables 86
Metadata ECC Variable 92
Metadata ECC Variable 92
Common Configuration for Differentiating VRUs Based on Dialed Number 93
Local SRV File Configuration Example for SIP Messaging Redundancy 93

CHAPTER 3 **Writing Scripts for Unified CVP 95**
Before You Begin 95
Making Unified ICME Work with Unified CVP 96
Scripts to Access Unified CVP From Unified ICME 96
Unified ICME Scripting 97
Unified CVP VoiceXML Scripting 97
Micro-Application Use versus VXML Server Use 98
Scripting for Unified CVP with Unified ICME 98
Micro-Applications 98
Simple Example Script: Welcome to XYZ Corporation 99
Unified ICME Unified CVP Micro-app Connection 100
Information Exchange Between Unified ICME and Unified CVP 102
Unified ICME Data Handling 102
Unified CVP Script Error Checking 103
Unified ICME Setup 108
Writing Unified ICME Applications for Unified CVP 108
Configure a Unified CVP Network VRU Script 108
Run External Script Node That Accesses a Unified CVP Micro-Application 109
Unified CVP Micro-Applications 110
Micro-Applications Automatic Speech Recognition and Text-to-Speech 110
Micro-Applications External Voice XML 111
Dynamic Audio File Support for Micro-Applications 111
Example of Using the Dynamic Prompt 112
Default Media Server for Micro-Applications 113

Capture Micro-Application 114
Play Media Micro-Application 115
Configure Network VRU Script for Play Media 115
Configure Play Media Micro-Application to Use Streaming Audio 117
Play Media Examples: Play Welcome Message 120
Play Data Micro-Application 122
Play Data and Data Storage 122
Configure Network VRU Script Settings for Play Data Micro-Application 122
Play Back Types for Voice Data 124
Play Data Configuration Examples 131
Get Digits Micro-Application 132
Configure Network VRU Script Settings for Get Digits Micro-Application 132
Get Digits Configuration Examples 134
Get Digits and Digit Entry Completion 138
Menu Micro-Application 139
Configure Network VRU Script Settings for the Menu Micro-Application 139
Menu Configuration Examples 141
Menu and Digit Entry Completion 144
Get Speech Micro-Application 145
Get Speech and Grammar Specification 145
Configure Network VRU Script Settings for the Get Speech Micro-Application 145
Get Speech Configuration Examples 148
Get Speech and DTMF Input Collection 151
Get Speech Data Format 152
Get Speech and Entry Completion 153
Get Speech and External VoiceXML 154
Passing Information to the External VoiceXML 154
ECC Variable Array Formula 156
Passing Data Back to Unified ICME with External Voice XML 157
External VoiceXML File Contents 160
Type-Ahead Support for ASR 162
Scripting for Unified CVP with Call Studio 162
About Call Studio 162
Call Studio ReqICMLabel Element to Pass Data 163

Integrate Call Studio Scripts with Unified ICME Scripts - Traditional Method 166
Integrate Call Studio Scripts with Unified ICME Scripts - Simplified Method 168
Call Studio Scripts in Unified CVP 169
Deploy Call Studio Scripts Using Call Studio 170
Deploy Call Studio Scripts Using the Operations Console 170
System Media Files 171

CHAPTER 4 **Cisco Serviceability Tools** 189
Web Services Manager 189
Create a WSM User 189
Unified System CLI 190
Access Unified System CLI and Its Help 191
Unified System CLI: System Mode 193
System CLI Automated Execution 193
System CLI Remote Execution 194
Help for the System CLI 194
System CLI Troubleshooting 195
System CLI Commands and Parameters 196
Details for Specific Options 197
Configure Analysis Manager with Unified CVP 198
System CLI Commands Map to IOS CLI Commands 199

CHAPTER 5 **Unified CVP Security** 201
Secure Communications Between Unified CVP Components 201
Before You Begin 202
Secure JMX Communications Between Unified CVP Components 202
Exchange Certificates Between Systems 204
Enable Security on Unified CVP Devices 206
Certificate Authority Signed Certificates 207
Add a Certificate Signed by a Certificate Authority to the Keystore 207
Add a Certificate Signed by a Certificate Authority for HTTPS Web Access 208
Secure Communications Between Unified CVP and IOS Devices 209
Export the Operations Console Certificate to a File 209
Import the Operations Console Certificate to a Gateway 210
Configure Secure Access to the Router Using Protocol SSH Version 2 210

Secure Communications Between VoiceXML Gateways and VXML Server 212
HTTPS support for unified CVP 212
Set Up Tomcat to Present CA-Signed Certificates to Inbound HTTPs Clients 212
Secure Communications Between Unified CVP and IOS Devices 213
Sensitive Customer Information 214

PART II Configuration Detail of Unified CVP Components 217

CHAPTER 6 Configure VXML Solution 219
About Unified CVP VXML Server 220
About VoiceXML Service 220
About Call Studio 221
VXML Server Reporting 222
Unified CVP VXML Server 223
Unified CVP VXML Server with a Call Server Association 224
Associate a Unified CVP VXML Server with a Call Server 224
Security for a Unified CVP VXML Server 225
Enable Security for a Unified CVP VXML Server 225
Configure Unified CVP VXML Server Applications: Filters 225
Enable Reporting for the Unified CVP VXML Server 226
QoS for the Unified CVP VXML Server 226
Create Policy Based QoS 227
Unified CVP VXML Server (Standalone) 227
Unified CVP VXML Server with Unified ICME 228
Integrate VoiceXML Scripts with Unified ICME Scripts 228
Correlate Unified CVP and Unified ICME Logs with Unified CVP VXML Server Logs 229
Error Codes for Unified CVP VXML Server 230
Unified CVP VXML Server (Standalone) Server Solutions 230
Configure the Unified CVP VXML Server (Standalone) with ICM Lookup Call Flow Model 231
Configure the Unified CVP VXML Server (Standalone) Call Flow Model (Without ICM Lookup) 232
Activate the Gateway Configuration 234
Takeback and Transfer in VoiceXML Scripts 234

Configure Two B-Channel Transfer 234
Configure Hookflash Relay 236
Configure SIP REFER 237
Configure the Gateway for TDM to IP Calls 238
Pass Data to Unified ICME 239
Configure the Connections 239
Configure the Gateway for IP to TDM Calls 240
Configure a Cisco Multiservice IP-to-IP Gateway for Unified CM Connections 241
SNMP Monitoring for the Unified CVP VXML Server 241

CHAPTER 7 **Configure Unified CVP Logging and Event Notifications** 243
Using Syslog 243
Using Logs to Interpret Events 243
Editing the Log Messages XML File 244
Unified CVP Event Severity Levels 245
VoiceXML Logs 246
About VoiceXML Logs 246
Correlate Unified CVP/Unified ICME Logs with VXML Server Logs 246
About Unified CVP VXML Server Logs 247
About VoiceXML Application Logging 248
About Event Statistics 249
Infrastructure Statistics 250
ICM Service Call Statistics 253
IVR Service Call Statistics 255
SIP Service Call Statistics 258
Gateway Statistics 260
Procedure 260
Gateway Statistics 261
Trunk Utilization Reporting 261
RAI Information on SIP OPTIONS (CVP Server Group Heartbeats) 262
Unified CVP VXML Server Statistics 262
Reporting Server Statistics 264
Unified CVP SNMP-Raise/Clear Mappings 265

PART III **Configuration Detail of Non-Unified CVP Components** 271

CHAPTER 8 **Cisco Unified ICME Warm Consult Transfer/Conference to Unified CVP** 273

About Unified ICME Warm Consult Transfer/Conference to Unified CVP Feature 273

Configure Unified ICME Warm Consult Transfer/Conference to Unified CVP Feature 274

Minimal Component Version Required 275

Warm Transfer Feature with SIP Calls 275

Set Up Unified ICME Warm Consult Transfer 276

CHAPTER 9 **Configure Unified CM** 279

Configure Unified CM for Use with Unified CVP 279

Prerequisite Configuration 279

Configure a Unified CM Server 279

CHAPTER 10 **Configure the SIP Devices** 281

Configure a SIP Proxy Server 281

Load-Balancing SIP Calls 282

Set Up the Ingress Gateway to Use Redundant Proxy Servers 283

Set Up the Unified CVP Call Server with Redundant Proxy Servers 283

Cisco Unified SIP Proxy (CUSP) Configuration 283

REFER Transfers 286

Configure Custom Streaming Ringtones 287

Procedure 287

sendtooriginator Setting in the SIP Service 289

100rel and SIP Outbound Dialer 290

Expiration Timeout Setting Using Dialed Number Patterns in SIP 290

Critical Error Message Playback on Abnormal Disconnects with SIP 290

How Critical Error Message Playback Works 291

Delivering SIP Incoming UUI to Unified ICME Routing Script 291

UUI Delivery 292

SIP Hookflash 292

SIP Hookflash Configuration 293

Pass User-to-User Information 295

Pass Information with SIP Headers 296

Configure SIP Headers 297

Example Dial-Peer Configuration for Connecting the Cisco IOS Gateway to Video Media Server 298

CHAPTER 11 **Transfer and Queue Calls with Unified CVP** 299

IVRs From Unified ICME's Perspective 299

Transfer Calls Using Unified CVP in Comprehensive Mode 300

Transfer Calls Using SIP Service 300

Example: Transfer to a Label 300

Example: Queue and Transfer to a Skill Group 302

Example: Network Transfer Script 305

Transfer a Call From Agent to Agent 305

Configure Network Transfer From IP Phone 306

Configure Network Transfer From CTI OS Agent Desktop 306

IP Transfer Example 306

CLI Field on Outgoing Transfers 307

Configure CLI Override 307

Configure Unified CCE Re-Route on No Answer for Unified CVP 308

Summary 308

Re-Route on No Answer Operation for Unified CCE with Unified IP IVR 308

Re-Route on No Answer Operation with Unified CVP 308

Re-Route on No Answer Agent Desk Settings Configuration 309

Router Requery Configuration 309

Limitations 312

Configure Re-Route on No Answer for Unified CM with Unified CVP 312

Limitations 312

Call Survivability 313

Install Call Survivability Script 313

Configure the Gateway for Call Survivability 315

Call Survivability Examples 317

Enhanced Location Call Admission Control 319

ELCAC Topic Definitions 319

Configuration of ELCAC Queue-at-the-Edge 320

Configure Locations-Based Call Admission Control 326

Configure Unified CM Service Configuration Settings 326

Unified CVP Bandwidth Utilization 326

VoiceXML Documents 327
Prompt Retrieval 327
Gateway Prompt Caching Considerations 327
Configure Caching on the Gateway 328
Determine If the Gateway Is Caching Properly 328
UUI to Be Used as the Correlation ID 329
How It Works 329
Configure the Gateway 329
Debugging Tips 330
Debug Trace Settings for the Gateway 330
GTD Values in the Gateway Log 330
External Transfers in Unified ICME 330
Unified ICME Script Label for Outpulse Transfer 330
Unified ICME Script Label for Two B-Channel Transfer 331
Unified ICME Script Label for Hookflash Transfer 331
Takeback and Transfer in VoiceXML Scripts 332
Configure Two B-Channel Transfer 332
Configure Hookflash Relay 334
Configure SIP REFER 335
Courtesy Callback 335
Callback Criteria 336
Modifiable Example Scripts and Sample Audio Files 336
Courtesy Callback Configuration 337
Configure the Ingress Gateway for Courtesy Callback 338
Configure the VXML Gateway for Courtesy Callback 339
Configure the Reporting Server for Courtesy Callback 341
Configure the Media Server for Courtesy Callback 344
Configure Call Studio Scripts for Courtesy Callback 344
CCE Script for Courtesy Callback 348
Overview of CCE Script Configuration for Courtesy Callback 349
Configure the CCE Script for Courtesy Callback 350
Multicast Music on Hold (MMoH) 353
Post Call Survey for SIP 354
Configure the Unified CVP Call Server for Post Call Survey 354
Configure Unified CCE for Post Call Survey 355

CHAPTER 12 **Configure High Availability for Unified CVP** 357

Server Groups 357

Configure Server Groups 357

Server Groups Diagnostics 360

Redundancy and Failover for Unified CVP 360

Redundancy for VXML Server Applications 360

Redundancy for Micro-App-Based Applications 361

IVR Service Failover Mechanism 361

Configure a Speech Server 362

Application Control Engine for Load Balancing in Unified CVP 363

General Probes 365

Unified CVP Media Servers 365

ASR/TTS Servers 366

ASR and TTS Server Location Setup 368

Specify ASR TTS Server Location Globally on the Gateway 368

Specify ASR and TTS Server Location with Individual VoiceXML Document 369

com.cisco.tts-server 369

com.cisco.asr-server 369

Set Up VoiceXML Document Properties 370

Example Gateway Configuration for MRCPv2 with Failover 370

Unified CVP Call Servers 371

Unified CVP VXML Servers 371

ACE Sample Configuration 373

CHAPTER 13 **Configure the Media Servers** 377

Media File Overview 377

Media Server 377

Media File Names and Types 378

Location of Media Files 378

Media File Address 379

Locale Backward Compatibility 382

System Media Files 383

Miscellaneous Files 400

System Media File Error Messages 402

Configure a Media Server 403

CHAPTER 14 Java Runtime Environment Minor Update 405

APPENDIX A Using the Helix Server 407

Broadcast Stream Using the Helix Server 407

Procedure 407

Preface

This document describes how to configure, run, and administer Cisco Unified Customer Voice Portal (Unified CVP).

Important Assumptions

The following bullets refer to user documentation available from the Cisco website.

Understanding Unified CVP and planning your Unified CVP solution are important parts of setting up the product. Therefore, this document assumes that you have:

- Carefully read Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND). This guide provides the following foundational material:
  - Unified CVP introductory material
  - Descriptions of Unified CVP Call Flow Models (deployment models)
  - Design and planning material
- Created the simplified Unified CVP lab setup and performed the basic exercises as described in Cisco Unified Customer Voice Portal.

This document is intended for Call Center managers, Unified CVP system managers, Cisco Unified Intelligent Contact Management Enterprise (Unified ICME)/Cisco Unified ICM Hosted (Unified ICMH) system managers, VoIP technical experts, and IVR application developers. Readers of this document should already have a general understanding of the Unified ICME products, as discussed in the *Pre-Installation Planning Guide for Cisco Unified Intelligent Contact Management Enterprise* and the *Product Description Guide for Cisco Unified ICM Hosted* . Readers should be familiar with general Unified ICME installation and setup procedures.

Organization

The manual is divided into the following parts and chapters.

Table 1: Part 1: Introduction to Unified CVP

Chapter	Description
Configuration Overview, on page 3	Provides background information, prerequisites, and references for understanding and planning a Unified CVP implementation, and check-off lists for the various call flow model options available for Unified CVP.
High-Level Configuration Instructions, on page 15	Provides high-level call flow model overviews, high-level sets of instructions for configuring specific call flow models, sample gateway and other configurations, and additional configuration task outlines. Provides cross-references to more detailed information.
Writing Scripts for Unified CVP, on page 95	Discusses Unified ICME configuration and script editing. This chapter also provides a detailed description of the Unified CVP micro-applications.
Cisco Serviceability Tools, on page 189	Describes serviceability features on Unified CVP on non-Windows boxes.
Unified CVP Security, on page 201	Describes Unified CVP security.

Table 2: Part 2: Configuration Detail of Customer Voice Portal Components

Chapter	Description
Configure VXML Solution, on page 219	Describes how to configure a VXML Server in either Standalone or Comprehensive deployment models.
Configure Unified CVP Logging and Event Notifications, on page 243	Describes how to configure external events and how to set trace levels and log levels.

Table 3: Part 3: Configuration Detail of Non-Customer Voice Portal Components

Chapter	Description
Cisco Unified ICME Warm Consult Transfer/Conference to Unified CVP, on page 273	Details how to configure the Unified ICME Warm Consult Transfer/Conference to Unified CVP feature.
Configure Unified CM, on page 279	Describes how to use the Unified CVP Operations Console menus to add a pre-configured Unified CM Server to the network map.
Configure the SIP Devices, on page 281	Provides the configuration tasks required for the SIP devices.

Chapter	Description
Transfer and Queue Calls with Unified CVP, on page 299 Transfer and Queue Calls with Unified CVP	Contains additional information about transferring and queuing calls using Unified CVP.
Configure High Availability for Unified CVP, on page 357	Provides information about how to accommodate load balancing and redundancy in Unified CVP deployments.
Configure the Media Servers, on page 377	Provides information about Unified CVP media file handling and details about the system media files distributed with the Unified CVP solution.

Table 4: Appendix

Appendix	Description
Appendix A, "Using the Helix Server"	Contains high-level steps for configuring a Helix server to create a broadcast stream.

Related Information

Note Planning your Unified CVP solution is an important part of the process in setting up Unified CVP. Cisco recommends that you read Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND) guide *before* configuring your Unified CVP solution.

The *Planning Guide for Cisco Unified Customer Voice Portal* has been incorporated into the SRND document.

Unified CVP provides the following documentation:

- Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND) provides design considerations and guidelines for deploying contact center voice response solutions based on Cisco Unified Customer Voice Portal (CVP) releases.
- Getting Started with Cisco Unified Customer Voice Portal provides instructions to create a simplified Unified CVP lab setup and perform basic call flow model exercises.
- Configuration and Administration Guide for Cisco Unified Customer Voice Portal describes how to configure, run, and administer the Cisco Unified CVP product, including associated configuration.
- Element Specifications for Cisco Unified CVP VXML Server and Unified Call Studio describes the settings, element data, exit states, and configuration options for Elements.
- Installation and Upgrade Guide for Cisco Unified Customer Voice Portal describes how to install Unified CVP software, perform initial configuration, and upgrade.
- Operations Console Online Help for Cisco Unified Customer Voice Portal describes how to use the Operations Console to configure Unified CVP solution components.

Note There is a printable (PDF) version of the *Operations Console online help*. See the Operations Console User Guide for Cisco Unified Customer Voice Portal. The user guide also explains how to log into the Operations Console.

- Port Utilization Guide for Cisco Unified Customer Voice Portal describes the ports used in a Unified CVP deployment.
- Programming Guide for Cisco Unified CVP VXML Server and Unified Call Studio describes how to build components that run on the Cisco Unified VXML Server.
- User Guide for Cisco Unified CVP VXML Server and Unified Call Studio describes the functionality of Call Studio including creating projects, using the Call Studio environment, and deploying applications to the Unified CVP VXML Server.
- Reporting Guide for Cisco Unified Customer Voice Portal describes the Reporting Server, including how to configure and manage it, and discusses the hosted database.
- Say It Smart Specifications for Cisco Unified CVP VXML Server and Unified Call Studio describes in detail the functionality and configuration options for all Say It Smart plugins included with the software.
- Troubleshooting Guide for Cisco Unified Customer Voice Portal describes how to isolate and solve problems in the Unified CVP solution.

For additional information about Unified ICME, see the Cisco web site listing Unified ICME documentation.

- Conventions, page xviii
- Support, page xix
- Feedback, page xix

Conventions

This manual uses the following conventions:

Convention	Description
boldface font	Boldface font is used to indicate commands, such as user entries, keys, buttons, and folder and submenu names. For example: • Choose **Edit** > **Find**. • Click **Finish**.

Convention	Description
italic font	Italic font is used to indicate the following: • To introduce a new term. Example: A *skill group* is a collection of agents who share similar skills. • For emphasis. Example: *Do not* use the numerical naming convention. • A syntax value that the user must replace. Example: IF (*condition, true-value, false-value*) • A book title. Example: See the *Cisco CRS Installation Guide*.
`window font`	Window font, such as Courier, is used for the following: • Text as it appears in code or that the window displays. Example: `<html><title>Cisco Systems,Inc. </title></html>`
`< >`	Angle brackets are used to indicate the following: • For arguments where the context does not allow italic, such as ASCII output. • A character string that the user enters but that does not appear on the window such as a password.

Support

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

Feedback

You can provide comments about this document by sending email to the following address:

ccbu_docfeedback@cisco.com

We appreciate your comments.

PART I

Introduction to Cisco Unified Customer Voice Portal (Unified CVP) Configuration

- Configuration Overview, page 3
- High-Level Configuration Instructions, page 15
- Writing Scripts for Unified CVP, page 95
- Cisco Serviceability Tools, page 189
- Unified CVP Security, page 201

CHAPTER 1

Configuration Overview

This chapter presents general Unified CVP configuration information and prerequisites needed before you begin configuring your Unified CVP solution. At the end of the chapter is a table summarizing the Unified CVP call flow models (deployment models) and links to the high-level configuration instructions for each call flow model, found in the guide.

- Call Flow Model Prerequisites, page 3
- Before You Begin, page 4
- Additional Information, page 10
- Operations Console, page 13

Call Flow Model Prerequisites

This topic describes information you need and tasks you should perform before you select one of the call flow model high-level configuration procedures and attempt to implement it.

Required Skills

The information in this chapter *assumes* that you are already familiar with the following:

- Configuring Cisco Gateways
- The ICM Configuration Manager and ICM Script Editor tools for call center operations and management
- Configuring Unified CM

Design Prerequisites

Understanding Unified CVP and planning your Unified CVP solution are important parts of setting up the product. Therefore, this manual assumes that you have completed the following prerequisite reading and design work:

- Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND)

- Study of the Unified CVP overview material and the detailed descriptions of the various call flow models.
- Careful examination of the design information in the SRND.
- Choose the appropriate call flow model for your desired Unified CVP implementation.

  For more information, See Getting Started with Cisco Unified Customer Voice Portal
- Create the simplified all-in-one-box step-by-step call model examples.
- Use the troubleshooting information to experiment with the working examples.

Additional Information

In addition to design information, you need the following configuration reference material:

- To successfully configure Unified ICME and use its features in conjunction with Unified CVP, see the following guides:
  - ICM Configuration Guide for Cisco ICM Enterprise Edition
  - Scripting and Media Routing Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted
  - The Unified ICME documentation
- Review the complete list of Unified CVP solution documentation (see Preface, Related Documentation), and then make any additional preparations

Other information included in this guide:

- For information on writing scripts for Unified ICME use with Unified CVP, see Writing Scripts for Unified CVP.
- For information on how to use the available scripting interface to create VoiceXML Scripts, see Integrate VoiceXML Scripts with Unified ICME Scripts .

Note For details on components required for your Unified CVP solution, but not covered or referenced in this chapter (such as the TDM side of the Voice Gateway configuration), see the component's documentation.

Before You Begin

The following topics relate to configuring your Unified CVP solution and should be examined before you begin following one of the specific call flow model configuration outlines in this chapter.

Developer Services

Cisco Technical Support is limited to standard Cisco product installation/configuration, and Cisco-developed applications. Questions and/or support issues related to such items as Call Studio scripting or ASR grammar are **not** covered by Cisco Technical Support.

Developers using these and similar components might be interested in joining the Cisco Developer Services Program. The Developer Services Program provides formalized services for Cisco Systems interfaces to enable developers, customers, and partners in the Cisco Technology Developer Program to accelerate their delivery of compatible solutions. The Developer Services engineers are an extension of the product technology engineering teams. They have direct access to the resources necessary to provide expert support in a timely manner.

A separate service agreement and subscription fee is required to participate in the Developer Services Program. For information on how to subscribe, go to **Getting Started** on the Developer Support Web site at http://developer.cisco.com/web/cdc.

Network Information

In order to use the information in this chapter, you need to know the following:

- The Unified CVP call flow model you will be implementing.

Note Be aware that some call flow model names have changed and new call flow models have been added since the last release. Refer to Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND).

- The network topology for your system, including addresses and names of the solution components.
- The failover strategy for Gateways, Unified CVP components, and media servers.
- The strategy for inbound call routing (that is, dial-peers versus Proxy Server).
- The naming resolution system for Gateways (DNS versus configured on the Gateway).
- Naming schemes to be used for Unified ICME PGs, Peripherals, and routing clients.
- If using a VRU other than Unified CVP, the VRU trunk group number and number of trunks.
- The locale values to be used for ASR and/or TTS.
- Whether the same set of VRUs are to be used for all cases, or whether that will be determined separately for each customer (dialed number).

Note If all dialed numbers will use the same VRUs, it is easiest to use a default Network VRU, rather than to configure multiple Network VRUs. For more information, refer to the section, "Common Configuration for Differentiating VRUs Based on Dialed Number."

H.323 Deprecation

The H.323 support has been deprecated in Unified CVP 9.0(1) onwards. For detailed information on H.323 to SIP migration, see the Installation and Upgrade Guide for Cisco Unified Customer Voice Portal Release 9.0(1).

SIP Proxy

Cisco Unified SIP Proxy (CUSP) server is recommended as proxy server in Unified CVP SIP deployments. The support for CUPS as SIP Proxy in Unified CVP deployment has been discontinued in Unified CVP 9.0(1) onwards.

Ring No Answer Settings with SIP

If you are using the CVP RNA settings in SIP (set RNA value between 5-60 seconds), ensure this value is 2 or 3 seconds greater than the Unified ICME Agent Desk Setting RNA timeout, in order to allow time for the signaling to the agent after the ICM Router picks the agent via the link with the Peripheral Gateway.

Unified CVP makes a call to the ringtone service on the VXML gateway, followed by sending the call to the Unified CM trunk for the agent, and this allows enough time for the agent to get the delivered event after being reserved, and ensuring that Unified ICME reporting is correct in terms of handled time and RNA call disposition reporting.

Unified CVP Installation

Before using the information in this chapter, you need to do the following:

- Install the Unified CVP software.

  Note See the Installation and Upgrade Guide for Cisco Unified Customer Voice Portal for complete installation instructions.

- Install all the solution components.
- If you are using Unified CVP as a Unified ICME queuing platform, make sure the VRU PGs use service control with Service Control reporting enabled. If you are using it strictly as a self-service platform, disable Service Control reporting. Also, take note of the VRU Connection Port used for each VRU PG peripheral (PIM).

  Note See the Reporting Guide for Cisco IPCC Enterprise & Hosted Editions and the Reporting Guide for Cisco Unified ICM Enterprise & Hosted for more information on IVR-related Service Control reporting and queue reporting.
  See the Reporting Guide for Cisco Unified Customer Voice Portal for Unified CVP reporting.

- Make sure the NIC cards, Voice GW, and network components all have the Ethernet interfaces configured with matching speed and duplex settings.

  Note Refer to the Values for Unified CVP for details about the required Ethernet Switch / Server NIC settings. Refer to the Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND for additional information.

Routing Calls Through the Network to the VRU

Most call flow models involve a step in which the call must be transferred to a VoiceXML gateway. Depending on the specific call flow model in use, one of two techniques is applied to direct that transfer. Both techniques involve a label or labels being provided by Unified ICME or Unified ICMH, and those labels must be configured in the solution's other call routing components to deliver the call to an appropriate VoiceXML gateway. Such labels are part of the overall dialed number plan of the contact center, and must be determined prior to configuring Unified CVP.

For call flow models that use Network VRUs of Type 3, 7, or 10:

- Determine the Network Routing Number. This number is the base for routing calls through the network to the VRU; a correlation ID is appended to this number to transfer calls to a Network VRU through the network.

For call flow models with a Customer VRU in Unified ICMH environments and for NIC Type 8 call flow models:

- Determine the translation route pools to use for each VRU.
- Determine the labels to be sent to the network to connect the call to the VRU and the corresponding DNIS that will be seen by the VRU. For example, the label for the network might be 18008889999 and the DNIS received by the VRU and sent back to Unified ICME to identify the call might be 9999.

Values for Unified CVP

Make sure your Ethernet Switch / Server NIC, gateways, and Call Server settings are set as follows:

Caution The **Auto**option, is *only* applicable for matched port/NIC at Gigabit Ethernet (1000Mbps). If you are unsure of the adjacent station configuration, select 1000/Full on the Gigabit interface. You can only use the Auto option if both stations supply Gigabit interfaces.

Ethernet Switch Speed	Server/Gateway NIC Speed	Speed/Duplex Setting for Switch Port MUST be:	Speed/Duplex Setting for Server/GW NIC MUST be:
1000 Mb	1000 Mb	1000/ Full	1000/ Full
1000 Mb	1000 Mb	Auto / Auto	Auto / Auto
1000 Mb	100 Mb	100 Mb / Full	100 Mb / Full
100 Mb	100 Mb	100 Mb / Full	100 Mb / Full
100 Mb	1000 Mb	100 Mb / Full	100 Mb / Full

Apply Contact Center Gateway Debug Settings

To enable required minimum contact center debug settings on the gateway, perform the following:

Procedure

Step 1 Log into the gateway.

Step 2 Type `enable` and the password to enter enable mode.

Step 3 Enter the configure terminal command to enter configuration mode.

Step 4 Type `ivr contact-center` to apply default debug settings.

Step 5 Configure the logging buffer size using `set logging buffer`

Example:

```
set logging buffer 1000000
```

The recommended logging buffer size is 1000000.

Step 6 Exit configuration mode and return to the enable prompt by pressing **Ctrl-Z**.
To see the current operating configuration, including the change you just made, enter the show running-config command: `show running-config`

Step 7 To persist configuration changes (make changes permanent), enter the copy running-config startup-config command at the enable prompt: `write`
Example:

```
User Access Verification
Password:
ccbu-doc-gw4>en
Password:
ccbu-doc-gw4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ccbu-doc-gw4(config)#ivr
ccbu-doc-gw4(config)#ivr contact-center
ccbu-doc-gw4(config)#^Z
ccbu-doc-gw4#show debug
....
```

Network VRU Types

In Unified ICME, Network VRU is a configuration database entity. It is accessed using the ICM Configuration Manager's Network VRU Explorer tool. A Network VRU entry has two pieces of information:

- Type - This is a number from 2 to 10, and corresponds to the types described above.
- Labels - This is a list of labels, which Unified ICME can use to transfer a call to the particular Network VRU being configured. These labels are only relevant for Network VRU's of Types *3*, *7*, and *10* (that use the Correlation ID mechanism to transfer calls). They are also required, but never used in the case of Type 5. (Labels for Types *8* and *2* are defined in the ICM Configuration Manager's Translation Route Explorer tool, and invoked using a Translation Route to VRU node.)

  Each label is made up of two parts:

  - A digit string, which becomes a DNIS that is understood by a SIP Proxy Server, or static route table or by gateway dial-peers.

- A routing client (also known as a switch leg peripheral). Each peripheral device which can act as a switch leg must have its own label, even *if* the digit strings are the same in all cases.

Unified ICME Release 7.1(1) introduced Network VRU Type 10, which simplifies the configuration of Network VRU's for Unified CVP. For most call flow models, a single Type 10 Network VRU can take the place of the Type 2, 3, 5, 7, or 8 Network VRUs, which were associated with the Customer Instance and/or the Switch and VRU leg peripherals. The VRU-Only call flow models still require Type 8; however, in one specific case Type 3 or 7 is still required.

Network VRU configuration entries themselves have no value until they are associated with active calls. There are three places in Unified ICME where this association is made:

- Advanced tab for a given peripheral in the ICM Configuration Manager's PG Explorer tool
- Customer Instance configuration in the ICM Configuration Manager's ICM Instance Explorer tool
- On every VRU Script configuration in the ICM Configuration Manager's Network VRU Script List tool

Depending on the call flow model, Unified ICME looks at either the peripheral or the customer instance to determine how to transfer a call to a VRU. Generally speaking, Unified ICME examines the Network VRU which is associated with the switch leg peripheral when the call first arrives on a switch leg, and the Network VRU which is associated with the VRU leg peripheral when the call is being transferred to VRU using the Translation Route mechanism. It examines the Network VRU, which is associated with the Customer Instance or the default Network VRU from the System Information tool, when the call is being transferred to the VRU using the Correlation ID mechanism.

Unified ICME also examines the Network VRU which is associated with the VRU Script every time it encounters a RunExternalScript node in its routing script. If Unified ICME does not believe the call is currently connected to the designated Network VRU, it will not execute the VRU Script.

Note The previously recommended VRU types still work as before, even in Unified ICME 7.1(1) and later; however, new installations must use Type 10 if possible, and existing installations can optionally switch to Type 10.

SIP DN Pattern Matching Algorithm

Throughout this document, when you are configuring static routes using dialed number (DN) patterns, you must consider a number of application and configuration concepts:

The following details refer to creating dialed number patterns:

- Wildcarded DN patterns can contain "." and "X" in any position to match a single wildcarded character.
- Any of the wildcard characters in the set ">*!T" will match multiple characters but can only be used trailing values because they will always match all remaining characters in the string.
- The highest precedence of pattern matching is an exact match, followed by the most specific wildcard match. When the number of characters are matched equally by more than one wildcarded pattern, precedence is given from top to bottom of the configured DN list.
- There is no explicit software limit on the number of items in the DN pattern list.

Additional Information

Once you complete the prerequisite study and preparation described in High-Level Configuration Instructions, choose your task from one of the lists below.

- The first table enables you to choose a call flow model and go to the related high-level instructions based on the links and reference points listed in the table.
- The second table provides references to additional configuration topics.

Note For a full description of each call flow model, see the Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND)

Unified CVP Call Flow Models

Standalone Call Flow Models

- VXML Server Standalone Call Flow Model

  Install the VXML Server as a standalone component, without Unified ICME, the Unified CVP Call Server components--a complete solution for rapidly creating and deploying dynamic VoiceXML applications.

- Unified CVP Standalone with ICM Lookup Call Flow Model, on page 18

  To the previous standalone solution, add the capability of invoking a routing script in Unified ICME and receiving a response.

Call Director Call Flow Models

- Unified CVP Call Director Call Flow Model Unified ICME

  Unified CVP only provides Unified ICME with VoIP call switching capabilities. You provide your own Service Control VRU, if you are using Unified ICME to queue calls or you might queue calls directly on your ACD. Calls might be transferred multiple times, from Ingress, to customer-provided VRU, to either Unified CCE or customer-provided ACD or agent, and back again.

  When calls are connected to customer-provided equipment (either VoIP or TDM), their voice paths must go to an egress gateway, which is connected by TDM to that equipment. If the signaling is SIP, then this call flow model works with customer-provided SIP endpoints which have been tested and certified to interoperate with Unified CVP.

  Note The Call Director call flow model is the most inclusive call flow because it potentially incorporates all components of the Unified CVP solution.

- Unified CVP Call Director Call Flow Model Unified ICMH

  Unified CVP only provides the Network Applications Manager (NAM) with VoIP call switching capabilities. You provide your own Service Control VRU, if you are using the NAM to queue calls or you might queue calls directly on your ACD. Calls might be transferred multiple times, from Ingress, to customer-provided VRU, to either the NAM or customer-provided ACD or agent, and back again.

When calls are connected to customer-provided equipment, their voice paths must go to an egress gateway, which is connected by TDM to that equipment. If the signaling is SIP, then this call flow model works with customer-provided SIP endpoints which have been tested and certified to interoperate with Unified CVP.

Note In a two-tier service bureau (carrier) configuration, the *NAM* is the tier providing direct communication with the carrier PSTN. Route requests arrive at the NAM from the IXC carrier network and are forwarded, based on specific call properties, to the appropriate Customer ICM (CICM). A NAM usually contains only a small configuration that allows it to directly route a subset of calls and dispatch other calls to the appropriate CICM. The NAM receives route responses from all CICMs and forwards them to the carrier network.

Comprehensive Call Flow Models

- Unified CVP Comprehensive Call Flow Model Unified ICME

  Unified CVP acts as the Switch, transferring the call to the Network VRU and to agents. The Unified CVP IVR Service in the Operations Console is also configured to work with the VoiceXML Gateway to provide VRU treatment, which might include ASR/TTS and VXML Server applications.

- Unified CVP Comprehensive Call Flow Model Unified ICMH

  Unified CVP is deployed at the NAM where it acts as the Switch, transferring the call to the Network VRU (using the Correlation ID transfer mechanism) and to agents. The Unified CVP IVR Service in the Operations Console is also configured to work with the VoiceXML Gateway to provide VRU treatment, which might include ASR/TTS and VXML Server applications.

Type 8 Call Flow Models

- Type 8 Unified CVP VRU-Only Call Flow Model Unified ICME

  Unified CVP works with the Voice Gateway to act as the VRU; VRU voice treatment is provided by the Gateway and can include ASR/TTS and VXML Server applications. This model would be used when call switching is provided by some component other than Unified CVP, such as a Unified ICME NIC.

- Type 8 Unified CVP VRU-Only Call Flow Model Unified ICMH

  The Unified CVP Call Server is deployed at the CICM level to act only as the VRU leg for the call. The VRU voice treatment is provided at the Voice Gateway, and might include ASR/TTS and VXML Server applications.

  This model would be used when call switching is provided by some component other than Unified CVP, such as a Unified ICME NIC.

Note The *CICM* is the Customer ICM. In the optional two-tier service bureau (carrier) configuration, the CICM is the tier providing the carrier customer-specific routing function. CICMs receive customer-specific call route requests from the NAM; they typically perform more elaborate scripted call routing using customer-specific advanced services or agent and skill context.

Type 3 or 7 Call Flow Models

- Type 3 or 7 Unified CVP VRU-Only Call Flow Model Network VRU Unified ICMH

  Unified CVP is deployed as a Network VRU at the NAM. The Unified CVP IVR Service in the Operations Console works with the Voice Gateway to act as the VRU; VRU voice treatment is provided at the Voice Gateway and can include ASR/TTS and VXML Server applications.

  This call flow model is used when Unified CVP is connected to the NAM.

  This model is also used when call switching is provided by some component other than Unified CVP, such as a Unified ICME NIC.

Configuration instructions for the following geographic and physical models are not listed in this guide. Refer to the Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND) at for details about these models.

Unified CVP Geographic models:

- Centralized Single-Site
- Centralized Multi-Site
- Centralized Branch
- Standalone Branch

Unified CVP Physical models:

- Typical for Unified ICME Integrated
- Typical for Standalone
- Streamlined for SIP Call Director
- Laboratory (All-in-a-Box)

  The laboratory all-in-one-box implementation is also documented in Getting Started with Cisco Unified Customer Voice Portal.

Links to Additional Configuration Instructions

- Unified CVP Comprehensive Call Flows for Pre-Routed Calls

  This class of call flows is similar to the Unified CVP Comprehensive call flow models, except that calls are first introduced into Unified ICME or Unified ICMH using some path other than through Unified CVP. A Unified ICME routing script is given the chance to "pre-route" such calls before Unified CVP ever sees them. Once the script transfers the call to Unified CVP, for either self-service or queuing, a more standard Unified CVP Comprehensive call flow model is used.

- Common Unified ICMH Configuration for Unified CVP Switch Leg

  Describes Unified ICMH configuration instructions common to Comprehensive Unified ICMH and VRU-Only with NIC routing, with Correlation ID call routing call flow models for Unified CVP switch legs.

- Define Unified CVP ECC Variables

  Provides instructions on how to set up ECC variables that Unified CVP uses to exchange information with Unified ICMH.

- Metadata ECC Variable

Defines the values for the *user.microapp.metadata* ECC variable.

- Common Configuration for Differentiating VRUs Based on Dialed Number

  Provides instructions on how to configure Unified ICME to differentiate the VRUs.

- SIP Proxy Redundancy

  Refer to "Set Up the Ingress Gateway to Use Redundant Proxy Servers" and "Set Up the Unified CVP Call Server with Redundant Proxy Servers" for details.

Operations Console

The Unified CVP Operations Console enables you to centrally operate, administer, maintain, and provision your Unified CVP solution. The Operations Console is a web-based tool that provides the capability to administer and monitor your Unified CVP components remotely.

For information on Operations console, see the *Operation Console online help*.

CHAPTER 2

High-Level Configuration Instructions

This chapter presents a set of high-level instructions for configuring many of the Unified CVP call flow models (deployment models).

Each set of call flow model instructions contains:

- A brief overview of that call flow model
- High-level instructions for configuring the components in that call flow model
- References to detailed instructions (elsewhere in this guide, in online help, or in other documents) for performing each high-level task

This chapter also includes information, or pointers to information, for configuring the Gateway, Unified ICME VRU handling and Unified CVP Call Server (including the SIP Service, ICM Service, and IVR Service).

- Best Practices for Order of Device Operations, page 16
- Standalone Call Flow Models, page 16
- Unified CVP Call Director Call Flow Models, page 25
- Unified CVP Comprehensive Call Flow Models, page 37
- Unified CVP VRU Call Flow Models with NIC Routing, page 63
- Unified CVP Comprehensive Call Flows for Pre-Routed Calls, page 80
- Common Unified ICMH Configuration for Unified CVP Switch Leg, page 84
- Define Unified CVP ECC Variables, page 86
- Metadata ECC Variable, page 92
- Common Configuration for Differentiating VRUs Based on Dialed Number, page 93
- Local SRV File Configuration Example for SIP Messaging Redundancy, page 93

Best Practices for Order of Device Operations

The following information is a guide for setting up the order of the device operations. Please apply the instructions accordingly, based on the call flow model.

Device Deployment:

- SIP Proxy Server device (optional)
- Unified CVP Call Server device
- Unified CVP VXML Server device
- Unified CVP Reporting Server device
- Other Devices (Gateways, Unified CM, etc.)

System Configuration:

- SIP Server Groups
- Dialed Number Pattern
- Locations
- Courtesy Callback

Miscellaneous:

- Transfer of licenses (required)
- Transfer of VXML applications (required)
- Bulk transfer of default Gateway files (required)

Device management order:

1. Add new CVP device
2. Configure CVP device
3. Save and deploy CVP device
4. Transfer license
5. Restart CVP device to activate license.
6. Verify CVP devices states are "Up" in OAMP Control Center.
7. Deploy system level configuration, Dialed Number Pattern, SIP Server Groups, Locations, and Courtesy Callback and confirm the status.
8. Save and deploy SNMP Configuration.

Standalone Call Flow Models

This section describes the following call flows and provides their high-level configuration instructions.

- VXML Server Standalone Call Flow Model
- Unified CVP Standalone with ICM Lookup Call Flow Model

VXML Server Standalone Call Flow Model

In this call flow model, the VXML Server is a J2EE-compliant server that provides a complete solution for rapidly creating and deploying dynamic VoiceXML applications. You can install the VXML Server as a standalone component without the Unified CVP Call Server component and with or without the Reporting.

The Unified CVP VXML Server (standalone) call flow model is available in the following variations:

- Standalone without reporting: Use the **VXML Server (Standalone)** option in the Operations Console. This call flow model *does not* require a Call Server and a Reporting Server.
- Standalone with reporting: Use the **VXML Server** option in the Operations Console. This call flow model *requires* a Call Server and a Reporting Server.
- Standalone, but adding reporting *after* the VXML Server (Standalone) version has already been configured: Configure the Unified CVP Call Server, delete the VXML Server (Standalone), and use the **VXML Server** option in the Operations Console to add the VXML Server.

Refer to Set Up Unified CVP VXML Server Standalone Call Flow Models for configuration instructions.

In this call flow model with reporting, the Unified CVP Call Server is used to route messages between the components. Calls arrive through a VoiceXML gateway and interact directly with a VXML Server to execute VoiceXML applications. The gateway performs both ingress and VoiceXML functions. This call flow model provides a sophisticated VoiceXML-based VRU, for applications which, in many cases, do not need to interact with a Unified ICME Server.

Some of the features of this option include:

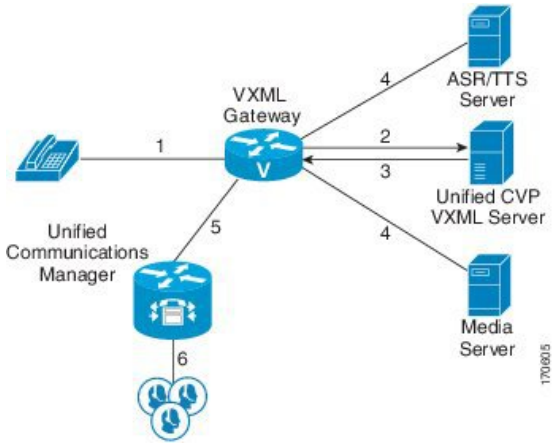
- VoiceXML features that control each of the following:
  - Audio input and output
  - Presentation logic
  - Call flow
  - Telephony connections
  - Event handling for errors
  - ASR/TTS
  - Digit detection and generation count
- Drag-and-drop Graphical User Interface for the rapid creation of voice applications.
- VXML Server, which provides a J2EE-compliant development framework.

In the Unified CVP VXML Server (standalone) call flow model, *only* the VXML Server, Call Studio, and a Gateway are required, except when using reporting which requires a Call Server and a Reporting Server.

This standalone model has functions similar to the Unified CVP VRU Call Flow Models with NIC Routing.

Note The CVP VXML standalone call flow model allows only one synchronous blind or bridged transfer. A synchronous blind transfer indicates that once the call has been transferred, a Unified CVP Standalone script has no ability to asynchronously take it back and deliver it somewhere else, whereas Unified ICME scripts, in the Unified ICME-integrated models, do have that ability.

The following figure displays the call flow for the Unified CVP VXML Server (standalone) call flow model.



The following, numbered, call flow description for the previous figure assumes:

- You installed and licensed the VXML Server.
- You created a Call Studio application and deployed it on the VXML Server.

The call flow shown in the previous figure is as follows:

1. The call arrives from the PSTN network to the Gateway.
2. The Gateway sends an HTTP URL request to the VXML Server.
3. The VXML Server returns the VoiceXML instructions to be executed on the Gateway Voice Browser.
4. The VoiceXML instructions returned to the Gateway can include references to ASR/TTS to recognize voice input and play TTS files, and references to Media Servers to play .wav files.
5. The gateway can, optionally, transfer the call to any destination that it can deliver a call to, such as Unified CM.
6. Unified CM can then send the call to an agent.

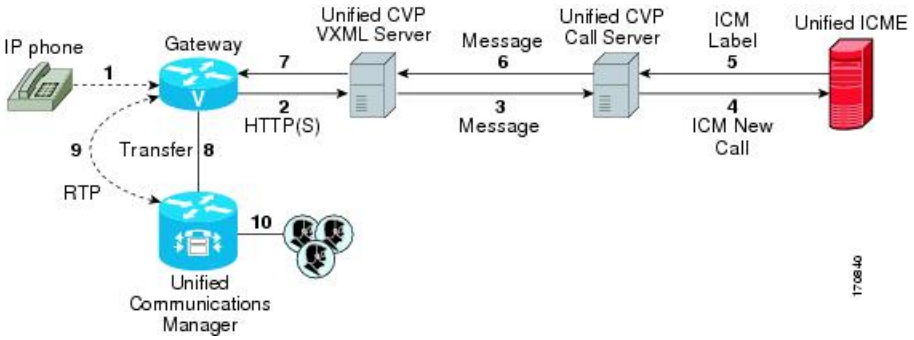
Unified CVP Standalone with ICM Lookup Call Flow Model

This call flow model invokes a new incoming dial number from a Unified CVP peripheral, which invokes a routing script in Unified ICME.

Note This call flow model only returns a label and call context variables; you cannot run scripts or do queuing.

The following figure displays the call flow for the Unified CVP Standalone with ICM Lookup call flow model.

Note In this diagram, solid lines indicate voice paths and dashed lines indicate signaling paths.



The following, numbered call flow description for the previous figure assumes:

- A Unified CVP Call Server has been defined using the Operations Console.
- A Call Studio application has been created that contains a Unified ICME request label element.
- The Unified ICME script must be set up to handle this request, so Unified ICME can interpret the label.
- The Call Studio application must be deployed on the VXML Server.

The call flow in the previous figure is as follows:

1 The call arrives from the PSTN network to the Gateway.

2 The Gateway sends a HTTP URL request to the VXML Server.

3 The VXML Server sends a message to the Unified CVP Call Server.

Note The request in Step#2 is only sent to the Call Server when the ReqICMLabel node is used in the Call Studio script. Otherwise, the call flow is the same as in standalone call flow model; that is, Steps #3 through #6 only happen if the ReqICMLabel node is used in the script.

4 Unified CVP Call Server sends a Unified ICME new call to Unified ICME.

5 Unified ICME returns a Unified ICME label to the Unified CVP Call Server.

6 Unified CVP Call Server sends a response message to the VXML Server.

7 The VoiceXML returned to the Gateway can include references to ASR/TTS and Media Servers to recognize voice input, play TTS files, and play .wav files, respectively.

8 The gateway can, optionally, transfer the call to any destination that it can deliver a call to, such as Unified CM.

9 Unified CM can then send the call to an agent.

Set Up Unified CVP VXML Server Standalone Call Flow Models

This task provides high-level steps for configuring Unified CVP as a standalone VXML Server. The instructions cover all three versions of Unified CVP standalone: without reporting, with reporting, and with ICM lookup and optional reporting.

There are four types of pre-instruction labels in the steps that follow. Complete the steps based on the type of standalone implementation you want. Skip steps that do not apply to your implementation, according to the following label descriptions:

- **All Versions:** Steps that apply to all standalone configurations.
- **Non-reporting:** Steps for the non-reporting version of standalone Unified CVP.
- **Reporting:** Steps for a standalone VXML Server **with** reporting. Includes steps to define a Call Server, Reporting Server, and reporting configurations.
- **ICM Lookup:** Steps, or unique portions of steps, required for the ICM Lookup version of Unified CVP standalone.

Note **Converting from a Non-reporting Configuration (no Call Server defined) to a Reporting or ICM Lookup Configuration:** If you have previously configured Unified CVP for standalone **without** reporting, the version of the VXML Server you defined cannot be associated with a Call Server, which is required for reporting and for ICM Lookup. Therefore you must **delete** the existing VXML Server definition and begin with Step #3 to incorporate a Call Server and the remaining Reporting Server and ICM Lookup configuration steps.

Note The Getting Started with Cisco Unified Customer Voice Portal guide provides a very detailed example of creating a standalone implementation. It includes both reporting configuration, and testing.

Procedure

Step 1 **All Versions:** Transfer the following script, configuration, and .wav files using the Unified CVP Operations Console (**or**, using the Unified CVP product CD):

- CVPSelfService.tcl

  Note This file contains a gateway configuration example.

- CVPSelfServiceBootstrap.vxml
- critical_error.wav

a) Select: **Bulk Administration** > **File Transfer** > **Scripts and Media**
b) From the Select device type drop-down list, select **Gateway**.
c) Highlight each required file in the Available pane and click the **right arrow icon** to move the file to the Selected pane.

d) Click **Transfer**.

Step 2 **All Versions:** Configure the gateway for VXML Server (Standalone) applications:

a) Define the VXML Server applications on the gateway.

Note Backup server is optional. For the Tomcat Application Server, set the port to 7000. The backup server cannot be the same server as the Primary Server.

b) Configure the base gateway settings.
See the Example Gateway Settings for Standalone Call Flow Model

c) Configure the service gateway settings.
See theExample Gateway Settings for Standalone Call Flow Model

d) Configure a dial-peer, which will call the service to reach the Unified CVP VXML Server.
See the Example Dial-Peer for Standalone Call Flow Model

e) Optionally, create additional dial-peers for any outgoing transfer destinations your application uses.

Review the updated gateway configuration by issuing the show run command to examine the running configuration.

Step 3 **All Versions, with Noted Difference for ICM Lookup:** Create an application using Call Studio and deploy it as a zip file.

Note **For ICM Lookup, Use the ReqICMLabel Element** as follows:

The application must use the ReqICMLabel Element. This element has two exit states: error and done. The *done* path must connect to a transfer element to transfer the caller to ReqICMLabel as referenced by the ReqICMLabel Element.

For details on the ReqICMLabel Element, see the Element Specifications for Cisco Unified CVP VXML Server and Unified Call Studio.

See the Deploy Call Studio Scripts Using Call Studio section in this guide.

For information about Unified Call Studio, see the User Guide for Cisco Unified CVP VXML Server and Unified Call Studio.

Step 4 **Reporting and ICM Lookup:** Enable logging.
See the User Guide for Cisco Unified CVP VXML Server and Unified Call Studio for details on configuring loggers using Call Studio.

Step 5 **ICM Lookup**: Enable the CVPSNMPLogger for SNMP monitoring.

Note By default, CVPSNMPLogger is enabled when a new Call Studio application is created and deployed to the VXML Server.

Step 6 **Reporting:** Configure a Call Server.
Configure a standard Call Server and *do not* enable any Unified CVP services.

a) Select **Device Management** > **Unified CVP Call Server** in the Operations Console.

b) Click **Help** > **This Page** for details.

Step 7 **ICM Lookup:** Configure a Call Server and select the ICM Service.
Configure a standard Call Server and enable the ICM Service.

a) Select **Device Management** > **Unified CVP Call Server** in the Operations Console.

b) Click **Help** > **This Page** for details.

Step 8 **Without Reporting:** Configure the VXML Server as a standalone server..
In the Operations Console, select **Device Management** > **VXML Server (standalone)** and add a new VXML Server.

For details, select **Help** > **This Page** and examine **Configuring a VXML Server** > **Adding a VXML Server**.

Step 9 **Reporting and ICM Lookup:** Configure the VXML Server.

a) In the Operations Console, select **Device Management** > **VXML Server** and add a VXML Server with an associated Primary Call Server.
Click **Help** > **This Page** to access the **Adding a VXML Server** topic for details.

b) To enable reporting for this VXML Server, in the Operations Console, select the **Configuration** tab and select **Enable Reporting for this VXML Server**.

c) Add appropriate filtering.
For help, from the Configuration tab select **Help** and select **VXML Server Configuration Properties** and scroll down to the filter discussion.

Step 10 **Non-reporting:** Deploy the Call Studio Application on the VXML Server.

Note This is the final step for a non-reporting, standalone call flow.

In the Operations Console, select **Device Management** > **VXML Server (standalone)**, select the VXML Server, and click **Save and Deploy**.

This completes the configuration for non-reporting standalone call flow model.

Step 11 **Reporting and ICM Lookup:** Deploy the Call Studio Application on the VXML Server.

a) Select **Device Management** > **VXML Server** in the Operations Console.

b) Select the VXML Server and click **Save and Deploy**

Step 12 **ICM Lookup:** Using the ICM Script Editor, create a Unified ICME script that returns a label.
In order to transfer information from Unified ICME to the VXML Server besides the label, use the ToExtVXML0 - 4 ECC Variables and/or Peripheral Variables 1 - 10. The format for using the ToExtVXML 0 - 4 is with name value pairs that are delimited by semi-colons.

Example:

```
ToExtVXML0 = "company=Cisco Systems;state=MA"
```

Use the Peripheral Variables 1 - 10 to pass information to the VXML Server. The values in these variables will be taken as is.

For more information about creating a Unified ICME script that returns a label in, see the Unified ICME documentation.

For more information about using the ReqICMLabel element, see the Pass Data to Unified ICME.

Step 13 **Reporting:** Configure the Reporting Server.

a) In the Operations Console, select **Device Management** > **CVP Reporting Server** > **General tab** and configure the Reporting Server.

b) Select a Call Server to associate with this Reporting Server.

c) Check the default values of the Reporting properties and change, if desired.

For more information, see the Reporting Guide for Cisco Unified Customer Voice Portal.

Example Gateway Settings for Standalone Call Flow Model

The first part of the following example provides the basic configuration for setting a VoiceXML Standalone gateway:

- Applies a timestamp to debugging and log messages
- Turns on logging
- Turns off printing to the command line interface console
- Sends RTP packets
- Configures ASR/TTS Server
- Configures gateway settings

The last part of this example provides the following:

- Standalone Service settings for hello_world application on the VXML Server
- Service requirements for configuring self-service call flow models

```
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
!
service internal
logging buffered 99999999 debugging
no logging console
!
ip cef
!
voice rtp send-recv

ip host tts-en-us <IP of TTS or MRCP Server>
ip host asr-en-us <IP of ASR or MRCP Server>

voice class codec 1
codec preference 1 g711ulaw
codec preference 2 g729r8

voice service voip
signaling forward unconditional
h323
!
gateway
timer receive-rtcp 6
!
ip rtcp report interval 3000
!
ivr prompt memory 15000
ivr prompt streamed none
ivr asr-server rtsp://asr-en-us/recognizer
ivr tts-server rtsp://tts-en-us/synthesizer

mrcp client timeout connect 10
mrcp client timeout message 10
mrcp client rtpsetup enable
rtsp client timeout connect 10
rtsp client timeout message 10
vxml tree memory 500
http client cache memory pool 15000
http client cache memory file 500
http client connection timeout 60
http client response timeout 30
http client connection idle timeout 10
```

```
application
service hello_world flash:CVPSelfService.tcl
param CVPPrimaryVXMLServer <ip address>
param CVPBackupVXMLServer <ip address>
param CVPSelfService-port 7000
param CVPSelfService-SSL 0
-OR-
param CVPSelfService-port 7443
param CVPSelfService-SSL 1
param CVPSelfService-app HelloWorld

service CVPSelfService
flash:CVPSelfServiceBootstrap.vxml
!
```

Note The optional param CVPSelfService-SSL 1 line enables HTTPS.

Important Calls may be rejected with a *403 Forbidden* response if Toll Fraud security is not configured correctly. The solution is to add the IP address as a trusted endpoint, or else disable the IP address trusted list authentication altogether using this configuration entry: **`voice service voip -> "no ip address trusted authenticate".`**

Example Dial-Peer for Standalone Call Flow Model

The following example provides the configuration for an incoming Pots and VoIP call for the Unified CVP VXML Server (standalone) call flow model:

Note VXML Server (Standalone) only supports an incoming call with a TDM through a T1 port. Using an FXS port is not supported.

```
dial-peer voice 8 pots
 description Example incoming POTS dial-peer calling HelloWorld VXML
Server app
 service hello_world
 incoming called-number <your DN pattern here>
 direct-inward-dial

dial-peer voice 800 voip
 description Example incoming VOIP dial-peer calling HelloWorld VXML
Server app
 service hello_world
 incoming called-number 800.......
 voice-class codec 1
 dtmf-relay rtp-nte
```

```
 no vad
!
```

Unified CVP Call Director Call Flow Models

This section describes the call director call flow models, broken down by Unified ICME and Unified ICMH products as follows:

- Unified CVP Call Director Call Flow Model Unified ICME
- Unified CVP Call Director Call Flow Model Unified ICMH

Unified CVP Call Director Call Flow Model Unified ICME

The SIP Call Director call flow model sends calls to other third-party ACDs, SIP gateways, or to a Unified CM server.

In this call flow model, Unified CVP only provides Unified ICME with VoIP call switching capabilities. You provide your own Service Control VRU if you are using Unified ICME to queue calls or you might queue calls directly on your ACD. Calls might be transferred multiple times, from Ingress, to customer-provided VRU, to either Unified CCE or customer-provided ACD or agent, and back again. When calls are connected to customer-provided equipment (either VoIP or TDM), their voice paths must go to an egress gateway, which is connected by TDM to that equipment. If the signaling is SIP, then this call flow model works with customer-provided SIP endpoints which have been tested and certified to interoperate with Unified CVP.

In this call flow model, Unified CVP *stays* in the signaling path after the transfer.

Note VRU scripts and transfer to a VRU leg are not available in this call flow model.

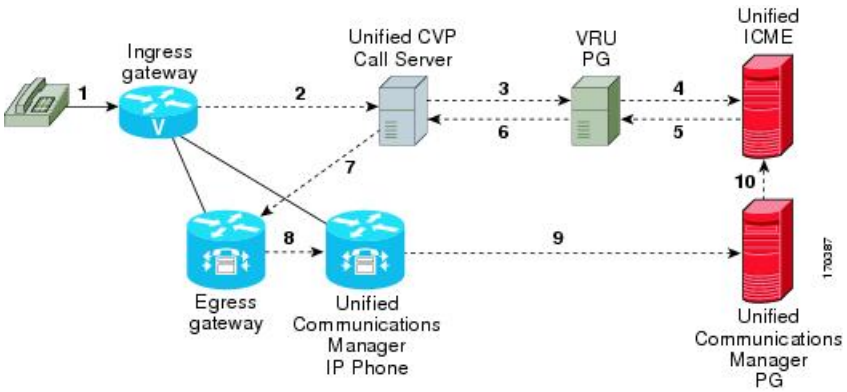
The Unified CVP Call Director call flow model requires the following components:

- Call Server
- Ingress Gateway
- Operations Console
- Unified ICME

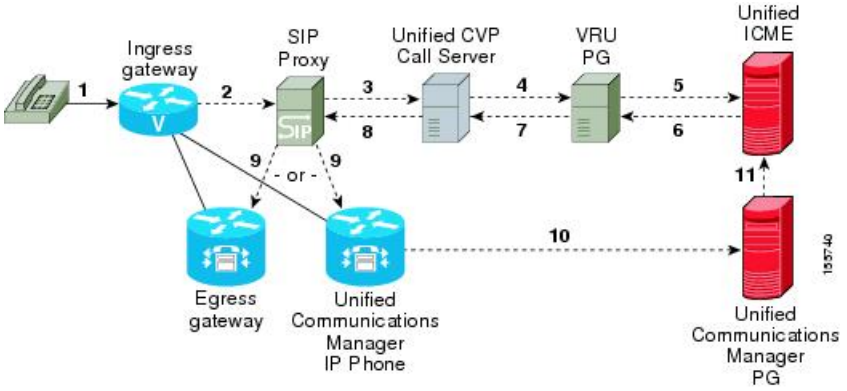
Note The Reporting Server is optional. If the Call Server is configured to use SIP signaling, a SIP Proxy Server is optional.

The following figure shows the call flow for this call flow model using SIP without a Proxy Server.

Note In the following figures, solid lines indicate voice paths and dashed lines indicate signaling paths.



The following figure shows the call flow for this call flow model using SIP with a Proxy Server.



Note See the REFER Transfers and sendtooriginator Setting in the SIP Service for more information.

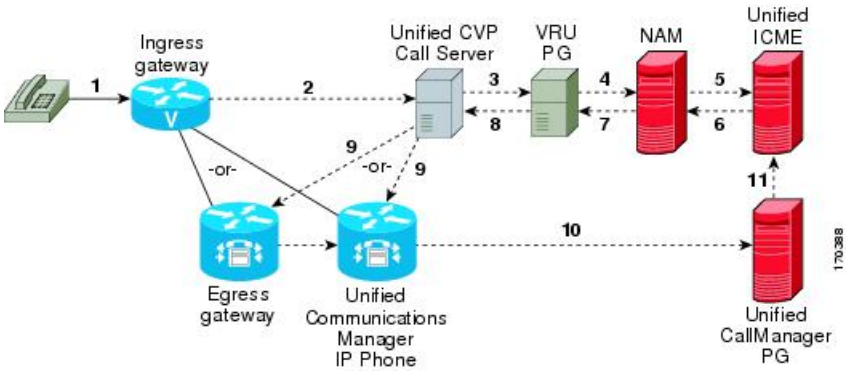
Unified CVP Call Director Call Flow Model Unified ICMH

In this call flow model, Unified CVP only provides the Network Applications Manager (NAM) with VoIP call switching capabilities. You provide your own Service Control VRU if you are using the NAM to queue calls, or you might queue calls directly on your ACD. Calls might be transferred multiple times, from Ingress, to customer-provided VRU, to either the NAM or customer-provided ACD or agent, and back again. When calls are connected to customer-provided equipment, their voice paths must go to an egress gateway, which is connected by TDM to that equipment. If the signaling is SIP, then this call flow model works with customer-provided SIP endpoints which have been tested and certified to interoperate with Unified CVP.

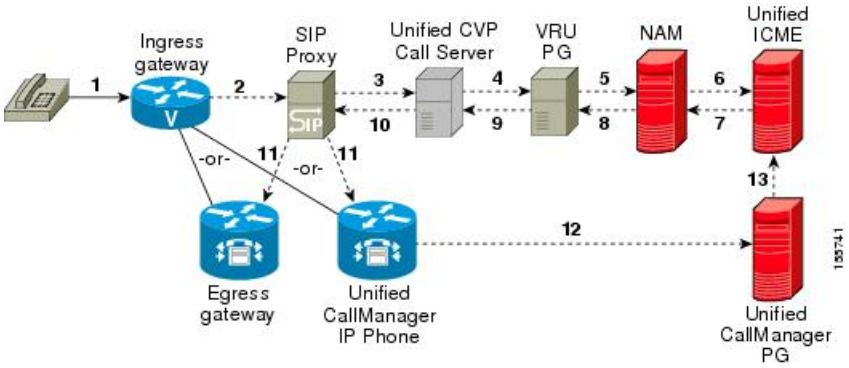
Note VRU scripts and transfer to a VRU leg are not available in this call flow model.

The following figure shows the call flow for this call flow model using SIP without a Proxy Server.

Note In the following diagrams, solid lines indicate voice paths and dashed lines indicate signaling paths.



The following figure shows the call flow for this call flow model using SIP with a Proxy Server.



See the Set Up Call Director Call Flow Model for configuration instructions for this call flow model.

Note see the REFER Transfers and sendtooriginator Setting in the SIP Service for more information.

Set Up Call Director Call Flow Model

The following task contains the configuration instructions for the Call Director Call Flow Model using **SIP** for both Unified ICME and Unified ICMH.

Procedure

Step 1 Install the IOS image on the Ingress Gateway.
For detailed information, see the Cisco IOS documentation

Step 2 Configure the Ingress Gateway:

a) Transfer the following script, configuration, and .wav files to the Ingress Gateway using the Unified CVP Operations Console (or the Unified CVP product CD):

- bootstrap.tcl
- handoff.tcl
- survivabilty.tcl
- bootstrap.vxml
- recovery.vxml
- ringtone.tcl
- cvperror.tcl
- ringback.wav
- critical_error.wav

In the Operations Console, select **Bulk Administration** > **File Transfer** > **Scripts and Media** and select **Help** > **This Page** for details.

b) Configure the Ingress Gateway base gateway settings.
c) Configure the Ingress Gateway service settings.
d) Configure the Ingress Gateway dial-peer for the Unified CVP Call Server.
e) Configure a dial-peer for ringtone and error.
f) If you are using a Proxy Server, configure your session target in the outbound dial peer to point to the Proxy Server.
g) If you are using the sip-server global configuration, then configure the sip-server in the sip-ua section to be your Proxy Server and point the session target of the dial-peer to the sip-server global variable.

Note Make sure your dial plan includes this information. You will need to see the Dial plan when you configure the SIP Proxy Server for Unified CVP.

The SIP Service voip dial peer and the destination pattern on the Ingress Gateway must match the DNIS in static routes on the SIP Proxy Server or Unified CVP Call Server.

See the Configure the SIP Devices, and to SIP DN Pattern Matching Algorithm for detailed information.

Step 3 For SIP *without* a Proxy Server, complete the following steps:

a) If you are using DNS query with SRV or A types from the gateway, configure the gateway to use DNS. See the *Operations Console online help* for detailed instructions. If you are using DNS query with SRV or A types from the gateway, use the gateway configuration CLI as shown below:

Non-DNS Setup:

```
sip-ua
sip-server ipv4:xx.xx.xxx.xxx:5060
!
```

DNS Setup:

```
ip domain name patz.cisco.com
ip name-server 10.10.111.16
!
sip-ua
sip-server dns:cvp.pats.cisco.com
!
```

b) Configure the DNS zone file for the separate DNS server that displays how the Service (SRV) records are configured.

Note SRV with DNS can be used in *any* of the SIP call flow models, with or without a Proxy server. Standard A type DNS queries can be used as well for the calls, without SRV, but they lose the load balancing and failover capabilities.

See the DNS Zone File Configuration for Call Director Call Flow Model for more information.

Step 4 For SIP *with* a Proxy Server, use one of the following methods:

Note You can configure the Gateway statically instead of using DNS.

The following example shows how both the A and SRV type records could be configured:

```
ip host cvp4cc2.cisco.com 10.4.33.132
ip host cvp4cc3.cisco.com 10.4.33.133
ip host cvp4cc1.cisco.com 10.4.33.131
```

For **SIP/TCP**:

```
ip host _sip._tcp.cvp.cisco.com srv 50 50 5060 cvp4cc3.cisco.com
ip host _sip._tcp.cvp.cisco.com srv 50 50 5060 cvp4cc2.cisco.com
ip host _sip._tcp.cvp.cisco.com srv 50 50 5060 cvp4cc1.cisco.com
```

For **SIP/UDP**:

```
ip host _sip._udp.cvp.cisco.com srv 50 50 5060 cvp4cc3.cisco.com
ip host _sip._udp.cvp.cisco.com srv 50 50 5060 cvp4cc2.cisco.com
ip host _sip._udp.cvp.cisco.com srv 50 50 5060 cvp4cc1.cisco.com
```

Note The DNS Server must be configured with all necessary A type or SRV type records.

See the Configure the SIP Devices.

If you are using the DNS Server, you can set your SIP Service as the Host Name (either A or SRV type).

Step 5 On the Unified CM server, CCMAdmin Publisher, complete the following SIP-specific actions:

a) Create SIP trunks.

- If you are using a SIP Proxy Server, set up a SIP trunk to the SIP Proxy Server.
- Add a SIP Trunk for the Unified CVP Call Server.
- Add a SIP Trunk for each Ingress gateway that will send SIP calls to Unified CVP that might be routed to Unified CM.

To add an SIP trunk, select **Device** > **Trunk** > **Add New** and use the following parameters:

- Trunk Type: **SIP trunk**
- Device Protocol: **SIP**
- Destination Address: IP address or host name of the SIP Proxy Server (if using a SIP Proxy Server). If not using a SIP Proxy Server, enter the IP address or host name of the Unified CVP Call Server.
- DTMF Signaling Method: **RFC 2833**
- Do **not** check the *Media Termination Point Required* check box.
- If you are using UDP as the outgoing transport on Unified CVP, also set the outgoing transport to **UDP** on the SIP Trunk Security Profile.

- Connection to CUSP Server: use 5060 as the default port.

b) Add route patterns for outbound calls from the Unified CM devices using a SIP Trunk to the Unified CVP Call Server. Also, add a route pattern for error DN.
Select **Call Routing** > **Route/Hunt** > **Route Pattern** > **Add New**

Add the following:

- Route Pattern: Specify the route pattern; for example: **3XXX** for a TDM phone that dials 9+3xxx and all Unified ICME scripts are set up for 3xxx dialed numbers.
- Gateway/Route List: Select the SIP Trunk defined in the previous substep.

Note For warm transfers, the call from Agent 1 to Agent 2 does not typically use a SIP Trunk, but you must configure the CTI Route Point for that dialed number on the Unified CM server and associate that number with your peripheral gateway user (PGUSER) for the JTAPI gateway on the Unified CM peripheral gateway. An alternative is to use the Dialed Number Plan on Unified ICME to bypass the CTI Route Point.

c) If you are sending calls to Unified CM using an SRV cluster domain name, select **Enterprise Parameters** > **Clusterwide Domain Configuration** and add the Cluster fully qualified domain name **FQDN**.

Step 6 (Optionally) Configure the **SIP Proxy Server**.

a) Configure the SIP static routes to the Unified CVP Call Servers, Unified CM SIP trunks, and Gateways. Configure the SIP static routes for intermediary transfers for ringtone, playback dialed numbers, and error playback dialed numbers.

Note For failover and load balancing of calls to multiple destinations, configure the CUSP server static route with priority and weight.

b) Configure Access Control Lists for Unified CVP calls.
Select **Proxy Settings** > **Incoming ACL**.

Address pattern: **all**

c) Configure the service parameters.
Select **Service Parameters**, then set the following:

- Add record route: **off**
- Maximum invite retransmission count: **2**
- Proxy Domain and Cluster Name: if using DNS SRV, set to the FQDN of your Proxy Server SRV name

d) Write down the IP address and host name of the SIP Proxy Server. (You need this information when configuring the SIP Proxy Server in Unified CVP.)

e) If using redundant SIP Proxy Servers (primary and secondary or load balancing), then decide whether to use DNS server lookups for SRV records or non-DNS based local SRV record configuration.

Note If a single CUSP Server is used, then SRV record usage is not required.

Configure the SRV records on the DNS server or locally on Unified CVP with a .xml file (local xml configuration avoids the overhead of DNS lookups with each call).

Note See the Local SRV File Configuration Example for SIP Messaging Redundancy section for detailed information.

The Call Director call flow model with SIP calls will typically be deployed with dual CUSP servers for redundancy. In some cases, you might want to purchase a second CUSP server. Regardless, the default transport for deployment will be UDP; make sure you *always* disable the record-route in a CUSP server as this advanced feature is not supported in Contact Center deployments.

For the required settings in the Unified CM Publisher configuration, see the Cisco Unified SIP Proxy documentation.

Step 7 Configure the PGs for the switch leg.
On Unified ICME, ICM Configuration Manager, **PG Explorer** tool:

a) Configure each peripheral gateway (PG) to be used for the **Switch** leg. In the tree view pane, select the applicable PG, and set the following:

1 **Logical Controller** tab:

- Client Type: **VRU**
- Name: A name descriptive of this PG

  For example: **<location>_A** for side A of a particular location

2 **Peripheral** tab:

- Peripheral Name: A name descriptive of this Unified CVP peripheral

  For example: **<location>_<cvp1> or <dns_name>**
- Client Type: **VRU**
- Select the check box: **Enable Post-routing**

3 **Routing Client** tab:

- Name: By convention, use the same name as the peripheral.
- Client Type: **VRU**

For more information, see the ICM Configuration Guide for Cisco ICM Enterprise Edition.

b) Configure a peripheral for each Unified CVP Call Server to be used for a Switch leg connected to each PG.

Step 8 Configure dialed numbers.
On the Unified ICME or Unified ICMH Server, in the ICM Configuration Manager, configure the following items:

a) **Dialed Number List Tool** tab: Configure the dialed numbers.
b) **Call Type List tool** tab: Configure the call types.
c) **ICM Instance Explorer tool** tab: Configure the applicable customers.

For more information, see the ICM Configuration Guide for Cisco ICM Enterprise Edition.

Step 9 Create a Routing Script.
On the Unified ICME or Unified ICMH Server in the ICM Script Editor tool:

Create a routing script that handles the incoming call. The routing script must execute a Label node or Select node (node that returns a label right away).

Note *Do not* use the Queue node in the routing script.

The label must be configured in the SIP Proxy Server to the IP address of the device that the label corresponds to. The Proxy Server is optional. If you do not have one, you must configure the Gateway dial-peer to point to the Call Server (refer to the first step in this process). Also, you must configure the **destination labels** in the SIP Service for the Call Server.

For more information about creating scripts, see the Writing Scripts for Unified CVP

See the Scripting and Media Routing Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted for more information.

Step 10 Configure the SIP Proxy Server using the Operations Console.
Select **Device Management** > **SIP Proxy Server**.

Step 11 In the Operations Console, install and configure Call Servers.

a) Enable the ICM and SIP Services on the Call Server.
In the Operations Console, select **Device Management** > **Unified CVP Call Server**.

Click **Help** > **This Page** for details.

Select the check boxes: **ICM** and **SIP**

b) Configure the SIP Service:
Select **Device Management** > **CVP Call Server** > **SIP tab**.

- If you are using a SIP Proxy Server, enable the Outbound Proxy and select the SIP Proxy Server. If using a SIP Proxy Server, configure Local Static Routes on the SIP Proxy Server itself.
- If you are not using a SIP Proxy Server, configure Local Static Routes using the Dialed Number Pattern system configuration in the Operations Console. A local static route must be configured for each SIP gateway/ACD, SIP endpoint in order to receive calls.
- Check the default values for the SIP Service and change, if desired.

See the Configure the SIP Devices and to SIP DN Pattern Matching Algorithm for detailed information.

c) Configure the ICM Service by setting the maximum length DNIS to the length of the Network Routing Number:

- Select **Device Management** > **CVP Call Server** > **ICM tab**.
- Set the Maximum Length of DNIS to length of the Network Routing Number.

Example: if the Gateway dial pattern is 1800******, the maximum DNIS length is 10.

For detailed information, see the *Operations Console online help*.

Step 12 Configure local static routes:
If an outbound proxy is enabled on the Operations Console, configure local static routes on the SIP Proxy Server.

If no outbound proxy is enabled, configure local static routes using the Operations Console Dialed Number Pattern system configuration. See the SIP DN Pattern Matching Algorithm for detailed information.

The following is an example of a local static route configuration. A local static route contains a dialed number pattern and a routing address (IP Address, Hostname, or SIP Server Group name):

- **22291>,cvp-ringtone.cisco.com**

- **22292>,cvp-error.cisco.com**
- 1>,ccm-subscribers.cisco.com
- 2>,ccm-subscribers.cisco.com
- 3>,ccm-subscribers.cisco.com

Step 13 (Optional) In the Operations Console, configure the **Reporting Server**. Select **Device Management** > **CVP Reporting Server** > **General tab**:

a) Configure the Reporting Server.
b) Select a Call Server to associate with this Reporting Server.
c) Check the default values of the Reporting properties and change, if desired.

For more information, see the Reporting Guide for Cisco Unified Customer Voice Portal.

Ingress Gateway Configuration Examples

Example Gateway Settings for Call Director Call Flow Model

The first part of the following example provides the basic configuration for setting an Ingress gateway:

- Applies a timestamp to debugging and log messages
- Turns on logging
- Turns off printing to the command line interface console
- Sends RTP packets
- Configures gateway settings

The last part of this example provides the following:

- Allows SIP to play a .wav file that enables caller to hear message from critical_error.wav
- Performs survivability
- Enables SIP to play ring tone to caller while caller is being transferred to an agent
- Logs errors on the gateway when the call fails
- Defines requirements for SIP Call Server

```
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
!
service internal
logging buffered 99999999 debugging
no logging console
!
ip cef
!
voice rtp send-recv
!
voice service voip
signaling forward unconditional
h323
sip
min-se 360
header-passing
!
voice class codec 1
codec preference 1 g711ulaw
codec preference 2 g729r8
!
application
service cvperror flash:cvperror.tcl
!
service cvp-survivability flash:survivability.tcl
!
service ringtone flash:ringtone.tcl
!
service handoff flash:handoff.tcl!gateway
!
gateway
timer receive-rtcp 6
!
ip rtcp report interval 3000
!
sip-ua
retry invite 2
timers expires 60000
sip-server ipv4:<IP of CUSP Server or Call Server>:5060
reason-header override
!
```

Example of Incoming Pots Dial-peer for Call Director Call Flow Model

The following example provides the configuration for an incoming Pots call for the Call Director call flow model:

```
dial-peer voice 8 pots
description Example incoming POTS dial-peer
service cvp-survivability
incoming called-number <your DN pattern here>
direct-inward-dial
!
```

Example of SIP Ringtone Dial-peer for Call Director Call Flow Model

The following example provides the configuration for a SIP ringtone for the Call Director callflow model:

```
dial-peer voice 9191 voip
description SIP ringtone dial-peer
service ringtone
voice-class codec 1
voice-class sip rel1xx disable
incoming called-number <your ringtone DN pattern here>
dtmf-relay rtp-nte
no vad
!
```

Example of SIP Error Dial-peer for Call Director Call Flow Model

The following example provides the configuration for a SIP error dial-peer for the Call Director call flow model:

```
dial-peer voice 9292 voip
description SIP error dial-peer
service cvperror
voice-class codec 1
voice-class sip rel1xx disable
incoming called-number <your error DN pattern here>
dtmf-relay rtp-nte
no vad
!
```

Example of Dial-peer to Reach the Unified CVP Call Server or CUSP Server for Call Director Call Flow Model

The following example provides the configuration for a dial-peer to reach the Unified CVP Call Server or CUSP Server for the Call Director call flow model:

```
dial-peer voice 800 voip
description Example Call Server Dialpeer with CUSP Server
destination-pattern <your DN pattern here>
voice-class codec 1
session protocol sipv2
session target sip-server
dtmf-relay rtp-nte
no vad
!
```

DNS Zone File Configuration for Call Director Call Flow Model

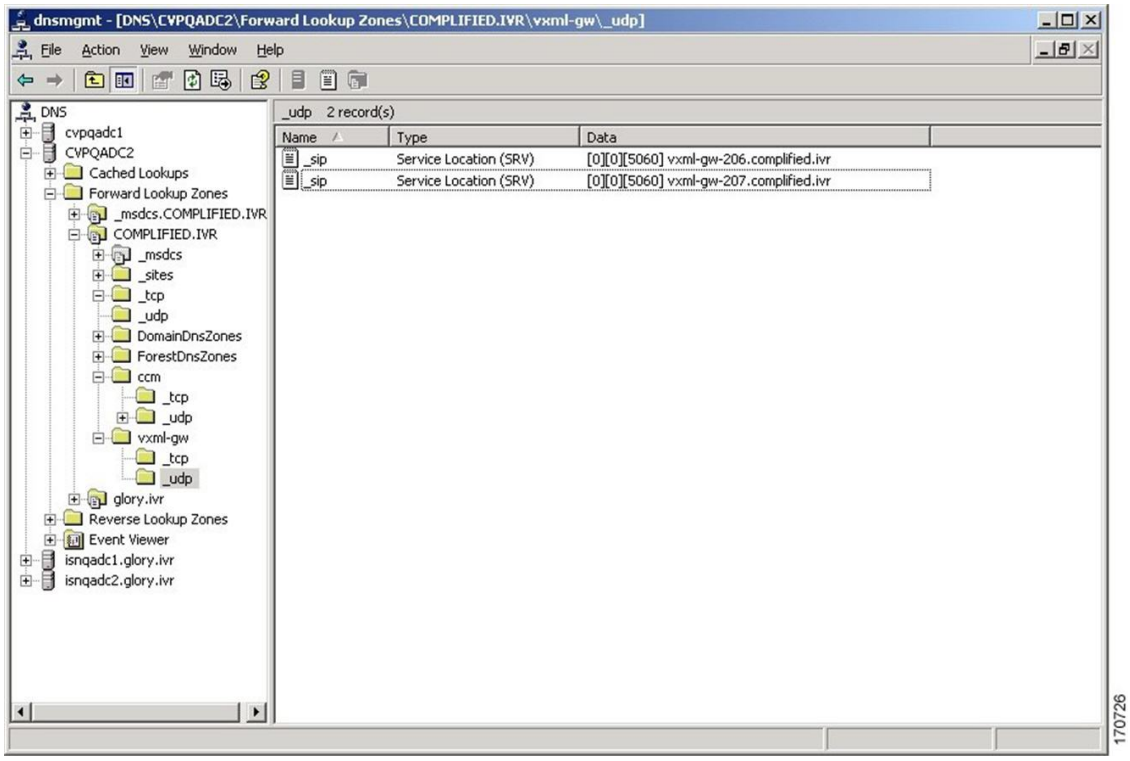
DNS Zone File Linux Configuration Example

The following is an example of a DNS zone file Linux NAMED configuration:

```
ringtone-1 IN A 10.86.129.20
         ringtone-2 IN A 10.86.129.229
         vxml-1 IN A 10.86.129.20
         vxml-2 IN A 10.86.129.229
         vxml-3 IN A 161.44.81.254
         cvp-1 IN A 10.86.129.211
         cvp-2 IN A 10.86.129.220
         cvp-3 IN A 161.44.81.254
         ; Priority Weight Port Target
         sip._tcp.ringtone.sox.cisco.com. SRV 1 1 5060 ringtone-1.sox.cisco.com.
         _
         SRV 1 1 5060 ringtone-2.sox.cisco.com.
         sip._udp.ringtone.sox.cisco.com. SRV 1 1 5060 ringtone-1.sox.cisco.com.
         _
         SRV 1 1 5060 ringtone-2.sox.cisco.com.
         _sip._tcp.vxml.sox.cisco.com. SRV 1 1 5060 vxml-1.sox.cisco.com.
         SRV 1 1 5060 vxml-2.sox.cisco.com.
         SRV 1 1 5060 vxml-3.sox.cisco.com.
         _sip._udp.vxml.sox.cisco.com. SRV 2 1 5060 vxml-1.sox.cisco.com.
         SRV 2 1 5060 vxml-2.sox.cisco.com.
         SRV 1 1 5060 vxml-3.sox.cisco.com.
         _sip._tcp.cvp.sox.cisco.com.  SRV 1 1 5060 cvp-1.sox.cisco.com.
         SRV 2 1 5060 cvp-2.sox.cisco.com.
         SRV 3 1 5060 cvp-3.sox.cisco.com.
         _sip._udp.cvp.sox.cisco.com. SRV 1 1 5060 cvp-1.sox.cisco.com.
         SRV 2 1 5060 cvp-2.sox.cisco.com.
         SRV 3 1 5060 cvp-3.sox.cisco.com.
```

DNS Zone File MS DNS Configuration Example

The following is an example of a DNS zone file MS DNS configuration:



Unified CVP Comprehensive Call Flow Models

This section describes the comprehensive call flow models used by Unified ICME and Unified ICMH products as follows:

- Unified CVP Comprehensive Call Flow Model Unified ICME
- Unified CVP Comprehensive Call Flow Model Unified ICMH

Note The SIP calls using the Unified CVP micro-applications will use the Call Server's IVR Service that has the switch leg of the call. VoiceXML fetches are always sent to the Call Server that has the switch leg of the call. The VoiceXML traffic for micro-applications must return only to the same Call Server as the switch leg.

Sending VoiceXML traffic to multiple application servers is implemented in the Unified CVP 4.0(1) and later releases by extracting the Call Server's IP from the SIP signaling messages in the bootstrap service rather than using IOS static configuration in the service parameter for the VoiceXML Gateway's bootstrap service, as was done in previous Unified CVP releases.

The service configuration parameters for the Call Server host and port are meant for the Unified CVP VRU-Only call flow model. These parameters are optional, if, for some reason, you need to override the IP address/port# of the Call Server that comes in through the SIP app-info header.

```
application
service vru-leg flash:bootstrap.tcl
param cvpserverhost xxx.xxx.xxx.xxx  <IP of primary Call Server>
param cvpserverbackup xxx.xxx.xxx.xxx  <IP of backup Call Server>
param cvpserverport 8000   <TCP Port # of Call Server>
```

The Unified CVP Comprehensive call flow model with SIP extracts the Call Server host from the signaling. The Unified CVP SIP Service is handling the switch legs of the call. If you are making a SIP call that does not involve the switch leg with Unified CVP, the service parameters below will apply for the "VRU leg only" call flow. Cisco requires that Comprehensive calls always use the same Call Server for both switch leg and VRU legs. Using the same Call Server simplifies the solution and makes it easier to troubleshoot and debug.

Note The app-info header is only for SIP calls. If this parameter is empty, then the primary Call Server IP, as configured on the service, will be used. If that server is "out-of-service," then it will try the backup Call Server.

Unified CVP Comprehensive Call Flow Model Unified ICME

The Unified CVP Comprehensive call flow model combines the Call Director and the VRU-Only call flow models. It provides initial prompt and collect, self service IVR, queuing, and VoIP switching among Unified ICME and TDM agents.

In this call flow model, Unified CVP acts as the switch, transferring the call to the Network VRU and to agents. The Unified CVP IVR service in the Operations Console is also configured to work with the VoiceXML Gateway to provide VRU treatment, which might include ASR/TTS.

The Unified CVP Comprehensive call flow model requires the following components:

- Call Server
- Unified ICME
- Ingress Gateway
- VoiceXML Gateway
- Operations Console

The following optional components can be used in this call flow model:

- Reporting Server
- VXML Server
- Call Studio
- Speech Servers

Caution The Operations Console can only manage speech servers installed on *Windows*, not on Linux. If the speech server is installed on Linux, the server cannot be managed.

- Media Servers
- DNS Server
- SIP Proxy Server (if the Call Server is configured to use SIP signaling, a SIP Proxy Server is optional)

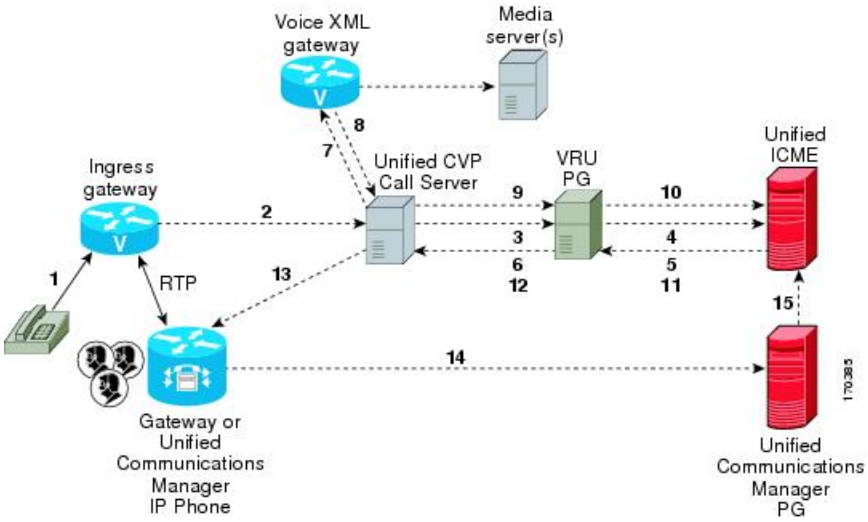
In this call flow model, both the Voice Gateway and the Call Server see two call legs for the same call:

- One for the Switch leg
- One for the VRU leg

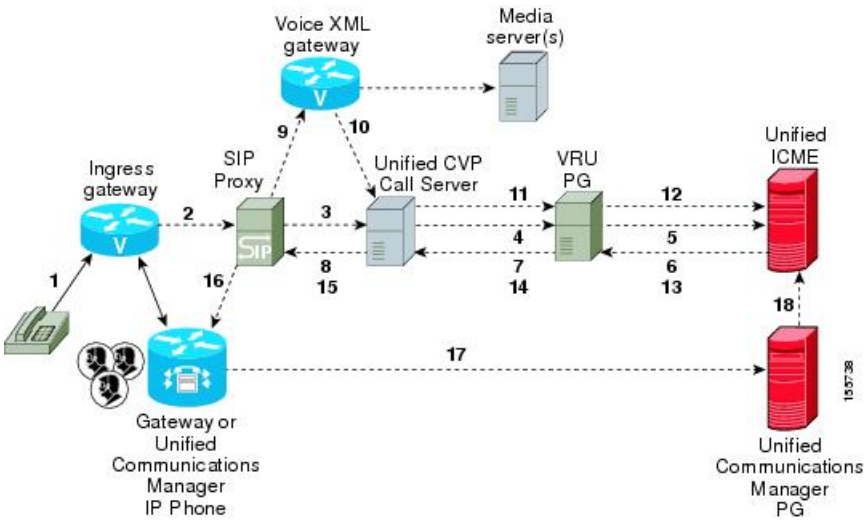
For the **Switch** leg, the Gateway simply provides Gateway capabilities from TDM to VoIP, and call-switching capabilities. For the **VRU** leg, the Gateway provides VRU voice treatment.

Note
- Although the following figures show two Gateways (one where the call arrives and a separate one for the VRU leg), these could be the same physical Gateway.
- For simplicity, the figures do not illustrate a call flow model for redundancy and failover.

The following figure shows the call flow for this call flow model using SIP without a Proxy Server. The solid lines indicate voice paths and dashed lines indicate signaling paths.



The following figure shows the call flow for this call flow model using SIP with a Proxy Server.



Configuration overview for this call flow model:

- The one Network VRU, Type 10, serves both the Switch and VRU legs.
- Use the ICM Script Editor's SendToVRU node to connect the call to the Network VRU.

Note See the REFER Transfers and sendtooriginator Setting in the SIP Service for more information.

Unified CVP Comprehensive Call Flow Model Unified ICMH

In this call flow model, Unified CVP is deployed at the NAM where it acts as the Switch, transferring the call to the Network VRU (using the Correlation ID transfer mechanism) and to agents. The Unified CVP IVR Service in the Operations Console is also configured to work with the VoiceXML Gateway to provide VRU treatment, which might include ASR/TTS.

Note This call flow model does not support calls that originate in IP.

Note See the "Calls Which Are Originated by Unified CM section for instructions on how to implement IP-originated calls in a way which is supplemental to this call flow model (Unified CVP Comprehensive Call Flow Model, Unified ICMH); however, this implementation requires that an additional Unified CVP Call Server be attached to the CICM.

In this call flow model, both the VoiceXML Gateway and the Unified CVP Call Server see two calls:

- One for the Switch leg
- One for the VRU leg

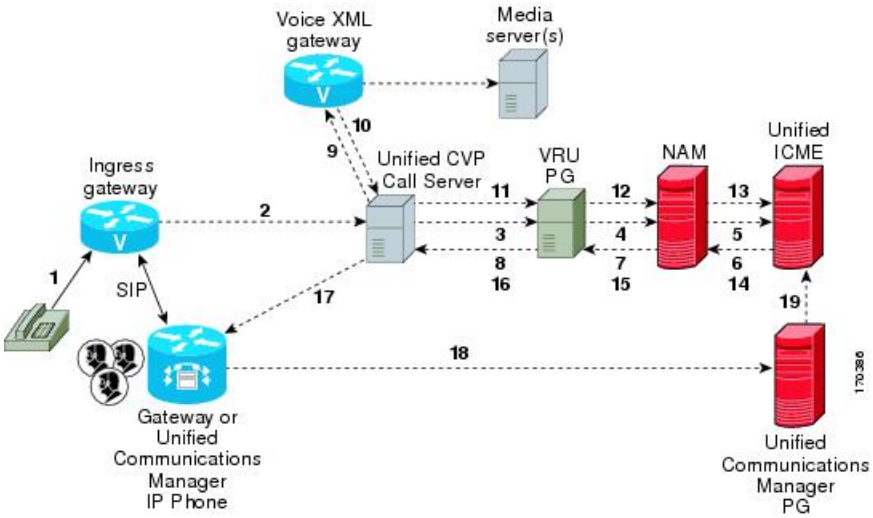
For the Switch leg, the Gateway simply provides Gateway capabilities from TDM to VoIP and call-switching capabilities. For the VRU leg, the Gateway provides VRU voice treatment.

Unified ICMH sees these as a single call routed through different peripherals for different purposes.

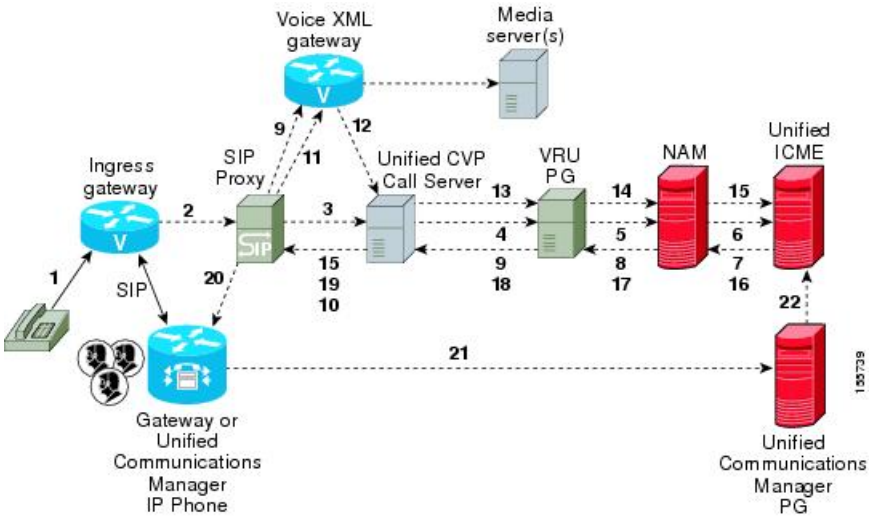
The following figure shows the call flow for this call flow model using SIP without a Proxy Server.

Note

- In the following figures, solid lines indicate voice paths and dashed lines indicate signaling paths. Although the figure shows two Gateways (one where the call arrives and a separate one for the VRU leg), these could be the same physical Gateway. Similarly, the Unified CVP IVR Service in the Operations Console and the PG could be the same physical machine.
- For simplicity, the figure does not illustrate a call flow model for redundancy and failover.



The following figure shows the call flow for this call flow model using SIP with a Proxy Server.



Note The numbers in the figure indicate call flow progression.

Configuration overview for this call flow model:

- There are two Network VRUs:
  - One on the NAM for the Switch leg and the VRU leg (Type 10)
  - One for the CICM for the INCRP connection
- The Network VRU names (where applicable) and the ECC variable configurations must be identical on the NAM and CICM. All labels must also be duplicated but their routing clients will be different.
- Use the ICM Script Editor's SendToVRU node to connect the call to the Network VRU.

See the Set Up Comprehensive Call Flow Model Using SIP for Both ICME and ICMH.

Note See the REFER Transfers and sendtooriginator Setting in the SIP Service for more information.

Set Up Comprehensive Call Flow Model Using SIP for Both ICME and ICMH

The following high-level configuration steps provide configuration instructions for the Comprehensive Call Flow Model using *SIP* for *both* Unified ICME and Unified ICMH.

Note See the Unified CVP Comprehensive Call Flow Model Unified ICMH for detailed information about the Unified ICMH version of this call flow model.

Procedure

Step 1 Install the IOS image on the Ingress Gateway.
For detailed information, see the Cisco IOS documentation.

Step 2 Transfer the following script, configuration, and .wav files to the Ingress gateway using the Unified CVP Operations Console (or the Unified CVP product CD):

- bootstrap.tcl
- handoff.tcl
- survivabilty.tcl
- bootstrap.vxml
- recovery.vxml
- ringtone.tcl
- cvperror.tcl
- ringback.wav
- critical_error.wav

In the Operations Console, select **Bulk Administration** > **File Transfer** > **Scripts and Media** and select **Help** > **This Page** for details.

Step 3 Configure the Ingress Gateway base settings.

Step 4 Configure the Ingress Gateway service settings.

Step 5 Configure an Ingress Gateway incoming Pots Dial-peer.

Step 6 (Optional) Configure a dial-peer for ringtone and error.

Step 7 If you are **using a Proxy Server**, configure your session target in the outbound Dial-peer to point to the Proxy Server.

Step 8 If you are using the sip-server global configuration, configure the sip-server in the sip-ua section to be your Proxy Server and point the session target of the dial-peer to the sip-server global variable.

Note
1. Make sure your Dial plan includes this information. See the Dial plan when you configure the SIP Proxy Server for Unified CVP.
2. The SIP Service voip dial-peer and the destination pattern on the Ingress Gateway must match the DNIS in static routes on the SIP Proxy Server or Unified CVP Call Server.

Note

See the Configure the SIP Devices and SIP DN Pattern Matching Algorithm for detailed information.

Step 9 For **SIP without a Proxy Server** , complete the following steps:

a) If you are using DNS query with SRV or A types from the gateway, configure the gateway to use DNS. Also, if you are using DNS query with SRV or A types from the gateway, use CLI as shown below:

Note Normally, non-DNS setup is: sip-server ipv4:xx.xx.xxx.xxx:5060
.

```
ip domain name pats.cisco.com
ip name-server 10.86.129.16
```

```
sip-ua
sip-server dns:cvp.pats.cisco.com
OR:
ipv4:xx.xx.xxx.xxx:5060
```

b) Configure the DNS zone file for the separate DNS server that displays how the Service (SRV) records are configured.

Note SRV with DNS can be used in *any* of the SIP call flow models, with or without a Proxy server. Standard A type DNS queries can be used as well for the calls, without SRV, but they lose the load balancing and failover capabilities.

See theDNS Zone File Configuration for Comprehensive Call Flow Model.

Step 10 For **SIP with a Proxy Server**, if you are using the DNS Server, you can set your SIP Service as the Host Name (either A or SRV type).

You can also configure the Gateway statically instead of using DNS. The following example shows how both the A and SRV type records could be configured:

```
ip host cvp4cc2.cisco.com 10.4.33.132
ip host cvp4cc3.cisco.com 10.4.33.133
ip host cvp4cc1.cisco.com 10.4.33.131
```

For SIP/TCP:

```
ip host _sip._tcp.cvp.cisco.com srv 50 50 5060 cvp4cc3.cisco.com
ip host _sip._tcp.cvp.cisco.com srv 50 50 5060 cvp4cc2.cisco.com
ip host _sip._tcp.cvp.cisco.com srv 50 50 5060 cvp4cc1.cisco.com
```

For SIP/UDP:

```
ip host _sip._udp.cvp.cisco.com srv 50 50 5060 cvp4cc3.cisco.com
ip host _sip._udp.cvp.cisco.com srv 50 50 5060 cvp4cc2.cisco.com
ip host _sip._udp.cvp.cisco.com srv 50 50 5060 cvp4cc1.cisco.com
```

Note The DNS Server must be configured with all necessary A type or SRV type records.

See the Configure the SIP Devices and the *Operations Console online help*, **Managing devices** > **Configuring a SIP Proxy Server** for detailed information.

Step 11 Transfer files to the **VXML** Gateway.

Transfer the following scripts, configuration, and .wav files to the gateway, using the Operations Console (or the Unified CVP product CD):

- bootstrap.tcl
- handoff.tcl
- survivabilty.tcl
- bootstrap.vxml
- recovery.vxml
- ringtone.tcl
- cvperror.tcl
- ringback.wav
- critical_error.wav

In the Operations Console, select **Bulk Administration** > **File Transfer** > **Scripts and Media** and select **Help** > **This Page** for details.

Step 12 Configure the VXML Gateway base settings.

Step 13 Configure the VXML Gateway service settings.

Step 14 Configure the ICM VRU Label.

Step 15 Optionally, enable security media fetches.

Note
- The VXML that the IVR Service returns as a response to an HTTP/HTTPS request from the VXML gateway contains URLs to media servers, so that the gateway knows where to fetch the media files from.
- The URLs to the media servers in the VXML returned by the IVR Service can be controlled so that they are either HTTP or HTTPS URLs. This property is a boolean property called **Use Security for Media Fetches**. By default, it is set to "false". A value of "true" means generate HTTPS URLs to media servers and a value of "false" means generate HTTP URLs to media servers.
  This property is only applicable, if the following condition is true:

  In the Unified ICME script, the media server (specified in ECC variable *call.user.microapp.media_server*) is not set to a URL that explicitly specifies an HTTP or HTTPS scheme. An example of a URL that explicitly specifies an HTTP scheme is `http://<servername>:80`. One that specifies an HTTPS scheme is `https://<servername>:443`. An example of a URL that does **not** specify the scheme is *<servername>*.

  In the Operations Console, the user-visible text for this property is "Use Security for Media Fetches." Do not restart the Call Server for this property to take effect.

Click the **Use Security for Media Fetches** check box on the IVR Service tab.

See the *Operations Console online help* for detailed information about the IVR Service.

See the Define Unified CVP ECC Variables for details about the user.microapp.media_server ECC variable.

Step 16 If using ASR and TTS Servers, specify IP addresses for those servers for each locale using the applicable name resolution system for the Gateway (DNS or "ip host" commands).

Note If ASR and TTS use the same server, the MRCP server might allocate one license for the ASR session and a second license for the TTS section. If you are hosting both ASR and TTS on the same speech server, you must select the **ASR/TTS use the same MRCP server** option in the IVR Service configuration tab in the Operations Console and follow the instructions in the step below.

Do **one** of the following:

- If you *are* using ACE, the server name is configured to the virtual IP (VIP) of the Call Server on ACE. For more information, see the Configure High Availability for Unified CVP.
- The primary and backup servers must be configured. If using name resolution local to the Gateway (rather than DNS) specify:

  ip host asr- *<locale> <ASR server for locale>*

  ip host asr- *<locale>*-backup *<backup ASR server for locale>*

  ip host tts- *<locale> <TTS server for locale>*

  ip host tts- *<locale>*-backup *<backup TTS server for locale>*

  Example for English US, use:

ip host asr-en-us 10.86.129.215

Step 17 If you want the ASR and TTS to use the same MRCP server option, you must configure the gateway as follows.

a) In the IVR Service in the Operations Console, select the **ASR/TTS use the same MRCP server** option.

b) Add the following two host names to the gateway configuration:

- ip host asrtts- *<locale> <IP Address Of MRCP Server>*
- ip host asrtts- *<locale>* -backup *<IP Address Of MRCP Server>*

  Where the *locale* might be something like en-us or es-es, resulting in `asrtts-en-us` or `asrtts-es-es`.

c) Change the 'ivr asr-server' and 'ivr tts-server' lines as follows:

- ivr asr-server rtsp://asr-en-server/recognizer
- ivr tts-server rtsp://tts-en-server/synthesizer

Step 18 Configure the speech servers to work with Unified CVP.

Caution The Operations Console can only manage speech servers installed on *Windows*, not on Linux. If the speech server is installed on Linux, the server cannot be managed.

To ensure that the speech servers work with Unified CVP, you must make the following changes on each speech server as part of configuring the Unified CVP solution.

If you are using Nuance SpeechWorks MediaServer (SWMS), the configuration file is osserver.cfg. If you are using Nuance Speech Server (NSS), the configuration file is NSSserver.cfg.

Make the following changes to the Nuance configuration file:

- **Change:** server.resource.2.url VXIString media/speechrecognizer

  To: server.resource.2.url VXIString recognizer
- **Change:** server.resource.4.url VXIString media/speechsynthesizer

  To: server.resource.4.url VXIString synthesizer
- **Change:** server.mrcp1.resource.3.url VXIString media/speechrecognizer

  To: server.mrcp1.resource.3.url VXIString /recognizer
- **Change:** server.mrcp1.resource.2.url VXIString media/speechsynthesizer

  To: server.mrcp1.resource.2.url VXIString media/synthesizer
- **Change:** server.mrcp1.transport.port VXIInteger 4900

  To: server.mrcp1.transport.port VXIInteger 554

If you are using Nuance Speech Server 5 and/or Nuance Vocalizer for Network 5, make changes to configuration files for each application. Make the following changes to the Nuance Speech Server 5 configuration file (NSSserver.cfg):

- **Change:** server.mrcp1.resource.3.url VXIString media/speechrecognizer

  To: server.mrcp1.resource.3.url VXIString /recognizer
- **Change:** server.mrcp1.resource.2.url VXIString media/speechsynthesizer

  To: server.mrcp1.resource.2.url VXIString /synthesizer

- **Change:** server.mrcp1.transport.port VXIInteger 4900

  To: server.mrcp1.transport.port VXIInteger 554

- **Change:** server.mrcp1.transport.dtmfPayloadType VXIInteger 96

  To: server.mrcp1.transport.dtmfPayloadType VXIInteger 101

- **Uncomment the following:** server.rtp.dtmfTriggerLeading VXIInteger 0

  If you are using the Nuance Vocalizer for Network 5 TTS System, the following configuration files will need to be updated:

  <install path>\Nuance Vocalizer for Network 5.0\config\ttsrshclient.xml

- **Change:** <ssml_validation>strict</ssml_validation>

  To:<ssml_validation>warn</ssml_validation>

  <install path>\Nuance Vocalizer for Network 5.0\config\ttssapi.xml

- **Change:** <ssml_validation>strict</ssml_validation>

  To: <ssml_validation>warn</ssml_validation>

If you are using Nuance Speech Server 10.0, make the following changes to the Nuance configuration file (NSSserver.cfg - C:\Program Files (x86)\Nuance\Speech Server\Server\config):

- **Change:** server.mrcp1.resource.3.url VXIString media/speechrecognizer

  To: server.mrcp1.resource.3.url VXIString /recognizer

- **Change:** server.mrcp1.resource.2.url VXIString media/speechsynthesizer

  To: server.mrcp1.resource.2.url VXIString /synthesizer

- **Change**: server.mrcp1.transport.port VXIInteger 4900

  To: server.mrcp1.transport.port VXIInteger 554

- **Change:** server.mrcp1.transport.dtmfPayloadType VXIInteger 96

  To: server.mrcp1.transport.dtmfPayloadType VXIInteger

Make the following change to the Baseline.xml file `C:\Program Files\Nuance\Recognizer\config`

Change: <ssml_validation>strict</ssml_validation>

To:<ssml_validation>warn</ssml_validation>.

Step 19 Configure the characteristics for the VRU leg.
These characteristics are for VRU legs requiring ASR and/or TTS treatment. If you have other requirements for DTMF relay, Codecs or VAD settings, you must modify the commands accordingly.

Step 20 Configure SIP-Specific Actions.
On the Unified CM server, CCMAdmin Publisher, configure **SIP-specific actions**:

a) Create SIP trunks:

- If you are using a SIP Proxy Server, set up a SIP trunk to the SIP Proxy Server.
- Add a SIP Trunk for the Unified CVP Call Server.

- Add a SIP Trunk for each Ingress gateway that will send SIP calls to Unified CVP that might be routed to Unified CM.

Select **Device** > **Trunk** > **Add New** and add the following:

- Trunk Type: **SIP trunk**
- Device Protocol: **SIP**
- Destination Address: IP address or host name of the SIP Proxy Server (if using a SIP Proxy Server). If not using a SIP Proxy Server, enter the IP address or host name of the Unified CVP Call Server.
- DTMF Signaling Method: **RFC 2833**
- Do **not** check the **Media Termination Point Required** checkbox.
- If you are using UDP as the outgoing transport on Unified CVP, also set the outgoing transport to **UDP** on the SIP Trunk Security Profile.

b) Add route patterns for outbound calls from Unified CM devices using a SIP Trunk to the Unified CVP Call Server. Also, add a route pattern for error DN.

Note CVP solution does not support 100rel. Under the SIP profile for the Trunk, confirm that SIP Rel1xx Options are disabled.

For warm transfers, the call from Agent 1 to Agent 2 does not typically use a SIP Trunk, but you must configure the CTI Route Point for that dialed number on the Unified CM Server and associate that number with your peripheral gateway user (PGUSER) for the JTAPI gateway on the Unified CM peripheral gateway. An alternative is to use the Dialed Number Plan on Unified ICME to bypass the CTI Route Point.

c) Select **Call Routing** > **Route/Hunt** > **Route Pattern** > **Add New**.

- Route Pattern: Specify the route pattern; for example: 3xxx for a TDM phone that dials 9+3xxx and all Unified ICME scripts are set up for 3xxx dialed numbers.
- Gateway/Route List: Select the SIP Trunk defined in Step 2.

d) If you are sending calls to Unified CM using an SRV cluster domain name, configure the cluster domain name.

- Select: **Enterprise Parameters** > **Clusterwide Domain Configuration**
- Add the Cluster fully qualified domain name: **FQDN**

For detailed instructions about using Unified CM and the CUSP Server, see the Cisco Unified SIP Proxy Server documentation.

Step 21 (Optional) Configure the **SIP Proxy Server**.
From the CUSP Server Administration web page (**http://<CUSP server>/admin**):

a) Configure the SIP static routes to the Unified CVP Call Server(s), Unified CM SIP trunks, and Gateways. Configure the SIP static routes for intermediary transfers for ring tone, playback dialed numbers, and error playback dialed numbers.

Note For failover and load balancing of calls to multiple destinations, configure the CUSP Server static route with priority and weight.

See the Configure the SIP Devices and to SIP DN Pattern Matching Algorithm for detailed information.

b) Configure Access Control Lists for Unified CVP calls.

- Select **Proxy Settings** > **Incoming ACL**.
- Set address pattern: **all**

c) Configure the service parameters.
Select **Service Parameters**, and set the following:

- Add record route: **off**
- Maximum invite retransmission count: **2**
- Proxy Domain and Cluster Name: if using DNS SRV, set to the FQDN of your Proxy Server SRV name.

d) Write down the IP address and host name of the SIP Proxy Server. You need this information when configuring the SIP Proxy Server in Unified CVP.

e) If using redundant SIP Proxy Servers (primary and secondary or load balancing), decide whether to use DNS server lookups for SRV records or non-DNS based local SRV record configuration.
The Comprehensive call flow model with SIP calls will typically be deployed with dual CUSP Servers for redundancy. In some cases, you might want to purchase a second CUSP Server. Regardless, the default transport for deployment will be UDP. Make sure you *always* set the AddRecordRoute setting to **Off** with CUSP Servers.

Configure the SRV records on the DNS server or locally on Unified CVP with an .xml file (local xml configuration avoids the overhead of DNS lookups with each call).

Step 22 Define Network VRUs.

a) On Unified ICME or the NAM, ICM Configuration Manager, select **Network VRU Explorer** tool, define a Network VRU for the VRU leg and labels for each Unified CVP Call Server.

b) On the *CICM only*, ICM Configuration Manager, select **Network VRU Explorer tool**, define a Network VRU for the VRU leg and labels for reaching the NAM.

For each of the two previous substeps, specify the following:

- Type: **10**
- Name: *<Network VRU Name>*

  For example: **cvp**
- Define a label for each Unified CVP Call Server that is handling the Switch leg:
  - Label: *<Network Routing Number>*
  - Type: **Normal**
  - Routing client for Unified ICME or the NAM: Select the routing client configured for that Unified CVP Call Server peripheral from the drop-down list.
  - Routing client for *CICM only* : Select the INCRP routing client from the drop-down list.

Note The Network VRU label in the NAM and CICM must be identical. The Network VRU Names on the NAM and CICM should also be identical to avoid confusion.

Step 23 Define network VRUs and PGs for the switch leg in the ICM Configuration Manager.

On Unified ICMH, on the NAM and CICMs, Network VRU Explorer tool, define one label per Unified CVP Call Server or NIC routing client.

Note Use the same Type 10 Network VRU that you defined in the previous steps for the VRU leg.

For more information, see the ICM Configuration Guide for Cisco ICM Enterprise Edition.

Step 24 Set the client type for the INCRP NIC.
On the **CICM**, ICM Configuration Manager, NIC Explorer tool, set the client type for the INCRP NIC.

- Client Type: **VRU**

Step 25 Define a VRU that uses INCRP.
On the **CICM**, ICM Configuration Manager, Network VRU Explorer tool:

a) Define a Network VRU with a label that uses INCRP as its routing client.
Specify the following:

- Type: **10**
- Name: *<name of Unified CVP VRU>*

  For example: **cvpVRU**

b) Define one label for the NAM routing client.
Specify the following:

- Type: **Normal**
- Label: *<Network Routing Number>*
- Routing client: **INCRP NIC**

For more information, see the ICM Configuration Guide for Cisco ICM Enterprise Edition .

Step 26 Configure Peripheral Gateways (PGs).
On the NAM, ICM Configuration Manager, **PG Explorer** tool, configure a peripheral gateway (PG) for the Unified CVP. Configure a PG for each Unified CVP Call Server as follows:

In the tree view pane, select the applicable PG.

Logical Controller tab:

- Client Type: **VRU**
- Name: A name descriptive of this PG

  For example: **<location>_A** for side A of a particular location

Peripheral tab:

- Peripheral Name: Descriptive name of this Unified CVP peripheral

  For example: **<location>_<cvp1> or <dns_name>**
- Client Type: **VRU**
- Select: **Enable Post-routing**

Advanced tab:

- Select the name of the Unified CVP VRU from the Network VRU field drop-down list.

  For example: **cvpVRU**

Routing Client tab:

- Name: By convention, use the same name as the peripheral
- Client Type: **VRU**
- If you are in a Unified ICMH environment and configuring the CICM, then do the following:
  - *Do not* select the **Network Transfer Preferred** checkbox
  - Routing client: **INCRP NIC**

Step 27 Define a default network VRU on Unified ICME or the NAM, in the ICM Configuration Manager, the **System Information** tool:

a) For Unified ICME or on the **CICM only**, define a default Network VRU.

- Define the Default Network VRU: *<Network VRU Name>*

  For example: **cvpVRU**

b) If there are Routing Scripts on the **NAM**, define a default Network VRU.

For more information, see the ICM Configuration Guide for Cisco ICM Enterprise Edition.

Step 28 Configure dialed numbers, call types, and customers on the Unified ICME or Unified ICMH Server in the ICM Configuration Manager:

a) **Dialed Number List Tool tab:** Configure the dialed numbers.
b) **Call Type List tool tab:** Configure the call types.
c) **ICM Instance Explorer tool tab:** Configure the applicable customers.

For more information, refer to ICM Configuration Guide for Cisco ICM Enterprise Edition.

Step 29 Configure ECC variables.
On Unified ICME, ICM Configuration Manager, configure ECC variables.

For more information, refer to Define Unified CVP ECC Variables.

Step 30 Create a routing script that handles the incoming calls.
On the Unified ICME or Unified ICMH Server in the ICM Script Editor tool, use the SendToVRU node to connect the call to the Network VRU.

For more information about creating scripts, refer to Writing Scripts for Unified CVP.

Refer to Scripting and Media Routing Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted for more information.

Step 31 Optionally, configure the **SIP Proxy**.
If using a SIP Proxy Server, configure it in the Unified CVP Operations Console.

Select: **Device Management** > **SIP Proxy Server**

Step 32 Install and configure the **Call Server(s)**.
In the Operations Console:

a) Enable the ICM and SIP Services on the Call Server.

- In the Operations Console select **Device Management** > **Unified CVP Call Server**.

  Click **Help** > **This Page** for details.

- Select the check boxes: **ICM** and **SIP**

b) In the Operations Console select **Device Management** > **Unified CVP Call Server** > **SIP**. Configure the SIP Service:

- If you are using a SIP Proxy Server, enable the Outbound Proxy and select the SIP Proxy Server.

  Select the **SIP tab** and configure the following:

  - Enable Outbound Proxy: **Yes**
  - Outbound Proxy Host: Select from drop-down list.
  - Configure Local Static Routes on the SIP Proxy Server itself.

- If you are **not** using a SIP Proxy Server, configure Local Static Routes using the Dialed Number Pattern system configuration on the Operations Console. A Local Static Route must be configured for each SIP gateway/ACD, SIP endpoint in order to receive calls.

  Local Static Routes, Dialed Number (DN): Specify the dialed number pattern for the destination.

  Valid number patterns include the following characters:

  - Use the period (**.**) or **X** character for single-digit wildcard matching in any position.
  - Use the greater than (**>**), asterisk (*****), or exclamation (**!**) characters as a wildcard for 0 or more digits at the end of the DN.
  - Do *not* use the **T** character for wildcard matching.
  - Dialed numbers must not be longer than 24 characters.
  - Refer to *Valid Formats for Dialed Numbers* in the Operations Console online help for format and precedence information.

  Example: **9>** (Errors are 9292 and ringtone is 9191)

  Refer to Configure the SIP Devices, and to SIP DN Pattern Matching Algorithm for detailed information.

  The following static route configuration is *not* correct because the least explicit routes must appear at the end. Load balancing and/or failover of calls will require DNS SRV domain names, not multiple routes with the same DN Pattern, but a single route to an SRV domain name.

  Incorrect Example:

```
1>,10.2.6.1
2>,10.2.6.2
3>,10.2.6.20
2229191>,10.2.6.241
2229292>,10.2.6.241
2229191>,10.2.6.242
2229292>,10.2.6.242
2>,ccm-subscribers.cisco.com
3>,ccm-subscribers.cisco.com
```

Correct static route configuration example:

```
22291>,cvp-ringtone.cisco.com
22292>,cvp-error.cisco.com
1>,ccm-subscribers.cisco.com
2>,ccm-subscribers.cisco.com
3>,ccm-subscribers.cisco.com
```

Note "91919191>" pattern will not match an exact DN of "91919191."

- Check the default values for the SIP Service and change, if desired.

c) Configure the ICM Service by setting the maximum length DNIS to the length of the Network Routing Number.
Select **Device Management** > **CVP Call Server** > **ICM tab**: Maximum Length of DNIS: length of the Network Routing Number.

Example: if the Gateway dial pattern is 1800******, the maximum DNIS length is **10**.

For detailed information, click **Help** > **This Page** .

Step 33 Configure Local Static Routes:
If an outbound proxy is enabled on the Operations Console, configure local static routes on the SIP Proxy Server.

If no outbound proxy is enabled, configure local static routes using the Operations Console Dialed Number Pattern system configuration. Refer to SIP DN Pattern Matching Algorithm for detailed information.

The following is an example of a local static route configuration. A local static route contains a dialed number pattern and a routing address (IP Address, Hostname, or SIP Server Group name):

- 22291>,cvp-ringtone.cisco.com
- 22292>,cvp-error.cisco.com
- 1>,ccm-subscribers.cisco.com
- 2>,ccm-subscribers.cisco.com
- 3>,ccm-subscribers.cisco.com

Step 34 Configure custom ringtone patterns.
In the Operations Console, configure the custom ringtone patterns to play different ring tones based on the dialed number destination.

Select **System** > **Dialed Number Pattern** and complete the following steps:

a) Click **Add New**. The Add New Dialed Number Pattern configuration page displays.
b) Fill in the appropriate configuration settings:

Table 5: Dialed Number Pattern Configuration Settings

Property	Description	Default	Value
General Configuration			

Property	Description	Default	Value
Dialed Number Pattern	The actual Dialed Number Pattern.	None	Must be unique Maximum length of 24 characters Can contain alphanumeric characters, wildcard characters such as exclamation point (!) or asterisk (*), single digit matches such as the letter X or period (.) Can end with an optional greater than (>) wildcard character
Description	Information about the Dialed Number Pattern.	None	Maximum length of 1024 characters

Property	Description	Default	Value
Enable Local Static Route	Enable local static routes on this Dialed Number Pattern. If Local Static Routes are enabled: • **Route to Device -** Select the device from the drop-down list which contains a list of configured, supported devices. Once a selection is made, the IP Address/Hostname/Server Group Name field is automatically updated with the IP Address of the selected device. • **Route to SIP Server Group -** Select the device from the drop-down list which contains a list of configured, support devices. Once a selection is made, the IP Address/Hostname/Server Group Name field is automatically updated with the IP Address of the selected device. • **IP Address/Hostname/Server Group Name -** If you have not selected a **Route to Device** or **Route to SIP Server Group**, enter the IP address, hostname, or the server group name of the route.	Disabled	Maximum length of 128 characters Must be a valid IP address, hostname, or fully qualified domain name
Enable Send Calls to Originator	Enables calls to be sent to originator.	Disabled	n/a

Property	Description	Default	Value
Enable RNA Timeout for Outbound Calls	Enables Ring No Answer (RNA) timer for outbound calls. • **Timeout -** Enter the timeout value in seconds.	Disabled none	n/a Valid integer in the inclusive range from 5 to 60
Enable Custom Ringtone	Enables customized ring tone. • **Ringtone media filename -** Enter the name of the file that contains the ringtone.	Disabled none	Maximum length of 256 characters Cannot contain whitespace characters
Enable Post Call Survey for Incoming Calls	Enables post call survey for incoming calls. • **Survey Dialed Number Pattern -** Enter the survey dialed number pattern.	Disabled none	n/a Maximum length of 24 characters Can contain alphanumeric characters, wildcard characters such as exclamation point (!) or asterisk (*), single digit matches such as period (.) or X, and can end with an optional greater than (>)wildcard character

c) Click **Save** to save the Dialed Number Pattern.
You are returned to the **Dialed Number Pattern** page. To deploy the Dialed Number Pattern configuration, click **Deploy** to deploy the configuration to all Unified CVP Call Server devices.

Step 35 (Optional) Configure the Reporting Server and associate it with a Call Server.
In the Operations Console, select **Device Management** > **CVP Reporting Server** > **General tab** and complete the following steps:

a) Configure the Reporting Server.
b) Select a Call Server to associate with this Reporting Server.
c) Check the default values of the Reporting properties and change, if desired.

For more information, see the Reporting Guide for Cisco Unified Customer Voice Portal.

Ingress and VoiceXML Gateway Configuration Examples

Example Gateway Settings for Comprehensive Call Flow Model

The first part of the following example provides the basic configuration for setting an Ingress gateway:

- Applies a timestamp to debugging and log messages
- Turns on logging
- Turns off printing to the command line interface console
- Sends RTP packets
- Configures gateway settings

The last part of this example provides the following:

- Allows SIP to play a .wav file that enables caller to hear message from critical_error.wav
- Performs survivability
- Enables SIP to play ringtone to caller while caller is being transferred to an agent
- Logs errors on the gateway when the call fails
- Defines requirements for SIP Call Server

Note CVP solution does not support 100rel. It can be disabled on the dial-peer level or on a global level under the voice service VoIP section.

```
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
!
service internallogging buffered 99999999 debuggingn
no logging console
!
ip cef
!voice rtp send-recv
!
voice service voip
signaling forward unconditional
sip
min-se 360
header-passing
!voice class codec 1
codec preference 1 g711ulaw
codec preference 2 g729r8
!
application
service cvperror flash:cvperror.tcl
!
service cvp-survivability flash:survivability.tcl
!
service ringtone flash:ringtone.tcl
!
service handoff flash:handoff.tcl
!gateway
timer receive-rtcp 4
!
ip rtcp report interval 2000
!sip-ua
retry invite 2
timers expires 60000
sip-server ipv4:<IP of CUSP server or Call Server>:5060
reason-header override
!
```

Example of Incoming POTS Dial-peer for Comprehensive Call Flow Model

The following example provides the configuration for an incoming Pots call for the Comprehensive call flow model:

```
dial-peer voice 8 pots
description Example incoming POTS dial-peer
service cvp-survivability
incoming called-number <your DN pattern here>
direct-inward-dial
!
```

Example of SIP Ringtone Dial-peer for Comprehensive Call Flow Model

The following example provides the configuration for a SIP ringtone for the Comprehensive call flow model:

```
dial-peer voice 9191 voip
description SIP ringtone dial-peer
service ringtone voice-class codec 1
voice-class sip rel1xx disable
incoming called-number <your ringtone DN pattern here>
dtmf-relay rtp-nte
no vad
!
```

Example of SIP Error Dial-peer for Comprehensive Call Flow Model

The following example provides the configuration for a SIP error dial-peer for the Comprehensive call flow model:

```
dial-peer voice 9292 voip
description SIP error dial-peer
service cvperror
voice-class codec 1
voice-class sip rel1xx disable
incoming called-number <your error DN pattern here>
dtmf-relay rtp-nte
no vad !
```

Example of Dial-peer to Reach the Unified CVP Call Server for Comprehensive Call Flow Model

The following example provides the configuration for a dial-peer to reach the Unified CVP Call Server for the Comprehensive call flow model:

```
dial-peer voice 800 voip
description Example Call Server Dialpeer with CUSP Server
destination-pattern <your DN pattern here>
voice-class codec 1
session protocol sipv2
session target sip-server
dtmf-relay rtp-nte
no vad
!
```

VoiceXML: Example Gateway Settings for Comprehensive Call Flow Model

The first part of the following example provides the basic configuration for setting a VoiceXML gateway:

- Applies a timestamp to debugging and log messages
- Turns on logging
- Turns off printing to the command line interface console
- Sends RTP packets
- Configures ASR/TTS Server
- Configures gateway settings

The last part of this example provides the following:

- Initiates the VoiceXML leg
- Initiates the switch leg of the call
- Plays a .wav file that enables caller to hear message from critical_error.wav
- Logs errors on the gateway when the call fails

```
service timestamps debug datetime msec
service timestamps log datetime msec
service internal
logging buffered 99999999 debugging
no logging console
ip cef
no ip domain lookup
ip host tts-en-us <IP of TTS or MRCP Server>
ip host asr-en-us <IP of ASR or MRCP Server>
voice rtp send-recv
!
voice service voip
signaling forward unconditional
sip
min-se 360
header-passing
voice class
codec 1 codec preference 1 g711ulaw
codec preference 2 g729r8
!
ivr prompt memory 15000
ivr prompt streamed none
ivr asr-server rtsp://asr-en-us/recognizer
ivr tts-server rtsp://tts-en-us/synthesizer
mrcp client timeout connect 10
mrcp client timeout message 10
mrcp client rtpsetup enable
rtsp client timeout connect 10
rtsp client timeout message 10
vxml tree memory 500
http client cache memory pool 15000
http client cache memory file 500
http client connection timeout 60
http client response timeout 30
http client connection idle timeout 10
gateway
timer receive-rtcp 6
!
ip rtcp report interval 3000
application
service new-call flash:bootstrap.vxml
service cvperror flash:cvperror.tcl
service handoff flash:handoff.tcl
service bootstrap flash:bootstrap.tcl
param cvpserverss1 1
!
```

Note The optional param cvpserverss1 1 line enables HTTPS.

VoiceXML: Example of Dial-peer for ICM VRU Label for Comprehensive Call Flow Model

The following example provides the configuration for an ICM VRU label dial-peer for the Comprehensive call flow model:

```
dial-peer voice 777 voip
description ICM VRU label
service bootstrap
codec g711ulaw
incoming called-number <your sendtovru label pattern here>
dtmf-relay rtp-nte
no vad
!
```

DNS Zone File Configuration for Comprehensive Call Flow Model

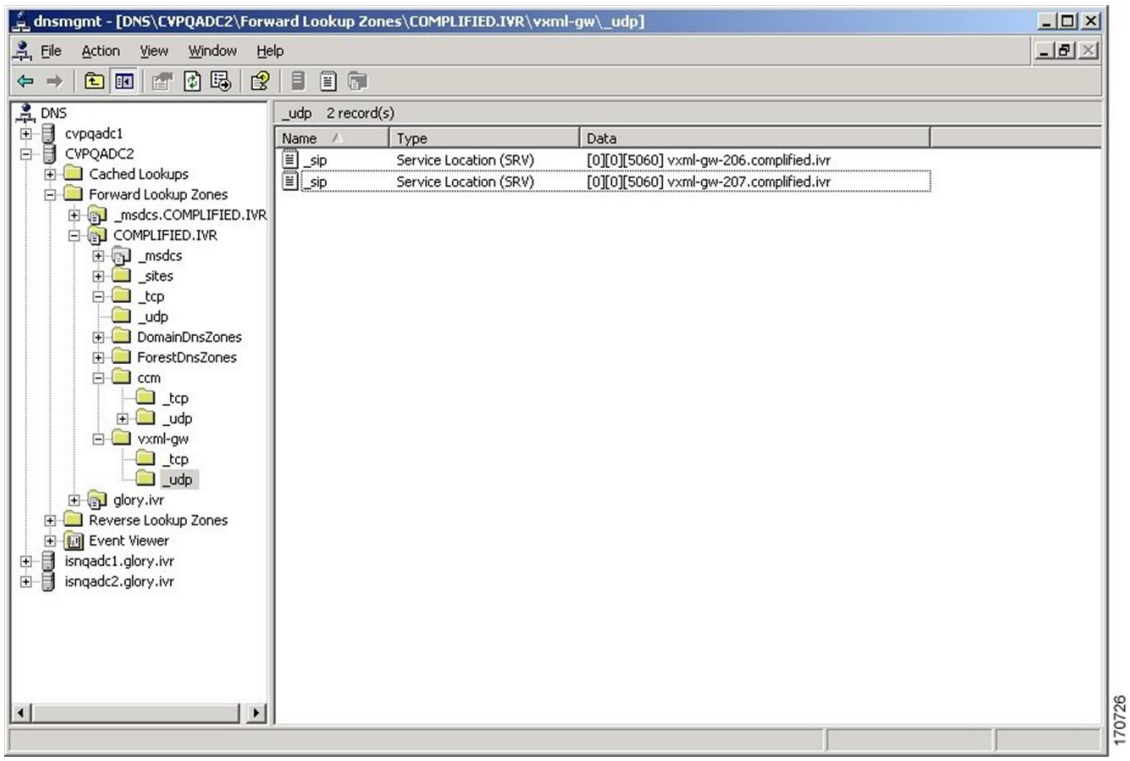
DNS Zone File Linux NAMED Configuration Example

The following is an example of a DNS zone file Linux NAMED configuration.

```
ringtone-1 IN A 10.86.129.20
ringtone-2 IN A 10.86.129.229
vxml-1 IN A 10.86.129.20
vxml-2 IN A 10.86.129.229
vxml-3 IN A 161.44.81.254
cvp-1 IN A 10.86.129.211
cvp-2 IN A 10.86.129.220
cvp-3 IN A 161.44.81.254
; Priority Weight Port Target
sip._tcp.ringtone.sox.cisco.com. SRV 1 1 5060 ringtone-1.sox.cisco.com.
_
SRV 1 1 5060 ringtone-2.sox.cisco.com.
sip._udp.ringtone.sox.cisco.com. SRV 1 1 5060 ringtone-1.sox.cisco.com.
_
SRV 1 1 5060 ringtone-2.sox.cisco.com.
_sip._tcp.vxml.sox.cisco.com. SRV 1 1 5060 vxml-1.sox.cisco.com.
SRV 1 1 5060 vxml-2.sox.cisco.com.
SRV 1 1 5060 vxml-3.sox.cisco.com.
_sip._udp.vxml.sox.cisco.com. SRV 2 1 5060 vxml-1.sox.cisco.com.
SRV 2 1 5060 vxml-2.sox.cisco.com.
SRV 1 1 5060 vxml-3.sox.cisco.com.
_sip._tcp.cvp.sox.cisco.com. SRV 1 1 5060 cvp-1.sox.cisco.com.
SRV 2 1 5060 cvp-2.sox.cisco.com.
SRV 3 1 5060 cvp-3.sox.cisco.com.
_sip._udp.cvp.sox.cisco.com. SRV 1 1 5060 cvp-1.sox.cisco.com.
SRV 2 1 5060 cvp-2.sox.cisco.com.
SRV 3 1 5060 cvp-3.sox.cisco.com.
```

DNS Zone File MS DNS Configuration Example

The following is an example of a DNS zone file MS DNS configuration.



Characteristics for the VRU Leg for Comprehensive Call Flow Model

Use the following commands to provide voice treatment:

Note new-call is a required name.

Continue with the VRU Leg Example.

```
service vru-leg flash:bootstrap.tcl
 !
service new-call flash:bootstrap.vxml
 !
service handoff flash:handoff.tcl
 !
service ringtone flash:ringtone.tcl
 !
service cvperror flash:cvperror.tcl
 !
service cvp-survivability flash:survivability.tcl
 !
```

Unified CVP VRU Call Flow Models with NIC Routing

This section describes the call flows and provides configuration instructions for the following Unified CVP call flow models:

- Type 8 Unified CVP VRU-Only Call Flow Model Unified ICME
- Type 8 Unified CVP VRU-Only Call Flow Model Unified ICMH

- Type 3 or 7 Unified CVP VRU-Only Call Flow Model Network VRU Unified ICMH

Note Be aware that in VRU-Only mode, Unified CVP by itself does not provide queuing capability; however, it can hold calls being queued when used with Unified ICME/Unified CCE with appropriate Unified ICME network interface controllers.

Type 8 Unified CVP VRU-Only Call Flow Model Unified ICME

In this call flow model, Unified CVP works with the Voice Gateway to act as the VRU; VRU voice treatment is provided by the Gateway and can include ASR/TTS.

When deployed with an NIC being used to queue and transfer calls (VRU Type 8), the NIC interfaces to the TDM switch or to the PSTN to transfer the call to an agent.

Note The Unified CVP SIP Service is part of this call flow model.

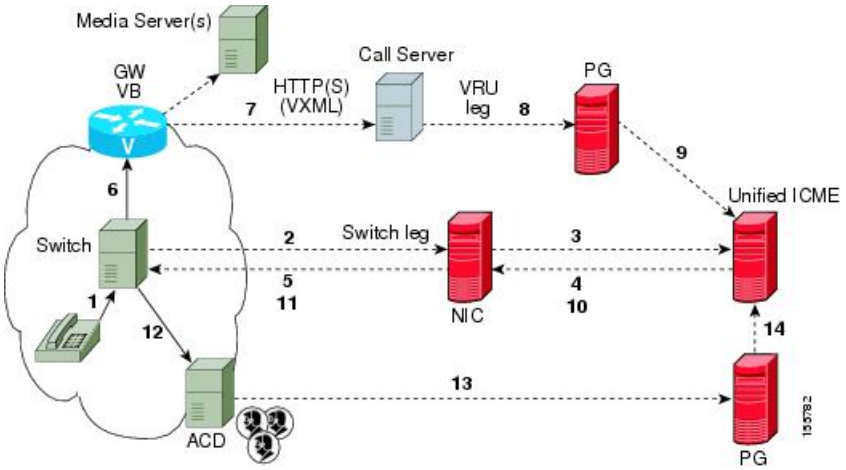
The Unified CVP VRU-Only call flow model requires the following components:

- Call Server with IVR and ICM Services enabled
- VoiceXML Gateway
- Operations Console
- Unified ICME

Note The VXML Server and the Reporting Server are optional.

The following figure shows the call flow for the Type 8 call flow model; in this figure, the NIC transfers the call.

Note In the following figure, solid lines indicate voice paths and dashed lines indicate signaling paths.



Note The numbers in the figure represent call flow progression.

Configuration overview for this call flow model:

- Confirm there is one Network VRU: a Type 8 when NIC is queuing and transferring calls.
- Define a Translation Route and labels for the VRU Peripheral (Network VRU labels do not need to be configured.)
- Use the ICM Script Editor's TranslationRouteToVRU node to connect the call to the Network VRU.

Type 8 Unified CVP VRU-Only Call Flow Model Unified ICMH

In this call flow model, the Unified CVP Call Server is deployed at the CICM level to act only as the VRU leg for the call. The VRU voice treatment is provided at the Voice Gateway, and might include ASR/TTS.

Note This call flow model is used when Unified CVP is connected to the CICM. The routing client in this call flow model is connected to the NAM.

When deployed with a NIC being used to queue and transfer calls (VRU Type 8), the NIC interfaces to the TDM switch to transfer the call to an agent.

Note The Unified CVP SIP Service is part of this call flow model.

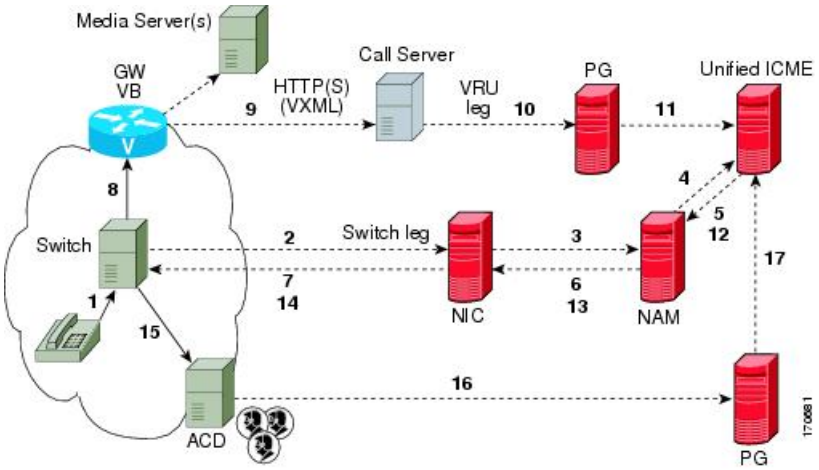
The Unified CVP VRU-Only call flow model requires the following components:

- Call Server with IVR and ICM Services enabled
- VoiceXML Gateway
- Operations Console
- Unified ICME

Note The VXML Server and the Reporting Server are optional.

The following figure shows the call flow for this call flow model.

Note In the following figure, solid lines indicate voice paths and dashed lines indicate signaling paths.



Note For simplicity, the figure does not illustrate a call flow model for redundancy and failover.

Configuration overview for this call flow model:

- There are two Network VRUs configured:
  - One on the NAM (Type 8).
  - One on the CICM for the INCRP connection (Type 8).
- Use the ICM Script Editor's TranslationRouteToVRU node to connect the call to the Network VRU.

Set Up Type 8 Unified CVP VRU-Only Unified ICME and Unified ICMH

The following task contains the configuration instructions for this call flow model for **both** ICME and ICMH.

Procedure

Step 1 Using the Unified CVP Operations Console (or the Unified CVP product CD), transfer the following script, configuration, and .wav files to the **VoiceXML Gateway** used for the VRU leg.
In the Operations Console, select **Bulk Administration** > **File Transfer** > **Scripts and Media** and select **Help** > **This Page** for details.

Transfer the following files:

- bootstrap.tcl
- handoff.tcl
- survivabilty.tcl
- bootstrap.vxml
- recovery.vxml
- ringtone.tcl
- cvperror.tcl
- ringback.wav
- critical_error.wav

Step 2 Configure the VXML gateway base settings.

Step 3 Configure the VXML gateway service settings.

Step 4 Configure the ICM VRU Label.

Step 5 Define a Network VRU on Unified ICME or (for Unified ICMH) on the NAM and each CICM.
Using the ICM Configuration Manager, the **Network VRU Explorer** tool, specify the following:

- Type: **8**
- Name: **cvpVRU**

Note Although any name will work, **cvpVRU** is used by convention, and is the example name referenced elsewhere in this document.

Step 6 Configure the Peripheral Gates (PGs) on Unified ICME or (for Unified ICMH) on each CICM.

a) Configure each PG.

b) Configure a peripheral for each Unified CVP ICM Service connected to each PG.

Use the ICM Configuration Manager, the **PG Explorer** tool. For each Unified CVP ICM Service connected to this PG, in the tree view pane, select the applicable PG and configure the following items:

Logical Controller tab:

- Client Type: **VRU**
- Name: A name descriptive of this PG

  Example: `<location>_A` for side A of a particular location

Peripheral tab:

- Peripheral Name: A name descriptive of this Unified CVP peripheral

  Examples: `<location>_<cvp1>` or `<dns_name>`
- Client Type: **VRU**
- Select the checkbox: **Enable Post-routing**

Advanced tab:

- From the Network VRU field drop-down list, select the name: **cvpVRU**

Routing Client tab:

- Name: By convention, use the same name as the peripheral.
- Client Type: **VRU**

Step 7 Configure a Service and Route for each VRU on Unified ICME or (for Unified ICMH) on each CICM.

Note You can also use service arrays. Refer to the Unified ICME documentation set for more information.

Using the ICM Configuration Manager, the **Service Explorer** tool, specify the following:

- Service Name: **cvpVRU**
- Route Name: **PeripheralName_cvpVRU**
- Peripheral Number: **2**

  Must match the "Pre-routed Call Service ID" in the Call Server configuration on the ICM tab in the Operations Console
- Select the checkbox: **Enable Post-routing**

Step 8 Define trunk groups.

Note You must configure one Network Transfer Group and one associated Trunk Group for each VRU leg Unified CVP ICM Service.

Define and configure the network trunk group on Unified ICME or (for Unified ICMH) on each CICM.

Using the ICM Configuration Manager, the Network **Trunk Group Explorer** tool:

a) Identify the network trunk group.

- Network Trunk Group Name: A name descriptive of this trunk group

b) For each Unified CVP ICM Service for the VRU leg, configure an associated trunk group.

- Peripheral Name: A name descriptive of this trunk group
- Peripheral Number: **200**

  Must match the "Pre-routed Call Trunk Group ID" in the Call Server configuration on the ICM tab in the Operations Console
- Trunk Count: Select **Use Trunk Data** from the drop-down list
- *Do not* configure any trunks

Step 9 Define translation route(s).

Define and configure a Translation Route for each VRU Peripheral on Unified ICME or (for Unified ICMH) on each CICM.

On Unified ICME, ICM Configuration Manager, **Translation Route Explorer** tool:

a) Define a Translation Route for each VRU Peripheral. Specify the following:
Translation Route tab:

- Set the **Name** field to the name of the target VRU peripheral. (This is by convention; this value must be unique in the enterprise)
- Set the **Type** field to **DNIS** and select the Service defined in the previous step

b) Configure translation route and label information for each VRU peripheral. Complete the following:
Route tab:

- Set the **Name**: by convention, this is the name of the target VRU peripheral, followed by the DNIS that this route will use, for example, MyVRU_2000

  This value must be unique in the enterprise

- Service Name drop-down list, select: **PeripheralName.cvpVRU**

Peripheral Target tab:

- Enter the first DNIS that will be seen by the VRU that you will be using for this translation route.

  Note The DNIS pool used for each VRU peripheral must be unique

- From the drop-down list, select a **Network Trunk Group** which belongs to the target VRU

Label tab:

- Enter the translation route label (which might or might not be the same DNIS you entered on the Peripheral Target tab)
- Type: **Normal**
- Routing Client: Select the NIC Routing Client

You must create an additional label for each NIC routing client.

Note Repeat the Route and corresponding Peripheral Target and Label information for each DNIS in the pool.

Step 10 Create VRU and routing scripts.
Create VRU scripts and routing scripts for IVR treatment and agent transfer on Unified ICME or (for Unified ICMH) on each CICM .

Using the ICM **Script Editor** tool, create the VRU scripts and routing scripts to be used for IVR treatment and agent transfer, as described in other sections of this manual and in the ICM manuals.

The VRU scripts are associated with the applicable Network VRU.

For example, **cvpVRU**

Use the ICM Script Editor's TranslationRouteToVRU node to connect the call to the Network VRU.

Step 11 Configure the ECC variables on Unified ICME or (for Unified ICMH) on the NAM and each CICM.
Using the ICM Configuration Manager, create the ECC variables.

For more information, refer to "Define Unified CVP ECC Variables."

Step 12 Configure dialed numbers and call types on Unified ICME or (for Unified ICMH) on the NAM and each CICM.
On Unified ICME, using the ICM Configuration Manager, configure dialed numbers and call types.

For more information, refer to ICM Configuration Guide for Cisco ICM Enterprise Edition.

Step 13 On Unified CM configure Unified CM.
For more information, refer to the Unified CM user documentation.

Step 14 Install and configure the Call Server(s).

Using the Operations Console, select **Device Management** > **CVP Call Server** and install and configure the **Call Server(s)**.

Select the check boxes: **ICM** and **IVR**

For detailed information, refer to the Operations Console online help.

Step 15 Configure the ICM service.
Using the Operations Console, select **Device Management** > **CVP Call Server** > **ICM tab**. On **each** Unified CVP Call Server, configure the **ICM Service** by specifying the following required information:

a) VRU connection port number.
Set the VRU Connection Port to match the VRU connection Port defined in ICM Setup for the corresponding VRU peripheral gateway (PIM).

b) Maximum Length of DNIS.
Set the maximum length DNIS to a number which is at least the length of the translation route DNIS numbers.

Example: if the Gateway dial pattern is 1800******, the maximum DNIS length is 10.

c) Call service IDs: New Call and Pre-routed.
Enter the new and pre-routed call service IDs. Configure the ports for both groups according to the licenses purchased, call profiles, and capacity by completing the required fields on this tab.

d) Trunk group IDs: New Call and Pre-routed.

- Enter the new and pre-routed call trunk group IDs
- Configure the group number for the Pre-routed Call Trunk group. The group number must match the trunk group number in the Network Trunk group used for the translation route
- Configure the number of ports according to the licenses purchased and capacity
- Configure each of the numbers used for translation routes. (The "New Call" group is not used since the calls are being sent to the VRU (Unified CVP) after some initial processing by the NIC/Unified ICME)

e) Dialed numbers used in the translation route.
Add the dialed numbers in the DNIS field.

f) Check the default values of the other settings and change, if desired.

Step 16 Configure the **IVR Service**.
In the Operations Console, select **Device Management** > **CVP Call Server** > **IVR** tab.

Check the default values and change, if desired.

Refer to the Operations Console online help for information about other settings you might want to adjust from their default values.

Step 17 (Optional) Configure the Reporting Server.
In the Operations Console, select **Device Management** > **CVP Reporting Server** > **General tab**:

1. Configure the Reporting Server.
2. Select a Call Server to associate with this Reporting Server.
3. Check the default values of the Reporting properties and change, if desired.

For more information, refer to Reporting Guide for Cisco Unified Customer Voice Portal

VoiceXML Gateway Configuration Examples

Example Gateway Settings for Type 8 Call Flow Model

The first part of the following example provides the basic configuration for setting a VoiceXML gateway:

- Applies a timestamp to debugging and log messages
- Turns on logging
- Turns off printing to the command line interface console
- Sends RTP packets
- Configures ASR/TTS Server
- Configures gateway settings

The last part of this example provides the following:

- Initiates the VoiceXML leg
- Plays a .wav file that enables caller to hear message from critical_error.wav
- Logs errors on the gateway when the call fails

```
service timestamps debug datetime msec
service timestamps log datetime msec
service internal
logging buffered 99999999 debugging
no logging console
ip cef
no ip domain lookup
ip host tts-en-us <IP of TTS or MRCP Server>
ip host asr-en-us <IP of ASR or MRCP Server>
voice rtp send-recv
!
voice service voip
allow-connections h323 to h323
signaling forward unconditional
h323
sip
min-se 360
header-passing
voice class codec 1
codec preference 1 g711ulaw
codec preference 2 g729r8
!
ivr prompt memory 15000
ivr prompt streamed none
ivr asr-server rtsp://asr-en-us/recognizer
ivr tts-server rtsp://tts-en-us/synthesizer
mrcp client timeout connect 10
mrcp client timeout message 10
mrcp client rtpsetup enable
rtsp client timeout connect 10
rtsp client timeout message 10
vxml tree memory 500
http client cache memory file 500
http client connection timeout 60
http client response timeout 30
http client connection idle timeout 10
gateway
timer receive-rtcp 6
!
ip rtcp report interval 3000
application
service new-call flash:bootstrap.vxml
service cvperror flash:cvperror.tcl
service handoff flash:handoff.tcl
```

Example of Dial-peer for ICM VRU Label for Type 8 Call Flow Model

The following example provides the configuration for an ICM VRU label dial-peer for the Type8 Unified CVP VRU-Only call flow model:

```
dial-peer voice 777 voip
 description ICM VRU label
 service bootstrap
 voice-class codec 1
 incoming called-number <your sendtovru label pattern here>
 dtmf-relay rtp-nte
 no vad
 !
```

Type 3 or 7 Unified CVP VRU-Only Call Flow Model Network VRU Unified ICMH

In this call flow model, Unified CVP is deployed as a Network VRU at the NAM. The Unified CVP IVR Service in the Operations Console works with the Voice Gateway to act as the VRU; VRU voice treatment is provided at the Voice Gateway and can include ASR/TTS. (This call flow model is used when Unified CVP is connected to the NAM.)

The NIC interfaces to the TDM switch to transfer calls to Unified CVP for VRU treatment and to queue and transfer calls using a VRU Type 3 or 7 call flow.

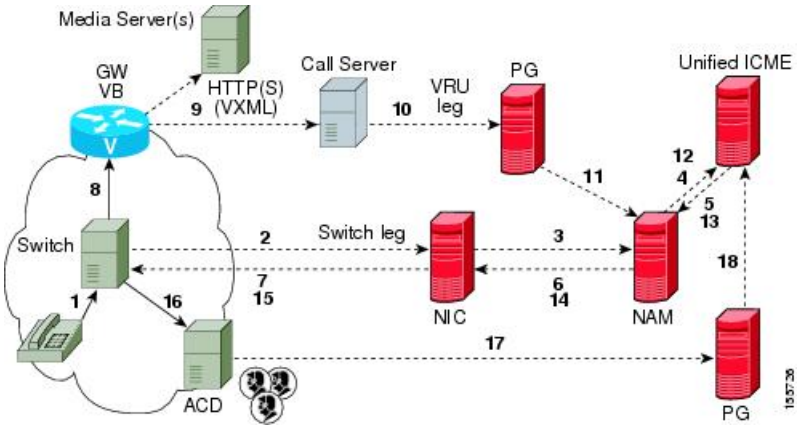
Note Use this call flow model only if the PSTN to which the NIC is connected can transport a Correlation ID when it transfers a call. If this is not the setup you are using, then use the Type 8 Unified CVP VRU-Only Call Flow Model Unified ICMH instead.

Note The Unified CVP SIP Service is part of this call flow model.

The following figure shows the call flow for this call flow model.

Note

- In the following figure, solid lines indicate voice paths and dashed lines indicate signaling paths.
- For simplicity, the figure does not illustrate a call flow model for redundancy and failover.



Note The numbers in the figure indicate call flow progression.

Configuration overview for this call flow model:

- Set the Network VRU Type to 3 or 7. There is no difference between these two types except that Type 7 causes Unified ICME to explicitly inform Unified CVP when it is about to transfer the call away from Unified CVP. (Most customers use Type 7.)
- The Network VRU names (where applicable), correlation IDs, and the ECC variable configurations must be identical on the NAM and CICM. All Labels must also be duplicated, although their routing clients will be different.
- Use the CICM Script Editor's SendToVRU node to connect the call to the Network VRU.

Set Up Type 3 or 7 VRU-Only Call Flow Model Network VRU Unified ICMH

The following table contains the high-level configuration instructions for this call flow.

Procedure

Step 1 Transfer the following script, configuration, and .wav files to the **VoiceXML Gateway** used for the VRU leg, using the Unified CVP Operations Console:
In the Operations Console, select **Bulk Administration** > **File Transfer** > **Scripts and Media** and select **Help** > **This Page** for details.

- bootstrap.tcl
- handoff.tcl
- survivabilty.tcl
- bootstrap.vxml
- recovery.vxml
- ringtone.tcl
- cvperror.tcl
- ringback.wav
- critical_error.wav

Note Optionally, you can transfer the files from the Unified CVP product CD.

Step 2 Configure the VoiceXML gateway base settings.

Step 3 Configure the VoiceXML gateway service settings.

Step 4 Configure the ICM VRU Label.

Step 5 Configure each PG.
On the **NAM**, ICM Configuration Manager, **PG Explorer** tool:

a) Configure each PG to be used for the **VRU Client** leg.

b) Configure a peripheral for each Unified CVP ICM Service to be used as a VRU leg connected to each PG. For each Unified CVP ICM Service connected to this PG, in the tree view pane, select the applicable PG.

Logical Controller tab, configure:

- Client Type: **VRU**
- Name: A name descriptive of this PG

  For example: `<location>_A` for side A of a particular location

Peripheral tab, configure:

- Peripheral Name: A name descriptive of this VRU peripheral.

  For example: `<location>_<cvp1>` or `<dns_name>`
- Client Type: **VRU**

- Select the checkbox: **Enable Post-routing**

Routing Client tab:

- Name: By convention, use the same name as the peripheral.
- Client Type: **VRU**

Step 6 Define a Network VRU and labels.
On the **CICM**, ICM Configuration Manager, **Network VRU Explorer** tool, define a Network VRU for the VRU leg and labels for reaching the NAM.

Specify the following:

- Type: **3** or **7**
- Name: **cvpVRU**

  Note This name is used by convention. Although any name will do, since it is referenced elsewhere in this document, **cvpVRU** is assumed.
- Define a **Label** for the NAM.
  - Label: Network routing number
  - Type: **Normal**
  - Routing client: Select the INCRP Routing Client from the drop-down list.

Step 7 Define a Network VRU and a label for each NIC.
On the **NAM**, ICM Configuration Manager, **Network VRU Explorer** tool, define a Network VRU and a label for each NIC that is using this VRU.

Specify the following:

- Type: **3** or **7**
- Name: **cvpVRU**

  Note This name is used by convention. Although any name will work, since it is referenced elsewhere in this document, **cvpVRU** is assumed.
- Define a **Label** for each NIC that is using this VRU:
  - Label: Network routing number
  - Type: **Normal**
  - Routing client: Select the Routing Client for that NIC from the drop-down list.

Note Make sure the Network VRU label is identical in the NAM and CICM. The Network VRU Name must be identical as well to avoid confusion.

Step 8 If there will be Routing Scripts on the NAM, define a default Network VRU.
On the **NAM**, ICM Configuration Manager, **System Information** tool, in the General section:

- Define the Default Network VRU: **cvpVRU**

Step 9 Define a default VRU.
On the **CICM**, ICM Configuration Manager, **System Information** tool, in the General section:

- Define a default Network VRU: **cvpVRU**

Step 10 Create the VRU and routing scripts.
On the **CICM**, ICM **Script Editor** tool:

Create the VRU scripts and routing scripts to be used for IVR treatment and agent transfer, as described in other sections of this manual and in the Unified ICME manuals. The VRU scripts are associated with the applicable Network VRU, that is, **cvpVRU**.

Use the ICM Script Editor's SendToVRU node to connect the call to the Network VRU.

Note A RunVRU Script or Queue node is an "implicit" SendToVRU node, although error handling will be easier if the explicit "SendToVRU" node is used.

Step 11 Configure the ECC variables.
On the **NAM** and **CICM**, ICM Configuration Manager, configure the ECC variables.

For more information, refer to "Define Unified CVP ECC Variables."

Step 12 Configure dialed numbers and call types.
On the **NAM** and **CICM**, ICM Configuration Manager, configure dialed numbers and call types.

For more information, refer to ICM Configuration Guide for Cisco ICM Enterprise Edition

Step 13 Define customers.
On the **NAM** and **CICM**, ICM Configuration Manager:

1. If necessary, differentiate VRUs (Unified CVPs) based on dialed number.
2. Define customers and their Network VRU.

For more information, refer to "Common Configuration for Differentiating VRUs Based on Dialed Number."

Step 14 On Cisco Unified CM, configure Unified CM.
For more information, refer to the Unified CM user documentation.

Step 15 Install and configure the Call Server(s).
In the Operations Console, select **Device Management** > **CVP Call Server**.

Click **Help** > **This Page** for details.

1. Install and configure the **Call Server(s)** .
2. To enable the ICM and IVR Services on the Call Server, select the check boxes: **ICM** and **IVR**

Click **Help** > **This Page** for details.

Step 16 Configure the ICM Service for each Call Server.
In the Operations Console, select **Device Management** > **CVP Call Server** > **ICM tab**. For each Unified CVP Call Server, configure the **ICM Service** by specifying the following required information:

1. VRU connection port number.

   Set the VRU Connection Port to match the VRU connection Port defined in ICM Setup for the corresponding VRU peripheral gateway (PIM).

2. Set the maximum length DNIS to the length of the Network Routing Number.

   Example: if the Gateway dial pattern is 1800******, the maximum DNIS length is 10.

3. Call service IDs: New Call and Pre-routed.

Enter the new and pre-routed call service IDs. Configure the ports for both groups according to the licenses purchased, call profiles, and capacity by completing the required fields on this tab

4 Trunk group IDs: New Call and Pre-routed.

Enter the new and pre-routed call trunk group IDs. Configure the group number for the Pre-routed Call Trunk group. The group number must match the trunk group number in the Network Trunk group used for the translation route.

Configure the number of ports according to the licenses purchased and capacity. Configure each of the numbers used for translation routes. (The "New Call" group is not used since the calls are being sent to the VRU (Unified CVP) after some initial processing by the NIC/Unified ICME.)

5 Check the default values of other settings and change, if desired.

Step 17 Configure the IVR service.
In the Operations Console, select **Device Management** > **CVP Call Server** > **IVR tab** and configure the **IVR Service**.

Check the default values and change, if desired.

Refer to the Operations Console online help for information about other settings you might want to adjust from their default values.

Step 18 (Optionally) Configure the Reporting Server.
In the Operations Console, select **Device Management** > **CVP Reporting Server** > **General** tab and configure the Reporting Server.

1 Configure the Reporting Server.

2 Select a Call Server to associate with this Reporting Server.

3 Check the default values of the Reporting properties and change, if desired.

For more information, refer to Reporting Guide for Cisco Unified Customer Voice Portal .

VoiceXML Gateway Configuration: Example Gateway Settings for Type 3 or Type 7

The first part of the following example provides the basic configuration for setting a VoiceXML gateway:

- Applies a timestamp to debugging and log messages
- Turns on logging
- Turns off printing to the command line interface console
- Sends RTP packets
- Configures ASR/TTS Server
- Configures gateway settings

The last part of this example provides the following:

- Initiates the VoiceXML leg

- Plays a .wav file that enables caller to hear message from critical_error.wav
- Logs errors on the gateway when the call fails

```
service timestamps debug datetime msec
service timestamps log datetime msec
service internal
logging buffered 99999999 debugging
no logging console
ip cef
no ip domain lookup
ip host tts-en-us <IP of TTS or MRCP Server>
ip host asr-en-us <IP of ASR or MRCP Server>
voice rtp send-recv
!
voice service voip
 allow-connections h323 to h323
 signaling forward unconditional
 h323
 sip
 min-se 360
 header-passing
voice class codec 1
 codec preference 1 g711ulaw
 codec preference 2 g729r8
!
ivr prompt memory 15000
ivr prompt streamed none
ivr asr-server rtsp://asr-en-us/recognizer
ivr tts-server rtsp://tts-en-us/synthesizer
mrcp client timeout connect 10
mrcp client timeout message 10
mrcp client rtpsetup enable
rtsp client timeout connect 10
rtsp client timeout message 10
vxml tree memory 500
http client cache memory pool 15000
http client cache memory file 500
http client connection timeout 60
http client response timeout 30
http client connection idle timeout 10
gateway
 timer receive-rtcp 6
!
ip rtcp report interval 3000
application
 service new-call flash:bootstrap.vxml
 service cvperror flash:cvperror.tcl
 service handoff flash:handoff.tcl
 service bootstrap flash:bootstrap.tcl
 !
```

VoiceXML Gateway Configuration: Example of Dial-Peer for ICM VRU Label for Type 3 or Type 7

The following example provides the configuration for an ICM VRU label dial-peer for the Type3 or 7 Unified CVP VRU-Only call flow model:

```
dial-peer voice 777 voip
 description ICM VRU label
 service bootstrap
 voice-class codec 1
 incoming called-number <your sendtovru label pattern here>
 dtmf-relay rtp-nte
 no vad
 !
```

Unified CVP Comprehensive Call Flows for Pre-Routed Calls

This class of call flows is similar to the Unified CVP Comprehensive call flow models, except that calls are first introduced into Unified ICME or Unified ICMH using some path other than through Unified CVP. A Unified ICME routing script is given the chance to "pre-route" such calls before Unified CVP ever sees them. Once the script transfers the call to Unified CVP, for either self-service or queuing, a more standard Unified CVP Comprehensive call flow model is used.

In this section we will discuss the following call flows:

- Calls Which Arrive at Unified ICME Through a Pre-Route-Only NIC
- Calls Which Are Originated by Unified CM
- Calls Which Are Originated by an ACD or Call Routing Interface. A VXML Server running as a Standalone with ICME Lookup call flow model also falls into this category, if the goal of the ICM Lookup is to transfer the call into a Unified CVP Comprehensive call flow model deployment.

All the above call flows are similar in that the original routing client (a NIC, Unified CM, an ACD, or a VRU) is capable of a single route request only. By definition, the routing client makes a single request to Unified ICME, the Unified ICME returns a destination label, and the routing client affects the transfer. At that point the route request dialog is ended, and Unified ICME has no ability to send a subsequent label or conduct any other form of third-party call control.

The routing script might continue, however. This would be the case if the returned label was a translation route to VRU label, or if it was a correlation ID label resulting from a SendToVRU node. In those cases the call is transferred to Unified CVP, and the routing script continues executing once Unified CVP successfully receives the call. The script then invokes micro-application requests as part of its queuing or self service treatment. If the call is then transferred to an agent or skill group, that label must go to Unified CVP rather than to the original routing client. If the call will later be blind-transferred to another agent or skill group, or back into Unified CVP for additional queuing or self service, that label too must go to Unified CVP rather than to the original routing client.

When the call is delivered to Unified CVP, in order for micro-applications to be supported, it must establish both the Switch and the VRU leg. In short, it must enter a normal Unified CVP Comprehensive call flow model. The only difference between the pre-routed case and a normal Unified CVP Comprehensive call flow model case is in how the call first arrives at Unified CVP. In the pre-routed case, it arrives using either a translation route or correlation-id transfer, whereas in the more typical case it arrives as a new call from the PSTN. In either case, a subsequent transfer to Unified CVP's VRU leg is required.

The following sections describe the important configuration points for each of the above call flows.

Calls Which Arrive at Unified ICME Through a Pre-Route-Only NIC

The following Unified ICME NICs fall into this category: ATT, GKTMP, MCI, Sprint, Stentor. The GKTMP NIC is a special case of this category and will be discussed in the next section. This call flow applies to both Unified ICME and Unified ICMH implementations. In the latter case, both Unified CVP and the NIC would be deployed at the NAM.

For Unified ICME Release 7.1 and later, do the following:

1. Configure a single Type 10 Network VRU and associate it with all Unified CVP peripherals in the PG Explorer configuration tool, and also define it as the default system Network VRU in the System Information tool.

2. In order to support the initial call transfer to Unified CVP from the pre-route routing client, configure Translation Route labels to target the Unified CVP peripherals.
3. In order to support the transfer to VRU leg, configure the Type 10 Network VRU that you defined above in Step 1 with Network Routing Number labels for each Unified CVP peripheral routing client.
4. Associate all micro-application VRU scripts with that same Type 10 Network VRU. When the routing script transfers the call into Unified CVP, it must use a TranslationRouteToVRU node. No subsequent node is necessary in order to perform the transfer to Unified CVP's VRU leg; this will take place automatically.

Note Non-prerouted calls can also share the same Network VRU and the same Unified CVP Call Servers.

For Unified ICME Release 7.0 and later, do the following:

1. Configure two Network VRUs: one Type 2 and one Type 7.
2. In the PG Explorer tool, assign all Unified CVP Call Servers to the Type 2 Network VRU.
3. Configure one set of Translation Route labels to target the Type 2 Call Servers; these will be used to transfer the call from the original routing client to the Unified CVP Switch leg.
4. Assign a label to the Type 7 Network VRU for each Unified CVP Call Server routing client, whose label string is set to the Network Routing Number.
5. Configure the Type 7 Network VRU as the system default Network VRU in the System Information configuration tool.
6. Associate all micro-application VRU scripts with the Type 7 Network VRU.
7. When the routing script transfers the call into Unified CVP, it must use *two* nodes in succession: first, a TranslationRouteToVRU, and then an explicit SendToVRU node. The first node transfers the call from the initial routing client to one of the Type 2 Call Servers (Unified CVP Switch leg); the second one transfers the call from the Type 2 Call Server to the Unified CVP VRU leg. (The VRU leg will usually end up running through the same Unified CVP Call Server as the Switch leg.)

Note Non-prerouted calls can also share the same Type 2 Call Servers and Type 2 and Type 7 Network VRUs; however, scripts which handle non-prerouted calls must also use an explicit SendToVRU node before they can execute any micro-applications.

Calls Which Are Originated by Unified CM

This category includes all of the following types of calls:

- "Internal Help Desk" calls, in which a Unified CM phone user calls a CTI Route Point, which starts a routing script that can optionally deliver the call to Unified CVP for queuing or self-service.
- Unified ICME Outbound Option calls, in which a dialer makes outbound calls and then transfers them to a CTI Route Point; this starts a routing script that can optionally deliver the call to Unified CVP for queuing or self-service.

- Consultative Warm Transfer, in which a Unified CM agent places the caller on hold and dials in to Unified ICME to reach a second agent; this starts a routing script that can optionally deliver the call to Unified CVP for queuing or self-service.

Note Refer to "Cisco Unified ICME Warm Consult Transfer/Conference to Unified CVP" for additional information about Consultative Warm Transfer.

Note If these call flows are used in a Cisco Unified Contact Center Management Portal environment, the target Unified CVP Call Servers are required to be connected to the same CICM as the Unified CM from which the call originates. Just as multiple CICMs will require multiple Unified CMs, so will they require multiple Unified CVP Call Servers.

Further configuration points differ depending on whether Unified CVP is being deployed with Unified ICME Release 7.0 or 7.1 and later.

For Unified ICME Release 7.1 and later, do the following:

1. Configure a single Type 10 Network VRU and defined as the default system Network VRU in the System Information tool.
2. Configure the Type 10 Network VRU with two sets of labels. Associate the first set with the Unified CM routing client, which contains a label that Unified CM uses to transfer the call to Unified CVP. Configure Unified CM with a series of route patterns, which include that label followed by one to five arbitrary digits. For example, if the selected label is 1111, then the following route pattern is needed: 1111!. The second set of Network VRU labels must contain the usual Comprehensive Model "Network Routing Number," which must be associated with each Unified CVP Call Server routing client.
3. Associate all micro-application VRU scripts with that same Type 10 Network VRU.
4. When the routing script transfers the call into Unified CVP, it should use a single SendToVRU node. No subsequent node is necessary in order to perform the transfer to Unified CVP's VRU leg; this will take place automatically. (The SendToVRU node can be omitted since any micro-application script node will invoke the same functionality automatically; however, you can include this node explicitly in the script for troubleshooting purposes.)

Note Non-prerouted calls can also share the same Network VRU and the same Unified CVP Call Servers as those calls which are transferred from Unified CM; however, note that the scripts which handle non-prerouted calls must also use an explicit SendToVRU node before they can execute any micro-applications.

For Unified ICME Release 7.0 and later, do the following:

1. Configure two Network VRUs: one Type 2 and one Type 7.
2. In the PG Explorer tool, assign the Unified CVP Call Servers to the Type 2 Network VRU.
3. Configure one set of Translation Route labels to target the Type 2 Call Servers; these will be used to transfer the call from the original routing client to the Unified CVP Switch leg.
4. Assign a label to the Type 7 Network VRU for each Unified CVP Call Server routing client, whose label string is set to the Network Routing Number.

5. Configure the Type 7 Network VRU as the system default Network VRU in the System Information configuration tool.
6. Associate all micro-application VRU scripts with the Type 7 Network VRU.
7. When the routing script to transfers the call into Unified CVP, it should use *two* nodes in succession: first, a TranslationRouteToVRU, and then an explicit SendToVRU node (which contrary to the Unified ICME 7.1 case, is *not* optional). The first node transfers the call from the initial routing client to one of the Type 2 Call Servers (Unified CVP Switch leg); the second one transfers the call from the Type 2 Call Server to the Unified CVP VRU leg. (The VRU leg will usually end up running through the same Unified CVP Call Server as the Switch leg.)

Note Non-prerouted calls can also share the same Type 2 Call Servers and Type 2 and Type 7 Network VRUs.

Calls Which Are Originated by an ACD or Call Routing Interface

These calls are very similar to those which arrive from a preroute-only NIC, except that the routing client is connected to Unified ICME using a PG rather than using a NIC. Therefore, if this call flow is used in a Unified ICMH environment, the target Unified CVP Call Servers are required to be connected to the same CICM as the ACD or CRI-based VRU from which the call originates. Just as multiple CICMs will require multiple ACD or VRU peripherals, so will they require multiple Unified CVP Call Servers.

Further configuration points differ depending on whether Unified CVP is being deployed with Unified ICME Release 7.0 or 7.1 and later

For Unified ICME Release 7.1 and later, do the following:

1. Configure a single Type 10 Network VRU and associate it with all Unified CVP peripherals in the PG Explorer configuration tool, and also define it as the default system Network VRU in the System Information tool.
2. In order to support the initial call transfer to Unified CVP from the pre-route routing client, configure Translation Route labels to target the Unified CVP peripherals.
3. In order to support the transfer to VRU leg, configure the Type 10 Network VRU with Network Routing Number labels for each Unified CVP peripheral routing client.
4. Associate all micro-application VRU scripts with that same Type 10 Network VRU.
5. When the routing script transfers the call into Unified CVP, it must use a TranslationRouteToVRU node. No subsequent node is necessary in order to perform the transfer to Unified CVP's VRU leg; this will take place automatically.

Note Non-prerouted calls can also share the same Network VRU and the same Unified CVP Call Servers.

For Unified ICME Release 7.0 and later, do the following:

1. Configure two Network VRUs: one Type 2 and one Type 7.
2. In the PG Explorer tool, assign all Unified CVP Call Servers to the Type 2 Network VRU.
3. Configure one set of Translation Route labels to target the Type 2 Call Servers; these will be used to transfer the call from the original routing client to the Unified CVP Switch leg.

4. Assign a label to the Type 7 Network VRU for each Unified CVP Call Server routing client, whose label string is set to the Network Routing Number.
5. Configure the Type 7 Network VRU as the system default Network VRU in the System Information configuration tool.
6. Associate all micro-application VRU scripts with the Type 2 Network VRU.
7. When the routing script transfers the call into Unified CVP, it should use *two* nodes in succession: first, a TranslationRouteToVRU, and then an explicit SendToVRU node. The first node transfers the call from the initial routing client to one of the Type 2 Call Servers (Unified CVP Switch leg); the second one transfers the call from the Type 2 Call Server to the Unified CVP VRU leg. (The VRU leg will usually end up running through the same Unified CVP Call Server as the Switch leg.)

Note Non-prerouted calls can also share the same Type 2 Call Servers and Type 2 and Type 7 Network VRUs.

Common Unified ICMH Configuration for Unified CVP Switch Leg

The steps in the following table describe Unified ICMH configuration instructions common to Comprehensive Unified ICMH and VRU-Only with NIC routing, with Correlation ID call routing call flow models for Unified CVP switch legs.

Set Up Common Unified ICMH Unified CVP Switch Leg

Procedure

Step 1 On the **NAM**, ICM Configuration Manager, **Network VRU Explorer** tool:

a) Define a Network VRU for Unified CVP.

- Type: **10**
- Name: **cvpVRU**

Note This name is used by convention. Any name will do; since it is referenced elsewhere in this document, **cvpVRU** will be assumed.

b) Assign labels.
Define one **Label** per Unified CVP or NIC routing client.

- Type: **Normal**
- Label: Network Routing Number

Step 2 Set the client type.
On the **CICM**, using the ICM Configuration Manager, **NIC Explorer** tool:

- Select the **Routing Client** tab for the INCRP NIC

- Specify the Client Type: **VRU**

Step 3 Define a Network VRU.
On the **CICM**, using the ICM Configuration Manager, **Network VRU Explorer** tool, define a Network VRU with a label that uses INCRP as its routing client.

Specify the following:

- Type: **10**
- Name: **cvpVRU**

  Note This name is used by convention. Although any name will work; since it is referenced elsewhere in this document, **cvpVRU** is assumed
- Define one **Label** for the NAM routing client:
  - Label: Network Routing Number
  - Type: **Normal**
  - Routing client: **INCRP NIC**

Step 4 Define the Peripheral Gates (PGs).
On the **NAM**, using the ICM Configuration Manager, **PG Explorer** tool, configure a peripheral gate for each ICM Service to be used for a Switch leg connected to each PG.

For each Unified CVP ICM Service connected to this PG, in the tree view pane, select the applicable PG.

Logical Controller tab:

- Client Type: **VRU**
- Name: A name descriptive of this PG

  For example: `<location>_A`, for side A of a particular location.

Peripheral tab:

- Peripheral Name: A name descriptive of this Unified CVP peripheral, for example, `<location>_<cvp1>` or `<dns_name>`
- Client Type: **VRU**
- Select the checkbox: **Enable Post-routing**

  On the **Advanced** tab, select the name **cvpVRU** from the Network VRU field drop-down list.

Routing Client tab:

- Name: By convention, use the same name as the peripheral
- Client Type: **VRU**
- *Do not* select the **Network Transfer Preferred** checkbox.

Define Unified CVP ECC Variables

You need to set up ECC variables that Unified CVP uses to exchange information with Unified ICME/ICMH.

Note Information about how Unified CVP uses these ECC variables can be found throughout the manual.

Procedure

Step 1 Within the ICM Configuration Manager, select **Tools** > **Miscellaneous Tools** > **System Information** and select the **Enable expanded call context** checkbox.

Step 2 Within the ICM Configuration Manager, select **Tools** > **List Tools** > **Expanded Call Variable List**.

Step 3 In the Expanded Call Variable List window, enable the **Add** button by clicking **Retrieve**.

Step 4 Click **Add**. The Attributes property tab is enabled.

Step 5 Create each of the variables in the following table, clicking **Save** after defining each variable.

Note Any time you change the configuration of any ECC variable with the Expanded Call Variable List tool, you must stop and restart the Unified CVP Call Server.

Caution It is important that you enter the ECC's **Name** values listed in following table exactly as specified. If you do not, the Unified ICME/ICMH software will not be able to successfully communicate with the micro-applications on the ICM Service.

Length values are more flexible. Unless the values listed in following table are *specifically* noted as "required," the value in the Length column is the *maximum* that Unified ICMH can handle for that ECC; you can specify a value between 1 and the maximum length.

Note In a Unified ICME/ICMH configuration, the ECC variable configuration, including the length, defined in the NAM must be defined *exactly* the same in the CICM.

If you change the length of the ECC variables while the Unified CVP ICM Service is running, you need to restart the Unified CVP ICM Service for it to work properly.

Step 6 When finished, click **Save** to apply your changes.

Table 6: Micro-Application ECCs

Name	Length	Definition
user.CourtesyCallbackEnabled	Required for using Courtesy Callback. Length:1	Used to determine if Courtesy Callback should be offered to a caller. Valid values are: "1" = Yes "0" = No

Name	Length	Definition
user.cvp_server_info	Length: 15	Used by Unified CVP to send the IP address of the Call Server sending the request to Unified ICME. Example: An IPv4 address like 192.168.150.181
user.microapp.currency	Value: 6	Currency type.
user.microapp.error_code	Value: 2	Return status error code to be returned from the Unified CVP to Unified ICME/ICMH upon a False return code in the Run Script Result.
user.microapp.fetchaudio	Recommended length: 20; but length depends on the file name.	Filename for audio to be played by the VXML gateway while the gateway loads and processes the requested resource when there is significant network latency. Default: none Example: "flash:holdmusic.wav"
user.microapp.fetchdelay	Length: 1	The length of time (in seconds) to wait at the start of the fetch delay before playing the audio specified by *user.microapp.fetchaudio*. This setting only takes effect if the value of fetchaudio is not empty. Default: 2 seconds; used to avoid a "blip" sound heard in a normal network scenario Setting this value to zero will play hold music immediately, for a minimum of five seconds. Values: 1 through 9
user.microapp.fetchminimum	Length: 1	The minimum length of time to play audio specified by *user.microapp.fetchaudio*, even if the requested resource arrives in the meantime. This setting only takes effect if value of fetchaudio is not empty. Default: 5 seconds Values; 1 through 9

Name	Length	Definition
user.microapp.isPostCallSurvey	Length: 1	Used to determine if post call survey should be offered to a caller after the agent hangs up. Valid values: "y" or "Y" is "Yes" "n" or "N" is "No" Default value is "Yes"
user.microapp.locale	Value: 5	Locale, a combination of language and country which defines the grammar and prompt set to use.
user.microapp.media_server	**Required** for any IVR scripting. Maximum length: 210 characters Recommended length: 30	Root of the URL for all media files and external grammar files used in the script. HTTP and HTTPS schemes can be specified as: • HTTP scheme is specified as "http://<servername>" • HTTPS scheme is specified as "https://<servername>"
user.microapp.play_data	40	Default storage area for data for Play Data micro-application.
user.microapp.sys_media_lib	10	Directory for all system media files, such as individual digits, months, default error messages, and so forth.
user.microapp.app_media_lib	Maximum length: 210 characters Recommended length: 10	Directory for all application-specific media files and grammar files. You can also set this value to "..", (literally two periods in quotes) which bypasses the user.microapp.app_media_lib and user.microapp.locale ECC Variables when writing a URL path. For example, if you set the user.microapp.app_media_lib to “..”, the path: http://server/locale/../hello.wav would really be: http://server/hello.wav
Note The system and application media libraries need message and prompt files created/recorded for each locale that will be referenced. For more information, refer to "Configure the Media Servers."		

Name	Length	Definition
user.microapp.grammar_choices	Configurable on Unified ICME. Maximum length: 210 characters.	Specifies the ASR choices that a caller can input for the Get Speech micro-application. Each option in the list of choices is delimited by a forward slash (/). **Note** If text is placed in this variable that is longer than the variable is configured to handle, only the first 210 characters are sent.
user.microapp.inline_tts	Configurable on the ICM. Maximum length: 210 characters.	Specifies the text for inline Text To Speech (TTS). **Note** If text is placed in this variable that is longer than the variable is configured to handle, only the first 210 characters are sent.
user.microapp.input_type	Value: 1	Specifies the type of input that is allowed. Valid contents are: • **D** - DTMF • **B** - (Both, the default) DTMF and Voice If you are not using an ASR, you need to set this variable to D. If you are using an ASR, you can set the variable to either D or B. **Note** With input_mode set to "B" (both), either DTMF or speech will be accepted, but mixed mode input is not. Once you begin entering with one mode, input using the other mode is ignored and has no effect.
user.microapp.caller_input	Configurable on Unified ICME/ICMH. Maximum length: 210 characters.	Storage area for any ASR input that is collected from Get Speech. **Note** Get Speech text results will be written to this ECC variable. Results from Get Digits or Menu micro-applications will be written to the CED.

Name	Length	Definition
user.microapp.pd_tts	Value: 1	Specifies whether Unified CVP's Text To Speech (TTS) or media files must be played to the caller. Valid contents are: • **Y** - Yes, use TTS capabilities • **N** - No, do not use TTS capabilities; play media files instead. **Note** Used only with Play Data micro-application.
user.microapp.UseVXMLParams	Value: 1	This parameter specifies the manner in which you pass information to the external VoiceXML. Set this parameter to either "Y" (for yes) or "N" (for no). Y uses the values in the user.microapp.ToExtVXML variable array. N appends the name/value pairs in user.microapp.ToExtVXML to the URL of the external VoiceXML. Default: "N" Refer to "Writing Scripts for Unified CVP," for more information.
user.microapp.ToExtVXML	210	This variable array sends information to the external VoiceXML file. Must be configured as Array variables, not Scalar variables, even if the array length is set to 1. Refer to "Passing Information to the External VoiceXML" for more information.
user.microapp.FromExtVXML	210	This variable array returns information from the external VoiceXML file. Must be configured as Array variables, not Scalar variables, even if the array length is set to 1. Refer to "Passing Data Back to Unified ICME with External Voice XML" for more information.

Name	Length	Definition
user.microapp.override_cli	Configurable on Unified ICME/ICMH. Maximum length: 200 characters.	Used by system to override the CLI field on outgoing transfers.
user.microapp.metadata	The variable length would normally be configured as 62 bytes, but if ECC space is restricted, you can configure it as 21 bytes.	Following the Menu (M), Get Data (GD) and Get Speech (GS) micro-applications, Unified CVP returns information about the execution of that micro-application. The user.microapp.metadata ECC variable is structured as follows: m|con|tr|to|iv|duratn|vruscriptname
user.microapp.uui	Configurable on Unified ICME/ICMH. Maximum length: 131 characters.	Used to pass user-to-user information back to Unified CVP from Unified ICME/ICMH.
user.pod.id		Used to carry POD ID at the solution level. For more information about `user.pod.id` ECC variable, see the *User Guide for Cisco Unified CVP VXML Server and Cisco Unified Call Studio*.
user.sip.refertransfer	Optional Maximum length: 1 character.	SIP Service will use REFERs when transferring to the agents: • y - Use REFER when transferring • n - Do not use REFER when transferring
user.suppress.sendtovru	Optional Length: 1	Suppress the Temporary Connect message generated by SendToVRU node (explicitly or implicitly, for example by a Translation Route to VRU node). Used in call flows where the Temporary Connect is generated right before the Connect message (that is, no Run Script messages expected) to avoid the extra overhead of setting up a VRU leg temporarily before the Connect arrives. Valid values are : "y" or "Y" (yes, suppress the message)

Metadata ECC Variable

Metadata ECC Variable

Following the Menu (M), Get Data (GD) and Get Speech (GS) micro-applications, Unified CVP returns information about the execution of that micro-application. The information is returned in the ECC variable **user.microapp.metadata**. Its format is defined in terms of a number of subfields, each separated by a vertical bar character ('|'). Also, the subfields are of fixed length in order to facilitate extraction either at reporting time or within the ICM routing script itself.

The **user.microapp.metadata** ECC variable is structured as follows:

m|con|tr|to|iv|duratn|vruscriptname

The following table shows the values for this variable.

Table 7: Metadata ECC Variable Values

Metadata	ECC Variable Value
m	D, V or N - Indicates whether the user responded with Voice (V), DTMF (D), or not at all (N). (Note that for the Menu micro-application, any successful single digit entry will result in m being set to V or D, even if the entry was an invalid menu selection.)
con	000 to 100 - Indicates the ASR percent confidence level at which the voice input was finally recognized. This field is only valid if m is Voice (V).
tr	00 to 99 - Indicates how many tries were required. 01 means user responded successfully after the first prompt.
to	00 to 99 - Indicates how many timeouts occurred. Does not include interdigit timeouts.
iv	00 to 99 - Indicates how many invalid entries were received, including interdigit timeouts.
duratn	000000 to 999999 - Indicates the micro-application duration in milliseconds. Duration is defined as the elapsed time between entering and exiting the RunExternalScript node, as measured in the IVR Service.
vru script name	Full name of the VRU script which was executed. This is the only variable length field.

This ECC variable is optional. If used, you must define it in the Unified ICME Expanded Call Context Variables configuration tool. The variable length would normally be configured as 62 bytes, but if ECC space is restricted, you can configure it as 21 bytes. This will drop the vruscriptname subfield. If you do define this variable, its contents get written to the Unified ICME database with every termination record, and can be used to provide a record of meta-information about the execution of each input micro-application.

Common Configuration for Differentiating VRUs Based on Dialed Number

The Network VRU configuration instructions in this guide assume that all callers will be routed to the same VRUs (Unified CVPs) for VRU treatment purposes. Under this assumption, it is always simplest to rely on the system default Network VRU. However, it is sometimes necessary to differentiate the VRUs (Unified CVPs) based on dialed number.

Note This section is only applicable to call flow models which use the SendToVRU node to transfer the call to Unified CVP's VRU leg (it does not apply to Translation Route transfers).

For example, some calls:

- Need to assign different customers or applications to their own Unified CVP machines.

To configure Unified ICME to differentiate the VRUs, do the following:

- Configure more than one Network VRU.
- On Unified ICME, ICM Configuration Manager, **ICM Instance Explorer** tool:
  - Configure the customer(s).
  - Configure the Network VRU for each customer if that customer will use a Network VRU other than the default.
- Associate the dialed number(s) to the customer in the Dialed Number List tool.
- Since each configured VRU script is specific to one specified Network VRU, you need to create a distinct set of VRU scripts for each Network VRU. Also, be sure that the ICM routing script calls the correct set of VRU scripts.

Local SRV File Configuration Example for SIP Messaging Redundancy

Follow the steps below to configure the SIP service on the Call Server.

1. Log in to the Operations Console and click **Device Management** > **Unified CVP Call Server**.
2. Select the **Call Server**.
3. Click the **SIP** tab and select **Yes** to **Use DNS SRV type query** for outbound proxy lookup. Otherwise, select **No**.
4. Enable **Resolve SRV records locally**.
5. Click **Save and deploy** to deploy the changes to the Call Server page.
6. Restart the Call Server service.

Below is an example local SRV configuration file. It must be named in a text file named srv.xml and manually placed in the c:\cisco\cvp\conf directory of the Call Server where the SIP Service is running.

```
<?xml version="1.0" encoding="UTF-8" ?>
<locater>
<host name="cusp.proxy.cisco.com">
```

```
 <record weight="30" priority="1" destination="10.86.129.23" port="5060"/>
 <record weight="30" priority="2" destination="10.86.129.109" port="5060"/>
</host>
<host name="ringtone.services.cisco.com">
 <record weight="30" priority="1" destination="10.86.129.23" port="5060"/>
 <record weight="30" priority="1" destination="10.86.129.109" port="5060"/>
</host>
</locater>
```

CHAPTER 3

Writing Scripts for Unified CVP

This chapter discusses using Unified ICME configuration and script editing to access the Unified CVP solution.

It includes information about how to:

- Set up Unified ICME to interact with Unified CVP
- Write applications for Unified CVP

Note This chapter contains important information for IVR application developers. It also may be of interest to Call Center Managers, Unified CVP System Managers, and Unified ICME System Managers.

- Before You Begin, page 95
- Making Unified ICME Work with Unified CVP, page 96
- Scripting for Unified CVP with Unified ICME, page 98
- Unified ICME Setup, page 108
- Writing Unified ICME Applications for Unified CVP, page 108
- Unified CVP Micro-Applications, page 110
- Scripting for Unified CVP with Call Studio, page 162
- System Media Files, page 171

Before You Begin

This chapter makes the following assumptions:

- The information in this chapter assumes that you are already familiar with using Unified ICME software's ICM Configuration Manager and Script Editor tools for call center operations and management.

Note You should have a copy of the following Unified ICME documentation available in addition to this manual in order to successfully configure Unified ICME and use its features in conjunction with Unified CVP: ICM Configuration Guide for Cisco ICM Enterprise Edition and *ICM Scripting and Media Routing Guide for Cisco ICM/IPCC Enterprise & Hosted Editions*.

- When creating applications that interact with Unified CVP, only use alphanumeric characters for application, element, and field names; *do not* use special characters such as periods, asterisks or brackets. Following this practice will avoid potential issues with data transfer between different systems.

Making Unified ICME Work with Unified CVP

Unified ICME determines where to route calls - whether to ACDs, specific agents, or to VRUs. However, the routing services themselves must be provided by an external routing client.

Traditionally, Unified ICME's routing clients were PSTN network switches, or in some cases, customer provided ACDs. Unified CVP provides VoIP routing capability for the Unified ICME and Unified CCE products. Unified CVP makes it possible for Unified ICME to use VoIP gateways as routing clients, as well traditional routing services.

Unified ICME and Unified CVP work together to perform such tasks as:

- Playing media, such as a recording stating office hours, to a caller.
- Playing streaming audio, such as a radio broadcast, to a caller.
- Retrieving caller-entered data, DTMF, or speech.
- Playing back different types of data, such as an account number or balance, to a caller.
- Moving calls to other destinations. For example, forwarding calls to an agent.

Unified ICME uses IVR messaging technology to direct Unified CVP and to receive the responses from Unified CVP.

Scripts to Access Unified CVP From Unified ICME

Both Unified ICME and Unified CVP use scripts to invoke their features. In fact, Unified ICME references Unified CVP scripts from *within* its own scripts . This method of invoking Unified CVP from within Unified ICME enables Unified ICME to take advantage of the features of Unified CVP.

The two products (Unified ICME and Unified CVP) provide two service creation (scripting) environments. Each environment is used for different purposes:

- **ICM Script Editor.** Use this scripting tool to develop agent routing scripts and to invoke the Unified CVP **micro-applications**: Play Media, Get Speech, Get Digits, Menu, Play Data, and Capture. These applications are the basic building blocks of a voice interaction design.
- **Call Studio.** Use Call Studio to develop sophisticated IVR applications.

Note For more information, refer to "Scripting for Unified CVP with Call Studio."

Unified ICME Scripting

The Unified ICME Script Editor is used to develop agent routing scripts, and to invoke Unified CVP micro-applications - basic building blocks of a voice interaction design. The Unified CVP micro-applications are: Play Media, Get Speech, Get Digits, Menu, Play Data, and Capture. These applications are combined and customized in the Unified ICME routing script to produce a viable voice interaction with the caller.

While it is possible to develop full scale IVR applications using micro-applications, it is not supported. Micro-application-based scripts are primarily used for initial prompt and collection operations, as well as for directing the playing of .wav files while calls are in queue. Instead, use the IVR scripts developed using Call Studio to create the IVR applications.

In an environment where Unified ICME script works with VXML script (the 2-script implementation for Unified ICME-integrated models described here), the Unified ICME script remains in control (and receives control back), even while it *delegates* the more complex self-service activity to the VXML Server script. Data can be passed from one script to the other and back through ECC variables.

Note The capability of using Unified ICME scripting for anything other than simple functions has been kept in support of legacy deployments. New customers are strongly advised to use the VoiceXML scripting environment of Unified CVP for creating IVR applications.

Unified CVP VoiceXML Scripting

Sophisticated IVR applications can be developed using Call Studio which is an Eclipse-based service creation environment whose output is an intermediary file describing the application flow. That file gets loaded onto the VXML Server machines for execution. To invoke a VXML Server application, the script writer includes a special micro-application in his Unified ICME routing script. This micro-application instructs the VoiceXML Gateway to interact with the VXML Server directly to execute the application. The final results are passed back to Unified ICME.

Some of the VoiceXML scripting environment features include:

- A drag-and-drop interface with a palette of IVR functions
- The ability to do database queries
- Extensibility with Java code written to perform any task a Java application can perform

Note Unified CVP does not support using the *MicroApp* nodes that are available in the ICM Script Editor. All MicroApp implementation must be done using the *Run External Script* node in ICM Script Editor. Refer to *ICM Scripting and Media Routing Guide for Cisco ICM/IPCC Enterprise & Hosted Editions* for detailed information about the Run External Script node. Refer to Writing Unified ICME Applications for Unified CVP for detailed information about setting Unified CVP-specific parameters in this node for each Unified CVP micro-application.

Note For more information about creating scripts, refer to "Scripting for Unified CVP with Unified ICME."

Micro-Application Use versus VXML Server Use

The same special micro-application that is used to invoke VXML Server applications can also be used to invoke arbitrary "External VXML" pages from a Media Server or other customer-provided source. However, only use this capability for very simple VoiceXML needs, because Cisco has no way to verify that *customer-provided* VoiceXML documents are compatible with the IOS Voice Browser. (As opposed to VoiceXML documents that are generated by VXML Server, which *are* guaranteed by Cisco to be compatible with the IOS Voice Browser.) **Although the capability to use the micro-application has not been removed from the Unified CVP 4.0 and later offerings, customers are discouraged from using it directly.**

Additionally, all the VoiceXML Gateway sizing metrics that Cisco provides are based on the specific VoiceXML documents that are generated using either micro-applications or VXML Server applications. Using VoiceXML from another source will require you to perform your own empirical performance and capacity testing in order to determine how to size the VoiceXML Gateways.

Scripting for Unified CVP with Unified ICME

The sections that follow include:

- A discussion of micro-applications.
- A sample Unified ICME script.
- A discussion of how Unified ICME and Unified CVP exchange information.

Micro-Applications

Micro-applications are a set of specific IVR functions that can be invoked by Unified ICME, enabling communication with the caller.

There are six Unified CVP micro-applications:

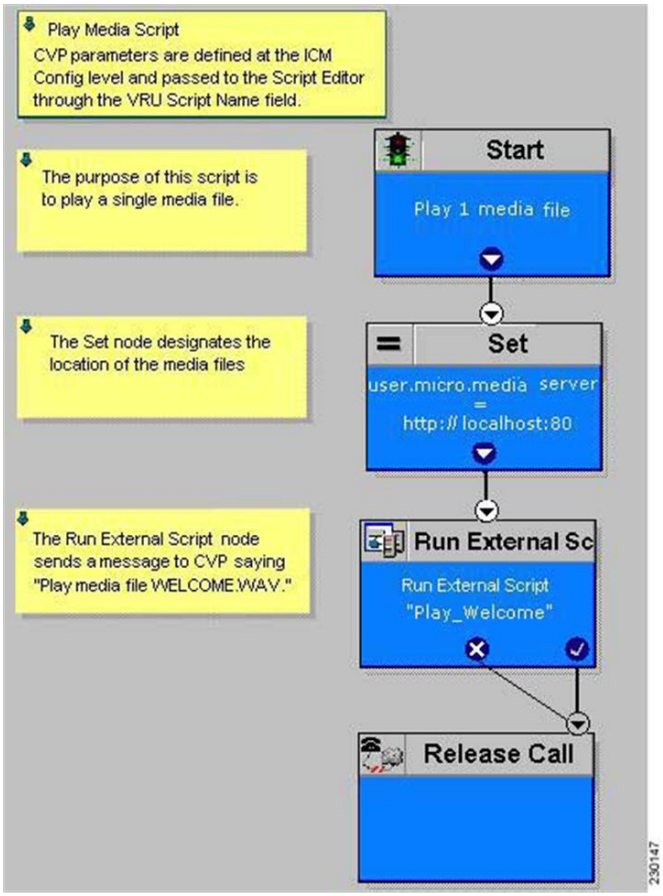
- **Play Media**. Plays a message to the caller.
- **Play Data** . Retrieves data from a storage area and plays it to the caller in a specific format called a data play back type.
- **Get Digits**. Plays a TTS or media file and retrieves digits from the caller.
- **Menu**. Plays a TTS or media menu file and retrieves a single telephone keypad entry from the caller.
- **Get Speech**. Collects ASR or DTMF input after prompting a caller.
- **Capture**. The Capture (CAP) micro-application enables you to trigger the storage of current call data at multiple points in the Unified ICME routing script.

Micro-applications are interpreted by the IVR Service, which resides on the Call Server. The IVR Service sends VoiceXML code to the VoiceXML Gateway Voice Browser.

The IVR Service also accepts HTTP requests from the VoiceXML Gateway's Voice Browser, and communicates those requests to Unified ICME's Service Control Interface using the ICM Service.

Simple Example Script: Welcome to XYZ Corporation

Suppose you want to create a Unified ICME script that simply plays a message, “Welcome to XYZ Corporation.” From the Unified ICME’s perspective, there is no difference between a script written for a standard “black box” IVR or the Unified CVP, so you can create a script such as the one shown in the following figure.



This simple script performs three functions:

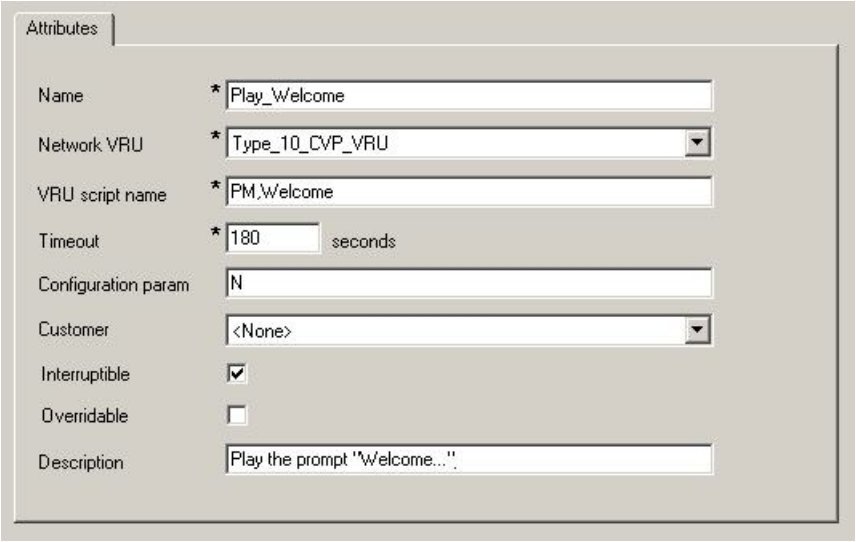
- Sends the Run External Script request to Unified CVP.
- Indicates the location of the "Welcome" media file.
- Releases the call.

Note In a “real life” application, any Unified ICME script you create would include error checking to ensure that micro-applications instructions are properly executed.

Unified ICME Unified CVP Micro-app Connection

Before the Unified CVP IVR solution can be accessible through the Script Editor's Run External Script node, you must first set up Unified ICME with special Unified CVP parameters using the ICM Configuration Manager tool.

Begin by using the ICM Configuration Manager's Network VRU Script window to define Unified CVP parameters.



In the figure above:

- **PM,Welcome**. (VRU Script Name field.) This means: "Use the instructions in the Play Media micro-application to play the Welcome.wav media file."
- **N**. (Configuration Param field.) This means: "Do not allow barge-in." (Barge-in is when the caller can interrupt message play by entering a digit, causing the script to move to the next prompt.)
- You *must* check the **Interruptible** checkbox as shown in the figure above. This specification allows the script to be interrupted by the Unified CVP script functions.

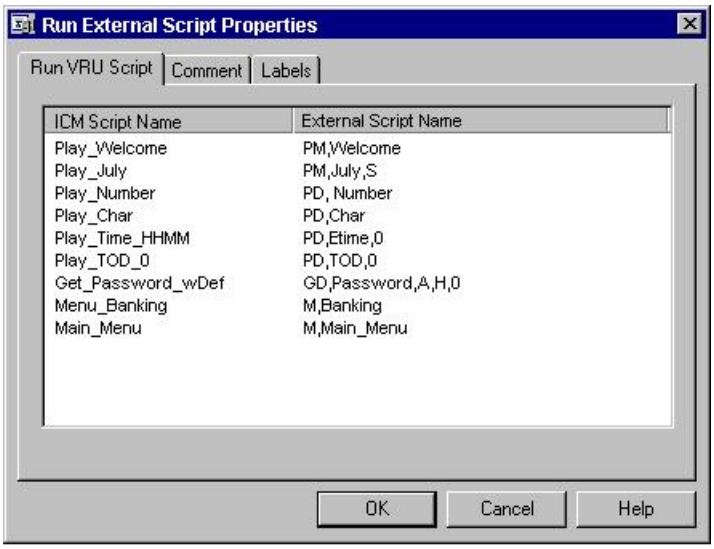
Note As shown in the two columns of the following table, certain entries for the VRU Script Name and Configuration Param fields are case-sensitive.

Network VRU Script Field Attributes That *Are* Case-Sensitive	Network VRU Script Field Attributes That *Are Not* Case-Sensitive
Applies to: All micro-applications. **Attribute:** Media File Name (for example, media or .vxml)	**Applies to:** All micro-applications. **Attribute:** VRU Script Name (for example, PM, GD).
Applies to: Get Speech (GS). **Attribute:** External Grammar File name.	**Applies to:** All micro-applications. **Attribute:** Media Library Type (A, S, V)

Network VRU Script Field Attributes That *Are* Case-Sensitive	Network VRU Script Field Attributes That *Are Not* Case-Sensitive
	Applies to: All micro-applications. **Attribute:** Barge-in Allowed (Y/N),
	Applies to: PlayData (PD). **Attribute:** Data playback type (for example, Number, Char).
	Applies to: PlayData (PD). **Attribute:** Time Format (HHMM, HHMMSS, HHMMAP),
	Applies to: Get Digits (GD), Get Speech (GS), Menu (M). **Attribute:** Timeout Message Override (Y/N)
	Applies to: Get Digits (GD), Get Speech (GS), Menu (M). **Attribute:** Invalid Entry Message Override (Y/N).
	Applies to: All micro-applications. **Attribute:** DTMF Termination Key (only N)
	Applies to: Get Speech (GS). **Attribute:** Type of Data to Collect (for example, boolean, date).

Once the ICM Configuration Manager's settings have been saved, the information is available to the Script Editor. When you place a Run External Script node in the Script Editor workspace and open the Properties dialog box, it displays all the script names defined in the system.

The Run External Script node below shows that the ICM Script Name Play_Welcome was selected.



Information Exchange Between Unified ICME and Unified CVP

When Unified ICME processes a Run External Script node, parameters are sent to Unified CVP.

These parameters contain instructions about how to interact with a caller, such as:

- What micro-application to use.
- The location of the media files to be played to the caller.
- Timeout settings to be used during caller digit entry.

Some IVR parameters are passed to Unified CVP through Expanded Call Context (ECC) variables and/or Call.Peripheral variables. Other parameters are sent in the normal VRU messaging interface (Unified ICME/IVR Service Control Interface).

Unified ICME Data Handling

In defining scripts, you might specify strings, numbers, or formulas to be sent to Unified CVP. When passing numbers to Unified CVP, always enclose them in quotes so that they will be processed as a string.

This is especially important if:

- Leading 0's are significant to the data type (times, character), enter the number as a quoted string (example: "031524").
- Trailing 0's after a decimal point are significant to the data type (number, character, currency), enter the number as a quoted string (examples: "42.00" or "42.10").
- The number is very large (example: a number normally expressed through exponential notation).

Unified CVP Script Error Checking

Unified CVP uses the **user.microapp.error_code** ECC variable to return information regarding problems encountered while running a script.

Unified CVP software tests for the following conditions when processing Unified ICME scripts:

ASR Error

Failure of an Advanced Speech Recognition component.

General error

General error occurred.

Invalid Configuration Param

Data passed from Unified ICME to the IVR Service is not consistent with what the micro-application requires for processing.

Invalid variable data

The variable data passed was not valid for the script type being processed.

Invalid VRU Script Name format

VRU Script Name data passed from Unified ICME to the IVR Service does not contain the expected components (micro-application name, media file name, media file type, uniqueness value).

Locale

Locale was not supported. (Only applies to Play Data micro-applications that use .wav files. Does not apply to Play Data micro-applications that use TTS, or to Play Media, Get Digits, Menu, Get Speech, or Capture micro-applications.)

Misconfigured ECC variable

An ECC variable was set to a value the IVR Service did not recognize. ECC variable definitions must be the same in Unified ICME and Unified CVP.

Network Error

Failure of an IP network connection.

Reached Maximum Invalid Tries

Caller was unsuccessful in entering digits during each of the tries allowed by the micro-application. (Only applies to Get Digits, Menu, and Get Speech micro-applications.)

Reached Maximum Number Entry Tries

Caller did not enter digits in response to the prompt for each of the tries allowed by the micro-application. (Only applies to Get Digits and Get Speech micro-applications.)

Semantic-Runtime

Semantic error occurred while running a micro-application.

System Error

Unexpected failure of a Unified CVP component.

Timed Out

Caller did not enter digits in response to the prompt in the time allowed by the micro-application.

TTS Error

Failure of a Text-to-Speech component.

Unavailable Media file

Media file name passed from Unified ICME to the IVR Service did not exist on the Media Server.

Unknown micro-application

Micro-application name passed from Unified ICME to the IVR Service did not exist on the IVR Service.

Unsupported locale

The VoiceXML Interpreter (that is, Gateway) did not recognize the locale passed from the IVR Service.

Unsupported VoiceXML element

The VoiceXML Interpreter (that is, Gateway) did not recognize a VoiceXML element passed from the IVR Service, VXML Server, or media server (using External VoiceXML).

Unsupported VoiceXML format

The VoiceXML Interpreter (that is, Gateway) did not recognize a VoiceXML format passed from the IVR Service, VXML Server, or media server (using External VoiceXML).

Each Unified CVP micro-application has individualized settings for **user.microapp.error_code** (non-video and video), as shown in the following table.

Table 8: Possible User.Microapp.error_code ECC Variable Settings for Non-Video

Error Code	Play Media	Play Data	Get Digits	Menu	Get Speech	Capture
0	No error	No error	No error	No error	No error	No error
1	Caller Hangup	Caller Hangup	Caller Hangup	Caller Hangup	Caller Hangup	N/A
2	Network Error	Network Error	Network Error	Network Error	Network Error	N/A
3	System Error	System Error	System Error	System Error	System Error	System Error
5	Unknown micro-application	Unknown micro-application	Unknown micro-application	Unknown micro-application	Unknown micro-application	Unknown micro-application
6	Invalid VRU Script Name format	Invalid VRU Script Name format	Invalid VRU Script Name format	Invalid VRU Script Name format	Invalid VRU Script Name format	N/A

Error Code	Play Media	Play Data	Get Digits	Menu	Get Speech	Capture
7	Invalid Configuration Param	Invalid Configuration Param	Invalid Configuration Param	Invalid Configuration Param	Invalid Configuration Param	N/A
8	Misconfigured ECC variable	Misconfigured ECC variable	Misconfigured ECC variable	Misconfigured ECC variable	Misconfigured ECC variable	N/A
9	One of the following: • Media or external VXML file does not exist • Invalid URL for Media or external VXML file • External VXML is in an invalid format	One of the following: • Media file does not exist • Invalid URL for Media L file	One of the following: • Media file does not exist • Invalid URL for Media L file	One of the following: • Media file does not exist • Invalid URL for Media L file	One of the following: • Media or external VXML file does not exist • Invalid URL for Media or external VXML file • External VXML is in an invalid format	N/A
10	Semantic-Runtime Error	Semantic-Runtime Error	Semantic-Runtime Error	Semantic-Runtime Error	Semantic-Runtime Error	N/A
11	Unsupported VoiceXML format	Unsupported VoiceXML format	Unsupported VoiceXML format	Unsupported VoiceXML format	Unsupported VoiceXML format	N/A
12	Unsupported VoiceXML element	Unsupported VoiceXML element	Unsupported VoiceXML element	Unsupported VoiceXML element	Unsupported VoiceXML element	N/A
13	N/A	Variable data is invalid	N/A	N/A	N/A	N/A
14	N/A	Location of variable data is empty	N/A	N/A	N/A	N/A
15	N/A	Time format is invalid	N/A	N/A	N/A	N/A
16	N/A	N/A	Reached Maximum Invalid Tries	Reached Maximum Invalid Tries	Reached Maximum Invalid Tries	N/A

Error Code	Play Media	Play Data	Get Digits	Menu	Get Speech	Capture
17	N/A	N/A	Reached Maximum No Entry Tries	Reached Maximum No Entry Tries	Reached Maximum No Entry Tries	N/A
20	N/A	Data value out of range	N/A	N/A	N/A	N/A
23	No answer	No answer	No answer	No answer	No answer	N/A
24	Busy	Busy	Busy	Busy	Busy	N/A
25	General transfer error	General transfer error	General transfer error	General transfer error	General transfer error	N/A
26	Invalid extension	Invalid extension	Invalid extension	Invalid extension	Invalid extension	N/A
27	Called party hung up	Called party hung up	Called party hung up	Called party hung up	Called party hung up	N/A
28	Error after transfer established	Error after transfer established	Error after transfer established	Error after transfer established	Error after transfer established	N/A
30	Unsupported locale	Unsupported locale	Unsupported locale	Unsupported locale	Unsupported locale	N/A
31	ASR error	ASR error	ASR error	ASR error	ASR error	N/A
32	TTS error	TTS error	TTS error	TTS error	TTS error	N/A
33	General ASR/TTS error	General ASR/TTS error	General ASR/TTS error	General ASR/TTS error	General ASR/TTS error	N/A
34	Unknown error	Unknown error	Unknown error	Unknown error	Unknown error	N/A
40	VXML Server system unavailable	N/A	N/A	N/A	VXML Server system unavailable	N/A
41	VXML Server application error	N/A	N/A	N/A	VXML Server application error	N/A
42	VXML Server application used hangup element instead of subdialog return element	N/A	N/A	N/A	VXML Server application used hangup element instead of subdialog return element	N/A

Error Code	Play Media	Play Data	Get Digits	Menu	Get Speech	Capture
43	VXML Server application is suspended	N/A	N/A	N/A	VXML Server application is suspended	N/A
44	VXML Server session error (for example, application has not yet been loaded)	N/A	N/A	N/A	VXML Server session error (for example, application has not yet been loaded)	N/A
45	VXML Server encounters a bad fetch error (for example, media or grammar file not found)	N/A	N/A	N/A	VXML Server encounters a bad fetch error (for example, media or grammar file not found)	N/A
46	Audio streaming error	N/A	N/A	N/A	N/A	N/A

Note **user.microapp.error_code** is always zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. Usually, if control proceeds out the X (failure) branch, Unified CVP sets this variable to one of the codes listed here. (Set up your routing script to always test the error code after an X branch is taken.)

Note However, if a configuration error, or a network or component failure of some sort, prevents the micro-application from being executed at all, then Unified CVP does not get a chance to set this variable at all. Such cases can be identified by using a Set node to pre-set **user.microapp.error_code** to some known invalid value such as -1, and then to test for that value using an If node, following the X branch of the Run External Script node.

Note Two classes of problems can prevent the micro-application from being executed at all: (1) inability to route the call to an appropriate VoiceXML-enabled gateway and IVR Service (VRU-Only call flow model only); (2) mismatch between Network VRU associated with the configured VRU script and Network VRU associated with Unified CVP that is handling the call. The second case can only be caused by an ICM configuration error, but the first case may also be caused by a temporary network outage or other component failure.

Unified ICME Setup

Before you can use Unified ICME features to access the Unified CVP solution, you must perform some initial setup tasks to enable communication between Unified ICME and Unified CVP. These setup tasks are determined by Unified CVP call flow model; see the *Configuration Guide for Cisco Unified Customer Voice Portal*, for setup procedure for each model.

Note For more information about the supported Unified CVP call flow models, see the *Configuration Guide for Cisco Unified Customer Voice Portal*.

Writing Unified ICME Applications for Unified CVP

Once Unified ICME-to-Unified CVP initial setup is complete, you can create Unified ICME applications to access Unified CVP micro-applications.

You do this using two Unified ICME software tools:

- Configuration Manager
- Script Editor

The sections that follow give a brief overview of how to use these tools to access Unified CVP functionality.

Configure a Unified CVP Network VRU Script

Procedure

Step 1 Within the ICM Configuration Manager, select **Tools** > **List Tools** > **Network VRU Script List**.

Step 2 In the Network VRU Script List window, enable the **Add** button by clicking **Retrieve**.

Step 3 Click **Add**.
The Attributes property tab is enabled.

Step 4 Complete the Attributes tab as described below.

Warning The format of the strings for the **VRU Script Name** and **Configuration Param** fields are *very specific* and vary for different micro-applications (Play Media, Play Data, Get Digits, Menu, and Get Speech).

- **Network VRU**. (Drop-down list.) The name of the Network VRU to be associated with the Network VRU script.
- **VRU Script Name**. A 39-character, comma-delimited string used by Unified CVP to pass parameters to the IVR Service. The content of string depends on the micro-application to be accessed. For more information on what to specify in this field, refer to the following sections:
  - Play Media Micro-Application
  - Play Data Micro-Application
  - Get Digits Micro-Application

- Menu Micro-Application
- Get Speech Micro-Application

- **Name**. A unique name for the VRU script. Unified ICME generates a name based on the Network VRU and script names.
- **Timeout**. The number of seconds Unified ICME waits for a response from the VRU after invoking the script before assuming that the Unified CVP script has failed.

  Warning This setting is designed to detect VRU failures only; attempting to use it as a technique for interrupting script processing can lead to unexpected results. Use the 180-second default or lengthen the setting to a duration that is longer than the longest time the script is expected to take.
- **Configuration Param**. A string used by Unified CVP to pass additional parameters to the IVR Service. The content of string depends on the micro-application to be accessed. For more information on what to specify in this field, refer to the following sections:
  - Play Media Micro-Application
  - Play Data Micro-Application
  - Get Digits Micro-Application
  - Menu Micro-Application
  - Get Speech Micro-Application
- **Description**. Any additional information about the script.
- **Customer**. (Optional.) A customer associated with the script. For Service Provider solutions, this field is mandatory, due to multiple tenancy solutions (customer-specific data needs to be separated).
- **Interruptible**. (Checkbox.) Whether Unified ICME can interrupt the script (for example, if a routing target becomes available).
- **Overridable**. (Checkbox.) Indicates whether the script can override its own Interruptible attribute. Options: This setting does not apply to Unified CVP micro-applications.

Step 5 When finished, click **Save** to apply your changes.

Run External Script Node That Accesses a Unified CVP Micro-Application

Procedure

Step 1 Within Script Editor, place the Run External Script object in the workspace, right-click, and open the Properties dialog box.
The Run External Script Properties dialog box lists all Network VRU scripts currently configured

Note The ICM Script Name column reflects the values defined through the Name field in ICM Configuration Manager's Network VRU Script List tool.

Step 2 Select the **ICM Script/VRU Script Name** you want to execute.

Step 3 Modify the Comments tab, if required.

Step 4 Modify the Labels tab, if required.

Step 5 When finished, click **OK** to submit the changes and close the dialog box.

Unified CVP Micro-Applications

Note Not all third-party Automatic Speech Recognition (ASR) servers use Unified CVP micro-application parameters the same way. This affects how third-party ASR servers interact with the Unified CVP micro-applications. For example, although Unified CVP allows timeout parameters to be set to a value from 1 to 99 seconds, an ASR server may only support a range of 1 to 32 seconds. Another ASR server requires a "#" to indicate that digits are to be collected before the inter-digit timeout is reached. Be sure to follow the instructions provided by your third-party vendor. Test your micro-applications before deploying them.

The sections that follow describe the parameters that can be defined through ICM Configuration Manager for each of the six Unified CVP micro-applications.

Keep the following in mind as you configure each Network VRU Script to be used with Unified CVP:

- Each micro-application parameter in fields of the Network VRU Script List's Attributes tab must be separated by a comma.
- If a parameter value is not specified, the micro-application uses its default.

Each section concludes with sample ICM Configuration Manager and Script Editor screen captures for the micro-application.

Note For examples of Unified CVP IVR scripts, see "Transfer and Queue Calls with Unified CVP."

Micro-Applications Automatic Speech Recognition and Text-to-Speech

Unified CVP micro-applications can use ASR in two ways:

- in Get Digits and Menu micro-applications, to recognize data for built-in data types, such as numbers, dates or currency, using digits and/or voice. The **user.microapp.input_type** ECC variable specifies the collection type. The script writer uses this variable in a Script Editor Set node to allow the caller to input DTMF only (**D**) or both DTMF and Voice (**B**, the default). If you are not using an ASR, you need to set this variable to **D**. If you are using an ASR, you can set the variable to either **D** or **B**. Regardless of the value of **user.microapp.input_type**, the recognized digit(s) are always returned to ICM in the CED variable.

Note With input_mode set to "B" (both), either DTMF or speech is accepted, but mixed mode input is not. When you enter with one mode, input by the other mode is ignored and has no effect.

- in Get Speech micro-applications, to collect voice input according to a specified grammar. The grammar is specified either as inline grammar (through the setting in the **user.microapp.grammar_choices** ECC variable) or as an external grammar file (through a text file, the name of which is given in the Network VRU Script's Configuration Param field). The recognized result is returned to ICM in the **user.microapp.caller_input** ECC variable.

Unified CVP micro-applications can use TTS for two purposes:

- As an alternative for playing recorded announcement prompts with the Play Media, Get Digits, Menu, and Get Speech micro-applications, using either the contents of the **user.microapp.inline_tts** or an external .vxml file. See "Micro-Applications External Voice XML." The ECC variable is useful if the amount of text is short and simple. The external .vxml file is useful for more lengthy text or text that needs to be changed frequently using tools other than the ICM Script Editor.
- As a method of playing data using the Play Data micro-application. If the **user.microapp.pd_tts** ECC variable contains **Y**, Unified CVP uses TTS to speak the data (depending on the TTS locale support and capabilities); if **N**, Unified CVP uses the system recorded announcements to speak the data (depending on IVR Service locale support and capabilities).

Note These ECC variables must be set in the Unified ICME script prior to executing the micro-application that they modify.

Micro-Applications External Voice XML

The Play Media and Get Speech micro-applications render external .vxml; that is, text Voice-XML files. To access the external file, the Media File Component of the Network VRU Script's VRU Script Name field must point to a .vxml file and specify **V** as the Media Library Type parameter.

The external VoiceXML file must contain particular call control catch blocks and must not execute call control, as Unified CVP and Unified ICME must be responsible for all call control. See External VoiceXML File Contents.

Dynamic Audio File Support for Micro-Applications

In ISN 2.0 (an earlier release of Unified CVP), audio files needed to be specified in the VRU Script Name of the Play Media, Menu, Get Digits and Get Speech micro-applications. Unified CVP lets you use a single micro-application and specify the prompt using call variables and the ICM formula editor.

To provide dynamic audio file capability, set the second VRU script parameter to a numeric value, 1-10, prefixed by a dash. You then set the Media Library to either "A", "S", or "V". Unified CVP looks in the corresponding Call.PeripheralVariable for the name of the audio file to play.

When you set the Media Library to "A" or "S", Unified CVP plays the audio file specified by the Call Variable after the "-(number)". For example, if the second VRU Script Parameter is set to "-4", it plays the audio file

specified in Call.PeripheralVariable4. This functionality is added for Play Media, Menu, Get Digits, and Get Speech micro-applications.

If you set the Media Library to “V”, Unified CVP calls the external VoiceXML file specified by the Call Variable after the “-(number)”. If the Script Parameter is set to “-7”, for example, it calls the external VoiceXML file specified in Call.PeripheralVariable7.

Note The “V” option is only supported for the Play Media and Get Speech micro-applications.

Second VRU Script Parameter	Corresponding Call Variable
-1 to -10	Call.PeripheralVariable (1 to 10)

For an example of how to use a dynamic audio file, see the following table.

VRU Script Parameter Example	Definition
PM, -3,V	**PM** - Uses the Play Media micro-application. **-3** - Plays the file specified in Call.PeripheralVariable3. **V** - Acquires the file from the external VoiceXML Media Library.

Example of Using the Dynamic Prompt

Use the dynamic prompt as follows:

- In the Set node in a Unified ICME script, set the value of ToExtVXML[0] to:

  `prompt=http://152.217.34.252/en-us/app/Welcome.wav`

- In the external VoiceXML file specify the following configuration:

```
<?xml version="1.0"?>
<vxml version="2.0">
<form id="BilingualMenu" scope="dialog">
<var name="prompt"/>
<field name="caller_input">
<prompt bargein="true" timeout="3s">
<audio expr="prompt"/>
</prompt>
```

Note A specific hostname, wav filename, and form ID, is used in this example. Replace these elements with your own configuration settings.

Notes

- If you do not specify a file extension for the file name in the Call.PeripheralVariable, the default media file extension is applied.

- If you set the second VRU script parameter to a value prefixed with a dash and don't specify a file name in the corresponding Call.PeripheralVariable, the IVR Service creates a VoiceXML that does not contain a media prompt.
- If you set the second VRU Script Parameter to a value prefixed with a dash and set the "App Media Library" to **V**, signifying external VoiceXML, you must specify a VoiceXML file in the corresponding Call.PeripheralVariable. If you do not, an "Invalid VRU Script Name" error is returned to ICM. If the specified VoiceXML filename does not contain an extension, and **user.microapp.UseVXMLParams** is not set to **N**, the default extension of .vxml is automatically added.
- You can only specify the name of a single file in the Peripheral Variable. You cannot set this value to a name/value pair.

For more information, refer to the sections on individual micro-applications in this chapter.

Default Media Server for Micro-Applications

In prior releases, the only way to specify a media server for a micro-application was to use the ECC variable `user.microapp.media_server`. You can now use the Operations Console to designate a default media server for the entire deployment.

The global default media server can be specified in the Operations Console by selecting **Device Management** > **Media Server**. The default media server is used by the micro-applications if the ECC variable `user.microapp.media_server` is missing or empty in the Unified ICM script.

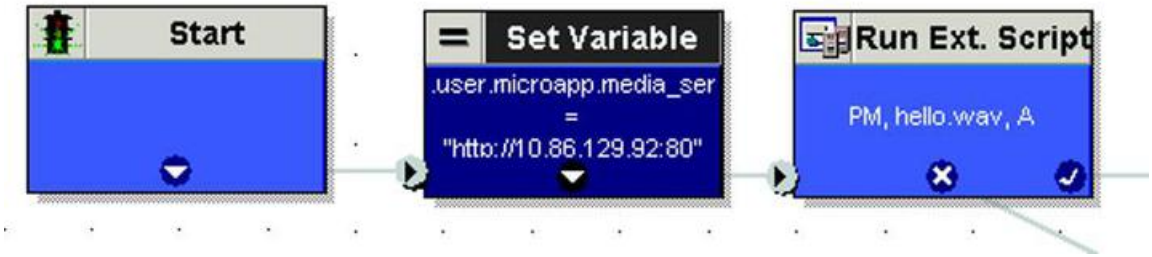
The following list specifies the order in which the micro-application tries to resolve which media server to use:

1. Media server is specified by the ECC variable: `user.microapp.media_server`
2. Global default media server is specified

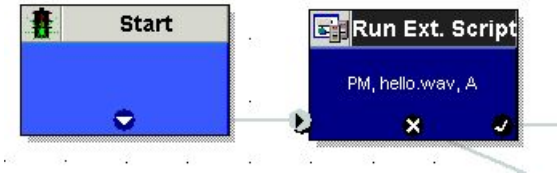
The first non-empty media server value encountered in the above order is used by the micro-application. This applies to all micro-applications including

- Play Media (PM)
- Play Data (PD)
- Get Digits (PD)
- Menu (M)
- Get Speech (GS) only if the Media Library Type in the VRU Script is set to A (Application) or `s` (System) but not `v` (ExternalVXML)

The following screen shot shows the Unified ICM script where Play Media micro-application plays a media file using the ECC variable `user.microapp.media_server`.



The following screen shot shows the Unified ICM script where Play Media micro-application plays a media file using a default media server configured in the Operations Console.



Capture Micro-Application

The Capture (CAP) micro-application allows you to trigger the storage of current call data at multiple points in the ICM routing script. The CAP micro-application must be configured as a VRU script, and it is executed using a RunExternalScript node, just as with any other Unified CVP micro-application. The VRU Script Name value is "CAP" or "CAP,xxx," where "xxx" is any arbitrary string to be used if necessary for uniqueness purposes. There is no VRU Script Config string.

Executing a Capture micro-application causes the ICM PG to produce an intermediate termination record. Specifically, it writes a record in the Termination_Call_Detail (TCD) table which includes all current call variables (not the VRUProgress variable), router call keys, date and time, and caller entered digits. Together with the TCD record, the Capture micro-application writes a set of records to the Termination_Call_Variable (TCV) table which includes the current values of all ECC variables.

Note To capture the ECC variables in the TCV table, enable the **persistent** check box in the **Expanded Variable List** window in the **ICM configuration manager**.

Unified ICME provides no standard reporting templates for TCD and TCV records. These tables are large and minimally indexed, and are optimized for writing rather than querying, to minimally impact call handling throughput. If you plan to report on this data, create off-hours extract processes which copy rows in their raw format into a database which is external to ICM. From there you can organize the tables in the way that best supports your querying requirements.

Information you need about these records includes:

- TCD records for a given call may be identified because they contain the same RouterCallKeyDay and RouterCallKey. Successive TCD records are ordered by incrementing RouterCallKeySequenceNumber.
- Intermediate TCD records may be identified because they contain a CallDisposition of 53, “PartialCall”. Only the last TCD record for the call contains the actual disposition.

- TCV records corresponding to a particular TCD record may be obtained by joining on TCV.TCDRecoveryKey. This key matches the RecoveryKey value in the TCD record.
- As of Unified ICME 6.0(0), the TCD record's CallTypeId is also populated for VRU peripherals. This means you can determine the call's current CallType at each Capture micro-application invocation, and at the end of the call.
- In Unified CVP Comprehensive call flow models, these records are associated with the VRU leg peripheral. If you are doing VRU application reporting, you can filter for TCD records which contain the PeripheralID of the ISN VRU leg.

The Capture micro-application, places a heavy demand on ICM resources. Each time you use it, ICM writes one TCD record and multiple TCV records. Though it can conveniently capture the information you need, it can also capture extra information which you do not require. If you overuse this micro-application, it can place a heavy load on ICM in terms of processing time and disk space, which despite the minimal indexing, may impact ICM's ability to handle the expected call load. Carefully choose where you need to capture information in your scripts. Spread data items into as many call variables as possible to maximize the usefulness of each invocation.

Play Media Micro-Application

The Play Media (PM) micro-application can be configured to play a message that is contained in a media file or streaming audio file.

Configure Network VRU Script for Play Media

Use the ICM Configuration Manager's Network VRU Script List tool's Attributes tab to specify parameters.

Note DTMF digit type-ahead is not supported by Play Media and Play Data micro-apps when executing in Comprehensive mode (Type 7). However, this feature is supported for Type 5 calls.

Procedure

Step 1 Configure VRU Script field parameters:

- **Micro-application type**. For Play Media, valid options are: **PM** or **pm**.
- **Media File Name**. Name of the media file to be played (that is, the prompt file) or the name of the external VoiceXML file.

  The valid options are:

  - A file name (for instance, a .wav file).

    Note Media file names are case-sensitive.

  - **null** - (default) If this field is empty, Unified CVP examines the contents of the **user.microapp.inline_tts** ECC variable. If this ECC variable contains a value, Unified CVP prompts using TTS. If the ECC is empty, no prompt is played.

- **-(number 1-10)** - Unified CVP plays the file in the corresponding Call.PeripheralVariable file. For example, a value of 2 instructs Unified CVP to look at Call.PeripheralVariable2.

  Note If you use the - (number 1-10) option and set the Media Library Type to "V," Unified CVP plays the external VoiceXML file specified in the corresponding Call.PeripheralVariable. If you set the value to - (no value) and set the Media Library Type to "A" or "S", the IVR Service creates VoiceXML without a media prompt.

- **-a** - Unified CVP automatically generates the media file name for agent greeting when this option is specified. The file name is based on GED-125 parameters received from Unified ICM. This option is only valid if the Media Library Type is not set to **V**.

- **Media Library Type**. Flag indicating the location of the media files to be played.

  The valid options are:

  - **A** - (default) Application
  - **S** - System
  - **V** - External VoiceXML

- **Uniqueness value**. Optional. A string identifying a VRU Script Name as unique.

Step 2 Configure the Configuration Param field parameters:

- **Barge-in Allowed**. Specifies whether barge-in (digit entry to interrupt media playback) is allowed.

  The valid options are:

  - **Y** - (default) barge-in allowed
  - **N** - barge-in not allowed

  Note Voice barge-in is not supported by Play Media and Play Data micro-applications. However, Dual Tone Multifrequency (DTMF) barge-in is supported for these micro-applications.

  Unified CVP handles barge-in as follows: If barge-in *is not* allowed, the SIP Service/Gateway continues prompt play when a caller starts entering digits, and the entered digits are discarded. If barge-in *is* allowed, the Gateway discontinues prompt play when the caller starts entering digits. See Get Speech and External VoiceXML.

- **RTSP Timeout**. Specifies the Real-time Streaming Protocol (RTSP) timeout - in seconds - when RTSP is used.

  The valid range is 0 - 43200 seconds (default is 10 seconds). If the value is set to 0 or a timeout value is not provided, the stream does not end.

  See Configure Play Media Micro-Application to Use Streaming Audio for more details.

- **Type-ahead Buffer Flush**. The Cisco VoiceXML implementation includes a type-ahead buffer that holds DTMF digits collected from the caller. When the VoiceXML form interpretation algorithm collects user DTMF input, it uses the digits from this buffer before waiting for further input. This parameter controls whether the type-ahead buffer is flushed after the prompt plays out. A false value (default) means that the type-ahead buffer is not flushed after the prompt plays out. If the prompt allows barge-in, the digit that barges in is not flushed.

  The valid options are

- **Y** - flush the type-ahead buffer
- **N** - (default) do not flush the type-ahead buffer

Note This parameter is only applicable when using the Cisco IOS gateway with DTMF barge-in. This parameter is not applicable when using external VXML. This parameter is normally used when two or more PM and/or PD microapps are used in a loop in the ICM script (such as while in queue for an agent). If the PM and/or PD microapps are enabled for barge-in, one would set this parameter to **Y** to prevent an uncontrolled looping in the ICM script when the user barges in.

Configure Play Media Micro-Application to Use Streaming Audio

Use the ICM Script Editor to configure Play Media (PM) micro-application to play .wav files from a streaming audio server.

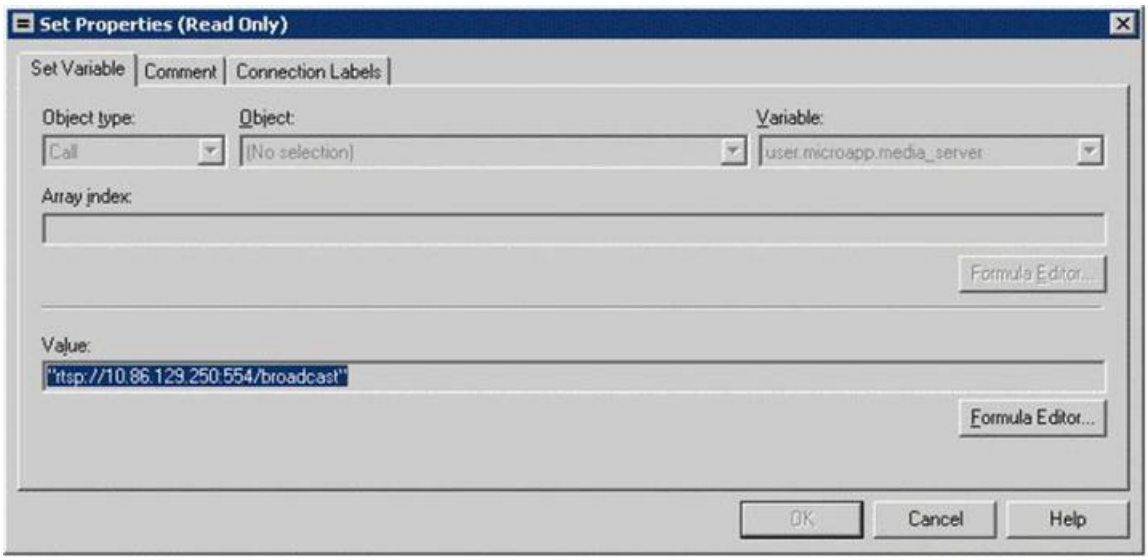
Cisco does not sell, OEM, or support any Media Servers. The IOS gateway only supports µ-law wav files in 8-bit format. Media Servers such as RealNetwork's Helix™ Server will serve RTSP broadcast audio streams in the µ-Law format. See *Configuration Guide for Cisco Voice Portal* to configure Helix Server for use with CVP.

Note Only RealNetwork's Helix™ server has been tested with CVP for RTSP broadcast audio streams in the µ-Law format, and none of the other streaming servers are supported.

Procedure

Step 1 Add a Set Node in the script to configure the media_server ECC variable.

- On the Set Variable tab of the Set Properties dialog box, select **Call** from the Object Type drop down and then set the Variable to user.microapp.media.server.



- In the Value field, specify the URL up to, but not including, the stream name.

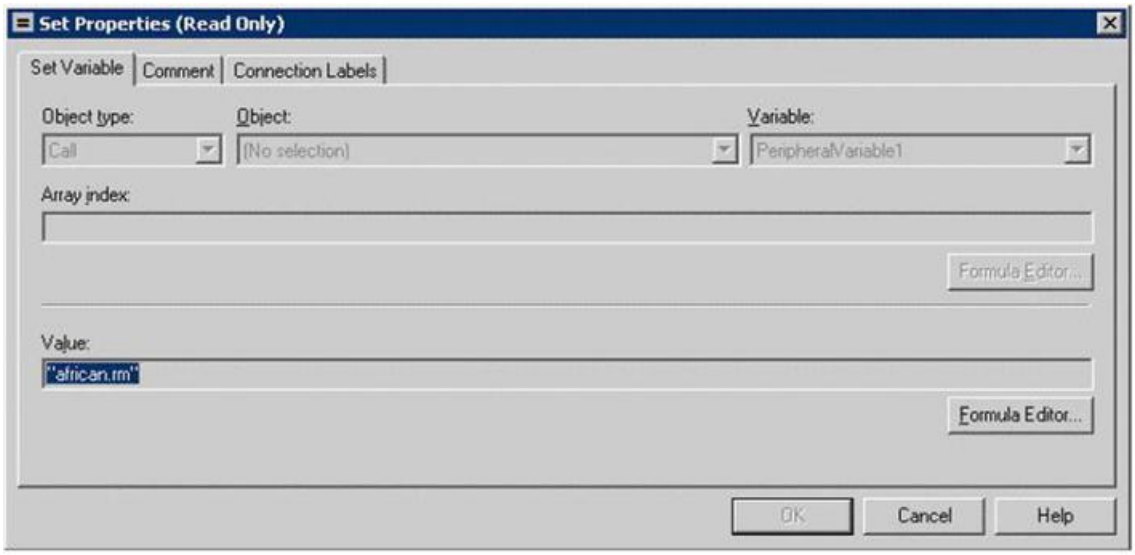
  Note The URL must begin with an *rtsp://* prefix (Real-time Streaming Protocol) to stream audio over the network. A trailing forward slash is not permitted in the URL.

- Click **OK**.

Step 2 Add another Set Node in the script to configure the stream name.

- On the Set Variable tab of the Set Properties dialog box, select Call from the Object Type drop down and set the Variable to **PeripheralVariable<1>**.

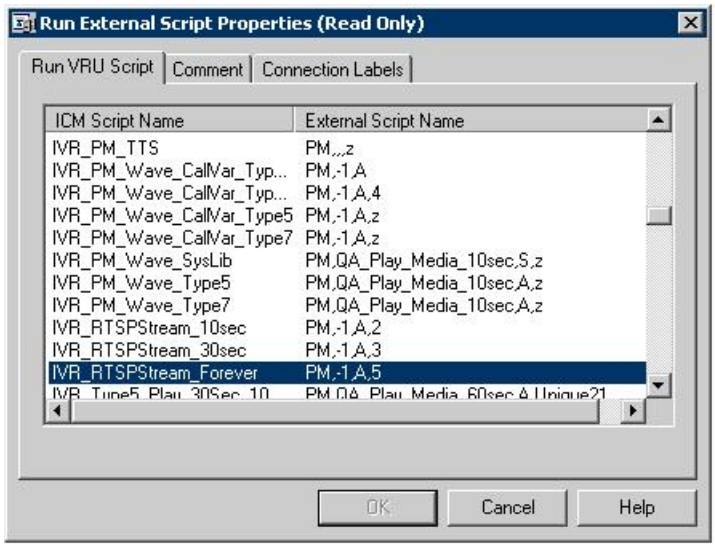
  The range for standard ICM Peripheral Variables is PeripheralVariable1 through PeripheralVariables10.



- In the Value field, specify the stream name and click **OK**.

Note Stream names are case-sensitive.

Step 3 Add a Run External Script node to the workspace and double-click Run External Script.
The Run External Script Properties dialog box lists all of the Network VRU scripts that are currently configured.



Note In the example above, the IVR_RTSPStream_Forever script's external script name contains four parameters: PM, -1, A, 5. The second parameter, **-1**, instructs CVP to play the stream name declared in **PeripheralVariable1** (shown in Step 2). It is recommended that you configure streaming audio following the steps outlined so that you may easily change the stream name within the Script Editor, if necessary.

You can also use the Run External Script node in the ICM Script Editor to configure ICM to failover to a new streaming server. For example, if you want to point to an alternate streaming server (IP address), use the X-path out of the Run External Script node to redefine the media_server ECC variable. In a failover situation, the script executes and the stream plays from the targeted streaming server and proceeds normally.

Step 4 From the Run VRU Script tab, select the ICM Script Name desired and click **OK**.

Step 5 Optionally, you can use the ICM Configuration Manager's Network VRU Script List tool's Attributes tab to configure the timeout value for the stream.
Configure the Configuration Param field parameter:

- In the RTSP Timeout field, enter a timeout value (in seconds).
  - The valid range is 0 - 43200 seconds.
  - If the value is set to 0 or a timeout value is not provided the stream does not end.

Step 6 Access the IOS device in global configuration mode and use the **rtsp client timeout connect** command to set the number of seconds the router waits before it reports an error to the Real-time Streaming Protocol (RTSP) server.

The range is 1 to 20. The recommended value is 10 seconds.

If the SIP Call with IVR Service is Terminated with **Reason Code: Q.850;Cause=38** then be sure that the network interface configuration is as follows:

```
ip route-cache same-interface
ip route-cache cef
ip route-cache
ip mroute-cache
no cdp enable
```

If specified, remove the following line from the network interface:

```
keepalive 1800
```

This issue arises if the Unified CVP loses network connectivity, then the VXML Server Gateway is not able to get information from the IVR Service, and as a result a code 38 rejection is generated in the Gateway logs.

Play Media Examples: Play Welcome Message

The following table shows some Network VRU Script configuration examples for Play Media.

Table 9: Network VRU Script Configuration Examples

Example	Field Name	Field Contents	Tells Unified CVP...
1	VRU Script Name	**PM,Welcome**	To use the Play Media (PM) micro-application to play the "Welcome.wav" Media file and accept the defaults for remaining settings. **Note** If no file extension is specified, .wav is assumed.
	Configuration Param	**N**	That Barge-in *is not* allowed.
2	VRU Script Name	**pm,July,S**	To use the Play Media (PM) micro-application to play the "July.wav" Media file, using the System (S) Media library.
	Configuration Param	**Null** (Accept default.)	That Barge-in *is* allowed.
3	VRU Script Name	**PM,WebSite,,0**	To use the Play Media (PM) micro-application to play the "Website.wav" Media file, using the default Media Type (Application library), and setting `0` as the Uniqueness value. **Note** A **,** (comma) indicates a skipped parameter. When a parameter is skipped, Unified CVP applies its default.
	Configuration Param	**Null** (Accept default.)	That Barge-in *is* allowed.

Example	Field Name	Field Contents	Tells Unified CVP...
4	VRU Script Name	**PM,WebSite,,1**	To use the Play Media (PM) micro-application to play the "Website.wav" Media file, using the default Media Type (Application library), and setting `1` as the Uniqueness value.
	Configuration Param	**N**	That Barge-in *is not* allowed.
5	VRU Script Name	**PM,customer.vxml,V,1**	To use the Play Media (PM) micro-application to the external VoiceXML file “customer.vxml”, using the VoiceXML Media library, and setting `1` as the Uniqueness value.
	Configuration Param	**Note** Any barge-in setting is ignored when using external VoiceXML.	
6	VRU Script Name	**PM**	To use the Play Media (PM) micro-application and accept the defaults for remaining settings. **Note** If the **user.microapp.inline_tts** ECC contains a value, the PM micro-application will play its contents (for example, “Hello world”).
	Configuration Param	**N**	That Barge-in *is not* allowed.
7	VRU Script Name	**PM, -3, A**	To use the Play Media (PM) micro-application, using the file listed in Call.PeripheralVariable3, acquiring the file from the Application (A) media library.
	Configuration Param	**N**	That Barge-in *is not* allowed.
8	VRU Script Name	**PM, -7, V**	To use the Play Media (PM) micro-application, Calls the external VoiceXML listed in Call.PeripheralVariable7, acquiring external VoiceXML (V) media library.
	Configuration Param	**Note** Any barge-in setting is ignored when using external VoiceXML.	

Example	Field Name	Field Contents	Tells Unified CVP...
9	VRU Script Name	**PM, stream.rm**	To use the Play Media (PM) micro-application to play "stream.rm" from a streaming audio server and accept the defaults for remaining settings.
	Configuration Param	**N, 30**	That Barge-in *is not* allowed, and the stream is configured to stop playing in 30 seconds.

Note Play Media sets the ECC variable **user.microapp.error_code** to zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. If control proceeds out the X (failure) branch, Play Media typically sets this variable to one of the codes listed in Unified CVP Script Error Checking.

Play Data Micro-Application

The Play Data micro-application retrieves data from a storage area and plays it to the caller in a specific format, called a data play back type.

Some possible sources of the data to be played back:

- Information retrieved from a database look-up
- Information entered by the caller

Play Data and Data Storage

Before this micro-application can be called, you must specify the location of the play back data. You do this with a Script Editor Set node that points to one of the following storage areas:

- One of the standard ICM Peripheral Variables (PeripheralVariable1 through PeripheralVariables10).
- The **user.microapp.play_data** elements.

Configure Network VRU Script Settings for Play Data Micro-Application

Use the ICM Configuration Manager's Network VRU Script List tool's Attributes tab to specify parameters.

Note DTMF digit type-ahead is not supported by Play Media and Play Data micro-apps when executing in Comprehensive mode (Type 7). This feature is supported for Type 5 calls.

Voice barge-in is not supported by Play Media and Play Data micro-applications. However, DTMF barge-in is supported for these micro-applications.

If you are using integers that are larger than nine digits, enclose the value in quotation marks, so it will be treated as a string.

Procedure

Step 1 Configure VRU Script field parameters:

- **Micro-application type**. For Play Data, valid options are: **PD** or **pd**.
- **Data Playback Type**. The type of the data to be returned ("played") to the caller. The valid options are:
  - **Number**
  - **Char** (character)
  - **Date**
  - **Etime** (elapsed time)
  - **TOD** (Time of Day)
  - **24TOD** (24-hour Time of Day)
  - **DOW** (Day of Week)
  - **Currency**

Note 24TOD and DOW data play back types are not supported when using TTS. Currency other than US dollar (USD) is not supported.

For more information about each of these playback types, including input format and output examples, see Play Back Types for Voice Data.

- **Uniqueness value**. Optional. A string identifying a VRU Script Name as unique.

Step 2 Configure the Configuration Param field parameters:

- **Location of the data to be played** . The valid options are:
  - *null* (default) - If you leave this option empty, uses the ECC variable **user.microapp.play_data**.
  - A **number** representing a Call Peripheral Variable number (for example, a 1 to represent Call.PeripheralVariable1).

Note For more information on data location, see Play Data and Data Storage.

- **Barge-in Allowed**. Specifies whether barge-in (digit entry to interrupt media playback) is allowed.

  The valid options are:
  - **Y** - (default) barge-in allowed
  - **N** - barge-in not allowed

Note Voice barge-in is not supported by Play Media and Play Data micro-applications. However, DTMF barge-in is supported for these micro-applications.

Unified CVP deals with barge-in as follows: If barge-in *is not* allowed, the SIP/Gateway continues prompt play when a caller starts entering digits and the entered digits are discarded. If barge-in *is* allowed, the Gateway discontinues prompt play when the caller starts entering digits. See Get Speech and External VoiceXML.

Barge-in works the same for ASR as DTMF. If the caller speaks during prompt play, the prompt play stops. Unlike DTMF input, ASR caller input is checked against the grammar that is defined. If a match is not found, an Invalid Entry error is generated and the caller input is deleted. Voice barge-in is not supported during a Play Media or Play Data script because there is not a grammar specified for these micro-applications.

- **Time Format**

  Valid only for the time Data Playback types (Etime, TOD, 24TOD).

  The available formats are:

  - *null* - leave this option empty for non-time formats
  - **HHMM** - default for time formats
  - **HHMMSS** - includes seconds
  - **HHMMAP** - includes am or pm; valid only for TOD

- **Type-ahead Buffer Flush** . The Cisco VoiceXML implementation includes a type-ahead buffer that holds DTMF digits collected from the caller. When the VoiceXML form interpretation algorithm collects user DTMF input, it uses the digits from this buffer before waiting for further input. This parameter controls whether the type-ahead buffer is flushed after the prompt plays out. A false value (default) means that the type-ahead buffer is not flushed after the prompt plays out. If the prompt allows barge-in, the digit that barges in is not flushed.

  The valid options are:

  - **Y** - flush the type-ahead buffer
  - **N** - (default) do not flush the type-ahead buffer

  Note This parameter is only applicable when using the Cisco IOS gateway with DTMF barge-in. This parameter is not applicable when using external VXML. This parameter is normally used when two or more PM and/or PD microapps are used in a loop in the ICM script (such as while in queue for an agent). If the PM and/or PD microapps are enabled for barge-in, one would set this parameter to **Y** to prevent an uncontrolled looping in the ICM script when the user barges in.

Play Back Types for Voice Data

Configuring how voice data is presented to a caller is an important part of setting up your Unified CVP IVR. The "Data Play Back Types" table below describes each type, along with sample valid values and formats for the supported locales when **not** using TTS:

- **en-us**. English (United States)
- **en-gb**. English (Great Britain)
- **es-mx**. Spanish (Mexico)
- **es-es**. Spanish (Spain)

Locale is selected by setting the **user.microapp.locale** variable.

Note For information about locale support when using TTS, check with your third-party vendor.

Any string of characters typically used in the language may need to be spoken back character by character (this includes special keyboard symbols and numbers). If a particular symbol is not used by a particular language, a string containing that symbol may be spelled out with a Play Data with Char data type.

For example, assume that an IVR application in the US (a locale of **en-us**) queries a database for an account owner's name and spells the name back to the caller. If the name pulled from the database was "Hänschen Walther," the media files that would need to be pulled from the Media Server would have been derived from a URL including the **en-us** locale. The symbol **ä** has a decimal value of 228, which is different than the symbol a which has a value of 97. It is the translator's task to record the proper word(s) for each symbol to be supported. For detailed information on character translation, refer to "System Media Files."

Note When using TTS, some output format may vary between Unified CVP playback types and TTS playback types. For example, TTS may pronounce a Play Data number "1234" as "twelve thirty four". When not using TTS, the output is "one thousand, two hundred, thirty four". Please check with your third-party vendor on TTS outputs for playback types.

Table 10: Data Play Back Types

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
Number	Play the stored data as a number.	-###############.###### The leading minus (-) is optional and is played as "minus." The whole number portion of the string can contain a maximum of 15 digits (for a maximum value of 999 trillion, 999 billion and so on). The decimal point is represented as a period (.) and played as "point." It is optional if there is no floating portion. The floating point portion of the number is optional and can contain a maximum of six digits. Trailing zeros are played.	**en-us** and **en-gb** typical spoken form: • -123 = "minus one hundred twenty three" • 35.67 = "thirty five point six seven" • 1234.0 = "one thousand, two hundred, thirty four point zero" **es-mx** and **es-es** typical spoken form: • -120 = "menos ciento veinte" • 10.60 = "diez coma seis cero" • 1,100 = "mil cien"
Char	Play the stored data as individual characters.	All printable American National Standards Institute (ANSI) characters are supported. **Note** Code Page 1252 is ANSI standard. It contains ASCII (characters 0-127) and extended characters from 128 to 255	**en-us** and **en-gb** typical spoken form: • abc123= "A, B, C, one, two, three" **es-mx** and **es-es** typical spoken form: • abc123 = "A, B, C, uno, dos, tres"

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
Date	Play the stored data as a date.	YYYYMMDD, regardless of locale. **YYYY** options: the range of 1800 through 9999. **MM** options: the range of 01 through 12. **DD** options: the range of 01 through 31. **Note** The software does not validate the date (for example, 20000231 is valid and played accordingly). However, a failure occurs if any bounds are broken (for example, 34 for month).	**en-us** typical spoken form: • MMDDYYYY format: 20000114 = “January fourteenth, two thousand” **en-gb** typical spoken form: • DDMMYYYY format: 20000114 = “Fourteenth of January, two thousand” **es-mx** and **es-es** typical spoken form: • DDMMYYYY format: 20001012 = "doce octubre dos mil" **Note** All spoken forms use the proper grammar for the locale.
Etime (elapsed time)	Play the stored data as an amount of elapsed time.	HHMM or HHMMSS Maximum 99 hours, 59 minutes, 59 seconds Leading zeros are ignored.	**en-us** and **en-gb** typical spoken form: • HHMM format: 0830= “eight hours thirty minutes” • HHMMSS format: 083020= “eight hours, thirty minutes, twenty seconds” **es-mx** and **es-es** typical spoken form: • HHMM format: 0205 = “dos horas cinco minutos” • HHMMSSS format: 020101 = “dos horas un minuto un segundo”

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
TOD (Time of Day)	Play the stored data as a time of day.	HHMM or HHMMSS 24 hour time **HH** options: 00 - 24 **MM** options: 00 - 59 **SS** options: 00 - 59	**en-us** and **en-gb** typical spoken form: • HHMM format: 0800 = “eight o’clock” 0830 = “eight thirty” 1430 = “two thirty” • HHMMSS format: 083020 = “eight thirty and twenty seconds” • HHMMAP format: 1430 = “two thirty p.m.” **es-mx** and **es-es** typical spoken form: • HHMM format: 0100 = “una a.m.” • HHMMAP format: 1203 = “doce y tres p.m.” • HHMMSS format: 242124 = “doce veintiuno a.m.”

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
24TOD (24-hour Time of Day)	Play the stored data as military time.	HHMM or HHMMSS 24 hour time. **HH** options: 00 - 24 **Note** 24-hour time and military time may have a discrepancy as to valid hours. Unified CVP plays back the value 00 or 24 "as is." The application developer is free to make alterations to the data passed to the micro-application, if so desired. **MM** options: 00 - 59 **SS** options: 00 - 59 Unified CVP validates the ranges as stated above. For example, if a time ends in 00 minutes (that is, 2300), one would say "hundred hours" (that is, "twenty-three hundred hours"). The range is 0000 (12 a.m.) through 2459 (after midnight) or 0059, if you prefer. 1300 equals 1 o'clock in the afternoon. **Note** The 24TOD play back type is not supported when using TTS.	**en-us** and **en-gb** typical spoken form: • HHMM format: 0815 = "eight fifteen" 2330 = "twenty three thirty" 2300 = "twenty three hundred hours" • HHMMSS format: 233029 = "twenty three thirty and twenty nine seconds" **es-mx** and **es-es** typical spoken form: • HHMM format: 2121 = "veintiuno veintiuno" 2100 = "veintiún horas" **Note** In Spanish, when a time ends in 00 minutes the spoken form is "hours," not "hundred hours." • HHMMSS format: 050505 = "cinco y cinco y cinco segundos"
DOW (Day of Week)	Play the stored data as a day of week.	An integer from 1 through 7 (1 = Sunday, 2 = Monday, et cetera). **Note** The DOW data play back type is not supported when using TTS.	**en-us** and **en-gb** typical spoken form: • 7 = "Saturday" **es-mx** and **es-es** typical spoken form: • 7 = "Sabado"

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
Currency	Play the stored data as currency.	Format is [-]15(X)[.2(Y)] where the minus sign is optional as well as the decimal point and the two digits after the decimal point. The whole number portion of the string can contain a maximum of 15 digits (for a maximum value of 999 trillion, 999 billion). **Note** No comma delimiters or currency symbols are recognized. Leading and trailing zeros are played. If a number does not have a decimal point, the "cents" portion of the amount will not be spoken. For example, the spoken form for the input 100 is "one hundred dollars." The grammar rules apply to the currency, not the locale. **Note** The **user.microapp.currency** ECC variable contains the currency indicator (USD, CAD, EUR, et cetera).	**USD** (US dollar) typical spoken form: • 15.05 = "fifteen dollars and five cents" • 3.00 = "three dollars and zero cents" **Note** Unified CVP uses the USD_dollar.wav and USD_dollars.wav media files; the dollar.wav and dollars.wav used by ISN Version 1.0 are no longer installed. **CAD** (Canadian dollar) typical spoken form: • 15.05 = "fifteen dollars and five cents" • 3.00 = "three dollars and zero cents" **EUR** (Euro dollar) typical spoken form: • 1.10 = "one point one zero euro" **GBP** (Great Britain pound) typical spoken form: • 1.10 = "one pound ten pence" **MXN** (Mexican pesos) typical spoken form: • 1.10 = "one peso and 10 centavos" **Note** The default spoken form for a negative amount (for all currency types) is "minus <amount>."

Note 24TOD and DOW data play back types are not supported when using TTS. Also, currency other than US dollar (USD) is not supported.

Play Data Configuration Examples

The following table shows several configuration examples for Play Data.

Table 11: Play Data Configuration Examples

If the VRU Script Name field setting is…	It means…	If the Configuration Param field is…	It means…
PD,Number **Note** If you are using integers that are larger than nine digits, enclose the value in quotation marks, so it will be treated as a string.	**PD** - Use the Play Data micro-app. **Number** - Play back the data as a number.	empty	Play the data in the default ECC, **user.microapp.play_data**, as a number.
PD, Char	**pd** - Use the Play Data micro-app. **Char** - Play back the data as individual characters.	1	**1** - Play the data in Call PeripheralVariable 1 as a character.
PD,Etime,0 **Note** If you are using integers that are larger than 9 digits, enclose the value in quotation marks, so it will be treated as a string.	**PD** - Use the Play Data micro-app. **Etime** - Play back the data as a Time.	1,,HHMM	**1** - Play the data in Call PeripheralVariable 1 as an elapsed time. **,** - (Skipped parameter) Accept default setting (Y) **HHMM** - Play the time in HHMM format (for example, 8 hours, 30 minutes).
PD,Date	**PD** - Use the Play Data micro-app. **Date** - Play back the data as a Date.	1,N	**1** - Play the data in Call Variable 1 as a date. **N** - No barge-in allowed.
PD,Currency	**PD** - Use the Play Data micro-app. **Currency** - Play back the data as a Currency.	4,N	**4** - Play the data in Call Variable 4 s currency. **N** - No barge-in allowed.

Note Play Data sets the ECC variable **user.microapp.error_code** to zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. If control proceeds out the X (failure) branch, Play Data typically sets this variable to one of the codes listed in Unified CVP Script Error Checking.

To enable Text-to-Speech (TTS) as the data play back source, you need to insert a Set node in the script prior to the Run External Script node setting **user.microapp.pd_tts** to **"Y"**.

Get Digits Micro-Application

The Get Digits (GD) micro-application plays a media file and retrieves digits. For example, you could use Get Digits in an application that prompts a caller to enter a password.

Unified Customer Voice Portal passes the retrieved digits back to Unified ICME for further processing using the Caller-Entered Digits (CED) field in the ICM/IVR Messaging interface. (This is available in the Unified ICME script through the variable Call.CallerEnteredDigits).

Note In Customer Voice Portal, the Collected digits will only be cached for a CAPTURE microapp following the digit collection node. If a non-digit collect nodes follows a digit collect node, Customer Voice Portal's CED variable will get nulled out and inturn **Call.CallerEnteredDigits** also get nulled out .

For example, if the script is **START** > **GD (Get Digit node)** > **CAPTURE NODE** > **PM (Play Media)** > **CAPTURE NODE**. In this case, user has to cache the collected data via ICM peripheral (or call) variables.

Here collected digits will be available only for first CAPTURE NODE. After PM node (non-digit collect node), the CED value is set to null. So second CAPTURE NODE or any node which require collected digits to process, will not be able access the collected digits through **Call.CallerEnteredDigits**.

Configure Network VRU Script Settings for Get Digits Micro-Application

Use the ICM Configuration Manager's Network VRU Script List tool's Attribute tab to specify parameters.

Procedure

Step 1 Configure VRU Script field parameters:

- **Micro-application type**. For Get Digits, valid options are: **GD** or **gd**.
- **Media File Name**. Name of the media file or external VoiceXML to be played (that is, the prompt file). The valid options are:
  - A file name (for instance, a .wav file).

    Note The file name is case-sensitive.
  - **null** - (default) If this field is empty, Unified CVP examines the contents of the **user.microapp.inline_tts** ECC variable. If this ECC variable contains a value, Unified CVP prompts using TTS. If the ECC is empty, no prompt is played.

- **-(number 1-10)** - Unified CVP plays the file in the corresponding Call.PeripheralVariable file. For example, entering -2 causes Unified CVP to look at Call.PeripheralVariable2.

- **Media Library Type** . Flag indicating the location of the media files to be played. The valid options are:
  - **A** - (default) Application
  - **S** - System

  Note This value is ignored if using TTS.

- **Uniqueness value**. Optional. A string identifying a VRU Script Name as unique.

Step 2 Configure the Configuration Param field parameters:

- **Minimum Field Length**. Minimum number of digits expected from the caller. The valid options are: **1-32** (the default is **1**)
- **Maximum Field Length**. Maximum number of digits expected from the caller. The valid options are: **1-32** (the default is **1**).

  Note For information about Maximum Field Length and the DTMF Termination Key, see Get Digits and Digit Entry Completion.

- **Barge-in Allowed** . Specifies whether barge-in (digit entry to interrupt media playback) is allowed.

  The valid options are:
  - **Y** - (default) barge-in allowed
  - **N** - barge-in not allowed

  Note Unified CVP deals with barge-in as follows: If barge-in *is* not allowed, the SIP/Gateway continues prompt play when a caller starts entering digits. If barge-in *is* allowed, the Gateway discontinues prompt play when the caller starts entering digits. See Get Speech and External VoiceXML.

- **Inter-digit Timeout** . The number of seconds the caller is allowed between entering digits. If exceeded, the system times-out. The valid options are: **1-99** (the default is **3**).

  Note This value is ignored if using ASR.

- **No Entry Timeout** . The number of seconds a caller is allowed to begin entering digits. If exceeded, the system times-out. The valid options are: **0-99** (the default is **5**).
- **Number of No Entry Tries**. Unified CVP repeats the "Get Digits" cycle when the caller does not enter any data after the prompt has been played. (Total includes the first cycle.) The valid options are: **1-9** (the default is **3**).
- **Number of Invalid Tries**. Unified CVP repeats the "Get digits" cycle when the caller enters invalid data (total includes the first cycle). The valid options are: **1-9** (default is **3**).
- **Timeout Message Override** . The valid options are:
  - **Y** - override the system default with a pre-recorded Application Media Library file

- **N** - (default) do not override the system default

Note This value is ignored if using TTS.

- **Invalid Entry Message Override**. The valid options are:
  - **Y** - override the system default with a pre-recorded Application Media Library file.
  - **N** - (default) do not override the system default

Note This value is ignored if using TTS.

For more information about Timeout and Invalid Entry Messages, see System Media Files.

- **DTMF Termination Key**. A single character that, when entered by the caller, indicates that the digit entry is complete. The valid options are:
  - **0-9**
  - ***** (asterisk)
  - **#** (pound sign, the default)
  - **N** (No termination key)

Note For information about Maximum Field Length and the DTMF Termination Key, see Get Digits and Digit Entry Completion.

This value is ignored if using ASR.

- **Incomplete Timeout**. The amount of time after a caller stops speaking to generate an invalid entry error because the caller input does not match the defined grammar. The valid options are: **0-99** (the default is **3**).

Note This value is ignored when not using ASR. If the value is set to 0, the IVR Service treats the NoEntry Timeout as NoError.

Get Digits Configuration Examples

The following table shows several configuration examples for Get Digits for an application that prompts using .wav files and retrieves input through DTMF.

Table 12: Get Digits Configuration Examples

If the VRU Script Name field setting is…	It means…	If the Configuration Param field setting is…	It means…
GD,Password,A,0	**GD** - Use the Get Digits micro-app. **Password** - Play the Media file named "Password.wav." **A** - Application Media Library. **0** - Uniqueness value.	6,12	**6** - Minimum field length **12** - Maximum field length Accept defaults for all other settings.
GD,Password,A,1	**gd** - Use the Get Digits micro-app. **Password** - Play the Media file named "Password.wav." **A** - Application Media Library. **1** - Uniqueness value.	6,12,N,3,5,2,2,N,Y,#	**6** - Minimum field length **12** - Maximum field length **N** - No barge-in allowed **3** - Inter-digit Timeout (seconds) **5** - No Entry Timeout (seconds) **2** - Number of no entry tries **2** - Number of invalid tries **N** - Timeout Msg Override **Y** - Invalid Entry Msg Override **#** - DTMF Termination key
Note The two examples above both play the Password.wav file ("Please enter your password followed by the pound sign.") and collect digits. They differ in that the first example accepts most of the default settings available through the Configuration Param field; the second field does not.			
GD,ssn	**GD** - Use the Get Digits micro-app. **ssn** - Play the Media file named "ssn.wav."	9,9,	**9** - Minimum field length **9** - Maximum field length Accept defaults for all other settings.

If the VRU Script Name field setting is…	It means…	If the Configuration Param field setting is…	It means…
GD	**GD** - Use the Get Digits micro-app. Since no Media field settings appear after **GD**, Unified CVP examines the contents of the **user.microapp.inline_tts** ECC variable. If this ECC variable contains a value - for example, "What is your account number?" - Unified CVP prompts using TTS. **Note** If the **user.microapp.inline_tts** is empty, no prompt is played. In turn, if the **user.microapp.input_type** ECC variable is D, Unified CVP is set to process any DTMF input the customer supplies.	6,12,N	**6** - Minimum field length **12** - Maximum field length **N** - No barge-in allowed Accept defaults for all other settings.
Note Type-ahead can only be used with the Get Digits micro-application when **user.microapp.input_type** is set to **D**. See Get Speech and External VoiceXML.			
GD, -4, S	**gd** - Use the Get Digits micro-app **-4** - Calls the file specified in Call.PeripheralVariable4 **S** - Acquires the file from the System media library	6,12,	**6** - Minimum field length **12** - Maximum field length Accept defaults for all other settings

The following table shows several configuration examples for Get Digits for an ASR/TTS application.

Table 13: Get Digits Configuration Examples

If ...	It means ...	If the Configuration Param field setting is…	It means ...
The **user.microapp.inline_tts** ECC variable contains "What is your account number?" *and* **user.microapp.input_type** contains: **D** (DTMF) *and* The VRU Script Name field contains: **GD**	Use the Get Digits micro-app to play the contents of the ECC variable and collect DTMF input.	6, 12, N,3,5,2,2,N,Y,#	**6** - Minimum field length **12** - Maximum field length **N** - No barge-in allowed **3** - Inter-digit Timeout (seconds) **5** - No Entry Timeout (seconds) **2** - Number of no entry tries **2** - Number of invalid tries **N** - Timeout Msg Override **Y** - Invalid Entry Msg Override **#** - DTMF Termination key
The **user.microapp.inline_tts** ECC variable contains "What is your account number?" *and* **user.microapp.input_type** contains: **B** (the default, both DTMF and voice) *and* The VRU Script Name field contains: **GD**	Use the Get Digits micro-app to play the contents of the ECC variable and collect either voice or DTMF input.	6,12,N,,,,4	**6** - Minimum field length **12** - Maximum field length **N** - No barge-in allowed **,,,,** - Accept defaults for Inter-digit Timeout (seconds), No Entry Timeout (seconds), Number of no entry tries, Number of invalid tries, Timeout Msg Override, Invalid Entry Msg Override, DTMF Termination key **4** - Incomplete timeout
The **user.microapp.inline_tts** ECC variable contains "What is your account number?" *and* **user.microapp.input_type** contains: **B** (the default, both DTMF and voice) *and* The VRU Script Name field contains: **GD**	Use the Get Digits micro-app to play the contents of the ECC variable and collect DTMF or voice input.	6,12,N	**6** - Minimum field length **12** - Maximum field length **N** - No barge-in allowed Accept defaults for all other settings.
Note	Type-ahead can only be used with the Get Digits micro-application when **user.microapp.input_type** is set to **D**. See Get Speech and External VoiceXML.		

Note Get Digits sets the ECC variable **user.microapp.error_code** to zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. If control proceeds out the X (failure) branch, Get Digits typically sets this variable to one of the codes listed in Unified CVP Script Error Checking.

Get Digits and Digit Entry Completion

Unified CVP tests GD digit entry input against several conditions to determine whether digit entry is complete.

Unified CVP considers digit entry to be complete if the caller enters any of the following:

- The maximum allowable number of digits (when terminator key is not used).
- The maximum number of digits, excluding a terminator key.
- Less than the maximum number of digits, followed by the terminator key.
- Less than the maximum number of digits and exceeding the inter-digit timeout.
- Nothing and reaching the no entry timeout.

Caution It is important that you set up your Unified ICME script to test for all the scenarios mentioned below.

If Digit Entry Input Is Complete

After digit-entry input is complete, Unified CVP validates the digit string to determine if it is >= (greater than or equal to) the minimum length and <= (less than or equal to) the maximum length.

In variable-length data entry, the Maximum Field Length value does not accommodate the termination key. For example, if a GD micro-application is configured to accept a password that is between 6 and 12 digits long and digit-entry completion is indicated through a termination key (or a timeout), the Minimum Field Length setting would be 6, the Maximum Field Length setting would be 12, and the DTMF Termination Key is defined as a single character.

Before passing the result back to the IVR Service, SIP Service discards the termination key (only the password digits are included in the CED returned to Unified ICME).

Note In this example, if the 13th digit is entered without reaching the interdigit timeout and the 13th digit is not the terminator key, the extra digits are buffered by the gateway VXML browser and will be consumed by the next digit collecting node (for example: GD or Menu micro-app).

This type-ahead behavior is described online in the Type-ahead Support section of the Cisco VoiceXML Programmer's Guide.

After validating the digit string, Unified CVP does the following:

- If the string is valid, Unified CVP stores the digit string (not including the terminator key) in the Call.CallerEnteredDigits variable, exits the node through the Checkmark (success) branch, and returns control to Unified ICME software.
- If the string is not valid, Unified CVP considers it an invalid entry and does the following:

- If the Number of Invalid Entry Tries value is not reached, Unified CVP plays an error message and re-plays the original prompt.
- If the Number of Invalid Entry Tries value is reached, Unified CVP stores the last-entered digit string in the Call.CallerEnteredDigits variable, exits the node through the X (failure) branch, sets the **user.microapp.error_code** ECC variable to **16** (Reached Maximum Invalid Tries), and returns control to Unified ICME.

If No Entry Timeout Occurs

If the caller does not enter input and No Entry Timeout period is exceeded, the following happens:

- If the Number of No Entry Tries value has not been reached, Unified CVP plays the "no entry" error message and re-plays the original prompt.
- If the Number of No Entry Tries value has been reached, Unified CVP exits the node through the X (failure) branch, sets the Call.CallerEnteredDigits variable to NULL, the **user.microapp.error_code** ECC variable to **17** (Reached Maximum No Entry Tries), and returns control to Unified ICME.

Menu Micro-Application

This micro-application plays a menu media file and retrieves a defined digit. (Menu is similar to the Get Digit micro-application except that it only accepts one digit, which it checks for validity.)

Unified CVP passes the retrieved digit back to Unified ICME for further processing using the Caller-Entered Digits (CED) field in the ICM/IVR Messaging interface.

Configure Network VRU Script Settings for the Menu Micro-Application

Use the ICM Configuration Manager's Network VRU Script List tool's Attribute tab to specify parameters.

Procedure

Step 1 Configure VRU Script field parameters:

- **Micro-application type** . For Menu, valid options are: **M** or **m**.
- **Media File Name**. Name of the media file or external VoiceXML to be played (that is, the prompt file). The valid options are
  - A file name (for instance, a .wav file)

    Note The file name is case-sensitive.
  - **null** - (default) If this field is empty, Unified CVP examines the contents of the **user.microapp.inline_tts** ECC variable. If this ECC variable contains a value, Unified CVP prompts using TTS. If the ECC is empty, no prompt is played.
  - **-(number 1-10)** - Unified CVP plays the file in the corresponding Call.PeripheralVariable file. For example, entering -2 causes Unified CVP to look at Call.PeripheralVariable2.

- **Media Library Type** . Flag indicating the location of the media files to be played. The valid options are:
  - **A** - (default) Application
  - **S** - System

  Note This value is ignored if using TTS.
- **Uniqueness value**. Optional. A string identifying a VRU Script Name as unique.

Step 2 Configure the Configuration Param field parameters:

- A list of **menu choices** . The valid options are:
  - **0-9**
  - ***** (asterisk)
  - **#** (pound sign)

  Formats allowed include:
  - Individual options delimited by a / (forward slash)
  - Ranges delimited by a - (hyphen) with no space
- **Barge-in Allowed** . Specifies whether barge-in (digit entry to interrupt media playback) is allowed.

  The valid options are:
  - **Y** - (default) barge-in allowed
  - **N** - barge-in not allowed

  Note Unified CVP deals with barge-in as follows: If barge-in *is not* allowed, the Gateway continues prompt play when a caller starts entering digits. If barge-in *is* allowed, the Gateway discontinues prompt play when the caller starts entering digits. See Get Speech and External VoiceXML.
- **No Entry Timeout** . The number of seconds a caller is allowed to begin entering digits. If exceeded, the system times-out. The valid options are: **0-99** (the default is **5**).
- **Number of No Entry Tries**. Unified CVP repeats the "Menu" cycle when the caller does not enter any data after the prompt has been played. (Total includes the first cycle.) The valid options are: **1-9** (the default is **3**).
- **Number of Invalid Tries** . Unified CVP repeats the prompt cycle when the caller enters invalid data. (Total includes the first cycle.) The valid options are: **1-9** (the default is **3**).
- **Timeout Message Override**. The valid options are:
  - **Y** - override the system default with a pre-recorded Application Media Library file
  - **N** - (default) do not override the system default
- **Invalid Entry Message Override** . The valid options are:
  - **Y** - override the system default with a pre-recorded Application Media Library file

- **N** - (default) do not override the system default

Note For more information about Timeout and Invalid Entry Messages, refer to "System Media Files."

Menu Configuration Examples

The following table shows several configuration examples for Menu for use in an ASR/TTS application:

Table 14: Menu Configuration Examples - ASR/TTS Application

If...	It means...	And, if the Configuration Param field contains...	It means...
The **user.microapp.inline_tts** ECC variable contains "Press 1 for Sales and 2 for Support." and **user.microapp.input_type** contains: **D** (DTMF) and The VRU Script Name field contains: **M**	Use the Menu micro-app to play the contents of the **user.microapp.inline_tts** ECC variable and collect DTMF input.	1-2,Y,4,3	**1-2** - Accept the DTMF digits 1 and 2. **Y** - Barge-in allowed. **4** - No Entry Timeout value (in seconds). **3** - Number of no entry tries allowed.
The **user.microapp.input_type** ECC variable contains: **D** (DTMF) and The VRU Script Name field contains: **M,SalesService,A**	Use the Menu micro-app to play the media file named "SalesService.wav" (which is located in the Application Media library) and collect DTMF input.	1-2,N,4,3,2,Y,Y	**1-2** - Accept the numbers 1 and 2. **N** - No barge-in allowed. **4** - No Entry Timeout value (in seconds). **3** - Number of no entry tries allowed. **2** - Number of invalid tries allowed. **Y** - Allow Timeout Msg Override. **Y** - Allow Invalid Entry Msg Override).

If...	It means...	And, if the Configuration Param field contains...	It means...
The **user.microapp.inline_tts** ECC variable contains "Press or Say 1 for Sales and 2 for Support." and **user.microapp.input_type** contains: **B** (the default, both DTMF and voice) and The VRU Script Name field contains: **M**	Use the Menu micro-app to play the contents of the **user.microapp.inline_tts** ECC variable and collect either DTMF or voice input.	1-2,Y,4,3	**1-2** - Accept the DTMF digits 1 and 2. **Y** - Barge-in allowed. **4** - No Entry Timeout value (in seconds). **3** - Number of no entry tries allowed.
The **user.microapp.inline_tts** ECC variable contains "Press 1 for Sales and 2 for Support." and **user.microapp.input_type** contains: **B** (the default, both DTMF and voice) and The VRU Script Name field contains: **M**	Use the Menu micro-app to play the contents of the **user.microapp.inline_tts** ECC variable and collect DTMF or voice input.	The Configuration Param field contains: **1-2,Y,4,3**	**1-2** - Accept the DTMF digits 1 and 2. **Y** - Barge-in allowed. **4** - No Entry Timeout value (in seconds). **3** - Number of no entry tries allowed.
Note Type-ahead can *only* be used with the Menu micro-application when **user.microapp.input_type** is set to **D**. See Get Speech and External VoiceXML.			

The following table shows several configuration examples for Menu for use in an application where input type is DTMF.

Table 15: Menu Configuration Example - DTMF Application

If the VRU Script Name field setting is…	It means...	If the Config Param setting is...	It means...
M,Banking	**M** - Use the Menu micro-app. **Banking** - Play the Media file named "Banking.wav." **Note** This file may contain a message such as: "For Checking, press 1. For Savings, press 2. For Money Market, press 3."	1-3	**1-3** - Accept numbers 1, 2, 3. Accept all other defaults (No Entry Timeout, Number of no entry tries, Number of invalid tries, Timeout Msg Override, Invalid Entry Msg Override).
M,Main_Menu	**M** - Use the Menu micro-app. **Main_Menu** - Play the Media file called "Main_Menu.wav." **Note** This file may contain a message such as: "For information or transactions on checking, press 1. For savings or club accounts, press 2. For other information, press 0. If you know your party's extension, press 9."	0-2/9,,4,2,2	**0-2/9** - Accept numbers 0, 1, 2, and 9. **,** (Skipped parameter) - Accept the default barge-in setting (Y). **4** - No Entry Timeout value (in seconds). **2** - Number of no entry tries allowed. **2** - Number of invalid tries allowed. Accept all other defaults (Timeout Msg Override, Invalid Entry Msg Override).
M,-2,S	**M** - Use the Menu micro-app. **-2** - Plays the file specified in Call.PeripheralVariable2. **S** - Acquires the file from the System media library.	1-3	**1-3** - Accept numbers 1, 2, 3. Accept all other defaults (No Entry Timeout, Number of no entry tries, Number of invalid tries, Timeout Msg Override, Invalid Entry Msg Override).

Note Menu sets the ECC variable **user.microapp.error_code** to zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. If control proceeds out the X (failure) branch, Menu typically sets this variable to one of the codes listed in Unified CVP Script Error Checking.

Menu and Digit Entry Completion

Unified CVP tests Menu digit entry input against two conditions to determine whether digit entry is complete:

- If a caller enters a digit, Unified CVP checks whether the digit is within the set of valid digits for this menu.
- If a caller does not enter a digit, Unified CVP checks whether the No Entry Timeout value has been reached.

Caution It is important that you set up your Unified ICME script to test for all the scenarios mentioned below.

Digit Entry Completion

After a caller enters a digit, Unified CVP validates the digit against the list of valid menu options that were defined through ICM Configuration Manager. Then Unified CVP does the following:

- If the digit is valid, Unified CVP stores the digit in the Call.CallerEnteredDigits variable, exits the node through the Checkmark (success) branch, and returns control to Unified ICME.
- If the digit is not valid, Unified CVP considers it an invalid entry and does the following:
  - If the Number of Invalid Entry Tries value *has not* been reached, Unified CVP plays the "invalid message" file and re-plays the menu prompt.
  - If the Number of Invalid Entry Tries value has been reached, Unified CVP stores the last-entered invalid digit in the **user.microapp.caller_input** variable, exits the node through the X (failure) branch, sets the **user.microapp.error_code** ECC variable to **16** (Reached Maximum Invalid Tries), and returns control to Unified ICME.

If No Entry Timeout Occurs

If the caller does not enter a digit within the No Entry Timeout period:

- If the Number of No Entry Tries value is reached, Unified CVP plays the "no entry" error message and re-plays the menu prompt.
- If the Number of No Entry Tries value has been reached, Unified CVP exits the node through the X (failure) branch, sets the Call.CallerEnteredDigits variable to NULL, the **user.microapp.error_code** ECC variable to **17** (Reached Maximum No Entry Tries), and returns control to Unified ICME.

Get Speech Micro-Application

The Get Speech (GS) micro-application collects input that can be DTMF-only, Speech, or both input modes, after prompting a caller. The prompt can be generated by a media file or a TTS source.

Note The Get Speech (GS) micro-application collects voice and DTMF input from the caller. Get Speech supports SRGS and built-in grammars with the exception of the "Digit" grammar which is handled by GetDigit. Use the ICM Configuration Manager's Network VRU Script List tool's Attribute tab to specify parameters. The prompt can be generated by a media file or a TTS source.

Unified CVP passes the input back to Unified ICME for further processing using the **user.microapp.caller_input** ECC variable.

Get Speech and Grammar Specification

There are three ways to specify a grammar in the Get Speech micro-application:

- Include a **Type of Data to Collect** setting in the Get Speech Configuration Param field for built-in grammars such as dates and numbers. If the "Type of Data to Collect" setting is specified, the other grammar options are not used by the IVR Service. Conversely, if you do not specify a "Type of Data to Collect" setting, then you must include either an inline or external grammar.
- Include an external grammar file name in the Get Speech Configuration Param field's "External Grammar File Name" setting.
- Include a list of inline grammar choices in the **user.microapp.grammar_choices** ECC variable. These grammar choices only used if a "Type of Data to Collect" or "External Grammar File Name" setting is not specified.

Note One of these grammar options must be used for each micro-application. If no grammar option is specified, an Invalid Config Param error is sent back to Unified ICME.

Note If you are using an external grammar, be sure to follow the instructions provided by your third-party vendor.

To write an external grammar file, see External VoiceXML File Contents.

Note For the following table, the Configuration Param field is not used if you are using external VoiceXML.

Configure Network VRU Script Settings for the Get Speech Micro-Application

Use the ICM Configuration Manager's Network VRU Script List tool's Attribute tab to specify parameters.

Procedure

Step 1 Configure VRU Script field parameters:

- **Micro-application type**. For Get Speech, valid options are: **GS** or **gs**.
- **Media File Name**. Name of the media file or external VoiceXML to be played (that is, the prompt file). The valid options are:
  - A file name (for instance, a .wav file)

    Note The file name is case-sensitive.
  - **null** - (default) If this field is empty, Unified CVP examines the contents of the **user.microapp.inline_tts** ECC variable. If this ECC variable contains a value, Unified CVP prompts using TTS. If the ECC is empty, no prompt is played.
  - **-(number 1-10)** - Unified CVP plays the file in the corresponding Call.PeripheralVariable file. For example, entering -2 causes Unified CVP to look at Call.PeripheralVariable2.

  Note If you use the -(number 1-10) option and set the Media Library Type to "V," Unified CVP plays the external VoiceXML file specified in the corresponding Call.PeripheralVariable. If you set the value to - (no value) and set the Media Library Type to "A" or "S", the IVR Service creates VoiceXML without a media prompt.
- **Media Library Type**. Flag indicating the location of the media files to be played. The valid options are:
  - **A** - (default) Application
  - **S** - System
  - **V** - External VoiceXML. Refer to "Get Speech and External VoiceXML."

  Note This value is ignored if using TTS.
- **Uniqueness value**. Optional. A string identifying a VRU Script Name as unique.

Step 2 Configure the Configuration Param field parameters:

Note This field does not apply if you are using external VoiceXML. For example, if you are using external VoiceXML, all Configuration Param settings (such as barge-in) will be allowed whether they're set to Y or N.

- **Type of Data to Collect**. A flag indicating the location of the media files to be played. The valid options are:
  - **null** - (default) Leave this option empty if you will be specifying an External Grammar File Name setting.
  - **boolean** - Affirmative and negative phrases appropriate to the current locale.
  - **date** - Phrases that specify a date, including a month, days and year.
  - **currency** - Phrases that specify a currency amount.

    Note Nuance 8.5 ASR does not support negative currencies in its built-in grammar of datatype "currency."

- **number** - Phrases that specify numbers. (For example, "one hundred twenty-three.")
- **time** - Phrases that specify a time, including hours and minutes.

Note For information about the format of the currency data returned to Unified ICME in the **user.microapp.caller_input** ECC variable, refer to "Get Speech Data Format."

- **External Grammar File Name**. The name of the grammar file that holds the grammar definition for the ASR. The valid options are:
  - **null** - (default) Leaving this option empty implies that an inline grammar, as given in the Type of Data to Collect setting, is used.
  - A grammar file name. The Gateway retrieves the grammar file from a Web Server using HTTP.

Note The file name is case-sensitive.

Note For more information about the "Type of Data to Collect" and "External Grammar" settings, see Get Speech and Grammar Specification.

- **Barge-in Allowed**. Specifies whether barge-in (digit entry to interrupt media playback) is allowed.

  The valid options are:
  - **Y** - (default) barge-in allowed
  - **N** - barge-in not allowed

Note Unified CVP deals with barge-in as follows: If barge-in *is not* allowed, the Gateway continues prompt play when a caller starts entering input. If barge-in *is* allowed, the Gateway discontinues prompt play when the caller starts entering input. See Get Speech and External VoiceXML.

- **No Entry Timeout** . The number of seconds a caller is allowed to begin entering digits. If exceeded, the system times-out. The valid options are: **0-99** (the default is **5**).
- **Number of No Entry Tries** Unified CVP repeats the “Get Digits” cycle when the caller does not enter any data after the prompt is played. (Total includes the first cycle.) The valid options are: **1-9** (default is **3**).
- **Number of Invalid Tries** Unified CVP repeats the prompt cycle when the caller enters invalid data. (Total includes the first cycle.) The valid options are: **1-9** (default is **3**).
- **Timeout Message Override** . The valid options are:
  - **Y** - override the system default with a pre-recorded Application Media Library file
  - **N** - (default) do not override the system default

Note This value is ignored if using TTS.

- **Invalid Entry Message Override**. The valid options are:
  - **Y** - override the system default with a pre-recorded Application Media Library file.
  - **N** - (default) do not override the system default

Note This value is ignored if using TTS.

Note For more information about Timeout and Invalid Entry Messages, see System Media Files.

- **Incomplete Timeout**. The amount of time after a caller stops speaking to generate an invalid entry error because the caller input does not match the defined grammar. The valid options are: **0-99** (default is 3).

  Note This value is ignored when not using ASR. If the value is set to 0, the IVR Service treats the NoEntry Timeout as NoError

- **Inter-digit Timeout** . The number of seconds the caller is allowed between entering DTMF key presses. If exceeded, the system times-out. The valid options are: **1-99** (the default is 3).

  Note This value is ignored if using ASR.

- **Pass FTP Information** Specifies whether to pass FTP server information to the VXML Server. This option is only useful if the VXML Server application uses the FTP_Client Element and the FTP server information is already configured using the Operations Console. Valid options are:

  - **Y** - Pass FTP server information to the VXML Server as VXML Server session variables.
  - **N** - (default) Do not pass FTP server information.

  If the **Pass FTP Information** parameter is set, the following information is passed:

  - **ftpServer** - A space separated string of FTP servers. For example, `ftp_host1|21|username|password ftp_host2`. Everything is optional except the host name. See FTP_Client Element settings located in the *Element Specifications for Cisco Unified CVP VXML Server and Cisco Unified Call Studio* guide for more information.
  - **ftpPath** - path on the FTP server. By default, this path is formed from the content of the ECC variable `user.microapp.locale` concatenated with path separator (`/`) and the content of the ECC variable `user.microapp.app_media_lib`. One exception is if the value of user.microapp.app_media_lib is `..`, then `app` is used instead. An example of a path is: `en-us/app`

Get Speech Configuration Examples

The following table shows several configuration examples for Get Speech.

Table 16: Get Speech Configuration Examples

If...	It means...	And, if...	It means...
The **user.microapp. inline_tts** ECC variable contains "What is your account value" and **user.microapp. input_type** contains: **B** (the default, both DTMF and voice) and The VRU Script Name field contains: **GS**	Use the Get Speech micro-app to play the contents of the **user.microapp.inline_tts** ECC variable and collect account balance in voice or DTMF input, which it passes in the **user.microapp.caller_input** ECC variable.	The Configuration Param field contains: **Currency,,N,5,2,1**	**Currency** - Collect a string of data in currency format , - Accept the default External Grammar File Name setting (empty). **Note** You accept the default because you are specifying a Type of Data to Collect parameter (Currency). **N** - No barge-in allowed. **5** - No Entry Timeout (seconds) **2** - Number of no entry tries **1** - Number of invalid tries

If...	It means...	And, if...	It means...
The **user.microapp. inline_tts** ECC variable contains “What department do you wish to speak to” and **user.microapp. input_type** contains: **B** (the default, both DTMF and voice) and **user.microapp. grammar_choices** contains **Sales/ Customer Service/ Help Desk** and The VRU Script Name field contains: **GS,Department,A**	Use the Get Speech micro-app play the contents of the **user.microapp.inline_tts** ECC variable and collect the answer, which will be passed in either voice or DTMF format in the **user.microapp.caller_input** ECC variable	The Configuration Param field contains: **,, N**	, - Accept the default Type of Data to Collect parameter (empty) **Note** You accept the default Type of Data to Collect parameter because you are specifying a value in the External Grammar File Name parameter. , - Accept the default External Grammar File Name setting (empty) and use the grammar in the **user.microapp. grammar_choices** ECC variable. **Note** This is an inline grammar example. Each option in the list of choices specified in the **user.grammar_choices** ECC must be delimited by a forward slash (/). **N** - No barge-in allowed.

If...	It means...	And, if...	It means...
The **user.microapp. inline_tts** ECC variable contains "What is your name" and **user.microapp. input_type** contains: **B** (the default, both DTMF and voice) and **user.microapp.media_server** contains `http://grammars.com` and **user.microapp.app_media_lib** contains **Boston** and The VRU Script Name field contains: **GS,YourName,A**	Use the Get Speech micro-app play the contents of the **user.microapp.inline_tts** ECC variable and collect the answer, which will be passed in either voice or DTMF format in the **user.microapp.caller_input** ECC variable.	The Configuration Param field contains: **,customers.grxml,N**	, - Accept the default Type of Data to Collect parameter (empty) **Note** You accept the default Type of Data to Collect parameter because you are specifying a value in the External Grammar File Name setting parameter. **customers.grxml** - Use the grammar in this file **Note** This is an external grammar file example. For details on writing an external grammar file, refer to External VoiceXML File Contents **N** - No barge-in allowed.

Note Get Speech sets the ECC variable **user.microapp.error_code** to zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. If control proceeds out the X (failure) branch, Get Speech typically sets this variable to one of the codes listed in Unified CVP Script Error Checking.

Get Speech and DTMF Input Collection

Contrary to its name, the Get Speech micro-application can also be used to collect DTMF input. For certain grammars, the caller could type a number, time, or currency rather than saying it.

Although the Get Digits micro-application is capable of providing the same type of functionality, it does not allow for validation at collection time. If a caller inputs 2 5 0 0 in response to a Get Speech prompt prompting the caller to enter a time, the Get Speech micro-application detects that "twenty-five hundred hours" is an invalid entry. With the Get Digits micro-application, this kind of validation is done using additional Script Editor nodes.

Note The caller cannot mix DTMF and speech in a single input, even if both are enabled. Once the caller starts talking, they cannot key-in characters, and vice versa.

The following table lists the rules associated with using DTMF collection in the Get Speech micro-application.

Table 17: DTMF Rules for Get Speech

Type of Data To Collect (as specified in the Config Params	Allows DTMF Input?	DTMF Rules
boolean	Yes	Valid DTMF inputs are: **1** (Yes) and **2** (No).
date	Yes	Valid DTMF inputs are: four digits for the year, followed by two digits for the month, and two digits for the day.*
currency	Yes	For DTMF input, the * (asterisk) key represents the decimal point.*
number	Yes	Valid DTMF input includes positive numbers entered using digits and the * (asterisk) key to represent a decimal point.
time	Yes	Since there is no DTMF convention for specifying AM/PM, in the case of DTMF input, the result will always end with **h** or **?**
External Grammars	No	None
Inline Grammars	No	None

Note Regardless of the collection type ("voice" or "DTMF"), caller input from Get Speech is always written to the **user.microapp.caller_input** ECC variable.

Get Speech Data Format

The data type determines the format of the information returned to Unified ICME in the user.microapp.caller_input ECC variable:

- **Boolean**. Returned to Unified ICME as "true" or "false."
- **Date**. Returned to Unified ICME as a fixed-length date string with the format *yyyymmdd* where *yyyy* is the year, *mm* is the month, and *dd* is the day.

- **Currency**. Returned to Unified ICME as a string with the format *UUUmmm.mm* or *mmm.mm*, where *UUU* is the three-character currency indicator (for example, USD), and *mmm.mm* is the currency amount with a decimal point.

Note Whether *UUU* is used depends on the ASR capabilities and on whether the caller said it unambiguously (for example, "dollar" and "dinar" are ambiguous, and so the UUU segment is not included in the return value).

- **Number**. Returned to Unified ICME as a string of digits from 0 to 9 which can optionally include a decimal point and/or a plus or minus sign as a prefix to indicate that the number is positive or negative.
- **Time**. Returned to Unified ICME as a five-character string in the format hhmmx, where hh is hours, mm is minutes and x is one of the following:

  a - AM

  p - PM

  ? - unknown/ambiguous

Get Speech and Entry Completion

The ASR Engine tests Get Speech input entry against two conditions to determine which entry is complete:

- If a caller enters input, the ASR Engine checks whether the input is within the set of valid grammar for this script.
- If a caller *does not* enter input, the ASR Engine verifies that the No Entry Timeout value has been reached.

Caution It is important that you set up your Unified ICME script to test for the scenarios that follow.

Input Entry Is Complete

After a caller enters input, the ASR Engine validates the input against the valid grammar set that was defined using the Set node to define the user.microapp.grammar_choices ECC variable.

Then Unified CVP does the following:

- If the input is valid, Unified CVP stores the input in the user.microapp.caller_input ECC variable, exits the node through the Checkmark (success) branch, and returns control to Unified ICME.
- If the input is not valid, Unified CVP treats it as an invalid entry and does the following:
  - If the Number of Invalid Entry Tries value *has not* been reached, Unified CVP plays the "invalid message" file and re-plays the menu prompt.
  - If the Number of Invalid Entry Tries value *has* been reached, Unified CVP stores the last-entered invalid digit in the **user.microapp.caller_input** variable, exits the node through the X (failure) branch, sets the **user.microapp.error_code** ECC variable to **16** (Reached Maximum Invalid Tries), and returns control to Unified ICME.

Note See Unified CVP Script Error Checking.

No Entry Timeout Occurs

When the caller does not enter input within the No Entry Timeout period and:

- The Number of No Entry Tries value has not been reached, Unified CVP plays the "no entry" error message and re-plays the prompt.
- The Number of No Entry Tries value *has* been reached, Unified CVP exits the node through the X (failure) branch, sets the **user.microapp.caller_input** ECC variable to NULL, the **user.microapp.error_code** ECC variable to **17** (Reached Maximum No Entry Tries), and returns control to Unified ICME.

Get Speech and External VoiceXML

You can use the Get Speech micro-application to pass information to and from an external VoiceXML file. The following table describes how to set the Get Speech script to use external VoiceXML.

To set up the Get Speech micro-application to use external VoiceXML, set the Media Library Type to "V". The IVR Service creates VoiceXML that calls the external VoiceXML that is specified in the external VoiceXML file name. The URL to the external VoiceXML is formed from a combination of the media_server, locale, App_Media_Lib and external VoiceXML file name. If the VoiceXML file name does not contain a file extension, the default "*.VoiceXML" is used.

If the external VoiceXML is used, the only GetSpeech VRU Script parameters that are used are:

- "Number of Invalid Entry" errors, and
- "Number of No Entry" errors.

The IVR Service "NoEntry" and "InvalidEntry" retry logic are used if the external VoiceXML returns a <noinput> or <nomatch> event.

Error Handling

Error handling
The error handling for an external VoiceXML called from the Get Speech micro-application includes the following:

- If you set the "Media Library Type" to "V" and you do not set an "External VoiceXML Name" parameter, an "Invalid VRU Script Name" error is returned to Unified ICME.

Passing Information to the External VoiceXML

There are two methods of passing information to the external VoiceXML, either by <param> elements or URL elements. You can pass up to 1050 characters to the external VoiceXML by using an ECC Variable array.

Table 18: To External VoiceXML ECC Variable Array

ECC Variable Name	Type	Max. Number of Elements	Max. Size of Each Element
user.microapp.ToExtVXML	Array	5	210

This variable array contains a list of semicolon delimited name/value pairs. The following is an example of the syntax:

Table 19: Sample Array Definition

Variable Name	Values
user.microapp.ToExtVXML[0]	"Company=Cisco;Job=technical writer"
user.microapp.ToExtVXML[1]	"Location=Boxborough;Street=Main"
user.microapp.ToExtVXML[2]	"FirstName=Gerrard;LastName=Thock"
user.microapp.ToExtVXML[3]	"Commute=1hour;Car=Isuzu"
user.microapp.ToExtVXML[4]	"BadgeID=2121212"

Unified CVP links all five elements of the "ToExtVXML" array, parses the contents, and then puts each of the name/value pairs in the VoiceXML that it creates.

You define an ECC variable to determine which method to use when passing information to the external VoiceXML.

Table 20: Use External VoiceXML ECC Variable

ECC Variable Name	Type	Max. # of Elements	Possible Values
user.microapp.UseVXMLParams	Scalar	1	• **Y** - (Yes) Use the values in the user.microapp.ToExtVXML variable array elements. • **N** - (No) Append the name/value pairs in user.microapp.ToExtVXML to the URL of the external VXML. Default: "N"

- If **user.microapp.UseVXMLParams** is set to a value other than "Y" or "N," the IVR Service sends a Misconfigured ECC Variable error message to Unified ICME.
- If the **user.microapp.UseVXMLParams** variable is not set, the default method is "VXML Parameters."

- If the **user.microapp.UseVXMLParams** is set to its default value "N," the .vxml file extension is not used.

Using <param> Elements

All name/value pairs declared in the **user.microapp.ToExtVXML** variable array are added to the VoiceXML file that the IVR Service creates in <param> elements. These <param> elements pass information to the external VoiceXML. Unlike the URL parameters, VoiceXML parameters are a complete VoiceXML solution that does not require media server side scripting.

You must declare names specified in the name/value pairs as variables in the external VoiceXML. Otherwise the VoiceXML Interpreter produces a error.semantic error that is returned to Unified ICME as error code "10." Using the example above in the table Table 19: Sample Array Definition, on page 155, you must define the following form level declarations in the external VoiceXML.

```
<var name="Company"/>
<var name="Job"/>
<var name="Location"/>
<var name="Street"/>
<var name="FirstName"/>
<var name="LastName"/>
<var name="Commute"/>
<var name="Car"/>
<var name="BadgeID"/>
```

This is why that Unified CVP does not allow the Script Writer to specify a name without a value. Specifying only a name in a ToExtVXML parameter is useless because the external VoiceXML already has the variable defined.

URL Parameter Element

When you use the URL parameter element option, the name/value pairs that you set in the **user.microapp.ToExtVXML** parameter are appended to the URL to the external VoiceXML. The media server side scripting logic parses the URL and passes the parameters to the external VoiceXML document. Unlike the “VXML Parameters,” this is not a VoiceXML solution and requires media-server side scripting.

Using the examples in Table 19: Sample Array Definition, on page 155, the URL to the external VoiceXML is in the following form:

```
http://server/en-us/app/MyVXML?Company=Cisco&Job=technical+writer&
Location=Boxborough&Street=Main&FirstName=Gerrard&LastName=Thock&
Commute=1+hour&Car=Isuzu&BadgeID=2121212
```

ECC Variable Array Formula

Use the following equation to determine how many bytes you are sending to and from the external VoiceXML. This calculation ensures that you do not overload the ECC Variable with too many bytes.

5 + (1 + Maximum_Length) * (Maximum_Array_Size)

For example, if you are sending three array elements to the external VoiceXML of the maximum 210 bytes, the equation looks like this:

```
5 + (1 + 210) * 3
5 + (211 * 3)
5 + 633
638
```

Although the maximum number of bytes you can set each variable to is 210, maxing out each variable goes over the 1050 character limit. Make sure you use this formula to keep your character limit under the 1050 maximum.

Use a single 210-byte array element in each direction.

Note When you are returning a call from the external VoiceXML, the user.microapp.caller_input variable is automatically returned.

Notes

- If the **user.microapp.ToExtVXML** array is either not defined on Unified ICME or empty, the IVR Service does not pass any parameters to the external VoiceXML.
- You do not need to define the **user.microapp.ToExtVXML** array to contain five elements. However, it must be defined as an ARRAY variable, not a SCALAR variable, even if you are using only one element. The IVR Service can handle five array elements.
- If the **user.microapp.ToExtVXML** is either undefined or partially defined, a warning message appears on the VRU PIM console window.
- Although the array elements are linked together, you can't span a name/value pair to multiple array elements. This is because before you parse for the name/value pairs, the IVR Service inserts a semicolon between two array elements if there is not one.
- The IVR Service produces a "Misconfigured ECC Variable" error if there is not a "=" symbol between two semicolons.
- The IVR Service produces a "Misconfigured ECC Variable" error if the "=" symbol is the first or last character between two semicolons (for example, if there is not a name or a value).
- The IVR Service produces a "Misconfigured ECC Variable" error if the "name" part of the name/value pair contains a space.
- The IVR Service treats each of the name/value parameters as strings. The IVR Service does not check to see if the value parameter is an integer.

Passing Data Back to Unified ICME with External Voice XML

Unified CVP can return 1050 characters for external VoiceXML.

Note All other Get Speech nodes are limited to the 210 characters returned in **user.microapp.caller_input**.

The following ECC Variable array is added:

Table 21: From External VoiceXML ECC Variable Array

ECC Variable Name	Type	Max. Number of Elements	Max. Size of Each Element
user.microapp.FromExtVXML	Array	4	210

The Get Speech micro-app returns up to 1050 characters by populating the **user.microapp.caller_input** variable and each element of the **user.microapp.FromExtVXML** array.

Note Use the ECC Variable array formula when defining the FromExtVXML variable. Do not define the user.microapp.FromExtVXML array to contain 5 elements. However, it must be defined as an ARRAY variable, not a SCALAR variable, even if you are using only one element.

Voice XML Requirements

External VoiceXML is called via the <subdialog> VoiceXML element. Because of the design of VoiceXML, the variable names in the customer defined VoiceXML must be coordinated with Unified CVP. Otherwise, it won't be possible to pass external VoiceXML data back to Unified CVP. The following table lists the VoiceXML variables and the ECC variables they correspond to.

Table 22: From External VoiceXML ECC Variable Definition

External VoiceXML Variable Name	Unified ICME ECC Variable	Max. Variable Size
caller_input	user.microapp.caller_input	210
FromExtVXML0	user.microapp.FromExtVXML[0]	210
FromExtVXML1	user.microapp.FromExtVXML[1]	210
FromExtVXML2	user.microapp.FromExtVXML[2]	210
FromExtVXML3	user.microapp.FromExtVXML[3]	210

Define the following form level declarations in the external VoiceXML.

```
<var name="caller_input"/>
```

When passing the information back to Unified ICME, use the following syntax for caller input:

```
<assign name="caller_input" expr="input$.utterance"/>
```

If the external VoiceXML sets "input" to "sales" and "FromExtVXML2" to "stocks", the user.microapp.caller_input is set to "sales" and user.microapp.FromExtVXML[2] is set to "stocks".

Notes

- The variables declared in the customer VoiceXML must be named exactly as specified. VoiceXML is case-sensitive.
- The "caller-input" variable must be declared and used in the VoiceXML. The only way that it is acceptable to not populate this variable is if the external VoiceXML is returning an error event. Examples of error events include <badfetch>, <noinput>, and <nomatch>. If the "caller_input" variable is not set to a value and an error event is not generated, the Unified CVP assumes that a "No Entry" error occurred.
- The FromExtVXML variables are optional. You can use these variables if the "caller_input" variable is not sufficient. If you are not using this additional data then the ECC Variable array does not need to be defined.

- If the user.microapp.FromExtVXML ECC Variable array is undefined or partially defined, a warning message appears when the IVR Service starts up on the VRU PIM console window.
- If you set a “FromExtVXML” variable in the external VoiceXML, the user.microapp.FromExtVXML must be defined on Unified ICME. If it is not, the Unified CVP does not attempt to set the value and an error appears on the VRU PIM console window. However, no error appears in the Unified CVP log files.
- If you need more than 210 characters, but less than the full 1050 characters, you can declare the user.microapp.FromExtVXML array to be less than four elements long. If you do this, you only use the corresponding number of External VoiceXML Variables. For example, if you configure a two-element “FromExtVXML” ECC Variable array, you can utilize the “caller_input”, “FromExtVXML0” and “FromExtVXML1” VoiceXML variables.
- If the external VoiceXML sets a FromExtVXML variable to a value that is longer than the maximum ECC Variable length, an error message appears in the VRU PIM console window and the value is not set. No error appears in the Unified CVP log files.

Sample External VoiceXML Code

This section provides sample external VoiceXML code.

```
<?xml version=”1.0”>
<vxml version=”2.0”>
<var name="caller_input"/>
 <form id=”getcredit”>
<field name=”input”>
 <prompt>
 What is your credit card type?
 </prompt>
 <help>
 I am trying to collect your credit card type.
 <reprompt/>
 </help>
 <nomatch>
 <return event=”nomatch”/>
 </nomatch>
 <grammar src=”cctype.grxml” type=”application/srgs+xml”/>
 </field>
<field name=”FromExtVXML0”>
 <prompt>
 What is your credit card number?
 </prompt>
 <help>
 I am trying to collect your credit card information.
 <reprompt/>
 </help>
 <nomatch>
 <return event=”nomatch”/>
 </nomatch>
 <grammar src=”ccn.grxml” type=”application/srgs+xml”/>
</field>
<field name=”FromExtVXML1”>
 <grammar type=”application/srgs+xml” src=”/grammars/date.grxml”/>
 </prompt>
<help>
 I am trying to collect the expiration date of the credit card number you provided.
 <reprompt/>
 </help>
 <nomatch>
 <return event=”nomatch”/>
 </nomatch>
</field>
<block>
 <assign name="caller_input" expr="input$.utterance"/>
```

```
 <return namelist="caller_input FromExtVXML0 FromExtVXML1"/>
</block>
<catch event="telephone.disconnect.hangup">
 <return event="telephone.disconnect.hangup"/>
</catch>
<catch event="error.badfetch">
 <return event="error.badfetch"/>
</catch>
<catch event="error.semantic">
 <return event="error.semantic"/>
</catch>
<catch event = "error.unsupported.format">
 <return event="error.unsupported.format"/>
</catch>
<catch event = "error.unsupported.element">
 <return event="error.unsupported.element"/>
</catch>
<catch event="error.unsupported.language">
 <return event="error.unsupported.language"/>
</catch>
<catch event = "error.com.cisco.media.resource.unavailable.asr">
 <return event=" error.com.cisco.media.resource.unavaliable.asr"/>
</catch>
<catch event = "error.com.cisco.media.resource.unavalable.tts">
 <return event=" error.com.cisco.media.resource.unavailable.tts"/>
</catch>
<catch event = "error.com.cisco.media.resource.failure.asr">
 <return event=" error.com.cisco.media.resource.failure.asr"/>
</catch>
<catch event = "error.com.cisco.media.resource.failure.tts">
 <return event=" error.com.cisco.media.resource.failure.tts"/>
</catch>
<catch event = "error.com.cisco.media.resource">
 <return event=" error.com.cisco.media.resource"/>
</catch>
<catch event = "error">
 <return event="error"/>
</catch>
<form>
</vxml>
```

External VoiceXML File Contents

An external VoiceXML file must adhere to the following rules:

- It must not use the <transfer>, <exit> or <disconnect> elements. However, it can use the <Goto> and <submit> elements.
- It must have <return> elements at all exit points in the document.
- It must check for all error events and the "telephone.disconnect.hangup" event. Each event handler must have a <return> element that includes the "event" attribute.
- It must contain <catch> event handlers for all events thrown by the Gateway. These catch handlers can have their own customer-defined logic, but they *must* include the statements that are listed in the sample VoiceXML provided in the previous section.

The following External VoiceXML document example illustrates the contents of a VoiceXML document that follows these rules.

Note This example assumes that the VRU Script Name value is **PM,CustomerVXML,V**.

External VoiceXML Document Example

```
<?xml version="1.0"?>
<vxml version="2.0">
 <form id="CustomerVXML" scope="dialog">
    <catch event="error.com.cisco.callhandoff.failure">
   <return/>
    </catch>

  <object name="dummyobj" classid="builtin://com.cisco.callhandoff">
   <param name="return" expr="true" valuetype="data"/>
   <param name="app-uri" expr="'builtin://dummyobj'" valuetype="data"/>
   <prompt bargein="true">
    <audio src="http://192.168.1.20:80/en-us/app/Hello_World.wav" />
   </prompt>
   <filled>
    <return/>
   </filled>
  </object>

  <catch event="error.badfetch">
   <return event="error.badfetch"/>
  </catch>
  <catch event="error.semantic">
   <return event="error.semantic"/>
  </catch>
  <catch event = "error.unsupported.format">
   <return event="error.unsupported.format"/>
  </catch>
  <catch event = "error.unsupported.element">
   <return event="error.unsupported.element"/>
  </catch>
  <catch event="telephone.disconnect.hangup">
   <return event="telephone.disconnect.hangup"/>
  </catch>
  <catch event="error">
   <return event="error"/>
  </catch>
  <block>
   <return/>
  </block>
 </form>
</vxml>
```

Note For a complete explanation of VoiceXML file grammar format, refer to http://www.w3.org/TR/speech-grammar/. Also, consult the ASR Server user documentation for a list of supported grammar elements.

The example that follows illustrates another external grammar file that prompts callers for the state that they live in.

External Grammar file

```
<?xml version = 1.0?>
<grammar version= 1.0 root= action xml:lang= en-us >
<rule id= action scope= public >
<one-of>
<item> California </item>
<item> Arizona </item>
<item> Connecticut </item>
```

```
</one-of>
</rule>
</grammar>
```

After a caller responds with the state the caller lives in, the ASR Engine determines if the caller said **California**, **Arizona**, or **Connecticut**. If the caller said the name of one of these states, the text listed in the **<item>** element is passed to the IVR Service and, **ICM** software. If a caller responds with a name not included in this list, an invalid entry error is returned to the IVR Service.

Type-Ahead Support for ASR

Type-ahead support for ASR is only supported for DTMF when the Gateway is the client and input type is set to **D**.

Scripting for Unified CVP with Call Studio

You can use Call Studio to build sophisticated IVR applications which can then be loaded onto a VXML Server machine for execution.

To invoke a VXML Server application, create a Unified ICME routing script that

- Includes a user.microapp.ToExtVXML[0] ECC variable instructing the VoiceXML Gateway to interact with the VXML Server directly to execute the application
- Instructs the application to pass back results to Unified ICME

This section describes

- Call Studio and how to use it to pass data to Unified ICME
- How to integrate Call Studio scripts with Unified ICME scripts
- How to deploy Call Studio Scripts in Unified CVP

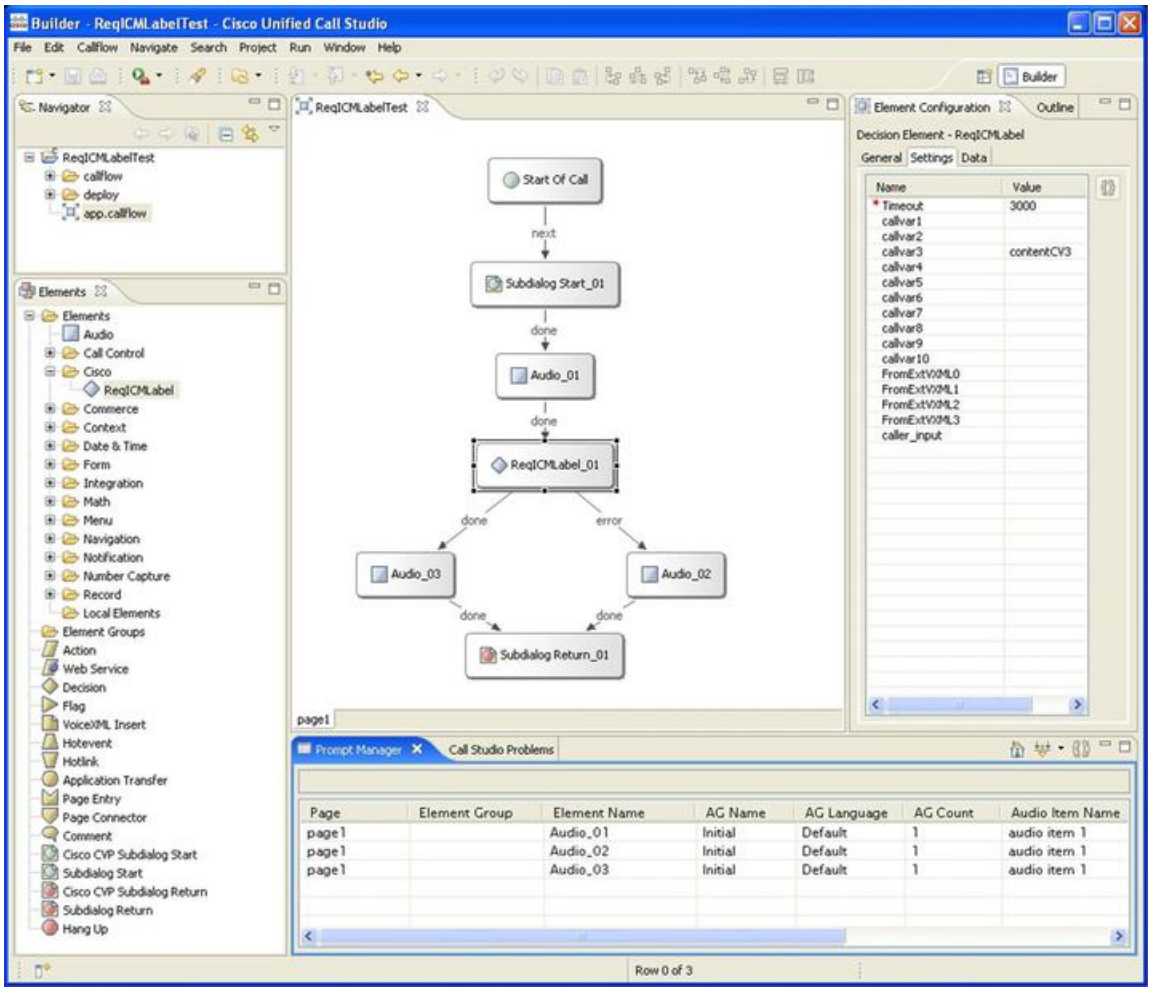
About Call Studio

Call Studio is an Eclipse-based service creation environment whose output is an intermediary file which describes the application flow.

Among its many features, the Call Studio scripting environment

- Has a drag-and-drop interface with a palette of IVR functions
- Can perform database queries
- Can be extended with Java code written to perform any task a Java application can perform

The following figure shows a Call Studio application that can be used with the Unified CVP Standalone with ICM Lookup call flow model. See High-Level Configuration Instructions.



Call Studio ReqICMLabel Element to Pass Data

The ReqICMLabel element allows a Call Studio script to pass caller input, Call Peripheral variables, and External Call Context (ECC) variables to a Unified ICME script. The ReqICMLabel must be inserted into a Call Studio script as a decision element. In Call Studio, the returned Unified ICME label result can be used by other elements in the same application, such as the Transfer or Audio element. The Transfer element sends instructions to the IOS Voice Browser to transfer the caller to the desired location.

After the ReqICMLabel exits its path, you can retrieve the values set by the Unified ICME script by selecting the Element Data tab for the ReqICMLabel element. The element data value is {Data.Element.ReqICMLabelElement.result}. ReqICMLabelElement is the name of the ReqICMLabel element in the Call Studio script. The default name for this element is ReqICMLabel_<n>. For example, if you changed ReqICMLabel to GetICMLabel, the value returned from Unified ICME is {Data.Element.GetICMLabel.result}, where *result* is the variable of the ReqICMLabel element that contains the Unified ICME label.

Table 23: Settings

Name (Label)	Type	Required	Single Setting Value	Substitution Allowed	Default	Notes
Call Peripheral Variables 1 - 10 (callvar1 - callvar10)	String	No	Yes	Yes		Call Peripheral variables passed by the Call Studio script to the Unified ICME Server. This setting can be a maximum of 40 characters. The Unified ICME Server returns a name-value pair for up to 10 Call Peripheral Variables in a result. Any value that is placed in callvar<n> from a Call Studio script is returned unchanged, if the Unified ICME Script does not change it.
Call Peripheral Variables Return 1 - 10 (callvarReturn1 - callvarReturn10)	String	No	Yes	Yes		Call Peripheral variables created upon the return of the Unified ICME Label request, regardless of whether or not these variables are filled by the Unified ICME Script. You need two sets of these variables to keep reporting the To ICM Call Peripheral Variables separate from what is returned from Unified ICME.
FromExtVXML0 - 3 (External VXML 0 - External VXML 3)	String Array	No	Yes	Yes		External Call Context (ECC) variables passed by the Call Studio script to the Unified ICME Server. Each variable is a string of name-value pairs, separated by semicolons, for up to four external VoiceXML variables. This setting can be a maximum of 210 characters.

Name (Label)	Type	Required	Single Setting Value	Substitution Allowed	Default	Notes
ToExtVXML0 - 4 (External VXML 0 - External VXML 4)	String Array	No	Yes	Yes		External Call Context (ECC) variables received from the Unified ICME script. The Unified ICME Server returns a string of name-value pairs, separated by semicolons, for up to five external VoiceXML variables.
Timeout	Integer	Yes	Yes	Yes	3000 (ms)	The number of milliseconds that the transfer request waits for a response from the Unified ICME Server before timing out. **Note** This value is increased or decreased by increments of 500 ms.
caller_input (Caller Input)	String	No	Yes	Yes		This setting can be a maximum of 210 characters. The caller_input is only passed to Unified ICME from Call Studio.

Table 24: Element Data

Name	Type	Notes
result	String	Unified ICME Label returned from a Unified ICME server. You can use this result as input to other Call Studio elements, such as Transfer or Audio. The element data value is {Data.Element.ReqICMLabelElement.result}.
callvar<*n*	String	Call Peripheral variables that the Call Studio scripts passes to the Unified ICME Server. Valid Call Peripheral Variables are callvar1 - callvar10.

Name	Type	Notes
callvarReturn<*n*	String	Call Peripheral variables that the Unified ICME script returns to the VXML Server. Valid Call Peripheral Variables are callvarReturn1 - callvarReturn10. For example, if a Unified ICME script contains Call Peripheral variable 3 with the string value “CompanyName=Cisco Systems, Inc”, you can access the value of CompanyName that is returned by the Unified ICME script by using Data.Element.ReqICMLabelElement.callvarReturn3 The returned value is “Cisco Systems, Inc.”

Table 25: Session Data

Name	Type	Notes
name	String	Value for a name-value pair contained in a ToExtVXML variable returned in the Unified ICME label. You must know which name-value pairs are set in the Unified ICME script to retrieve the correct value from the Call Studio script. For example, if a Unified ICME script contains a user.microapp.ToExtVXML0 variable with the string value “CustomerName=Mantle”, specify Data.Session.CustomerName. If the same Unified ICME script contains a user.microapp.ToExtVXML0 variable with the string value “BusinessType=Manufacturing”, you can access the customer business type returned by the Unified ICME script by using Data.Session.BusinessType.

Table 26: Exit States

Name	Notes
done	The element execution is complete and the value is successfully retrieved.
error	The element failed to retrieve the value.

Studio Element Folder is "Cisco."

Integrate Call Studio Scripts with Unified ICME Scripts - Traditional Method

This section describes how to integrate the VXML Server into the Unified CVP solution in the traditional way. This process involves

- Creating a Unified ICME script with ECC variables configured for VXML Server

- Creating a VRU Script to run in the Unified ICME script

Note There are two ways of integrating VXML Server into the Unified CVP solution. The first method is the same as in previous releases where ECC variables are used for specification of the parameters needed for the integration. This method is named as the 'traditional' method in the description below. Unified CVP continues to support the first method. The second method is described in the Integrate Call Studio Scripts with Unified ICME Scripts - Simplified Method.

Procedure

Step 1 Specify the URL and port number of the VXML Server that you want to reach, for example: http://10.78.26.28:7000/CVP/Server?application=HelloWorld

In the example above, 10.78.26.28 is the URL and 7000 port number; the values are delimited by a colon (:).

Note 7000 is the default port number for a VXML Server.

Step 2 In the Unified ICME script, first set the media_server ECC variable to: http://10.78.26.28:7000/CVP

Step 3 Set the app_media_lib ECC Variable to "..", (literally two periods in quotes).

Step 4 Set the user.microapp.ToExtVXML[0] ECC variable to application=HelloWorld.

Note This example indicates that the VXML Server will execute the "HelloWorld" application. To execute a different application, change the value of user.microapp.ToExtVXML[0].

Step 5 Set the UseVXMLParams ECC Variable to "N."

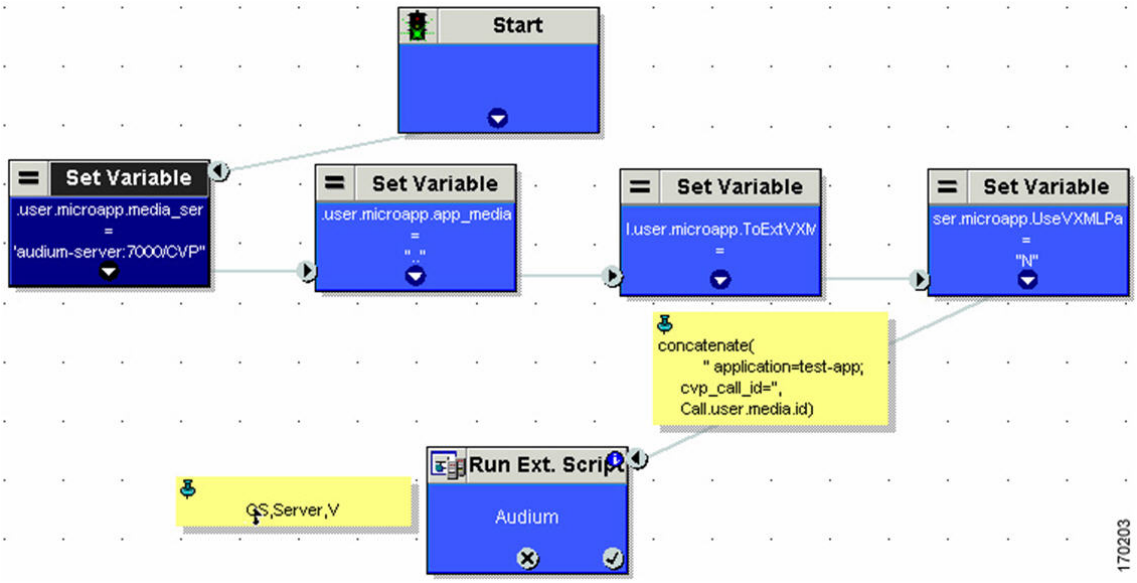
Step 6 Set the concatenate element by following the instructions in "Correlate Unified CVP and Unified ICME Logs with Unified CVP VXML Server Logs."

Step 7 Create a Run External Script node within the Unified ICME script with a VRU Script Name value of GS,Server,V.

Note Link this node to the nodes configured in previous steps.

- Configure the timeout setting in the Network VRU Script to a value substantially greater than the length of the timeout in the VXML Server application. (This timeout is only be used for recovery from a failed VXML Server.)
- Always leave the **Interruptible** check box in the Network VRU Script Attributes checked. Otherwise, calls queued to a VXML Server application may stay in the queue when an agent becomes available.

Figure 1: Example Script Configuration



After you configure the Unified ICME script, configure a corresponding VXML Server script with Call Studio. The VXML Server script must

- Begin with a Unified CVP Subdialog_Start element (immediately after the Call Start element)
- Contain a Unified CVP Subdialog_Return element on all return points (script must end with a Subdialog_Return element)
- Must include a value for the call input for the Unified CVP Subdialog_Return element
- Must add Data Feed/SNMP loggers to enable reporting

Integrate Call Studio Scripts with Unified ICME Scripts - Simplified Method

It is applicable only if the CVP Call Server and the VXML Server are co-located. This method simplifies the Unified ICME script configuration and reduces the script nodes that need to be configured. The GS micro-application assumes that it invokes the VXML Server that is co-located with the Call Server if the following conditions are met:

- The GetSpeech (GS) micro-application invokes a VXML Server application as follows:
  - The "Media File Name" of the GS is set to "Server" e.g., "GS,Server,V". (“Server” is case-sensitive).
  - - The "Media Library Type" of the GS is set to "V" (External VoiceXML).
- The ECC variable user.microapp.media_server has not been encountered in the Unified CCE script when the GS RunScript node is run.

With this method, the following ECC variables are not needed:

- user.microapp.media_server
- user.microapp.UseVXMLParams
- user.microapp.app_media_lib

Procedure

Step 1 Set the user.microapp.ToExtVXML[0] ECC variable to application=HelloWorld.

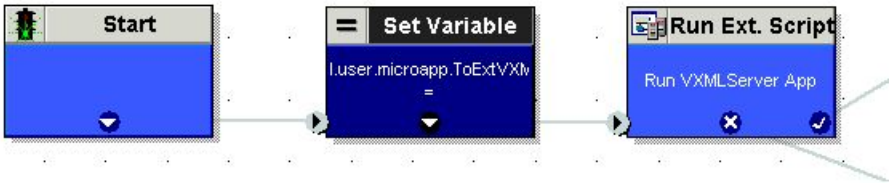
Note This example indicates that the VXML Server executes the "HelloWorld" application. To execute a different application, change the value of user.microapp.ToExtVXML[0] accordingly.

Step 2 Create a Run External Script node within the Unified ICME script with a VRU Script Name value of GS,Server,V.

- Configure the timeout setting in the Network VRU Script to a value greater than the timeout value in the VXML Server application. (This timeout is only used for recovery from a failed VXML Server.)
- Always leave the **Interruptible** checkbox in the Network VRU Script Attributes checked. Otherwise, calls queued to a VXML Server application may stay in the queue when an agent becomes available.

Step 3 After you configure the Unified ICME script, configure a corresponding VXML Server script with Call Studio. The VXML Server script must

- Begin with a Unified CVP Subdialog_Start element (immediately after the Call Start element)
- Contain a Unified CVP Subdialog_Return element on all return points (script must end with a Subdialog_Return element)
- Must include a value for the call input for the Unified CVP Subdialog_Return element
- Must add Data Feed/SNMP loggers to enable logging, as shown:



Call Studio Scripts in Unified CVP

Call Studio scripts can be deployed in one of the following ways:

- In Call Studio, create and deploy the Call Studio scripts to the local machine using the **Archive** option.
- In the Operations Console, upload the archived Call Studio script file from the local machine to the Operations Server and deploy it to other VXML Server machines.

Deploy Call Studio Scripts Using Call Studio

Procedure

Step 1 Create or modify one or more VoiceXML application scripts.

Step 2 Deploy one or more VoiceXML application scripts to the local machine using the archive option. The archived scripts are saved as a zipped file under a user-specified directory, for example: C:\Program Files\Cisco\CallStudio

Note The sample folder is C:\Cisco\CallStudio, which is also the default folder.

Step 3 Use Call Studio to set up the loggers using the ActivityLogger, ErrorLogger, and Admin Logger tools. Set up the Unified CVP Datafeed logger for each application.

Note Call Studio also includes CVPDatafeedLogger and CVPSNMPLogger. Call Studio lets you change other parameters for these loggers, such as log file size, log lever, et cetera.

See the Call Studio documentation for more information.

Deploy Call Studio Scripts Using the Operations Console

Procedure

Step 1 From the web browser, enter the following URL: https://ServerIP:9443/oamp or http://ServerIP:9000/oamp

Step 2 Enter your user ID in the User Name field.

Note The first time you log in after installing Unified CVP, enter Administrator, the default user account.

Step 3 In the Password field, enter your password, as follows:

- If you are logging in to the default Administrator account, enter the password that was set for this account during installation.
- If the user ID or password is invalid, the Operations server displays the message, "Invalid Username or password." Click the link, enter your user ID and password again, and click **OK**.

The Operations Console Welcome window appears.

Step 4 Select **Bulk Administration** > **File Transfer** > **Scripts and Media**.

Step 5 From the Device Association drop-down menu, select **Gateway**.

Step 6 In the Available pane, select one or more archived script files to deploy.

Step 7 Click the **arrow icon** to move the file from *Available* to **Selected**.

Step 8 Click **Transfer** to transfer the selected archived scripts file(s) to the selected device.

System Media Files

The following tables describe the English System Media Files installed by Unified CVP. These system media files are intended as samples only. It is the Customer/Media Administrator's responsibility to record all the system prompts for all the locales.

The table that follows lists the System Media File information for cardinal numbers.

Table 27: System Media Files, Cardinal Numbers

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		point	point	Number
		minus	minus	Number
0	48	0	zero	All except DOW
1	49	1	one (masculine version), uno (es-mx and es-es)	All except DOW
2	50	2	two	All except DOW
3	51	3	three	All except DOW
4	52	4	four	All except DOW
5	53	5	five	All except DOW
6	54	6	six	All except DOW
7	55	7	seven	All except DOW
8	56	8	eight	All except DOW
9	57	9	nine	All except DOW
		10	ten	Same for the rest of all the numbers
		11	eleven	
		12	twelve	
		13	thirteen	
		14	fourteen	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		15	fifteen	
		16	sixteen	
		17	seventeen	
		18	eighteen	
		19	nineteen	
		20	twenty	
		21	twenty-one	
		22	twenty-two	
		23	twenty-three	
		24	twenty-four	
		25	twenty-five	
		26	twenty-six	
		27	twenty-seven	
		28	twenty-eight	
		29	twenty-nine	
		30	thirty	
		31	thirty-one	
		32	thirty-two	
		33	thirty-three	
		34	thirty-four	
		35	thirty-five	
		36	thirty-six	
		37	thirty-seven	
		38	thirty-eight	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		39	thirty-nine	
		40	forty	
		41	forty-one	
		42	forty-two	
		43	forty-three	
		44	forty-four	
		45	forty-five	
		46	forty-six	
		47	forty-seven	
		48	forty-eight	
		49	forty-nine	
		50	fifty	
		51	fifty-one	
		52	fifty-two	
		53	fifty-three	
		54	fifty-four	
		55	fifty-five	
		56	fifty-six	
		57	fifty-seven	
		58	fifty-eight	
		59	fifty-nine	
		60	sixty	
		61	sixty-one	
		62	sixty-two	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		63	sixty-three	
		64	sixty-four	
		65	sixty-five	
		66	sixty-six	
		67	sixty-seven	
		68	sixty-eight	
		69	sixty-nine	
		70	seventy	
		71	seventy-one	
		72	seventy-two	
		73	seventy-three	
		74	seventy-four	
		75	seventy-five	
		76	seventy-six	
		77	seventy-seven	
		78	seventy-eight	
		79	seventy-nine	
		80	eighty	
		81	eighty-one	
		82	eighty-two	
		83	eighty-three	
		84	eighty-four	
		85	eighty-five	
		86	eighty-six	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		87	eighty-seven	
		88	eighty-eight	
		89	eighty-nine	
		90	ninety	
		91	ninety-one	
		92	ninety-two	
		93	ninety-three	
		94	ninety-four	
		95	ninety-five	
		96	ninety-six	
		97	ninety-seven	
		98	ninety-eight	
		99	ninety-nine	
		oh	oh	24TOD, Date
		hundred	hundred	Number, 24TOD, Date, Currency
		thousand	thousand	Number, Date, Currency
		million	million	Number, Currency
		billion	billion	Number, Date, Currency
		trillion	trillion	Number, Currency

The table that follows lists the System Media File information for ordinal numbers.

Note If ordinal system prompts are to be used in a script for a purpose other than dates, they should be recorded as application prompts with the true ordinal values.

Table 28: System Media Files, Ordinal Numbers

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		1ord	first	Date
		2ord	second	Date for all ordinal numbers
		3ord	third	
		4ord	fourth	
		5ord	fifth	
		6ord	sixth	
		7ord	seventh	
		8ord	eighth	
		9ord	nineth	
		10ord	tenth	
		11ord	eleventh	
		12ord	twelveth	
		13ord	thirteenth	
		14ord	fourteenth	
		15ord	fifteenth	
		16ord	sixteenth	
		17ord	seventeenth	
		18ord	eighteenth	
		19ord	nineteenth	
		20ord	twentieth	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		21ord	twenty-first	
		22ord	twenty-second	
		23ord	twenty-third	
		24ord	twenty-fourth	
		25ord	twenty-fifth	
		26ord	twenty-sixth	
		27ord	twenty-seventh	
		28ord	twenty-eight	
		29ord	twenty-nineth	
		30ord	thirtieth	
		31ord	thirty-first	

The table that follows lists the System Media File information for measurements.

Table 29: System Media Files, Measurements

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
½	189	one_half	one half	Char
¼	188	one_quarter	one quarter	Char
¾	190	three_quarters	three quarters	Char
A, a	65,97	a	A	Char
B,b	66,98	b	B	Char
C, c	67,99	c	C	Char
D, d	68,100	d	D	Char
E, e	69,101	e	E	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
F, f	70,102	f	F	Char
G, g	71,103	g	G	Char
H, h	72,104	h	H	Char
I, I	73,105	I	I	Char
J, j	74,106	j	J	Char
K, k	75,107	k	K	Char
L, l	76,108	l	L	Char
M, m	77,109	m	M	Char
N, n	78,110	n	N	Char
O, o	79,111	o	O	Char
P, p	80,112	p	P	Char
Q, q	81,113	q	Q	Char
R, r	82,114	r	R	Char
S, s	83,115	s	S	Char
T, t	84,116	t	T	Char
U, u	85,117	u	U	Char
V, v	86,118	v	V	Char
W, w	87,119	w	W	Char
X, x	88,120	x	X	Char
Y, y	89,121	y	Y	Char
Z, z	90,122	z	Z	Char
Œ, œ	140,156	oe_140_156	Ligature OE	Char
À,à	192,224	a_192_224	A grave	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
Á,á	193,225	a_193_225	A acute	Char
Â,â	194,226	a_194_226	A circumflex	Char
Ã,ã	195,227	a_195_227	A tilde	Char
Ä,ä	196,228	a_196_228	A umlaut	Char
Å,å	197,229	a_197_229	A with ring above	Char
Æ,æ	198,230	ae_198_230	Ligature AE	Char
È,è	200,232	e_200_232	E grave	Char
É,é	201,233	e_201_233	E acute	Char
Ê,ê	202,234	e_202_234	E circumflex	Char
Ë,ë	203,235	e_203_235	E umlaut	
Ì,ì	204,236	i_204_236	I grave	Char
Í, í	205,237	i_205	I acute	Char
Î,î	206,238	i_206	I circumflex	Char
Ï,ï	207,239	i_207	I umlaut	Char
Ð	208	char_208	character 208	Char
ð	240	char_240	character 240	
Ò,ò	210,242	o_210_242	O grave	Char
Ó,ó	211,243	o_211_243	O acute	Char
Ô,ô	212,244	o_212_244	O circumflex	Char
Õ,õ	213,245	o_213_245	O tilde	Char
Ö,ö	214,246	o_214_246	O umlaut	Char
x	215	multiply	multiplication sign	Char
Ø,ø	216,248	o_216_248	oh stroke	Char
Ù,ù	217,249	u_217_249	U grave	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
Ú,ú	218,250	u_218_250	U acute	Char
Û,û	219,251	u_219_251	U circumflex	Char
Ü,ü	220,252	u_220_252	U umlaut	Char
Ý,ý	221,253	y_221_253	Y acute	Char
Þ	222	char_222	character 222	Char
ß	223	ss	double s	Char
÷	247	divide	division sign	Char
þ	254	char_254	character 254	Char
Ÿ,ÿ	159,255	y_159_255	character 159 or 255	Char

The table that follows lists the System Media File information for month values.

Table 30: System Media Files, Months

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		January	January	Date
		February	February	Date
		March	March	Date
		April	April	Date
		May	May	Date
		June	June	Date
		July	July	Date
		August	August	Date
		September	September	Date
		October	October	Date

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		November	November	Date
		December	December	Date

The table that follows lists the System Media File information for month values.

Table 31: System Media Files, Days

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		Sunday	Sunday	DOW
		Monday	Monday	DOW
		Tuesday	Tuesday	DOW
		Wednesday	Wednesday	DOW
		Thursday	Thursday	DOW
		Friday	Friday	DOW
		Saturday	Saturday	DOW

The table that follows lists the System Media File information for month values.

Table 32: System Media Files, Time

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		hour	hour	Etime, 24TOD per locale, TOD per locale
		hours	hours	Etime,24TOD per locale,TOD per locale
		minute	minute	Etime
		minutes	minutes	Etime

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		second	second	Etime,24TOD
		seconds	seconds	Etime,24TOD
		on	on	per locale(unused for en-us)
		at	at	per locale(unused for en-us)
		am	am	TOD
		pm	pm	TOD
		oclock	oclock	TOD

The table that follows lists the System Media File information for currency values.

Note The customer's Media Administrator may prefer to replace the contents of "currency_minus" (for the negative amount) and "currency_and" (the latter can even be changed to contain silence).

Table 33: System Media Files, Currency

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		currency_ minus	minus	Currency
		currency_and	and	Currency
$	36	USD_dollar	dollar	Currency
		USD_dollars	dollars	Currency
		Note Unified CVP uses the USD_dollar.wav and USD_dollars.wav media files; the dollar.wav and dollars.wav used by ISN Version 1.0 are no longer installed.		
$	36	CAD_dollar	dollar	Currency
		CAD_dollars	dollars	Currency
		HKD_dollar	dollar	Currency

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		HKD_dollars	dollars	Currency
¢	162	cent	cent	Currency
		cents	cents	Currency
		euro	euro	Currency
£	163	GBP_pound	pound	Currency
		GBP_pounds	pounds	Currency
		penny	penny	Currency
		pence	pence	Currency
		MXN_peso	peso	Currency
		MXN_pesos	pesos	Currency
		centavo	centavo	Currency
		centavos	centavos	Currency

The table that follows lists the System Media File information for gaps of silence and miscellaneous phrases.

Table 34: System Media Files, Silence and Miscellaneous Phrases

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		silence_.1_ sec	(.1 second of silence)	Used for pauses where needed
		silence_.25_ sec	(.25 second of silence)	Used for pauses where needed
		silence_.5_ sec	(.5 second of silence)	Used for pauses where needed
		silence_1_sec	(1 second of silence)	Used for pauses where needed
		and	and	Etime,TOD,25TOD

The table that follows lists the System Media File information for ANSI characters.

Table 35: System Media Files, ANSI Characters

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
	32	space	space	Char
!	33	exclamation_ mark	exclamation mark	Char
"	34	double_ quote	double quote	Char
#	35	pound	pound	Char
%	37	percent	percent	Char
&	38	ampersand	ampersand	Char
'	39	apostrophe	apostrophe	Char
(	40	open_ parenthesis	open parenthesis	Char
)	41	close_ parenthesis	close parenthesis	Char
*	42	asterisk	asterisk	Char
+	43	plus	plus	Char
,	44	comma	comma	Char
-	45	hyphen	hyphen	Char
.	46	period	period	Char
/	47	slash	slash	Char
:	58	colon	colon	Char
;	59	semicolon	semicolon	Char
<	60	less_than	less than	Char
=	61	equal	equal	Char
	62	greater_than	greater than	Char
?	63	question_ mark	question mark	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
@	64	at_symbol	at	Char
[	91	left_square_bracket	left square bracket	Char
\	92	backslash	backslash	Char
]	93	right_square_bracket	right square bracket	Char
^	94	caret	caret	Char
_	95	underscore	underscore	Char
`	96	single_quote	single quote	Char
{	123	open_brace	open brace	Char
|	124	pipe	pipe	Char
}	125	close_brace	close brace	Char
~	126	tilde	tilde	Char
’	130	char_130	low single quote	Char
ƒ	131	char_131	F with hook	Char
”	132	low double quote	low double quote	Char
…	133	ellipsis	ellipsis	Char
†	134	char_134	character 134	Char
‡	135	char_135	character 135	Char
ˆ	136	char_136	character 136	Char
‰	137	per_mille	per mile	Char
Š	138	char_138	character 138	
<	139	left_pointing _angle	left pointing angle	Char
‘	145	left_single_ quote	left single quote	Char
’	146	right_single_ quote	right single quote	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
“	147	left_double_ quote	left double quote	Char
”	148	right_double _quote	right double quote	Char
·	149	bullet	bullet	Char
–	150	en_dash	en dash	Char
—	151	em_dash	em dash	
˜	152	small_tilde	small tilde	Char
™	153	trade_mark	trade mark	Char
š	154	char_154	character 154	Char
›	155	char_155	character 155	Char
¡	161	exclamation_ mark_ inverted	inverted exclamation mark	Char
¤	164	char_164	character 164	Char
¦	166	broken_pipe	broken pipe	Char
§	167	section	section	Char
¨	168	char_168	character 168	Char
©	169	copyright	copyright	Char
ª	170	char_170	character 170	Char
«	171	left_double_ angle_ quote	left double angle quote	Char
¬	172	not	not	Char
-	173	char_173	character 173	Char
®	174	registered	registered	Char
¯	175	char_175	character 175	Char
°	176	degree	degree	Char
±	177	plus_minus	plus or minus	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
²	178	superscript_ 2	superscript two	Char
³	179	superscript_ 3	superscript three	Char
´	180	acute_accent	acute accent	Char
µ	181	micro	micro	Char
¶	182	paragraph	paragraph	Char
·	183	middle_dot	middle dot	Char
¸	184	cedilla	cedilla	Char
¹	185	superscript_ 1	superscript one	Char
º	186	char_186	character 186	Char
»	187	right_double _angle_ quote	right double angle quote	Char
¿	191	question_ mark_ inverted	inverted question mark	Char

CHAPTER 4

Cisco Serviceability Tools

This chapter presents an overview of Cisco serviceability tools, including the tools available with Unified CVP solutions running on Windows environments. It also presents the serviceability tools that use the Web Services Manager (WSM).

- Web Services Manager, page 189
- Unified System CLI, page 190
- Configure Analysis Manager with Unified CVP, page 198
- System CLI Commands Map to IOS CLI Commands, page 199

Web Services Manager

Unified CVP supports a new service layer called the Web Services Manager (WSM). WSM interacts with various subsystems and infrastructure handlers, consolidates the responses, and publishes an XML result. The Web Services Manager supports HTTPS requests and sends a predefined XML response. WSM is installed on each Unified CVP device and runs automatically as a Windows service. For a device to be managed by WSM, the device must be deployed from the Operations Console.

Note System CLI uses WSM to collect and present the data available to WSM from the various Unified CVP components.

Create a WSM User

When Unified CVP is installed, a new user called wsmadmin is created with the same password as the Operations Console user. You can create and manage additional WSM users using the Operations Console.

When you have devices deployed in the Operations Console, log on to any server where WSM is installed and access the System CLI. See Unified System CLI.

Procedure

Step 1 Log into the Unified CVP Operations Console and select **User Management** > **Users**.

Step 2 Click **Add New**.

Step 3 Provide a **Username** and **Password**.

Step 4 Click the **User Groups** tab.

Step 5 In the **Available** panel, highlight **ServiceabilityAdministrationUserGroup**, then click the right arrow to move that group to the **Selected** panel.

Step 6 Click **Save**.

Unified System CLI

Unified CVP supports a new serviceability CLI called Unified System CLI (System CLI). The System CLI lets you collect diagnostic information (health and status) on Unified CVP servers and collect device-specific information from each supported node connected to the Unified CVP server from which you are using System CLI. The System CLI accesses a new web services layer in Unified CVP called the Web Services Manager. You can run System CLI commands on a local server, or a remote server. You can obtain information from all the devices in your CVP system by switching to *system* mode. (Devices must first be configured and deployed in the Operations Console.)

Note To quickly access and use the System CLI, see Access Unified System CLI and Its Help

Note In addition to the System CLI, which is automatically installed with the Unified CVP installation, you can also obtain a GUI-based client, . This client is called the Analysis Manager, and it is part of Unified CM. For more information on the Analysis Manager, refer to Cisco Unified Communications Analysis Manager User Guide. For instructions specific to configuring Analysis Manager with Unified CVP, see Configure Analysis Manager with Unified CVP, on page 198

The System CLI is installed on all CVP servers. You can leverage the WSM and CLI functionality to collect diagnostic details such as server map, version information, licenses, configuration, components, sessions, logs, traces, performance factors, and platform information for each Unified CVP Device, on a component and sub-component level. You can also set or reset debug levels using CLI on a component and sub-component level.

The System CLI provides a local mode and a system mode:

- The local mode accesses data about the devices associated with the server that you are logged into. Local mode is the default mode accessed automatically when you log into the Unified CLI.
- The system mode, accessed by typing the system command at the CLI prompt, provides access to all the devices in your Unified CVP deployment solution. In system mode, the System CLI automatically detects the Operations Console and extracts solution topology based on the devices configured in the Operations Console. Based on options you enter for a given command, System mode can be limited to a certain device group or list of servers.

Note Before you can use the system CLI to obtain information about a device, that device must be listed in and deployed by the Operations Console.

The System CLI commands (for example `show all` and `show component`) enable you to view and zip necessary logs or configurations on a specific server, servers, or groups of servers, and store that data on a local disk.

Access Unified System CLI and Its Help

The Unified System CLI is installed on all Unified CVP Servers. Using the CLI client on a Unified CVP device enables you to connect to the local server or a remote server. You can also connect to servers defined and deployed in the Operations Console when using system mode.

To launch Unified System CLI on any CVP server, log into a Unified CVP server through windows. You can use tools such as VNC or Remote Desktop console.

Note To be able to log in to the System CLI on Unified CVP, the WSM service must be up and running. By default WSM service is always running.

Note The System CLI only provides information on devices that have been configured, saved, and deployed in the Operations Console. If you change the configuration of a device, you must save and deploy the revised configuration before it is available to the System CLI.

Complete the following example session to quickly learn how to use Unified System CLI.

Procedure

Step 1 Launch Unified System CLI.
Select **Start** > **Programs** > **Cisco Unified Customer Voice Portal** > **Unified System CLI**.

A CMD window displays with an Enter Username prompt.

Step 2 Log into the Unified System CLI by entering the default username, **wsmadmin**, or a username and password that you created. See Create a WSM User.
After logging, in you see the following message and prompt:

```
Welcome to the Platform Command Line Interface
admin:
```

Step 3 You can now receive Web Services Manager data from the local machine using the system CLI commands. The following CLI Usage example shows a user issuing the show tech-support command.

```
Enter username[wsmadmin]: wsmadmin

Enter password:

Welcome to the Platform Command Line Interface
```

```
admin: show tech-support
Warning: Because running this command can affect system performance,
Cisco recommends that you run the command during off-peak hours.
Do you want to continue? [y/n]: y

Retrieving [version] data from device [localhost] ProductType [cvp] ...
Retrieving [component] data from device [localhost] ProductType [cvp] ...
Retrieving [log] data from device [localhost] ProductType [cvp] ...
Default time range is last 24 hours.
...
Output is saved to "C:\Cisco\CVP\wsm\CLI\download\clioutput0.zip"
```

Note The show tech-support command creates a single zip file in the directory *%CVP_HOME%\wsm\CLI\download*.

Note By default, this command collects the traces for the last 24 hours. Use the **reltime** parameter to change the time period. For example, if you want to pull traces for the last three days, use the following command: show tech-support reltime 3 days.

Step 4 Each System CLI command has a set of **command options**. Each option consists of a keyword and set of values. The following is an example of how to use a command with options.
In this example, the keyword *component* has a value *cvp:CallServer* and the keyword *subcomponent* has a value *cvp:ICM*. The following command tells the System CLI to get the configuration data for component *CallServer* and subcomponent *ICM* in the Unified CVP deployment.

```
admin: show config component cvp:CallServer subcomponent cvp:ICM
Downloading Configuration file: [ICM: icm.properties] ...
ICM.icmGarbageCollectorInterval : 120
ICM.locationDelimeter : --
ICM.icmHeartbeatInterval : 5000
...
ICM.preRoutedCallServiceID : 2
ICM.icmVxmlIdleTimeout : 30
```

Step 5 Almost all commands have a *redirect* option. This option tells the System CLI to redirect the command output to a directory (or a file). If the output is saved in a directory, it is saved in a zip file in that specific directory. The following example saves the zip file to **c:\temp\clioutput.zip**:

```
admin: show version redirect dir c:\temp
```

The output zip file provides a consistent directory structure when you save to a directory. When unzipped, the directory structure enables you to find the required diagnostic data quickly.

If you redirect the output to a file, the information is stored in the form a of "flat" file similar to what you see for the window output. An example of redirecting to a file is:

```
admin: show version redirect file <filename>
```

Step 6 To obtain information from the Web Services Managers that are part of your Unified CVP environment, you can change to *system* mode to issue system-wide commands.
To enable system mode, type *system* at the prompt.

Note See Unified System CLI.

Unified System CLI: System Mode

Enter the CLI system mode by typing **system** at the command prompt and then execute the commands exactly like the local version of the CLI for interactive mode.

In system mode, the Unified System CLI automatically detects the Operations Console, which acts as a seed device, and extracts the solution topology automatically based on devices configured in the Operations Console. System mode enables the System CLI to iteratively go to each supported box in the background and run the command that was executed by you in system mode.

Note System CLI initialization for the first system mode command execution, or for the **system init** command, may take a few minutes to complete, especially when there are a lot of devices in a Unified CVP solution, or if devices are unreachable, or network timeouts are involved.

Optionally, you can limit the system command to execute only on a certain device group or list of servers.

Device group is automatically populated based on:

- Device type (Unified CVP, Unified ICM, IOS Firewall , Unified CM as an example)
- Device IP/hostname wildcard (LOC-1*, 10.86.129.* as an example for branch office deployments)
- Device pool (defined within the Operations Console)

When the System CLI executes the *system* command, the System CLI queries each device in the list and caches the responses locally during its first time initialization process, or when **system init** is executed. The cache enables the system command to be executed quickly for subsequent sessions.

If an error is reported due to an unreachable destination or incorrect credentials for a specific device during the execution of a system command , then the device is marked OFFLINE in the System CLI cache. Use *system init* to retry this device.

The most common command to receive all information from all the components in a Unified CVP solution deployment is:

```
system show tech-support dtcomponent "ucm:Cisco CallManager|cusp:Cisco Unified SIP Proxy"
```

This command collects everything from all device types except the devices *ucm* and *cusp*. For ucm, it applies the device type filter *Cisco CallManager* and for the device *cusp* it applies the device type filter *Cisco Unified SIP Proxy*.

System CLI Automated Execution

To automatically execute Unified System CLI commands, create a plain text file with the .bat extension as shown in the example below and replace the <password> as highlighted with the Operations Console password.

```
 REM TECH-SUPPORT-COLLECTION
 echo show tech-support > clicmds.txt
 echo exit >> clicmds.txt
 type clicmds.txt | systemcli.bat inplace nointeractive novalidation user:wsmadmin
passwd:<password>
```

In a batch file, the system command can also be executed by prefixing *system* on any regular command. For example, **system show tech-support**, the entire example is:

```
REM SYSTEM-TECH-SUPPORT-COLLECTION
echo system show tech-support > clicmds.txt
echo exit >> clicmds.txt
```

```
   type clicmds.txt | systemcli.bat inplace nointeractive novalidation user:wsmadmin
passwd:<password>
```

The most commonly used command in a Unified CVP solution deployment, to get data from all solution components, is given below:

```
   REM SYSTEM-TECH-SUPPORT-COLLECTION
   echo system show tech-support dtcomponent "ucm:Cisco CallManager\|cusp:Cisco Unified SIP
 Proxy"> clicmds.txt
   echo exit >> clicmds.txt
   type clicmds.txt \| systemcli.bat inplace nointeractive novalidation user:wsmadmin
passwd:<password>
```

This command collects everything from all device types except for the device *ucm* and the device *cusp*. For the ucm device, it applies the device type filter *Cisco CallManager* and for the device cusp, it applies the device type filter *Cisco Unified SIP Proxy*.

Note You can run a Windows scheduled job and collect traces periodically from one or multiple servers using a schedule. See How to Schedule Tasks in Windows XP for information about scheduling a Windows job.

System CLI Remote Execution

To launch System CLI remotely from your laptop, (or any Windows system), to connect to any Unified CVP server or other solution component box (for example, Unified CM, ICM, IOS, CUSP Server, etc.), complete the following procedure:

1. Install Unified CVP Remote Operations, if no other CVP software is installed.

   For information about only installing the Remote Operations feature of Unified CVP, see Installation and Upgrade Guide for Cisco Unified Customer Voice Portal.

2. In the Operations Console, complete the following steps:

   a. Select **System** > **Web Services** > **Remote Operations Deployment (tab)**.

   b. Enter the IP address, hostname, and description information for the remote device.

   c. Click **Add** to add the device and click **Save & Deploy** to make this device available for remote operations.

3. Start the System CLI on the remote system.

Note You can also run a Windows scheduled job and collect traces periodically from one or multiple servers, using a schedule from a remote system.

Help for the System CLI

The following types of help are available:

- For an overview of the CLI system help, type help at the admin prompt and press enter.

  admin:**help<enter>**

- For a list of main commands, enter "?" at the admin prompt.

admin:**?<enter>**

- To obtain the syntax of the command, type a "?" at the end of the current syntax. For example:

  admin: **capture start ?**

  The preceding entry then informs you that there are two options in the syntax: **duration** and **<cr>**

- To obtain detailed help, enter a portion of the command preceded by the word help. For example:

  admin: **help show version**

  That command provides information about additional options both local and system modes.

The following entries show an example of "drilling down" to obtain more detailed help. At the option level, placing *help* in front of the command, provides detailed information, as in the example. Because this command can be used in "local" mode and in "system" mode, the detailed help also gives information about how to limit the command when using system mode.

```
admin: show version ?
Options: redirect
<cr>

show version redirect ?
Options: dir
file
```

The help system displays the actual components for your Cisco Unified product as shown in the following example for Unified CVP using system mode of the System CLI.

```
admin(system):show log component ?
Options:cvp:CallServer
cvp:OAMP
cvp:ORM
cvp:Reporting
cvp:VXMLServer
```

System CLI Troubleshooting

The System CLI shows two types of errors:

- Errors from servers (displayed unchanged)
- Errors from the System CLI for which you need to check the log files in the System CLI directory

Sometimes it is necessary to change the System CLI debug level to "debug" to collect more data about a System CLI error.

1. Open CLI\conf\cli_log4j.xml and change the word "info" to "debug".
2. Restart the System CLI to reproduce the error.

See the Doc Wiki troubleshooting page for more CLI troubleshooting information: System CLI troubleshooting tips.

System CLI Commands and Parameters

The Unified System CLI is designed to work across multiple Cisco Unified products. The meanings of some of the parameters for commands such as *show* vary from product to product. An example is the *components* parameter which vary because the components of systems such as Unified CVP and Unified ICM are different.

The CLI online help provides product-specific information for each command's parameters. Type the command and its parameter, followed by the "?" symbol. An example of this mechanism is given in Help for the System CLI.

The following table provides information about the Unified System CLI commands.

Command or Parameter	Description
capture	Sets up, starts, and stops, a network packet capture. Capturing network packets with the System CLI can be performed in either local mode (to run on a single machine), or system mode (to run across several machines simultaneously). The capture start command has an optional duration parameter, while the capture stop command has no parameters. The duration parameter indicates when the capture should stop. If no duration is provided, the capture process stops after one day. The capture command starts the packet capture on a single Unified CVP device or multiple devices. The capture operation defaults to the interface card that is receiving packets and is used for the Unified CVP server socket and IP address binding. The default setting of the capture command will capture the network packets and save the capture information in the Unified CVP logs folder which can be retrieved using the regular CLI trace command.
help	Accesses the online help system overview. Use the "?" character to access command-specific and parameter-specific help. See Help for the System CLI.
exit	Exits the CLI.
show	Accesses, displays and saves (to a file) data about system configuration and operation. See the next table for descriptions of the sub-level show commands.
system	Enter system mode, which provides access to all the devices in your Unified CVP deployment solution. Use the exit command to return to local mode.

The following table describes the variations of the *show* command.

Show Commands	Descriptions
show all	Shows all available information in all sub-categories listed in this table. However, you can still enter qualifying parameters to restrict the information retrieved.
show component	Shows component-specific information. Available components are based on the Cisco Unified product type and can be listed by entering the command: show component ?
show config	Displays the application configuration.

show debug	Shows the current debug level.
show devices	Shows a list of the devices in your deployment (system mode only).
show license	Shows license information.
show log	Shows log contents. You can narstrow this command to specific logs.
show perf	Shows system performance statistics.
show platform	Shows platform information.
show sessions	Shows the current active sessions or calls.
show tech-support	Shows information to help tech support in solving issues. This command is equivalent to show all. **Note** When you issue the show tech-support command, the System CLI issues the other show commands needed to provide all of the system information. During the process of executing the other show commands, the System CLI passes component and sub-component parameters to the show trace command, but does not include component and sub-component parameters when it executes commands such as show configuration and show log.
show trace	Shows trace file information.
show version	Shows Unified CVP component version information.

Details for Specific Options

This section provides detailed information of certain options that require additional explanations.

Results of the *match* Option when Information is Sent to a Directory Instead of a File

You can use the *match* option with the *show trace* and the *show log* commands to send selected output to a directory. This option enables you to specify text that the CLI should match when examining data in text files. The output is then limited to log information that matches the specified criteria. However, because the system cannot perform a text match to include or exclude information in binary files such as .zip files, these files are *included* in your *show* command execution, since they may contain pertinent information.

If you send the output of the *match* selection process to a file instead of a directory, the output only includes the actual text the CLI command would return to the screen.

Comparison of the *component* Option with the*dtcomponent*Option

The *component* limits the output of a command to the results from specific system components.

Using the command **show trace component ucm:CallManager** and the system components shown in the following list, you receive the following information:

- Unified CVP Call Server: No Information Returned
- Unified Call Manager: Information Returned
- Unified CM Tomcat Server: No Information Returned

The command **show trace component ucm:CallManager** only returns information from the Call Manager.

The option *dtcomponent* (device type component option) only restricts the information from the given device type, while allowing data to be returned from other device types.

Using the above system and the command **show trace dtcomponent ucm:CallManager** would restrict its *ucm* component output to just the Call Manager, but returns information for other device types. In this example, it returns the following information for the Unified CVP Call Server and the Call Manager, but not the Unified CM Tomcat Server:

- Unified CVP Call Server: **Information Returned**
- Unified Call Manager: **Information Returned**
- Unified CM Tomcat Server: **No Information Returned**

Configure Analysis Manager with Unified CVP

To configure Analysis Manager with Unified CVP, follow this procedure:

Procedure

Step 1 From the **Unified Analysis Manager** menu, choose **Inventory** > **Node**.
The Node window appears.

Step 2 Click **Add** to add a node or select a node from the list. Click **Edit** to edit an existing configuration.
The Add or Edit Node window appears.

Note Fields on this window that are marked with an asterisk (*) are required fields.

Step 3 From the Product Type drop-down list box, select **CVP**.

Step 4 In the IP/Host Name field, enter the host name or the IP address of the Unified CVP OAMP box.

Step 5 In the Transport Protocol field, select the protocol you want to use. Use the default value populated (HTTPS).

Step 6 In the Port Number field, enter the port number of the node you are using.

Note Use the default value populated (8111).

Step 7 Enter the user name as **wsmadmin** and enter the OAMP password. Re-enter the password in the Confirm Password field.

Step 8 In the Description field, you can optionally provide a brief description of the node you are adding.

Step 9 In the Associated Call Record Server and Associated Trace File Server fields, use the drop-down list to select the respective servers you want to use for the node.

Step 10 To add a node to an existing group, check the **Associated Group** check box.

Step 11 If you have a NAT or Terminal Server configuration, use the Advanced button to display the Add Node-Advanced screen. Enter the appropriate information in the Alternate IP/Hostname and Alternate Port fields.

Step 12 Click **Save** to save the node to the list. Click **Cancel** to end the operation without adding the node.

System CLI Commands Map to IOS CLI Commands

The following table maps the System CLI commands to their corresponding IOS CLI commands:

System CLI Commands	IOS CLI Commands
show config	show running-config
show version	show version show clock
show license	show license
show perf	show call resource voice stat show memory statistics show processes cpu history show processes memory sorted show voice dsp group all show voice dsp voice
show debug	show debug
show log	N/A
show sessions	show call active voice compact show voice call status | inc calls show voip rtp connections | inc connect sh sip-ua calls | inc calls
show tech-support	show tech-support <Everything else given above>
show trace	show logging
show platform	show diag sh inventory sh int | inc media type|Ethernet|address sh controllers T1 | inc T1 sh controllers E1 | inc E1
debug	0 no debug all 1 - deb ccsip err

	deb cch323 err deb voip app vxml err deb http client err deb mrcp err deb rtsp err deb h225 asn1 err deb h245 asn1 err 2 - debug isdn q931 debug h225 events debug h245 events debug voip ccapi inout debug vtsp events 3 - debug ccsip messages debug h225 q931 debug h225 asn1 debug h245 asn1

CHAPTER 5

Unified CVP Security

This chapter describes security considerations for Unified CVP deployments.

Note For detailed information about security issues in Unified ICME, see *Security Best Practices Guide for ICM and IPCC Enterprise Hosted Editions* .

- Secure Communications Between Unified CVP Components, page 201
- Secure Communications Between Unified CVP and IOS Devices, page 209
- Secure Communications Between VoiceXML Gateways and VXML Server, page 212
- HTTPS support for unified CVP, page 212
- Sensitive Customer Information, page 214

Secure Communications Between Unified CVP Components

During the configuration of a Unified CVP device - such as the Call Server, Reporting Server, or VXML Server - the Operations Console Server uses Java Management Extensions (JMX) to communicate to the managed Unified CVP device.

Note The Operations Console Server is the underlying software that manages component configurations. The Operations Console is a Web-based interface that you use to configure Unified CVP components. See Operations Console User Guide for Cisco Unified Customer Voice Portal.

Unified CVP installation uses a default JMX communications setting of *non-secured*, so communications are not encrypted. However, you can modify this setting to secure communications using Secure Sockets Layer (SSL).

Note Modifying this setting requires that you stop and restart services.

Before You Begin

Secure JMX communications by importing:

- Self-signed certificates that are created automatically from information that you specify during Unified CVP installation.
- Signed certificates available from a Certificate Authority (CA).

You manage certificates using:

- The *keystore*, a database for keys and trusted certificate information. For all keystore operations it is assumed that

  For Windows 2008 R2 server machines:

  - Keystore resides in: `%CVP_HOME%\conf\security\.keystore`
  - Resource Manager keystore resides in: `%CVP_HOME%\conf\security\.ormKeystore`
  - Keystore password resides in: `%CVP_HOME%\conf\security.properties`

- The *keytool*, a command-line utility for managing keys and trusted certificates. The keytool is installed in:
- For Windows 2008 R2 server machines:

  `Keytool resides in: %CVP_HOME%/jre`

Note On Windows systems, the keystore and the keytool passwords are in a folder that is protected with Access Control List (ACL), so only a user who has Administrator privileges can import trusted certificates.

Note For more information about the keytool and keystores, See your *Java documentation*.

Secure JMX Communications Between Unified CVP Components

Secure JMX communication by using SSL between the Unified CVP Operations Console service, a managed Unified CVP device, and other CVP-related JMX clients.

Procedure

Step 1 Stop the *Unified CVP Operations Console* service. On a *Windows* system, do the following:

1. On a **Windows** system, Select **Start** > **Administrative Tools** > **Services**.
2. The **Services** windows appears. In the list of Service names, highlight the Cisco services listed below.

   In Call Server Window system

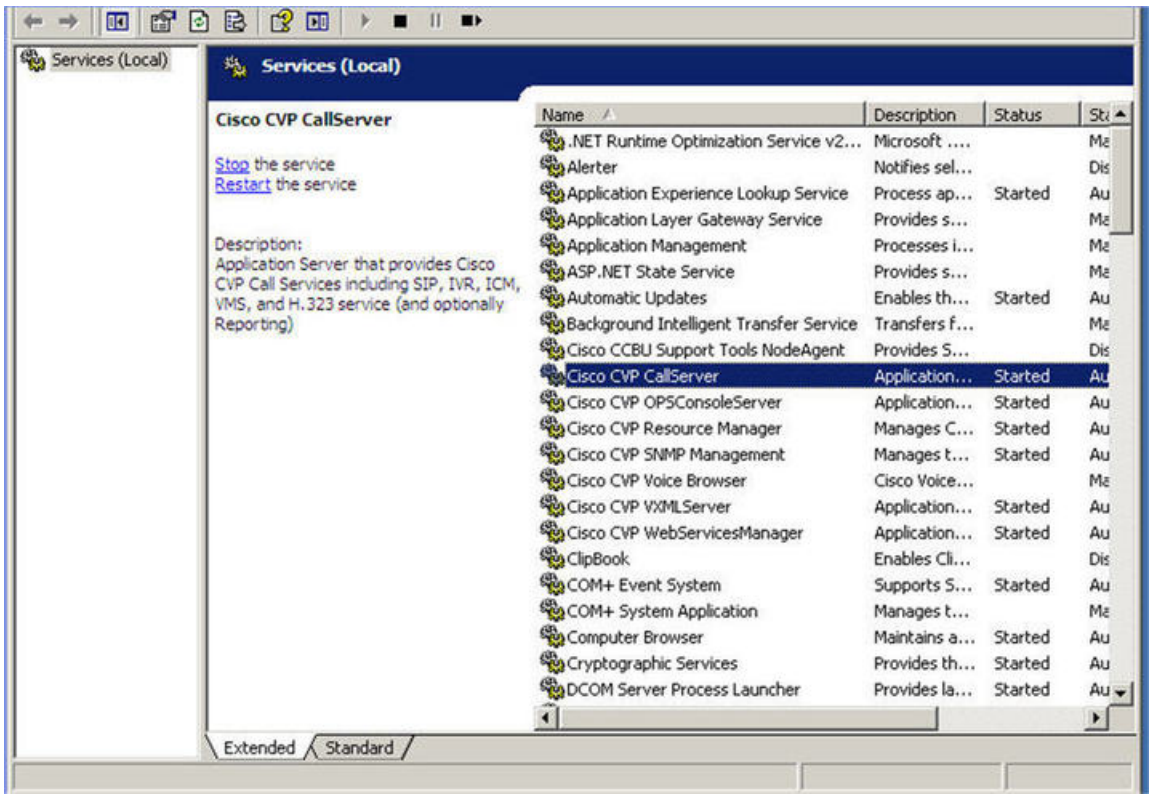
   - CVP CallServer

- Cisco CVP Resource Manager
- CVP VXMLServer
- WebServicesManager

In OAMP Window system

- Cisco CVP OPSConsoleServer
- Cisco CVP Resource Manager
- Cisco CVP WebServicesManager

Note In Step Step #4, when you enable secure communication and save the setting, the Operations Console provides a list of the services you must restart for secure communication to take effect.



3 Click the **Stop** link to stop service.

Step 2 Use the instructions in "Exchange Certificates Between Systems" for details on using the keystore and keytool Java tools to exchange trusted certificates between the Operations Console and the device being managed.

Note For information about prerequisites and assumptions regarding keystore and keytool, see Before You Begin. For instructions about using these tools, see the Java documentation.

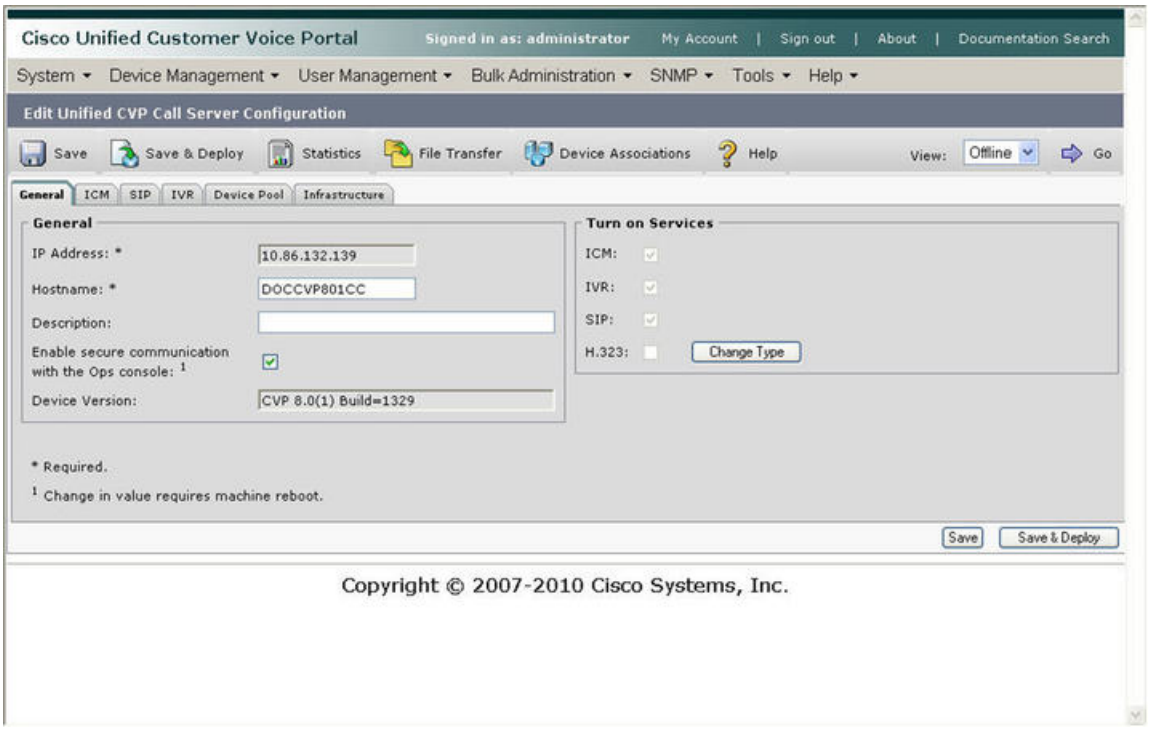
Step 3 Restart just the *Cisco CVP OpsConsoleServer* service.

Restart this procedure, selecting the **Start** link instead of **Stop** on a Windows system.

Step 4 Use the **Enable secure communication with the Ops Console** checkbox on the Operations Console Device Management configuration page to enable security for devices that require secure communication. (see "Enable Security on Unified CVP Devices" for more information.)

Note Checking this box for any server on a specific box enables security for all the servers on that box. You will be asked to restart the servers that have security enabled.

Note After you have enabled secure communication between Unified CVP components, any devices or clients that are not set up for secure communication do not work until modified for secure communication. See Exchange Certificates Between Systems to complete the setup.



Step 5 Restart the **Cisco CVP Resource Manager** service on the Unified CVP device machines on which communications needs to be secure by selecting **Start** > **Control Panel** > **Administrative Tools** > **Services**

Exchange Certificates Between Systems

The following procedures describes how to move certificates between keystores.

Note The keytool commands shown below use the JRE relative path for the Windows platform.

Procedure

Step 1 Import the Operations Console Server certificate as trusted on the managed Unified CVP device:

a) Retrieve the keystore password from the **security.properties** file (resides in the %CVP_HOME%\conf directory) on the *Operations Console* Server.
b) Export the certificate from the keystore on the Operations Console Server. Open a command prompt and navigate to the %CVP_HOME%\conf\security directory, and then enter the following command:

```
..\..\jre\bin\keytool -export -v -keystore .keystore -storetype
JCEKS -alias oamp_certificate -file <oamp_cert_XXX>
```

The file argument in <> (angle brackets) is user-defined and unique.

Note *Do not* modify the oamp_certificate alias name.

When prompted, enter the keystore password.

c) Copy the exported certificate file <oamp_cert_XXX> from the Operations Console service to the %CVP_HOME%\conf\security folder on the machine where the Unified CVP Resource Manager service is running.
d) Retrieve the keystore password from the **security.properties** file on the managed Unified CVP device.
e) For Windows, import the Operations Console certificate <oamp_cert_XXX> into the keystore on the managed Unified CVP device. Open a command prompt and navigate to the %CVP_HOME%\conf\security directory, and then enter the following command:

```
..\..\jre\bin\keytool -import
 -keystore .keystore -storetype JCEKS -trustcacerts -alias
 <orm_oamp_certificate> -file <oamp_cert_XXX>
```

Where the alias argument in <> (angle brackets) is user-defined and unique, and the file argument in <> (angle brackets) is the exported Operations Console certificate filename.

When prompted, enter the keystore password and then enter yes to confirm.

f) Repeat these steps for every machine where the Unified CVP Resource Manager service is running if the JMX communication from the Operations Server to that managed Unified CVP device needs to be secured.

Step 2 Import the managed Unified CVP device certificate as trusted in the keystore on the Operations Console Server:

a) Retrieve the keystore password from the security.properties file (resides in the %CVP_HOME%\conf directory) on the managed Unified CVP device.
b) For Windows, export the Unified CVP device certificate from the keystore. Open a command prompt and navigate to the %CVP_HOME%\conf\security directory, and then enter the following command:

```
..\..\jre\bin\keytool -export -v
-keystore .ormKeystore -storetype JCEKS -alias orm_certificate -file
<orm_cert_file_XXX>
```

The file argument in <> (angle brackets) is user-defined and unique.

Note *Do not* modify the orm_certificate alias name.

c) Append an IP address to the file name to make it unique. The IP address can be replaced with any value as long as it makes the filename unique when copied to the Operations Console Server.
d) Copy the exported certificate file <orm_cert_file> from the managed Unified CVP device to the %CVP_HOME%\conf\security folder on the Operations Console service.
e) Retrieve the keystore password from the security.properties file in the Operations Console Server.

f) Import the certificate <orm_cert_file> into the keystore on the Operations Console Server. Open a command prompt and navigate to the %CVP_HOME%\conf\security directory, and then enter the following command:

```
..\..\jre\bin\keytool -import -keystore .keystore -storetype
JCEKS -trustcacerts -alias <oamp_orm_certificate_XXX> -file
<orm_cert_XXX>
```

The alias argument in <> (angle brackets) is user-defined and unique, and the file argument in <> angle brackets is the exported managed device certificate filename.

g) Append an IP address to the certificate alias to make the alias unique in the keystore. The IP address can be replaced with any value as long as it makes the certificate name unique when imported to the keystore.

h) Repeat these steps for every machine where the Unified CVP Resource Manager service is running if the JMX communication from the Operations Console Server to that managed Unified CVP device needs to be secured.

Note For self signed certificates , import the certificate <orm_cert_file> (generated using the option "b" in Step 2) into the keystore on the CVP managed device. Open a command prompt, navigate to the %CVP_HOME%\conf\security directory, and then enter the following command:

```
..\..\jre\bin\keytool -import -keystore .keystore -storetype
JCEKS -trustcacerts -alias <cvp_orm_certificate_XXX>-file
<orm_cert_XXX>
```

Enable Security on Unified CVP Devices

Once you have completed the procedure described in Exchange Certificates Between Systems, you must *enable* security on the Unified CVP components that you want to accept only secure SSL communications.

Note For information about enabling security on non-Unified CVP components that are part of the Unified CVP solution, see Secure Communications Between Unified CVP and IOS Devices.

By default, the communication channel between the Unified CVP Operations Console and the Resource Manager on Unified CVP devices is non-secure at the completion of Unified CVP installation. You use the Operations Console's Device Management configuration page to enable or disable secure SSL communications.

Note Whenever this security setting is modified, you must restart the Unified CVP Resource Manager service on the machine where the device is running.

Caution If you do not check the Enable secure communication with the Ops console checkbox, the communication link between the Operations Console and the managed Unified CVP device *will not* be secure.

Procedure

Step 1 Open the Operations Console.

Step 2 Select a device type from the **Device Management** menu.

Step 3 Click **Add New** or select an existing device name and click **Edit**.
The General tab appears.

Step 4 Select the **Enable secure communication with the Ops console** checkbox.

Step 5 Click **Save** to save the settings in the Operations Server database and click **Save and Deploy** to apply the changes to the device.

Step 6 Restart the Unified CVP Resource Manager service on the machine where the device is running.

Step 7 Repeat these steps for all Unified CVP components that are accepting only secure SSL communications.

Certificate Authority Signed Certificates

This section describes how to:

- Generate a Certificate Signing Request
- Obtain the signed certificate
- Import the signed certificates on all machines managed by the Unified CVP Operations Console

Add a Certificate Signed by a Certificate Authority to the Keystore

Follow the steps below to generate and import CA-signed certificates for secure communications between the Operations Console and the CVP ResourceManager on other devices in your Unified CVP solution.

- This section *does not* discuss how to accommodate HTTPS connections to the Operations Console; for information on that topic, refer to Add a Certificate Signed by a Certificate Authority for HTTPS Web Access, on page 208.
- The **keytool** commands use the JRE relative path for the Windows platform.
- If you have already exchanged certificates to secure Unified CVP device communications, that process must be repeated after importing the signed certificates.

Procedure

Step 1 Retrieve the keystore password from the security.properties file.

Step 2 Generate a Certificate Signing Request (CSR).

a) From the %CVP_HOME%\conf\security directory, enter the following:

```
..\..\jre\bin\keytool -keystore .keystore -storetype JCEKS -certreq -keyalg RSA -sigalg
 MD5withRSA -alias orm_certificate -file ormcertreq.csr
```

b) When prompted, enter the keystore password.

Step 3 Send the ormcertreq.csr certificate file to a Certificate Authority (CA) for sign-off. Once the certificate is signed, it is returned with a CA root certificate and, depending on the signing CA, some optional intermediate certificates.

Step 4 Install the signed certificate into keystore.

a) Enter the following command to install any Intermediate CA Certificates:

```
keytool -keystore .keystore -storetype JCEKS -import -alias root -trustcacerts -file
<filename_of_intermediate_CA_certs>
```

b) Enter the following command to install the root certificates (not in the Unified CVP keystore by default):

```
keytool -keystore .keystore -storetype JCEKS -import -alias root -trustcacerts -file
<filename_of_root_cert>
```

Note Examine the contents of any root certificate file before installing it to your keystore as a trusted certificate.

The Java root certificates are installed in %CVP_HOME%\jre\lib\security\cacerts.

c) Enter the following command to install the CA Signed Certificate:

```
keytool -keystore .keystore -storetype JCEKS -import -alias orm_certificate -trustcacerts
 -file <filename_of_your_signed_cert_from_CA>
```

Step 5 Repeat these steps on every machine running Unified CVP Services.

Add a Certificate Signed by a Certificate Authority for HTTPS Web Access

The following procedure describes how to present a CA-signed certificate to inbound Operations Console HTTPS clients.

Note The OAMP and ORM certificates provided in the keystore do not provide TLS encryption for inbound HTTPS traffic; those certificates provide secure connections between the Operations Console and the CVP Resource Manager on other devices in your Unified CVP solution.

The certificate and private key used for Operations Console HTTPS are

- Self-signed certificate: %CVP_HOME%\conf\security\oamp.crt
- Private key for self-signed certificate: %CVP_HOME%\conf\security\oamp.key

Procedure

Step 1 Access the OpenSSL command line.

Note You must first install OpenSSL (http://www.openssl.org), as it is not included with Unified CVP. See the OpenSSL documentation for details.

Step 2 Generate a Certificate Signing Request (CSR) by entering the following command:

openssl req -new -key xxxx.key -out xxxx.csr

xxxx represents the key and the certificate files.

Step 3 Send the xxxx.csr certificate file to a Certificate Authority (CA) for sign-off. Once the certificate is signed, it is returned with a root certificate of a CA.

Step 4 Replace the original oamp.crt file with the signed certificate.

Secure Communications Between Unified CVP and IOS Devices

To secure file transfer between Cisco Gateways and the Unified CVP Operations Console, import the Operations Console Server certificate on the IOS device during device configuration and enable SSH on the router. Otherwise, any user-requested action through the Operations Console (for example, file transfer to an IOS device) will fail. For example, to copy a file to the IOS device, the Operations Console expects SSH to be enabled on the device. If SSH is not enabled, a failure will occur.

Export the Operations Console Certificate to a File

Procedure

Step 1 From the web browser, enter https://*ServerIP*:9443/oamp
The Security Alert dialog box appears.

Step 2 Click **View Certificate** .
The Certificate dialog box appears.

Step 3 Select the Details tab.
<All> are highlighted in the Show drop-down list.

Step 4 Click **Copy to File**.
The Certificate Export Wizard dialog appears.

Step 5 Select the Base-64 encoded X.509 (.CER) radio button and click **Next**.

Step 6 Specify a file name in the File to Export dialog and click **Next**.

Step 7 Click **Finish**.
An Export was Successful message appears.

Step 8 Click **OK** and close the Security Alert dialog.

Step 9 Open the exported file in Notepad and copy the text that appears between the ----BEGIN CERTIFICATE--- and ----END CERTIFICATE--- tags.

You are now ready to copy the Operations Console certificate information to the IOS device.

Import the Operations Console Certificate to a Gateway

Procedure

Step 1 Access the IOS device in privileged EXEC mode.
Note For more information, see the Cisco IOS CLI documentation.

Step 2 Access global configuration mode by entering:
configure terminal

Step 3 Create and enroll a trustpoint by entering the following commands:

```
crypto pki trustpoint xxxx
en terminal
exit
```

where xxxx is a trustpoint name.

The IOS device exits conf t mode and returns to privileged EXEC mode.

Step 4 Copy the certificate exported to the Notepad to the IOS device:

1. Enter: cryp pki auth xxxx

   where xxxx is the trustpoint name specified in the previous step.
2. Paste the certificate from the Notepad clipboard.
3. Enter: quit

   A message displays describing the certificate attributes. A confirmation prompt appears.

Step 5 Enter: Yes
A message informs you that the certificate is imported.

Configure Secure Access to the Router Using Protocol SSH Version 2

The Unified CVP administrator can configure secure access to the router using the following example:

Procedure

Step 1 Log into the IOS devices.

Step 2 On the gateway, enter the command show ip ssh to determine if the SSH Server is enabled.
If the SSH Server is not enabled, continue this procedure.

Step 3 Access the configurable terminal and enter the following commands:

```
gw(config)# ip domain name <domain name>
gw(config)# crypto key generate rsa
gw(config) # ip ssh version 2
```

where the domain name is your configured domain name.

Step 4 Enter 768 or higher for how many bits to use.

Important If you enter a value lower than 768, IOS automatically uses SSH Version 1, which is not compatible with the Operations Console. The Operations Console is trying to connect with SSH Version 2.

Step 5 Select the key module size from 360 to 2048 for your General Purpose Keys.

Step 6 Enter the following commands:

```
gw(config)# ip ssh time-out 60
gw(config)# ip ssh authentication-retries 5
```

Step 7 Enter the show ip ssh command and confirm that "SSH version enabled -version 1.99" or "SSH version enabled - version 2.0" displays.

Step 8 Configure a user name and password in IOS by entering the following command:

```
gw(config)# username <username> password
<password>
```

where the username and password are your configured username and password.

Step 9 Configure the IOS settings to allow SSH logins by entering the following commands.

Note These commands are located in the running configuration towards the end. Search for the line **vty configurations**, in the configuration. The important settings are those indicated by an asterisk (*).

```
line vty 0 4*
exec-timeout 35 0
password <password>*
session-limit 10
login local*
transport input telnet*
transport output all
```

```
gw# configure terminal
gw(config)# line vty 0 4
gw(config-line)# password <password>
gw(config-line)# login local
gw(config-line)# transport input ssh
```

Step 10 To configure these settings, access the configurable terminal and enter the following:

```
line vty 0 4*
exec-timeout 35 0
password <password>*
session-limit 10
login local*
transport input telnet*
transport output all
```

Where the password is your configured password.

Important Enabling these settings only allow SSH. If you want telnet and SSH available, then enter transport input all or transport input telnet for telnet-only access.

Secure Communications Between VoiceXML Gateways and VXML Server

To prevent malicious requests from being sent to the VXML Server, configure the application server on which it runs to only accept connections from the IP addresses of the VoiceXML Gateways.

Note The following instructions are specifically written for Tomcat users.

The following procedure describes how to configure Tomcat to only accept connections from a known set of IP addresses:

Procedure

Step 1 Edit the `CATALINA_HOME\conf\context.xml` file, where *CATALINA_HOME* is the Tomcat installation directory.

Step 2 Between the existing <Context> tag and its closing tag, insert:

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="X.X.X.X" deny=""/>
```

where *X.X.X.X* is the IP address of your VoiceXML Gateway(or Application Control Engine IP).

If multiple IP addresses need access, indicate them as a comma-delimited list, such as:

```
allow="X.X.X.X,Y.Y.Y.Y"
```

Step 3 Restart Tomcat to enable the changes.

HTTPS support for unified CVP

Set Up Tomcat to Present CA-Signed Certificates to Inbound HTTPs Clients

Due to the large processing overhead of HTTPs, the Tomcat application server only achieves up to 100 simultaneous connections dependent on the configuration (see the Cisco Unified Customer Voice Portal (CVP) Release Solution Reference Network Design (SRND) for further information).

Tomcat users must follow the steps below to present a CA-signed certificate to inbound HTTPs clients.

Procedure

Step 1 Open the **security.properties** file to retrieve the .keystore password. You will need to copy and paste the value of this property when managing the .keystore.

1. Open the `%CVP_HOME%\conf\security.properties` file where **%CVP_HOME%** is the target installation directory for Unified CVP. In the default case, this is **C:\Cisco\CVP**.

   Note The property file should contain one property: **Security.keystorePW**.

2. After entering a command, the keytool prompts you to "Enter keystore password." Copy the value of the **Security.keystorePW** property and paste it into the command-line window.

For example, if the `%CVP_HOME%\conf\security.properties` file contains the property line:

- **Security.keystorePW = [3X}}E7@nhMXGy{ou.5AL!+4Ffm868** .
- The password to copy would be: **[3X}}E7@nhMXGy{ou.5AL!+4Ffm868**.

Step 2 Create a back up of the **%CVP_HOME%\conf\security** directory.

Step 3 Open a command-line prompt window and change to the security configuration directory: **cd\cisco\cvp\conf\security**.

Step 4 Create the certificate signing request to use the private key entry for your certificate, remembering to enter in the keystore password when prompted. A new csr file is created on the file system.
Examples:

- Call Server: `%CVP_HOME%\jre\bin\keytool.exe - certreq - alias callserver_certificate - storetype JCEKS - keystore .keystore - file callserver_certificate.csr .`
- VXML Server: `%CVP_HOME%\jre\bin\keytool.exe - certreq - alias vxml_certificate - storetype JCEKS - keystore .keystore - file vxml_certificate.csr .`

Step 5 Give the certificate signing request file to a trusted Certificate Authority. They will sign it and return one or more trusted certificates.

Step 6 Import the signed certificate file from your trusted Certificate Authority to the .keystore file, and enter in the keystore password when prompted. If more than one certificate is delivered, certificates must be imported in order of the chained certificate hierarchy. For example: root, intermediate, signed certificate.
Examples:

- Call Server: `%CVP_HOME%\jre\bin\keytool.exe - import - v - alias callserver_certificate - storetype JCEKS - trustcacerts - keystore .keystore - file signed_callserver_certificate.crt .`
- VXML Server: `%CVP_HOME%\jre\bin\keytool.exe - import - v - alias vxml_certificate - storetype JCEKS - trustcacerts - keystore .keystore - file signed_vxml_certificate.crt .`

Secure Communications Between Unified CVP and IOS Devices

To secure HTTPS between Cisco Gateways and the Call Server and VXML Server to the gateway for HTTPS, you need to import the Call Server or VXML Server certificate on the IOS device during device configuration.

Procedure

Step 1 From the web browser, access the secure Call Server with [https://<ServerIP>:8443/] or the secure VXML Server with [https://<ServerIP>:7443/].
The Security Alert dialog box displays.

Step 2 Click **View Certificate**.
The Certificate dialog box displays.

Step 3 Select the Details tab.
<All> is highlighted in the Show drop-down list.

Step 4 Click **Copy to File**.
The Certificate Export Wizard dialog appears.

Step 5 Click **Base-64 encoded X.509 (.CER)** and then click **Next**.

Step 6 Specify a file name in the File to Export dialog box and then click **Next**.

Step 7 Click **Finish**.
A message indicates that the export was successful.

Step 8 Click **OK** and close the Security Alert dialog box.

Step 9 Open the exported file in Notepad and copy the text that appears between the ---BEGIN CERTIFICATE-- and --END CERTIFICATE-- tags.
You are now ready to copy the Operations Console certificate information to the IOS device.

Step 10 Access the IOS device in privileged EXEC mode.
For more information, refer to the Cisco IOS CLI documentation.

Step 11 Access global configuration mode by entering configuration terminal .

Step 12 Create and enroll a trustpoint by entering the following commands:

```
crypto pki trustpoint xxxx
en terminal
exit
```

where *xxxx* is a trustpoint name.

The IOS device exits conf t mode and returns to privileged EXEC mode.

Step 13 Copy the certificate exported to the Notepad to the IOS device:

1. Enter crypto pki auth xxxx

   where *xxxx* is the trustpoint name specified in the previous step.

2. Paste the certificate from the Notepad clipboard.

3. Enter quit.

   - A message displays describing the certificate attributes.
   - A confirmation prompt appears.

Step 14 Enter yes.
A message indicates that the certificate was successfully imported.

Sensitive Customer Information

Use the VXML Server Inclusive and Exclusive filters to control the sensitive customer information, such as PIN numbers, that are sent to the Reporting Server.

By default, all items except the Start and End element are filtered from information the VXML Server feeds to the Reporting Server unless they are added to an Inclusive Filter. If you create Inclusive filters that are broad enough to allow sensitive information to be passed, you then have the option to:

- Adjust the Inclusive filters so that the sensitive information is not included.
- Add Exclusive filters to prevent the sensitive information from being included.

See to the Operations Console online help topic *Configuring a VXML Server > Inclusive and Exclusive VoiceXML Filters for Reporting* for detailed information about configuring filters.

PART II

Configuration Detail of Unified CVP Components

- Configure VXML Solution, page 219
- Configure Unified CVP Logging and Event Notifications, page 243

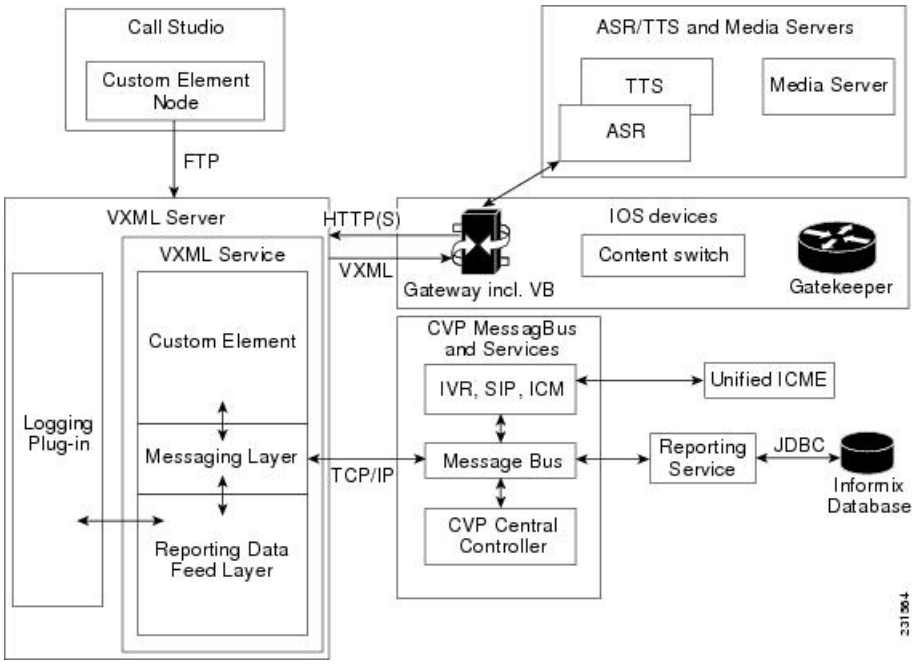
CHAPTER 6

Configure VXML Solution

Unified CVP provides a platform for developing powerful, speech-driven interactive applications accessible from any phone.

VXML solution consists of:

- *VXML Server*, a J2EE-compliant application server that dynamically drives the caller experience.
- *VoiceXML Service*, provides Unified ICME with call control capabilities and allows data to be sent to the Reporting Service.
- *Call Studio*, a drag-and-drop graphical user interface (GUI) for the rapid creation of advanced voice applications.



Note This section describes how to *integrate* the Unified CVP solution with other Cisco products. For complete information on how to use the VXML Server software, see User Guide for Cisco Unified CVP VXML Server and Unified Call Studio .

- About Unified CVP VXML Server, page 220
- Unified CVP VXML Server, page 223
- Unified CVP VXML Server (Standalone), page 227
- Unified CVP VXML Server with Unified ICME, page 228
- Unified CVP VXML Server (Standalone) Server Solutions, page 230
- Pass Data to Unified ICME, page 239

About Unified CVP VXML Server

The Unified CVP VXML Server is a J2EE-compliant application server that provides a complete solution for rapidly creating and deploying dynamic VXML-based self service applications.

You can use the Unified CVP VXML Server:

- As an autonomous standalone component, without the Unified CVP Call Server and Unified ICME components. In this configuration, the Unified CVP VXML Server handles self-service VXML applications. In a deployment that includes a Call Server the Unified CVP VXML Server can consult Unified ICME to obtain information or transfer destination labels.
- As an add-on component to a SIP Service or Call Server, in an integrated Unified CVP deployment with Unified ICME. In this configuration, the SIP Service:
  - Maintains control of the call flow, transferring calls to appropriate targets, under the direction of a script running on Unified ICME.
  - Can use the Cisco Voice Gateway's prompt-and-collect features by making requests to the Unified CVP VXML Server.

Note For information about deployment options for Unified CVP VXML Server, see High-Level Configuration Instructions. For instructions on how to install a Unified CVP VXML Server, see *Installation and Upgrade Guide for Cisco Unified Customer Voice Portal* .

About VoiceXML Service

The VoiceXML Service provides Unified ICME call control capabilities and data to the Reporting Service.

The VoiceXML Service

- Resides outside of the Call Server that gives call control capabilities to the Standalone Mode.

- Is the connection between the VXML Server and the ICM Service that feeds data to the Reporting Service.
- In a Standalone Mode with ICM Lookup deployment:
  - Interacts with the VXML Server and the ICM Service to do call control piece
  - Interacts with VXML Server and Reporting Service to populate the Reporting database.

Note For more information, see Call Studio ReqICMLabel Element to Pass Data.

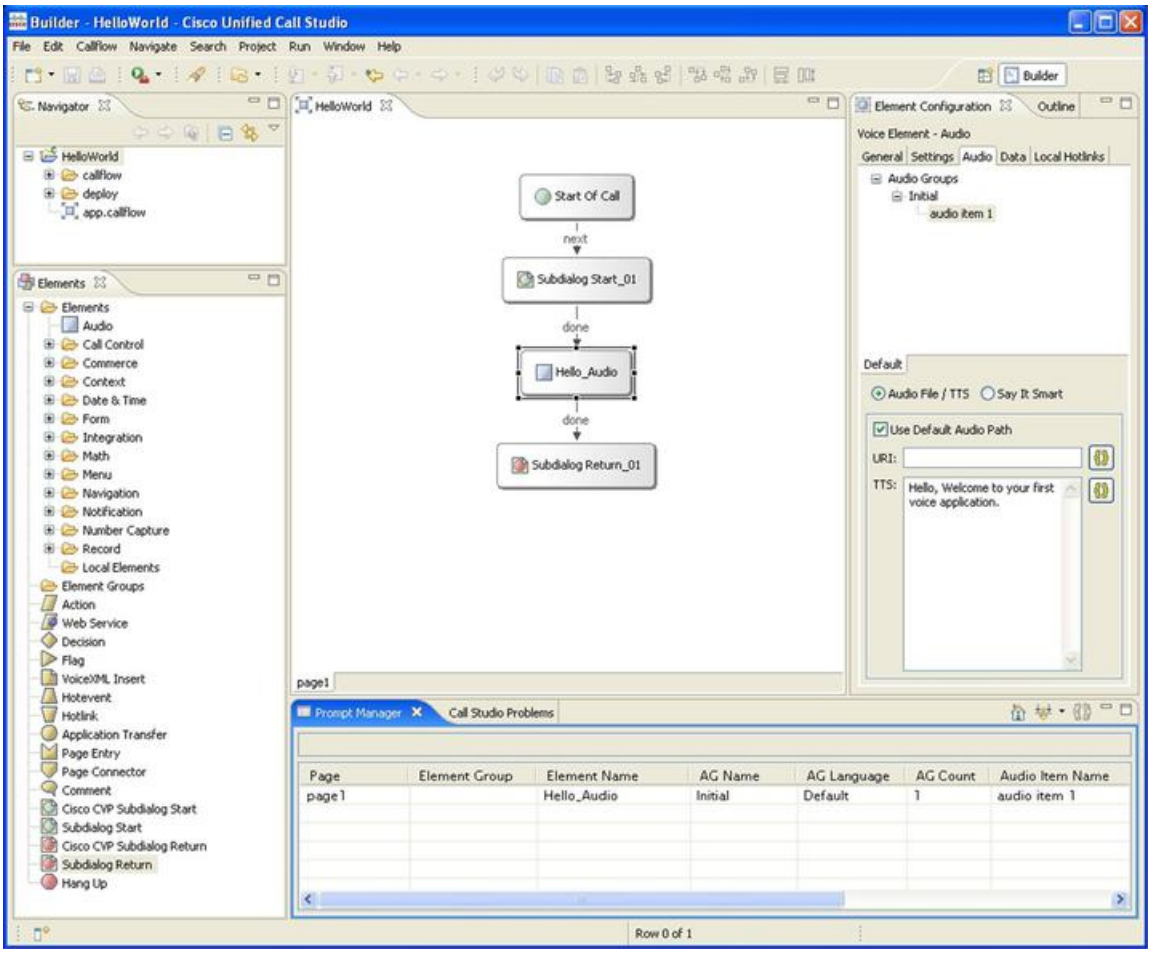
About Call Studio

Call Studio is a development platform for the creation of voice applications.

Call Studio provides

- A framework on which a host of Unified CVP and third-party tools appear with a robust, consistent interface for voice application designers and developers to use.
- A true control panel for developing all aspects of a voice application, with each function implemented as a plug-in to the Call Studio platform.

The following figure shows an example of a Call Studio application:



Note For more information about Call Studio applications, see *User Guide for Cisco Unified CVP VXML Server and Cisco Unified Call Studio*.

VXML Server Reporting

VXML Server applications can function in a wide range of paradigms, from the VXML Server virtually controlling the entire user interaction to performing individual interactions on a scale similar to that of the Unified CVP micro-applications. Between these extremes, you can design the VXML Server applications to implement specific transactions. For example, in a banking application a transaction can consist of all the user interactions required to successfully complete a balance transfer or a telephone bill payment. The high-level menus which the user can use to select a particular type of transaction is controlled by the Unified ICME routing script, using standard Unified CVP micro-applications, such as Menu and Play Media. Once a particular transaction type is chosen, the Unified ICME routing script issues an External VoiceXML micro-application to invoke the appropriate VXML Server application which implements that transaction type. Once the VXML Server application completes, control returns to the Unified ICME routing script for further menus. Typically, audit information about the transaction is returned, and can be stored in the Unified ICME database. It is also

determines whether the transaction was successful, or it needs to be transferred or queued to an agent, and so on.

While Unified ICME VRU Progress reporting capabilities are always in effect, they compliment VXML Server applications most effectively when this transaction-oriented design is used. The customer defines a Unified ICME CallType for each type of transaction, and uses the audit information returned from the VXML Server to determine how to set the Unified ICME's VRUProgress variable. The setting selected dictates how the transaction is counted in the aggregate VRU reporting fields in the CallTypeHalfHour table.

VRU reporting enhancements are described in the Unified ICME 6.0(0) and online help.

Unified CVP VXML Server

The Unified CVP VXML Server is an optional J2EE-compliant application server that provides a complete solution for rapidly creating and deploying dynamic VoiceXML applications. If you installed a Unified CVP VXML Server, you must configure it before using it to deploy VoiceXML applications or licenses.

If you are using a VoiceXML gateway to route calls from the Unified CVP VXML Server, but want to use the Unified CVP reporting feature, install the Call Server and Reporting Server on the same physical machine. Configure the Call Server with no call services enabled. Then configure the Reporting Server and select the Call Server that is installed on the same machine (same IP address) as the primary Call Server for the Reporting Server.

If you need to make requests to an ICM server, without relinquishing control of the call or use Unified CVP reporting, you must configure the Unified CVP VXML Server to use a Call Server with at least the ICM Service enabled.

The Operations Console online help topic *Configuring a Unified CVP VXML Server* provides details for performing the following tasks:

- Adding a Unified CVP VXML Server
- Editing a Unified CVP VXML Server
- Deleting a Unified CVP VXML Server
- Uploading a Syslog XML File to a Unified CVP VXML Server
- Downloading a Syslog XML File from a Unified CVP VXML Server
- Editing the Log Messages XML File
- Transferring Script Files to a Unified CVP VXML Server
- Applying a License to a Unified CVP VXML Server

  When you are creating a new Unified CVP VXML Server, you must apply a valid license file before using the server. You can browse for and upload the license file to the Operations Console, and then transfer the license to the Unified CVP VXML Server. Select either an existing license file in the Operations Console database or a new license file from your local PC. For more information on licensing, see *Installation and Upgrade Guide for Cisco Unified Customer Voice Portal* .
- Finding a Unified CVP VXML Server

  The Operations Console lets you locate a Unified CVP VXML Server from a list of configured servers or on the basis of specific search criteria.
- Viewing Device Status

The online help also contains two tables describing the Unified CVP VXML Server properties:

- VXML General Properties (see the help topic *Unified CVP VXML Server General Properties*)

  The table includes settings that identify the Unified CVP VXML Server and choose a primary, and an optional backup Call Server to communicate with the Reporting Server. It also includes the setting to enable secure communications between the Operations Console and the Unified CVP VXML Server.

- Unified CVP VXML Server Configuration Properties (see the help topic *Unified CVP VXML Server Configuration Properties*

  From the Unified CVP VXML Server Configuration tab, you can enable the reporting of Unified CVP VXML Server and call activities to the Reporting Server. When enabled, the Unified CVP VXML Server reports on call and application session summary data.

  Call summary data includes call identifier, start and end timestamp of calls, ANI, and DNIS. Application session data includes application names, sessionid, and session timestamps.

  If you choose detailed reporting, Unified CVP VXML Server application details are reported, including element access history, activities within the element, element variables and element exit state. Customized values added in the Add to Log element configuration area in Call Studio applications are also included in reporting data. Optionally, you can create report filters that define which data to include in the report.

Use Inclusive and Exclusive VoiceXML filters to control the data that the Unified CVP VXML Server feeds to the Reporting Server. Data feed control is crucial for:

- Saving space in the reporting database
- Preserving messaging communication bandwidth

See the Operations Console online help topic *Inclusive and Exclusive VoiceXML Filters for Reporting* for step-by-step configuration details.

You can also transfer script, and other files, to the Unified CVP VXML Server using the Operations Console. See the Operations Console online help subtopic *Configuring a Unified CVP VXML Server > Transferring Script Files* .

Unified CVP VXML Server with a Call Server Association

The Unified CVP VXML Server uses the message service on a Call Server to communicate with the Reporting Server.

Associate a Unified CVP VXML Server with a Call Server

This section describes how to specify a primary and backup Call Server for the Unified CVP VXML Server.

Procedure

Step 1 Open the Operations Console.

Step 2 Choose **Device Management** > **VXML Server**.

Step 3 Click **Add New** or select an existing Unified CVP VXML Server name, and click **Edit**.

The General tab appears.

Step 4 Select the **Primary Call Server** from the drop-down list.

Step 5 Select the **Backup Call Server** from the drop-down list.

Step 6 Click **Save** to save the settings in the Operations Server database, and click **Save & Deploy** to apply the changes to the Unified CVP VXML Server.

Step 7 Restart the Unified CVP VXML Server and the Call Servers.

Security for a Unified CVP VXML Server

By default, the communication channel between Operations Console and Unified CVP Resource Manager is non-secure at the completion of Unified CVP installation. To set up the Unified CVP VXML Server to accept *only* secure SSL connections from the Operations Console, you must manually enable a security setting.

Note For more information, see Unified CVP Security.

Note Whenever this security setting is modified, you must restart the CVP Resource Manager service on the machine where the device is running.

Enable Security for a Unified CVP VXML Server

This section describes how to enable security for the Unified CVP VXML Server.

Procedure

Step 1 Open the Operations Console.

Step 2 Choose **Device Management** > **VXML Server**.

Step 3 Click **Add New** or select an existing Unified CVP VXML Server name and click **Edit**.
The General tab appears.

Step 4 Select the **Security On** check box.

Step 5 Click **Save** to save the settings in the Operations Server database, and click **Save & Deploy** to apply the changes to the Unified CVP VXML Server.

Step 6 Restart the Unified CVP VXML Server.

Configure Unified CVP VXML Server Applications: Filters

This section describes how to configure filters for Unified CVP VXML Server applications detail reporting.

Procedure

Step 1 From the Operations Console, choose **Device Management** > **VXML Server**.

Step 2 Click **Add New** or select an existing Unified CVP VXML Server name and click **Edit**.

Step 3 Select the **Configuration Tab** and use the **Inclusive Filters** and **Exclusive Filters** fields to specify the list of applications, element types, element names, element fields, and ECC variables to include and exclude from reporting data.
See the Operations Console online help topic *Configuring a Unified CVP VXML Server > Inclusive and Exclusive VoiceXML Filters for Reporting*. This topic includes the subtopic *Rules for VoiceXML Inclusive and Exclusive Filters* .

Step 4 Click **Save**.

Step 5 Restart the VoiceXML Service.

Enable Reporting for the Unified CVP VXML Server

This section describes how to configure reporting settings for the Unified CVP VXML Server.

Procedure

Step 1 Open the Operations Console.

Step 2 Choose **Device Management** > **VXML Server**.

Step 3 Click **Add New** or select an existing Unified CVP VXML Server name, and click **Edit**.

Step 4 Select the **Configuration Tab** and set the following:

- Click the **Yes** button to **Enable reporting for this VXML Server**. Indicates whether or not the Unified CVP VXML Server sends data to the Reporting Server. If disabled, no data is sent to the Reporting Server and reports do not contain any VoiceXML application data.
- Click the **Yes** button to **Enable reporting for VXML application details**. Indicates whether or not VoiceXML application details are reported.
- Specify **Max. Number of Messages** to save in a file if failover occurs and the Reporting Server is unreachable. This value is limited by the amount of free disk space.

Step 5 Click **Save**.

Step 6 Restart the VoiceXML Service and the Call Server.

QoS for the Unified CVP VXML Server

Quality of Service (QoS) is the measure of transmission quality and service availability of a network (or internetworks).

Note For more information about defining QoS criteria, see the latest Enterprise QoS Solution Reference Network Design Guide.

For more information to create policy based QoS, see section Create Policy Based QoS

Create Policy Based QoS

This section describes how to create policy based QoS.

Procedure

Step 1 From the **Local Group Policy Editor** on Windows 2008 Server, select **Computer Configuration** > **Windows Settings**.

Step 2 In the **Group Policy Object Editor** window right-click the Policy-based QoS node, and click **Create a new policy**.

Step 3 On the **Policy-based QoS** wizard specify a policy name. Specify a DSCP value, and click **Next**.

Step 4 Select **all application**, and click **Next**.

Step 5 Check the **Any source IP address** and **Any destination IP address** check box, and click **Next**.

Step 6 If the policy is for Call Server QoS, then from the **Select the protocol this QoS policy applies to** drop-down list, select the same protocol that was set in the Outbound transport type on the Unified CVP Operations Console.
If the policy is for VXML Server QoS, then from the **Select the protocol this QoS policy applies to** drop-down list, select TCP.

Step 7 If the policy is for Call Server QoS, check the **To destination port number or range** check box. Assign the same port number as configured in the **Port number for outgoing SIP requests** in the Unified CVP Operations console. By default the port number is 5060.
If the policy is for VXML Server QoS, check the **From this source port number or range** check box. Assign the port number 7000.

Step 8 Click on Finish.

Unified CVP VXML Server (Standalone)

In the Unified CVP VXML Server (standalone) call flow model, the Call Server routes messages between the components. Calls arrive through a VoiceXML gateway and interact directly with a Unified CVP VXML Server to execute VXML applications. The gateway performs both Ingress and VoiceXML functions. This call flow model provides a sophisticated VoiceXML-based VRU for applications which do not always need to interact with an ICM Server.

The Unified CVP VXML Server (standalone) does not provide statistics, and no database can be configured to capture data from standalone VoiceXML applications.

In this call flow model, configure the Call Server with no services enabled.

See the Operations Console online help topic *Managing Devices > Configuring a Unified CVP VXML Server (Standalone)* for instructions on performing the following tasks:

- Adding a Unified CVP VXML Server (standalone)

  Before adding a Unified CVP VXML Server (standalone) in the Operations Console, collect the hostname and IP address of the Unified CVP VXML Server.

- Editing a Unified CVP VXML Server (standalone)
- Deleting a Unified CVP VXML Server (standalone)

  Deleting a Unified CVP VXML Server (standalone) from the Operations Console deletes its configuration data in the Operations Console database and removes the Unified CVP VXML Server from the displayed list of Unified CVP VXML Servers.

- Finding a Unified CVP VXML Server (standalone)

  You can locate a Unified CVP VXML Server on the basis of specific criteria entered into the search tool within the Operations Console, or you can identify the server from the list of servers displayed in the Operations Console.

- Applying a License to a Unified CVP VXML Server (standalone)

Unified CVP VXML Server with Unified ICME

This section describes how to integrate VoiceXML and Unified ICME scripts.

Integrate VoiceXML Scripts with Unified ICME Scripts

This section describes how to integrate the Unified CVP VXML Server into the Unified CVP solution. This process involves:

- Creating a Unified ICME script with ECC variables configured for Unified CVP VXML Server.
- Creating a VRU Script to run in the Unified ICME script.

Procedure

Step 1 Specify the URL (remove and port number) of the Unified CVP VXML Server that you want to reach, for example:
http://10.78.26.28:7000/CVP/Server?application=HelloWorld

In the example, **10.78.26.28** is the IP address of the Unified CVP VXML Server, **7000** is the port number, and the application name is HelloWorld. The values are delimited by a colon (:).

Note 7000 is the default port number for a Unified CVP VXML Server. The new port for Unified CVP 4.0 and later is 7000 for Tomcat with Unified CVP VXML Server.

Step 2 In the Unified ICME script, first set the media_server ECC variable to:
http://10.78.26.28:7000/CVP

Step 3 Set the app_media_lib ECC Variable to "..", (literally two periods in quotes).

Step 4 Set the user.microapp.ToExtVXML[0] ECC variable to: application=HelloWorld

Note This example indicates that the Unified CVP VXML Server will execute the *HelloWorld* application. To execute a different application, change the value of user.microapp.ToExtVXML[0].

Step 5 Set the UseVXMLParams ECC Variable to **N**.

Step 6 Create a Run External Script node within the Unified ICME script with a VRU Script Name value of GS,Server,V.

Note Remember to link this node to the nodes configured in the previous steps.

- The timeout value set in the Network VRU Script should be substantially greater than the length of the timeout in the Unified CVP VXML Server application. Use this timeout only for recovery from a failed Unified CVP VXML Server.
- Always leave the **Interruptible** check box in the Network VRU Script Attributes tab checked. Otherwise, calls queued to a Unified CVP VXML Server application might stay in the queue when an agent becomes available.

Step 7 After you configure the Unified ICME script, configure a corresponding Unified CVP VXML Server script with Call Studio.

The Unified CVP VXML Server script must:

- Begin with a Unified CVP Subdialog_Start element (immediately after the Call Start element)
- Contain a Unified CVP Subdialog_Return element on all return points (script must end with a Subdialog_Return element)
- The Unified CVP Subdialog_Return element must include a value for the call input
- To enable reporting, you must add Data Feed/SNMP loggers

Correlate Unified CVP and Unified ICME Logs with Unified CVP VXML Server Logs

When using the Unified CVP VXML Server option in the Unified CVP solution, you can correlate Unified CVP/Unified ICME logs with VoiceXML logs by passing the Call ID to the Unified CVP VXML Server by URL. Building upon the URL used in the previous example, the URL is as follows:

http://10.78.26.28:7000/CVP/Server?application=Chapter1_HelloWorld&callid=XXXXX-XXXXX-XXXXXX-XXXXXX

Note Unified CVP VXML Server (by default) receives callid (which contains the call GUID), _dnis, and _ani as session variables in comprehensive mode even if the variables are not configured as parameters in the ToExtVXML array. If the variables are configured in ToExtVXML then those values are used. These variables are available to VXML applications as session variables, and they are displayed in the Unified CVP VXML Server log. This change is backwards compatible with the following script. That is, if you have added the following script, you do not need to change it. However, if you remove this script, you save an estimated 40 bytes of ECC variable space .

To configure logging, do the following:

In the Unified ICME script, use the formula editor to set ToExtVXML[1]. Set the value of ToExtVXML[1] variable to:

```
concatenate("callid=",Call.user.media.id)
```

Note

- Always include "callid" when sending the call to the Unified CVP VXML Server using the Comprehensive call flow model. The Call ID can also be used in Unified CVP VXML Server (standalone) solutions.
- When you concatenate multiple values, use a comma for the delimiter.
- The value of ICMInfoKeys must contain RouterCallKey, RouterCallDay, and RouterCallKeySequenceNumber separated by a "-".

  For example,
  concatenate("ICMInfoKeys=",Call.RouterCallKey,"-",Call.RouterCallDay,"-",Call.RouterCallKeySequenceNumber).

  See Passing Information to the External VoiceXML for more information.

Error Codes for Unified CVP VXML Server

The following are some of the error codes that you may see with the VXML Server application:

- Error Code 40 -- System Unavailable

  This is returned if the VXML Server is unavailable (shutdown, network connection disabled, and so forth).
- Error Code 41 -- App Error

  This is returned if a Unified CVP VXML Server application error occurs (For example, a java exception).
- Error Code 42 -- App Hangup

  This is returned if the Hang Up element is used instead of the Unified CVP Subdialog_Return element.

  Note If the application is configured correctly, this does not occur.
- Error Code 43 -- Suspended

  This is returned if the Unified CVP VXML Server application is suspended.
- Error Code 44 -- No Session Error

  This is returned when an emergency error occurs (for example, an application is called that has not been loaded in the Unified CVP VXML Server application).
- Error Code 45 -- Bad Fetch

  This is returned when the Unified CVP VXML Server encounters a bad fetch situation. This code is returned when either a .wav file or an external grammar file is not found.

Unified CVP VXML Server (Standalone) Server Solutions

The Unified CVP VXML Server is a J2EE-compliant application server that provides a complete solution for rapidly creating and deploying dynamic VoiceXML applications. You can install the Unified CVP VXML Server as a standalone component, without the Call Server component. The Unified CVP VXML Server (Standalone) is designed to handle self-service VoiceXML applications.

Note see Hardware and Software System Specification for Cisco Unified Customer Voice Portal Software for information about supported versions of Cisco IOS Firewall software.

Configure the Unified CVP VXML Server (Standalone) with ICM Lookup Call Flow Model

The following procedure describes how to configure the Unified CVP VXML Server (standalone) with ICM Lookup call flow model.:

Procedure

Step 1 Copy the following files from the Unified CVP VXML Server CD to the gateway flash memory using tftp: CVPSelfService.tcl

critical_error.wav

For example:

copy tftp: flash:CVPSelfService.tcl

copy tftp: flash:CVPSelfServiceBootstrap.vxml

copy tftp: flash:critical_error.wav

Step 2 Define the Unified CVP VXML Server applications on the gateway. The following lines show an example configuration:

```
service CVPSelfService flash:CVPSelfServiceBootstrap.vxml
!
service [gateway application name] flash:CVPSelfService.tcl
param CVPBackupVXMLServer 12.34.567.890
param CVPSelfService-port 7000
param CVPSelfService-app [name of application on the VXML Server, exactly how it appears]
param CVPPrimaryVXMLServer 12.34.567.891
```

Note CVPSelfService is required. Backup server is optional. For Tomcat Application Server, set the port to 7000.

After completing the gateway configuration, run the following to load and activate the applications:

```
call application voice load CVPSelfService
call application voice load [gateway application name]
```

Step 3 Define a dial-peer for the gateway application, for example:

```
dial-peer voice [dial-peer unique ID] voip /* for IP originated call */
service [gateway application name]
incoming called-number [dialed number]
dtmf-relay rtp-nte
codec g711ulaw
!
dial-peer voice [dial-peer unique ID] pots /* for TDM originated calls */
service [gateway application name]
incoming called-number [dialed number]
direct-inward-dial
```

Step 4 Optionally, create another dial peer to do transfers using the Unified ICME label that is returned.

Step 5 Create the application in Call Studio. In the Call Studio application, the ReqICMLabel has two exit states: error and done. The done path grabs a transfer element to transfer the caller to that label. The gateway needs another dial peer to transfer the label it gets from this process (refer to previous Step step 4). If you want to do real transfers, you must have the transfer element set up inside the Call Studio application.

Step 6 Drag the ReqICMLabel element onto the application created in Call Studio and configure it.

Note This step is necessary to obtain a label from Unified ICME. For more information, see Call Studio ReqICMLabel Element to Pass Data.

Step 7 Save and deploy the application from Call Studio using the VoiceXML Service on the Operations Console.

Step 8 Install the Call Server, selecting only the Core Software component.

Step 9 Configure the Unified CVP VXML Server to communicate with the Call Server through the Operations Console.

Step 10 Transfer the application using File Transfer to the Unified CVP VXML Server. This automatically deploys the application on the selected Unified CVP VXML Server.

Configure the Unified CVP VXML Server (Standalone) Call Flow Model (Without ICM Lookup)

The following procedure describes how to configured Unified CVP VXML Server (standalone) call flow model:

Procedure

Step 1 Copy the following files from the Unified CVP VXML Server CD to the gateway flash memory using tftp: CVPSelfService.tcl

critical_error.wav

For example:

```
copy tftp: flash:CVPSelfService.tcl
copy tftp: flash:CVPSelfServiceBootstrap.vxml
copy tftp: flash:critical_error.wav
```

Step 2 Define the Unified CVP VXML Server applications on the gateway. The following lines show an example configuration:

```
service CVPSelfService flash:CVPSelfServiceBootstrap.vxml
!
service [gateway application name] flash:CVPSelfService.tcl
param CVPBackupVXMLServer 10.78.26.28
param CVPSelfService-port 7000
param CVPSelfService-app [name of application on the VXML Server, exactly how it
appears]
param CVPPrimaryVXMLServer 10.78.26.28
```

Note CVPSelfService is required. Backup server is optional. For the Tomcat Application Server, set the port to 7000.

After completing the gateway configuration, run the following to load and activate the applications:

```
call application voice load CVPSelfService
call application voice load [gateway application name]
```

Step 3 Define a dial-peer for the gateway application, for example:

```
dial-peer voice [dial-peer unique ID] voip /* for IP originated call * /
service [gateway application name]
incoming called-number [dialed number]
dtmf-relay rtp-nte
codec g711ulaw
!
dial-peer voice [dial-peer unique ID] pots /* for TDM originated calls */
service [gateway application name]
incoming called-number [dialed number]
direct-inward-dial
```

Step 4 Create the application in Call Studio. This application *must* have the same name as the CVPSelfService-app defined in the gateway configuration above.

Step 5

Options	Description
If there is an Operations Console	Save and deploy the Call Studio application locally. Create a Unified CVP VXML Server (Standalone) configuration,and upload and transfer the application script file to the required Unified CVP VXML Server or Unified CVP VXML Server (standalone).
If there is *not* an Operations Console	Save and deploy the Call Studio Application to the desired installed Unified CVP VXML Server. Then, on the Unified CVP VXML Server, run the deployallapps.bat file (c:/Cisco/CVP/VXMLServer/admin directory).

Note See User Guide for Cisco Unified CVP VXML Server and Unified Call Studio.

Sample Gateway Configuration

Unified CVP VXML Server:

```
application
 service CVPSelfService flash:CVPSelfServiceBootstrap.vxml
 service HelloWorld flash:CVPSelfService.tcl
 param CVPBackupVXMLServer 10.78.26.28
 param CVPSelfService-app HelloWorld
 param CVPSelfService-port 7000
 param CVPPrimaryVXMLServer 10.78.26.28
dial-peer voice 4109999 voip /* for IP originated call */
 service HelloWorld
 incoming called-number 88844410..
 dtmf-relay rtp-nte
 codec g711ulaw
 dial-peer voice 4109999 voip /* for TDM originated call */
 service HelloWorld
 incoming called-number 88844420..
 direct-inward-dial
```

Activate the Gateway Configuration

Activate the gateway configuration by entering these commands:

Procedure

Step 1 call application voice load CVPSelfService

Step 2 call application voice load HelloWorld

Takeback and Transfer in VoiceXML Scripts

Unified CVP provides the following takeback and transfer methods that you invoke from a VoiceXML script:

- Two B-Channel Transfer (TBCT) - A call transfer standard for ISDN interfaces. This feature enables a Cisco voice gateway to request an NI-2 switch to directly connect two independent calls. The two calls can be served by the same PRI or by two different PRIs on the gateway.
- Hookflash Relay - A brief interruption in the loop current that the originating call entity (PBX or Public Switch Telephone Network switch) does not interpret as a call disconnect. Instead, once the PBX or Public Switch Telephone Network switch senses the hookflash, it puts the current call on hold and provides a secondary dial tone, which allows Unified CVP VXML Server to transfer the caller to another destination.
- SIP Refer - VoiceXML applications can use a SIP REFER transfer instead of a blind or bridged transfer. This allows Unified CVP to remove itself from the call, to free up licensed Unified CVP VXML Server ports. Unified CVP cannot execute further call control or IVR operations after the label has been executed.

Configure Two B-Channel Transfer

This procedure describes now to configure Two B-Channel Transfer (TBCT) with Unified CVP from a VoiceXML script.

Procedure

Step 1 Configure the originating gateway for TBCT call transfer.

Step 2 Locate the following files on the Unified CVP VXML Server and copy them to flash memory on the gateway, using the tftp command:

en_holdmusic.wav

en_pleasewait.wav

survivability.tcl

CVPSelfService.tcl

CVPSelfServiceBootstrap.vxml

Step 3 Add the following lines to the gateway:

```
service takeback flash:survivability.tcl
param icm-tbct 1
```

Step 4 Configure the CVPSelfService application, as follows:

```
service [gateway application name] flash:CVPSelfService.tcl
param CVPBackupVXMLServer 10.78.26.28
param CVPSelfService-port 7000
param CVPSelfService-app [name of application on the VXML Server, exactly how it appears]
param CVPPrimaryVXMLServer 12.34.567.891
```

Note CVPSelfService is required. Backup server is optional. For Tomcat Application Server set the port to 7000.

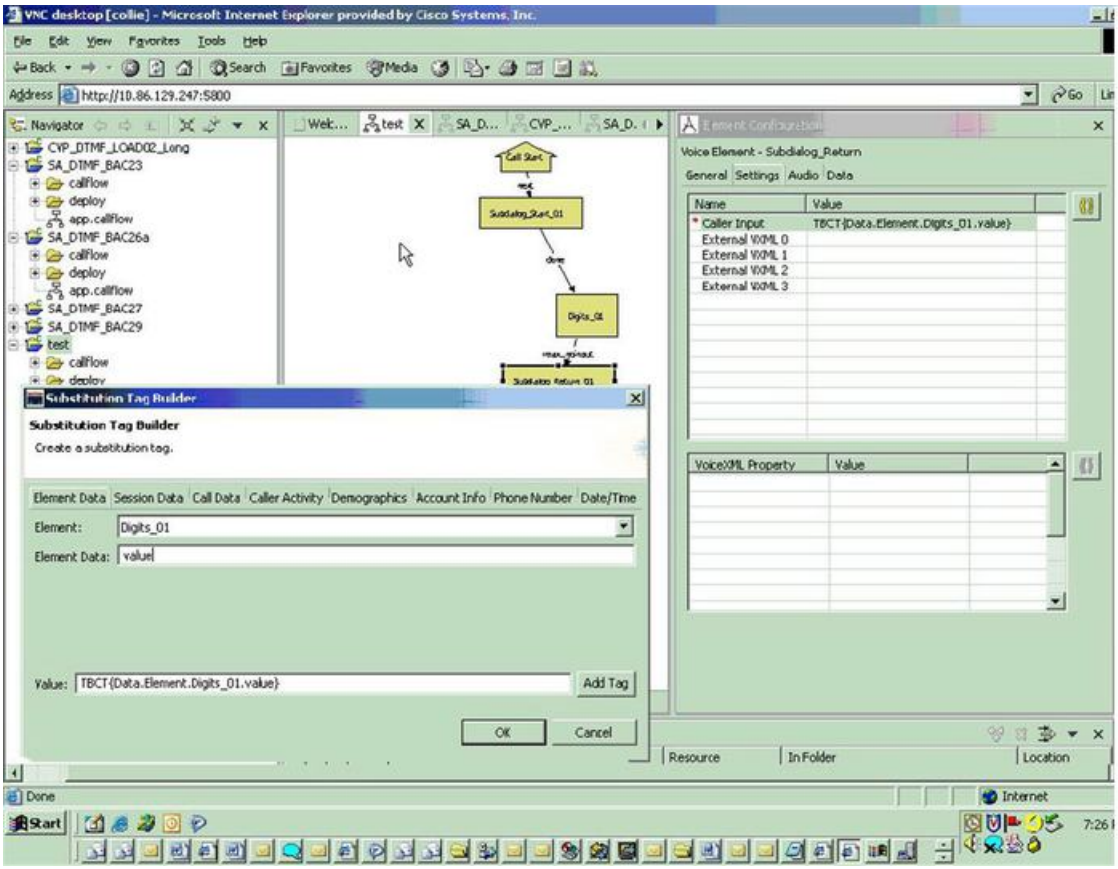
Step 5 From command line mode:

```
call application voice load takeback
call application voice load CVPSelfService
```

Step 6 Specify the target destination for the TBCT transfer either by entering the number manually, or dynamically by using caller input.

a) Manually. In the SubdialogReturn node in the Unified CVP VXML Server application, next to Caller Input in the Settings Tab, enter **TBCT<target_destination_number>**, where *target_destination_number* is the target destination of the TBCT transfer. For example:
TBCT8005551212

b) Dynamically. The target destination is created dynamically using input entered by the caller during the call. Click the **Substitution** icon next to the Caller Input variable and select substitution values. For example:



Configure Hookflash Relay

The following procedure describes how to configure Hookflash Relay for use with Unified CVP from VoiceXML scripts.

Procedure

Step 1 Configure the originating gateway for Hookflash Relay call transfer.

Step 2 Locate the following files on the Unified CVP VXML Server and copy them to flash memory on the gateway.

en_holdmusic.wav

en_pleasewait.wav

survivability.tcl

en_0.wav en_1.wav

en_2.wav en_3.wav

en_4.wav

en_5.wav

en_6.wav

en_7.wav

en_8.wav

en_9.wav

en_pound.wav

en_star.wav

Step 3 Add the following lines to the gateway:

```
service hookflash flash:survivability.tcl
```

Step 4 If you have not already done so, configure the CVPSelfService application:

```
service [gateway application name] flash:CVPSelfService.tcl
param CVPBackupVXMLServer 10.78.26.28
param CVPSelfService-port 7000
param CVPSelfService-app [name of application on the VXML Server, exactly how it appears]
param CVPPrimaryVXMLServer 10.78.26.28
```

Note CVPSelfService is required. Backup server is optional. For the Tomcat Application Server set the port to 7000.

Step 5 From the command line mode:

```
call application voice load hookflash
call application voice load CVPSelfService
```

Step 6 In the SubdialogReturn node in the Unified CVP VXML Server application, next to Caller Input in the Settings Tab, enter HF8005551212, replacing 8005551212 with the target destination of the hookflash transfer. The label can also be defined dynamically using digits entered by the caller in conjunction with the Unified CVP VXML Server substitution tags. If the switch requires a pause after the hookflash, insert commas between the HF and the transfer number. Each comma represents 100ms.

Configure SIP REFER

To configure SIP REFER for use with Unified CVP VXML Server from a VoiceXML script, follow this procedure:

Procedure

Step 1 Configure the gateway according to Configure the Unified CVP VXML Server (Standalone) Call Flow Model (Without ICM Lookup) or Configure the Unified CVP VXML Server (Standalone) with ICM Lookup Call Flow Model, according to your implementation.

Note The incoming dial-peer running the CVPSelfService application must be a VoIP dial-peer, not a POTS dial-peer.

Step 2 Specify the target destination for the REFER transfer in the Call Studio application by entering the number manually, or dynamically using caller input.

a) Manually. In the SubdialogReturn node in the Unified CVP VXML Server application, next to CallerInput in the Settings tab, enter RF<target_destination_number>, where target_destination_number is the target destination of the REFER transfer. For example, RF8005551212.
b) Dynamically. The target destination is created dynamically using input entered by the caller during the call. Click the **Substitution** icon next to the Caller Input variable and select the substitution values.

Step 3 The following configuration must be added to the gateway configuration for the handoff to survivability.tcl to occur and to send the REFER:

```
service takeback flash:survivability.tcl
```

Configure the Gateway for TDM to IP Calls

To transfer TDM call to IP, configure the gateway as follows:

Procedure

Step 1 To configure the number dialed from the TDM phone to match, use the following configuration:

```
dial-peer voice [dialpeer name] pots
service [gateway application name]
incoming called-number [number dialed from phone]
direct-inward-dial
```

Step 2 To configure the number dialed from the IP phone to match, use the following configuration:

```
dial-peer voice [dialpeer name] voip
service [gateway application name]
incoming called-number [number dialed from phone]
dtmf-relay rtp-nte
codec g711ulaw
```

Step 3 To configure the transfer node to transfer to a specific extension (for example, 5080001):

Note The gateway sends the call to Unified CM as a blind transfer.

```
dial-peer voice [dialpeer name] voip destination-pattern 508....

session target ipv4:[call manager subscriber IP]

dtmf-relay rtp-nte

codec g711ulaw

no vad
```

Step 4 To hear a ring tone on a blind transfer call:

- Select **Service** > **Service Parameters** in Unified CM.

- Select your server and the Unified CM Service.
- Set the Send H225 User Info Message setting to H225 Info for Call Progress Tone.

Pass Data to Unified ICME

In the Unified CVP VXML Server (standalone) with ICM Lookup call flow model, Unified ICME sends a label to Unified CVP. This process requires some configuration, which is explained below.

The Standalone with Request ICM Label variation of the Standalone call flow model performs a route request to Unified ICME, and then Unified ICME starts a script (new call). Unified ICME sees whatever the device puts in the new call message, then Unified ICME chooses a target, such as an agent, and sends a label back to the device. That route request to Unified ICME sends other information, such as ECC variables. Unified ICME can pass other ECC variables to Unified CVP. Also, you need to configure a Unified CVP VXML Server in the Unified CVP Call Server for the call flow model.

Configure the Connections

The following procedure describes how to set up a Unified CVP VXML Server that connects to the Unified CVP Call Server through the ICM Service, and the connection from the ICM Service to the PG.

Note The VRU PIM initiates the connection from the PG to the Call Server. The ICM Service listens for a connection from the VRU PIM.

Procedure

Step 1 Start the Unified CVP VXML Server. The Unified CVP VXML Server starts the VoiceXML Service using the DataFeed mechanism or the ReqICMLabel element.
The ReqICMLabel element allows a Call Studio script to pass caller input, call variables, and External Call Context (ECC) variables to a Unified ICME script. The ReqICMLabel must be inserted into a Call Studio script as a decision element. In Call Studio, the returned Unified ICME label contains a result which can be used by other elements in the same application, such as the Transfer or Audio element. The Transfer element sends instructions to the IOS Voice Browser to transfer the caller to the desired location.

Once the VoiceXML Service starts, it communicates with the ICM Service.

Step 2 From the Operations Console, add a Call Server.

Step 3 On the General tab, enable the ICM Service by clicking the **ICM** check box.

Step 4 On the ICM General Configuration tab, specify the VRU Connection Port (default is 5000). This is the port on which the ICM Service listens for a connection from the VRU PIM upon startup of the Call Server.

Note See Unified ICME documentation for instructions on configuring the VRU PIM to connect to a VRU (for example, Unified CVP).

Step 5 Set up the following items in Unified ICME: Dialed Number (DN), customer records which must match the Network VRU, call type, and a script.

When setting up the script, be aware that only the following script nodes are allowed:

- Label
- Select node
- LAA and MED agents

Caution The Run External Script node and queuing are not allowed.

Configure the Gateway for IP to TDM Calls

The following components are required for the gateway to process IP to TDM calls:

- Phones and numbers must be configured on the TDM switch.
- Gateway must be defined on Unified CM.
- Route pattern on Unified CM that sends the call to the gateway.
- Dial peer on the gateway that sends calls that must be configured.
- Dial 888800605x on the IP phone (this is a specific physical phone extension).

Procedure

Step 1 Configure the gateway to send the call to a particular Unified CVP VXML Server application, as follows:

```
dial-peer voice 8888 voip
service [gateway application name]
incoming called-number 888800....
dtmf-relay rtp-nte
codec g711ulaw
no vad
```

Step 2 To match the number in the Unified CVP VXML Server transfer node and send it out the T1 port to the G3 to its destination, use the following configuration:

```
dial-peer voice 8880 pots
destination-pattern 888800....
incoming called-number
direct-inward-dial
port 1/0:D
```

Configure a Cisco Multiservice IP-to-IP Gateway for Unified CM Connections

For information on configuring the Cisco IOS gateway for Unified CM connections, refer to the Cisco Multiservice IP-to-IP Gateway Software documentation.

SNMP Monitoring for the Unified CVP VXML Server

SNMP monitoring for the Unified CVP VXML Server is provided by default when a Call Studio application is created. CVPSNMPLogger logs error events it receives from the Unified CVP VXML Server. For example, this process allows you to configure sending a page to a technical support representative when a particular error alert is triggered on the customer site.

Note By default, CVPSNMPLogger is enabled when a new Call Studio application is created and deployed to the Unified CVP VXML Server.

Procedure

Step 1 To view CVPSNMPLogger for the Unified CVP VXML Server, access the Call Studio interface.

Step 2 Right-click the application and select **Properties** > **Cisco Unified CVP** > **General Settings** from Call Studio for each Call Studio application. CVPSNMPLogger displays in the Loggers list box.

Caution Do not remove CVPSNMPLogger. Doing so disables viewing of SNMP events and alerts.

CHAPTER 7

Configure Unified CVP Logging and Event Notifications

Unified CVP provides information about component device status and interaction through

- Logs, which are presented in text format and can be viewed using Cisco serviceability tools
- Statistics, which can be viewed using the Unified CVP Operations Console

This chapter also provides information about Unified CVP SNMP-Raise/Clear Mappings, and contains the following topics:

- Using Syslog, page 243
- Using Logs to Interpret Events, page 243
- VoiceXML Logs, page 246
- About Event Statistics, page 249
- Unified CVP SNMP-Raise/Clear Mappings, page 265

Using Syslog

Unified CVP allows you to configure the primary and backup syslog servers with the forked primary and forked backup servers. Failover from primary to backup server is not guaranteed. When the primary syslog server goes down (the entire machine, not just the syslog receiver application), Unified CVP relies on the host operating system and the Java Runtime Environment for notification that the destination is not reachable. Because the semantics of this notification do not guarantee delivery, Unified CVP cannot guarantee failover.

Using Logs to Interpret Events

You can use the **CVPLogMessages.xml** file to help interpret events. This file contains all messages (or notifications) on SNMP events and/or through Syslog.

Note The **CVPLogMessages.xml** file applies to all Unified CVP Services.

Each event in the **CVPLogMessages.xml** field containing information that must be useful for correcting any problems indicated by the event.

Note Be aware that the <resolution> field might not always contain as much information as the Troubleshooting Guide for Cisco Unified Customer Voice Portal or other Unified CVP documentation, and should be considered with all other resources when troubleshooting a problem.

The sections that follow provide information about editing, uploading, and downloading the CVPLogMessages.xml file from the Operations Console.

Editing the Log Messages XML File

The log messages XML file, `CVPLogMessages.xml`, defines the severity, destination (SNMP management station or Syslog server) and possible resolution for Unified CVP log messages. This file also identifies an event type identifier and message text identifier for each event. The text for these identifiers is stored in the resource properties file, `CVPLogMessagesRes.properties`.

Each Unified CVP Call Server, VXML Server, and Reporting Server has a log messages XML file and log message file. You can edit the `CVPLogMessages.xml` file on a particular Unified CVP server to customize the severity, destination and possible resolution for each event that the server generates. You can also edit the `CVPLogMessagesRes.properties` file to change the text of the message that is generated when an event occurs on that server.

Use any plain-text editor (one that does not create any markup) or XML editor to edit the `CVPLogMessages.xml` file. Use a resource file editor to edit the `CVPLogMessagesRes.properties` file. If a resource file editor is not available, use a text editor.

Message Element	Possible Values	What it Means
Name	Resource="identifier"	Identifies the event type described in the `CVPLogMessagesRes.properties` file.
Body	Resource="identifier"	Identifies the message text described in the `CVPLogMessagesRes.properties` file.
Severity	0 to 6	Identifies the Unified CVP Event Severity Levels of the event.
SendToSNMP	True or false	Set to true, to send this message, when logged, to an SNMP manager, if one is configured.
SendToSyslog	True or false	Set to true to send this message, when logged, to a Syslog server, if one is configured.

Message Element	Possible Values	What it Means
SNMPRaise	True or false	Set to true to identify this message, when logged, as an SNMP raise event, which the SNMP management station uses to initiate a task or automatically take an action. Set to false to identify this message as an SNMP clear when sent to an SNMP management station. An SNMP clear event usually corresponds to an SNMP raise event, indicating that the problem causing the raise has been corrected. An administrator on an SNMP management station can correlate SNMP raise events with SNMP clear events.

Unified CVP Event Severity Levels

The following table describes the available severity levels for Unified CVP events. You can set the severity level for an event by editing the log messages XML file, CVPLogMessages.xml, on the server that generates events. For instructions on editing this file, see Editing the Log Messages XML File.

Level	Severity	Purpose
EMERGENCY	0	System or service is unusable
ALERT	1	Action must be taken immediately
CRITICAL	2	Critical condition, similar to ALERT, but not necessarily requiring an immediate action
ERROR	3	An error condition that does not necessarily impact the ability of the service to continue to function
WARN	4	A warning about a bad condition, which is not necessarily an error
NOTICE	5	Notification about interesting system-level conditions, which are not errors
INFO	6	Information about internal flows or application or per-request information, not system-wide information

VoiceXML Logs

About VoiceXML Logs

VoiceXML logs record Unified CVP system-specific information, such as heartbeat status. By default, VoiceXML logs are stored in the \Cisco\CVP\logs\VXML folder.

The table that follows describes the logs that VoiceXML creates.

Log Type	Log Name	Description
Infrastructure	CVP.<timestamp>.log	Unified CVP logs for the VoiceXML Service: This includes Notice, Info, and Debug logs. With Debug turned on, you can also see Call, Message, and Method trace types of logs.
Error messages	Error.<timestamp>.log	Unified CVP error log: This contains any error that Unified CVP Services and message layer has generated.

Correlate Unified CVP/Unified ICME Logs with VXML Server Logs

When using the VXML Server option in the Unified CVP solution, you can correlate Unified CVP/Unified ICME logs with VoiceXML logs. Pass the Call ID to the VXML Server by URL. Building upon the URL used in the previous example, the URL is as follows:
http://10.78.26.28:7000/CVP/Server?application=Chapter1_HelloWorld&callid=XXXXX-XXXXX-XXXXXX-XXXXXX

Note Unified CVP VXML Server (by default) receives callid (which contains the call GUID), _dnis, and _ani as session variables in comprehensive mode even if the variables are not configured as parameters in the ToExtVXML array. If the variables are configured in ToExtVXML then those values are used. These variables are available to VXML applications as session variables, and displayed in the Unified CVP VXML Server log. This change is backwards compatible with the following script. That is, if you have added the following script, you do not change it. However, if you remove this script, you save an estimated 40 bytes of ECC variable space.

The following procedure describes how to configure logging.

Procedure

In the Unified ICME script, use the formula editor to set ToExtVXML[1]. Set the value of ToExtVXML[1] variable to `concatenate("callid=",Call.user.media.id)`

Note

- Always include Call ID when sending the call to the Unified CVP VXML Server using the Comprehen call flow model. The Call ID can also be used in Unified CVP VXML Server (standalone) solutions.
- When you concatenate multiple values, use a comma for the delimiter.
- The value of ICMInfoKeys must contain RouterCallKey, RouterCallDay, and RouterCallKeySequenceNu separated by a "-".

  For example,
  concatenate("ICMInfoKeys=",Call.RouterCallKey,"-",Call.RouterCallDay,"-",Call.RouterCallKeySequenceNu

See Passing Information to the External VoiceXML.

About Unified CVP VXML Server Logs

Unified CVP VXML Server logs record interactions between the Unified CVP VXML Server and the server that hosts the VoiceXML applications. By default, Unified CVP VXML Server logs are stored in the /Cisco/CVP/VXMLServer/logs folder.

The following table describes the logs that Unified CVP VXML Server creates:

Log Type	Log Name	Description
Unified CVP VXML Server Call Log	call_log<timestamp>.txt	Records a single line for every application visit handled by the Unified CVP VXML Server.
Unified CVP VXML Server Call Error Log	error_log<timestamp>.txt	Records errors that occur outside the realm of a particular application.
Unified CVP VXML Server Administration History Log	admin_history<timestamp>.txt	Records information from Unified CVP VXML Server administration scripts.

The Unified CVP VXML Server Call Error Log contains the following error codes:

- Error Code 40 -- System Unavailable

  This is returned if the application server is unavailable (shutdown, network connection disabled, and so forth)

- Error Code 41 -- App Error

  This is returned if some Unified CVP VXML Server-specific error occurs (For example, java exception).

- Error Code 42 -- App Hangup

  This is returned to Unified CVP if the Hang Up element is used without being preceded by a Subdialog_Return element.

  Note If the application is configured correctly, this does not occur.

- Error Code 43 -- Suspended

  This is returned if the Unified CVP VXML Server application is suspended.

- Error Code 44 -- No Session Error

  This is returned when an emergency error occurs (for example, an application is called that has not been loaded in the Unified CVP VXML Server application).

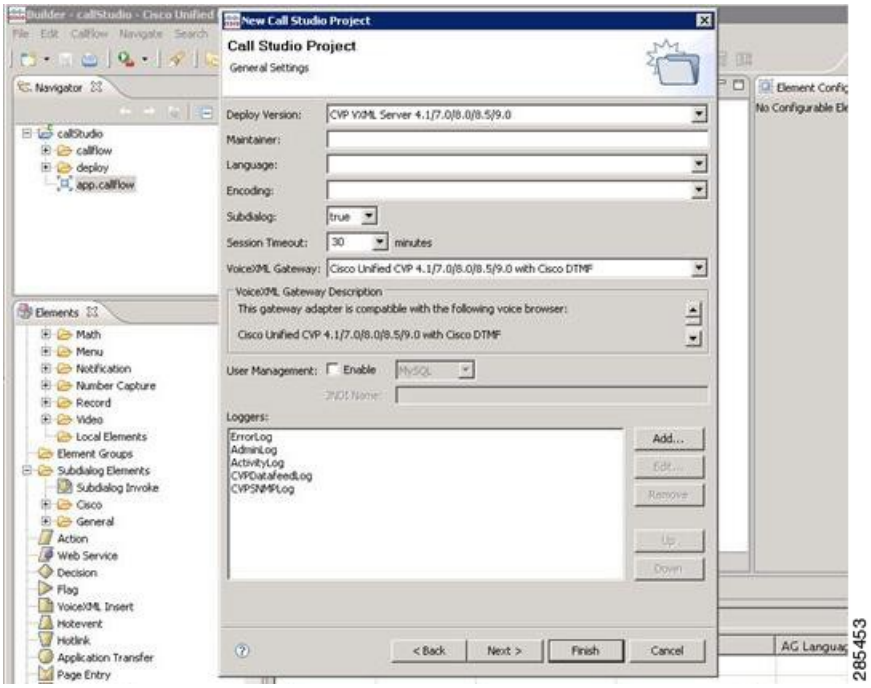
- Error Code 45 -- Bad Fetch

  This is returned when the Unified CVP VXML Server encounters a bad fetch situation. This code is returned when a .wav file or an external grammar file is not found.

About VoiceXML Application Logging

The Unified CVP VXML Server creates several logs for each individual VoiceXML application. By default, these application logs - with the exception of CVPDatafeedLog and CVPSNMPLog - are stored in the /Cisco/CVP/VXMLServer/applications/<NAME of APPLICATION>/logs folder.

Configure these logs using Call Studio:



Note See Element Specifications for Cisco Unified CVP Unified CVP VXML Server and Unified Call Studio for information about configuring loggers.

The following table describes the logs that are created for each application:

Application Logger Type	Log Name	Description
ActivityLog	activity_log<timestamp>.txt **Note** Log files are stored in the ActivityLog directory.	Records all application activity, showing which elements are entered and exited during a call. Default setting: on

Application Logger Type	Log Name	Description
ErrorLog	error_log<timestamp>.txt **Note** Log files are stored in the ErrorLog directory.	Records all error messages for the application. Default setting: on
AdminLog	admin_history<timestamp>.txt **Note** Log files are stored in the AdminLog directory.	Records information from application-specific administration scripts. Default setting: on
CVPDatafeedLog	CVPDatafeed.log. **Note** This log is stored in /Cisco/CVP/logs/VXML folder.	Listens for logging events and provides Unified CVP VXML Server and VoiceXML Service data to the Unified CVP Reporting Server. The Unified CVP Reporting Server stores this information in a reporting database so that it is available for later review. One CVPDatafeedLog is created per application. Default setting: on **Note** The VoiceXML Service can be started by adding this logger in the VoiceXML application.
CVPSNMPLog	CVPSNMP.log. **Note** This log is stored in /Cisco/CVP/logs/VXML folder.	Listens for a set of events and sends information about these events to the SNMP log, Syslog, or Unified CVP log. Default setting: on
DebugLog	debug_log<timestamp>.txt. **Note** Log files are stored in the DebugLog directory.	Creates a single file per call that contains all HTTP requests and responses that occurred between a IOS Gateway and Unified CVP VXML Server during the call session. Default setting: off

About Event Statistics

You can monitor the following statistics through the Operations Console Control Center:

- Device statistics
- Infrastructure statistics
- ICM Service call statistics
- IVR Service call statistics
- SIP Service call statistics

- Gateway statistics
- VXML Server statistics
- Reporting Server statistics

Infrastructure Statistics

Unified CVP infrastructure statistics include realtime and interval data on the Java Virtual Machine (JVM), threading, and Licensing.

You can access these statistics by choosing Control Center from the System menu and then selecting a device. See the Operations Console topic *Viewing Infrastructure Statistics* for more information.

Access infrastructure statistics either by:

- Selecting **System** > **Control Center**, selecting a device, clicking the Statistics icon in the toolbar, and then selecting the **Infrastructure** tab.
- Selecting a device type from the **Device Management** menu, selecting a device, clicking the Statistics icon in the toolbar, and then selecting the **Infrastructure** tab.

The following table describes Licensing statistics.

Table 36: Licensing Statistics

Statistic	Description
Realtime Statistics	
Port Licenses Available	The number of port licenses available for the processing of new calls. Exactly one port license is used per call, independent of the call's traversal through the individual Call Server services.
Current Port Licenses in Use	The number of port licenses currently in use on the Call Server. One port license is used per call, independent of the call's traversal of the individual Call Server services.
Current Port Licenses State	There are four threshold levels of port license usage: safe, warning, critical and failure. An administrator may set the required percentage of port licenses in use needed to reach a given threshold level, with the exception of the failure level which is reached when the number of ports checked out is equal to the number of licenses ports.
Interval Statistics	
Start Time	The time the system started collecting statistics for the current interval.
Duration Elapsed	The amount of time that has elapsed since the start time in the current interval.

Statistic	Description
Interval Duration	The interval at which statistics are collected. The default value is 30 minutes.
Total New Port License Requests	The number of port license checkout requests made in the current interval. For each port license checkout request, this metric is increased by one, regardless of whether if checks out a new port license.
Average License Requests/Minute	The average number of port license checkout requests made per minute in the current interval. This metric is calculated by dividing the port license requests metric by the number of minutes elapsed in the current interval.
Maximum Port Licenses Used	The maximum number of port licenses used during this time interval.
Aggregate Statistics	
Start Time	The time the service started collecting statistics.
Duration Elapsed	The amount of time that has elapsed since the service start time.
Total New Port License Requests	The number of port license checkout requests made since the system was started. For each port license checkout, this metric is increased by one, regardless of whether if checks out a new port license.
Average License Requests /Minute	The average number of port license checkout requests made per minute since the system was started. This metric is calculated by dividing the aggregate port license requests metric by the number of minutes elapsed since the system was started.
Peak Port Licenses Used	The peak number of simultaneous port licenses used since the start of the system. When a port checkout occurs, this metric is set to the current port licenses in use metric if that value is greater than this metric's current peak value.
Total Denied Port License Requests	The number of port license checkout requests that were denied since the start of the system. A port license checkout request is denied if the number of port licenses checked out at the time of the request is equal to the total number of port license available. When a port license checkout is denied, the call does not receive regular treatment (the caller may hear a busy tone or an error message).

The following table describes thread pool system statistics. The thread pool is a cache of threads, used by Unified CVP components only, for processing relatively short tasks. Using a thread pool eliminates the waste of resources encountered when rapidly creating and destroying threads for these types of tasks.

Table 37: Thread Pool Realtime Statistics

Statistic	Description
Realtime Statistics	
Idle Threads	The number of idle threads waiting for some work
Running Threads	The number of running thread pool threads currently processing some work.
Core Threads	The number of thread pool threads that will never be destroyed no matter how long they remain idle
Maximum Threads	The maximum number of thread pool threads that will ever exist simultaneously
Peak Threads Used	The peak number of thread pool threads ever simultaneously tasked with some work to process

The following table describes Java Virtual Machine statistics.

Table 38: Java Virtual Machine (JVM) Realtime Statistics

Statistic	Description
Realtime Statistics	
Peak Memory Usage	The greatest amount of memory used by the Java Virtual machine since startup. The number reported is in megabytes and indicates the peak amount of memory ever used simultaneously by this Java Virtual Machine.
Current Memory Usage	The current number of megabytes of memory used by the Java Virtual Machine.
Total Memory	The amount of memory in megabytes available to the Java Virtual Machine. The number indicates how much system memory is available for the Java Virtual Machine.
Available Memory	The amount of available memory in the Java Virtual Machine. The number reported is in megabytes and indicates how much of the current system memory claimed by the Java Virtual Machine is not currently being used.
Threads in Use	The number of threads currently in use in the Java Virtual Machine. This number includes all of the Unified CVP standalone and thread pool threads, and those threads created by the Web Application Server running within the same JVM.

Statistic	Description
Peak Threads in Use	The greatest amount of threads used simultaneously in the Java Virtual Machine since startup. The peak number of threads used by thw Java Virtual Machine includes all Unified CVP standalone and thread pool threads, and threads created by the Web Application Server running within the same JVM.
Uptime	The time that the Java Virtual Machine has been running. This time is measured in hh:mm:ss and shows the amount of elapsed time since the Java Virtual Machine process began.

ICM Service Call Statistics

The ICM Service call statistics include data on calls currently being processed by the ICM service, new calls received during a specified interval, and total calls processed since start time.

Access ICM Service statistics either by:

- Selecting **System** > **Control Center**, selecting a CVP Call Server, clicking the **Statistics** icon in the toolbar, and then selecting the **ICM** tab.
- Selecting **Device Management** > **CVP Call Server**, selecting a Call Server, clicking the **Statistics** icon in the toolbar, and then selecting the **ICM** tab.

The following table describes ICM Service call statistics.

Table 39: ICM Service Call Statistics

Statistic	Description
Realtime Statistics	
Active Calls	The current number of calls being serviced by the Unified Intelligent Contact Management (Unified ICM) Server for a Unified CVP Call Server. This value represents the calls currently being serviced by the Unified ICM for the Unified CVP Call Server for follow-on routing to a Contact Center agent.
Active SIP Call Legs	The ICM Server can accept VoIP calls that originate using either the Session Initiation Protocol (SIP). Active SIP Call Legs indicates the number of calls received by the Unified ICM Server from the Unified CVP Call Server using the SIP protocol.
Active VRU Call Legs	The current number of calls receiving Voice Response Unit (VRU) treatment from the Unified ICM Server. The VRU treatment includes playing pre-recorded messages, asking for Caller Entered Digits (CED) or Speech Recognition Techniques to understand the customer request.

Statistic	Description
Active ICM Lookup Requests	Calls originating from an external Unified CVP VXML Server need call routing instructions from the Unified ICM Server. Active Lookup Requests indicates the current number of external Unified CVP VXML Server call routing requests sent to the ICM Server.
Active Basic Service Video Calls Offered	The current number of simultaneous basic service video calls being processed by theUnified ICM service where video capability was offered.
Active Basic Service Video Calls Accepted	The current number of simultaneous calls that were accepted as basic service video calls and are being processed by the Unified ICM service.
Interval Statistics	
Start Time	The time at which the current interval began.
Duration Elapsed	The amount of time that has elapsed since the current interval began.
Interval Duration	The time interval at which statistics are collected. The default value is 30 minutes.
New Calls	The number of new calls received by the Unified ICM application for follow-on Voice Response Unit (VRU) treatment and routing to a Contact Center agent during the current interval.
SIP Call Legs	The Unified ICM application accepts VoIP calls that originate from the Session Initiation Protocol (SIP) Protocol. Interval SIP Call Legs is an interval specific snapshot metric indicating the number of calls received by the ICM application from SIP during the current interval.
VRU Call Legs	The number of calls receiving VRU treatment from the Unified ICM application. The VRU treatment includes playing pre-recorded messages, asking for Caller Entered Digits (CED) or speech recognition techniques to understand the customer request during the current interval.
ICM Lookup Requests	Calls originating in an external Unified CVP VXML Server need call routing instructions from the Unified ICM application. Interval Lookup Requests is an interval specific metric indicating the number of external Unified CVP VXML Server call routing requests sent to the Unified ICM application during the current interval.
Basic Service Video Calls Offered	The number of offered basic service video calls processed by the Unified ICM service during the current interval.
Basic Service Video Calls Accepted	The number of basic service video calls accepted and processed by the Unified ICM service during the current interval.
Aggregate Statistics	

Statistic	Description
Start Time	The time the service started collecting statistics.
Duration Elapsed	The amount of time that has elapsed since the service start time.
Total Calls	The total number of new calls received by the Unified ICM application for follow-on VRU treatment and routing to a Contact Center agent since system start time.
Total SIP Call Legs	The Unified ICM application can accept VoIP calls that originate from the Session Initiation Protocol (SIP) Protocol. Total SIP Switch Legs is a metric indicating the number of calls received by the ICM application by SIP since system start time.
Total VRU Call Legs	The number of calls that have received VRU treatment from the Unified ICM application since system start time. The VRU treatment includes playing pre-recorded messages, asking for Caller Entered Digits (CED) or Speech Recognition Techniques to understand the customer request.
Total ICM Lookup Requests	Calls originating in an external Unified CVP VXML Server need call routing instructions from the Unified ICM application. Total Lookup Requests is a metric indicating the total number of external Unified CVP VXML Server call routing requests sent to the Unified ICM application since system start time.
Total Basic Service Video Calls Offered	The number of newly offered basic service video calls processed by the Unified ICM service since system start time.
Total Basic Service Video Calls Accepted	The number of new basic service video calls accepted and processed by the Unified ICM service since system start time.

IVR Service Call Statistics

The IVR service call statistics include data on calls currently being processed by the IVR service, new calls received during a specified interval, and total calls processed since the IVR service started.

Access IVR Service statistics either by:

- Selecting **System** > **Control Center**, selecting a Call Server, clicking the **Statistics** icon in the toolbar, and then selecting the **IVR** tab.
- Selecting **Device Management** > **CVP Call Server**, selecting a Call Server, clicking the **Statistics** icon in the toolbar, and then selecting the **IVR** tab.

The following table describes the IVR Service call statistics.

Table 40: IVR Service Call Statistics

Statistic	Description
Realtime Call Statistics	
Active Calls	The number of active calls being serviced by the IVR service.
Active HTTP Requests	The number of active HTTP requests being serviced by the IVR service.
Interval Statistics	
Start Time	The time the system starts collecting statistics for the current interval.
Duration Elapsed	The amount of time that has elapsed since the start time in the current interval.
Interval Duration	The interval at which statistics are collected. The default value is 30 minutes.
Peak Active Calls	Maximum number of active calls handled by the IVR service simultaneously.
New Calls	Metric that counts the number of New Call requests received from the IOS Gateway. A New Call includes the Switch leg of the call and the IVR leg of the call. This metric counts the number of New Call Requests received by the IVR Service.
Calls Finished	Metric that counts the number of Unified CVP Calls that have finished during this interval. A Call, for the purpose of the Call Finished metric, includes both the Switch leg and the IVR leg of the Unified CVP call. When both legs of the call are finished, the *Calls Finished* metric increases.
Average Call Latency	The average amount of time in milliseconds that it takes the IVR Service to process a New Call or Call Result Request.
Maximum Call Latency	The maximum amount of time in milliseconds that it has taken for the IVR Service to process a New Call Request or a Request Instruction Request.
Minimum Call Latency	The minimum amount of time in milliseconds it took for the IVR Service to process a New Call Request or a Request Instruction Request.

Statistic	Description
Peak Active HTTP Requests	Active HTTP Requests is a metric that indicates the current number of simultaneous HTTP requests being processed by the IVR Service. Peak Active Requests is a metric that represents the maximum simultaneous HTTP requests being processed by the IVR Service.
Total HTTP Requests	The number of HTTP Requests received from a client by the IVR Service.
Average HTTP Requests/second	The average number of HTTP Requests the IVR Service receives per second.
Peak Active HTTP Requests/second	HTTP Requests per Second is a metric that represents the number of HTTP Requests the IVR Service receives each second from all clients. Peak HTTP Requests per Second is the maximum number of HTTP Requests that were processed by the IVR Service in any given second. This is also known as high water marking.
Aggregate Statistics	
Start Time	The time the service started collecting statistics.
Duration Elapsed	The amount of time that has elapsed since the service start time.
Total New Calls	Metric that counts the number of New Call requests received from the IOS GatewayUsing Unified ICME Warm. A New Call includes the Switch leg of the call and the IVR leg of the call. Total New Calls is a metric that represents the number of new calls received by the IVR Service since system startup.
Peak Active Calls	The maximum number of simultaneous calls processed by the IVR Service since the service started.
Total HTTP Requests	Metric that represents the number of HTTP Requests received from all clients. This metric is the total number of HTTP Requests received by the IVR Service since system startup.
Peak Active HTTP Requests	Peak Active HTTP Requests is a metric that indicates the current number of simultaneous HTTP requests processed by the IVR Service. Maximum number of active HTTP requests processed at the same time since the IVR service started. This is also known as high water marking.
Total Agent Video Pushes	The number of videos pushed by agents since system start time.
Total Agent Initiated Recordings	The number of video recordings by agents since system start time.

Statistic	Description
Total Agent VCR Control Invocations	The number of video VCR controls invoked by agents since system start time.

SIP Service Call Statistics

The SIP service call statistics include data on calls currently being processed by the SIP service, new calls received during a specified interval, and total calls processed since the SIP service started.

Access SIP Service statistics either by:

- Selecting **System** > **Control Center**, selecting a Call Server, clicking the **Statistics** icon in the toolbar, and then selecting the **SIP** tab.
- Selecting **Device Management** > **CVP Call Server**, selecting a Call Server, clicking the **Statistics** icon in the toolbar, and then selecting the **SIP** tab.

The following table describes the SIP Service call statistics.

Table 41: SIP Service Call Statistics

Statistic	Description
Realtime Statistics	
Total Call Legs	The number of SIP call legs being handled by the SIP service. A call leg is also known as a SIP dialog. The metric includes incoming, outgoing and ringtone type call legs. For each active call in the SIP service, there is an incoming call leg, and an outgoing call leg to the destination of the transfer label.
Active Basic Service Video Calls Offered	The number of basic service video calls in progress where video capability was offered.
Active Basic Service Video Calls Answered	The number of basic service video calls in progress where video capability was answered.
Interval Statistics	
Start Time	The time the system started collecting statistics.
Duration Elapsed	The amount of time that has elapsed since the start time.
Interval Duration	The interval at which statistics are collected. The default value is 30 minutes.
New Calls	The number of SIP Invite messages received by Unified CVP in the current interval. It includes the failed calls, and calls rejected due to the SIP service being out of service.

Statistic	Description
Connects Received	The number of CONNECT messages received by SIP service to perform a call Transfer, in the last statistics aggregation interval. Connects Received includes the regular Unified CVP transfers, and Refer transfers. Any label coming from the ICM service is a CONNECT message, whether it is a label to send to the VRU or a label to transfer to an agent.
Avg Latency Connect to Answer	The period of time between the CONNECT from ICM and when the call is answered. The metric includes the average latency computation for the calls that have been answered in the last statistics aggregation interval.
Failed SIP Transfers (Pre-Dialog)	The number of failed SIP transfers since system start time. When Unified CVP attempts to make a transfer to the first destination of the call, it sends the initial INVITE request to set up the caller with the ICM routed destination label. The metric does not include rejections due to the SIP Service not running. The metric includes failed transfers that were made after a label was returned from the ICM Server in a CONNECT message.
Failed SIP Transfers (Post-Dialog)	The number of failed re-invite requests on either the inbound or outbound legs of the call during the interval. After a SIP dialog is established, re-INVITE messages perform transfers. Re-invite requests can originate from the endpoints or else be initiated by a Unified CVP transfer from the Unified ICME script. This counter includes failures for both kinds of re-invite requests.
Basic Service Video Calls Offered	The number of basic service video calls offered in the current interval.
Basic Service Video Calls Answered	The number of basic service video calls answered in the current interval.
Aggregate Statistics	
Start Time	The time the service started collecting statistics.
Duration Elapsed	The amount of time that has elapsed since the service start time.
Total New Calls	The number of SIP Invite messages received by Unified CVP since system start time. It includes the failed calls, and calls rejected due to the SIP service being out of service.
Connects Received	The number of Connect messages received by SIP service to perform a Unified CVP Transfer, since system start time. Connects Received includes the regular Unified CVP transfers, and Refer transfers. Any label coming from the ICM service is a Connect message, whether it is a label to send to the VRU or a label to transfer to an agent.

Statistic	Description
Avg Latency Connect to Answer	The time between the Connect from ICM and when the call is answered. The metric includes the average latency computation for all the calls that have been answered since system start up time.
Failed SIP Transfers (Pre-Dialog)	The total number of failed transfers on the first CVP transfer since system start time. A SIP dialog is established after the first CVP transfer finishes. The metric does not include rejections due to SIP being out of service. The metric includes failed transfers that are after a label is returned from the ICM in a CONNECT message.
Failed SIP Transfers (Post-Dialog)	The number of failed re-invite requests on the inbound or outbound legs of the call since start time. After a SIP dialog is established, re-INVITE messages perform transfers. Re-invite requests can originate from the endpoints or initiated by a Unified CVP transfer from the Unified ICME script. This counter includes failures for re-invite requests.
Total Basic Service Video Calls Offered	The number of basic service video calls offered since system start time.
Total Basic Service Video Calls Answered	The number of basic service video calls answered since system start time.

Gateway Statistics

Gateway statistics include the number of active calls, available memory, and CPU utilization.

Procedure

To obtain gateway statistics:

Procedure

Step 1 Choose **System** > **Control Center**.

Step 2 Select the **Device Type** tab in the left pane, then select **Gateways**.
Gateways are listed in the right pane.

Step 3 Select the gateway by clicking on its link under the Hostname column.
the Edit Gateway Configuration window opens.

Step 4 Select the Statistics icon in the toolbar.

What to Do Next

See *Administration Guide for Cisco Unified Customer Voice Portal* for device statistics.

Gateway Statistics

The following table describes gateway statistics.

Statistic	Description
Active Calls	Number of currently active calls handled by the gateway. For example, Total call-legs: 0 no active calls

Free Memory	Free memory, for example:
	`Processor memory free: 82%`
	`I/O memory free: 79%`

CPU Utilization	CPU utilization, for example: `CPU utilization for five seconds: 3%/3%; one minute: 3%; five minutes: 4%`

Trunk Utilization Reporting

You can configure IOS gateways to report on truck utilization. The configuration involves two pieces:

- Configuring the Call Server using the Operations Console to request reporting from a given gateway.
- Configuring the gateway to respond to trunk utilization reporting requests.

To configure Unified CVP to provide trunk utilization reporting, complete these steps:

1. In the Operations Console, select: **Device Management** > **Call Server** > **ICM (tab)** > **Advanced Configuration**..
2. Under *Trunk Utilization*, select **Enable Gateway Trunk Reporting**
3. In the same section, associate the gateway(s) that you want to send truck information to the Call Server.
4. Add the following configuration to the gateway configuration:

```
voice class resource-group 1
resource cpu 1-min-avg threshold high 80 low 60
resource ds0
resource dsp
resource mem total-mem
periodic-report interval 30

sip-ua
rai target ipv4:10.86.129.11 resource-group 1
rai target ipv4:10.86.129.24 resource-group 1
```

RAI Information on SIP OPTIONS (CVP Server Group Heartbeats)

If a resource availability indicator (RAI) is desired on SIP OPTIONS, the option override host setting can be used with server group heartbeating. When one or more Unified CVPs are sending OPTIONS heartbeats to the gateway, RAI trunk utilization information is not normally sent in the 200 OK response, unless an RAI target is configured.

CLI like the following can be added in IOS to have RAI information sent to CVP in the response:

```
sip-ua
rai target dns:cvp.cisco.com resource-group 1
```

Note Trunk Utilization data is only written to the Unified CVP database when RAI OPTIONS are sent from the gateway to Unified CVP targets. When Unified CVP is using server group heartbeats to the gateway, the RAI data in the response is only marks the element as UP or DOWN (overloaded resources) in the server group.

Unified CVP VXML Server Statistics

The Operations Console displays realtime, interval, and aggregate Unified CVP VXML server statistics.

Access Unified CVP VXML server statistics either by:

- Selecting **System** > **Control Center**, selecting a Unified CVP VXML server, and then clicking the Statistics icon in the toolbar.
- Selecting **Device Management** > **VXML Server** (or Unified CVP VXML server (Standalone)), selecting a Unified CVP VXML server, and then clicking the Statistics icon in the toolbar.

The following table describes the statistics reported by the Unified CVP VXML server.

Table 42: Unified CVP VXML Server Statistics

Statistic	Description
Real Time Statistics	
Active Sessions	The number of current sessions being handled by the Unified CVP VXML server.
Active ICM Lookup Requests	The number of current ICM requests being handled by the Unified CVP VXML server.
Interval Statistics	
Start Time	The time when the current interval began.
Duration Elapsed	The time that has elapsed since the start time in the current interval.

Statistic	Description
Interval Duration	The interval at which statistics are collected. The default is 30 minutes.
Sessions	The number of sessions in the Unified CVP VXML server.
Reporting Events	The number of events sent to the Reporting Server from the Unified CVP VXML server.
ICM Lookup Requests	The number of requests from the Unified CVP VXML server to the ICM Service.
ICM Lookup Responses	The number of responses to failed and successful ICM Lookup Requests that the ICM Service sends to the Unified CVP VXML server. In the case that multiple response messages are sent back to the Unified CVP VXML server to a single request, this metric increases per response message from the ICM Service.
ICM Lookup Successes	The number of successful requests from the Unified CVP VXML server to the ICM Service in the current interval.
ICM Lookup Failures	The number of requests from the Unified CVP VXML server to the ICM Service in the current interval. This metric increases when an ICM failed message is received or when the Unified CVP VXML server generates the failed message.
Aggregate Statistics	
Start Time	The time when the current interval has begins.
Duration Elapsed	The time since the current interval began.
Total Sessions	The number of sessions in the Unified CVP VXML server since startup.
Total Reporting Events	The number of reporting events sent from the Unified CVP VXML server since startup.
Total ICM Lookup Requests	The number of requests from the Unified CVP VXML server to the ICM Service. For each ICM lookup request (successful or failed), this metric increases by one.
Total ICM Lookup Responses	The number of responses the ICM Service has sent to the Unified CVP VXML server since startup. For each ICM lookup request (successful or failed), this metric increases by one. When multiple response messages are sent back to the Unified CVP VXML server to a single request, this metric increases per response message from the ICM Service.

Statistic	Description
Total ICM Lookup Success	The number of requests from the Unified CVP VXML server to the ICM Service since startup. For each ICM lookup request that succeeded, this metric increases one.
Total ICM Lookup Failures	The number of requests from the Unified CVP VXML server to the ICM Service since startup. For each ICM lookup request that failed, this metric increases by one. This metric will increase when an ICM failed message was received or in the case the Unified CVP VXML server generates a failed message.

See the *Administration Guide for Cisco Unified Customer Voice Portal* for Infrastructure Statistics and Device Statistics.

Reporting Server Statistics

Reporting Server statistics include the total number of events received from the IVR, SIP, and VoiceXML services.

Access Reporting Server statistics either by:

- Selecting **System** > **Control Center**, selecting a Reporting Server, and then clicking the **Statistics** icon in the toolbar.
- Selecting **Device Management** > **CVP Reporting Server**, selecting a Reporting Server, and then clicking the Statistics icon in the toolbar.

The following table describes the Reporting Server statistics.

Table 43: Reporting Server Statistics

Statistic	Description
Interval Statistics	
Start Time	The time the system began collecting statistics.
Duration Elapsed	The amount of time that has elapsed since the start time.
Interval Duration	The interval at which statistics are collected. The default value is 30 minutes.
VXML Events Received	The number of reporting events received from the VoiceXML Service. For each reporting event received from the VoiceXML Service, this metric increases by one.
SIP Events Received	The number of reporting events received from the SIP Service during this interval. For each reporting event received from the SIP Service, this metric increases by one.

Statistic	Description
IVR Events Received	The number of reporting events received from the IVR service in the interval. For each reporting event received from the IVR service, this metric increases by one.
Database Writes	The number of writes to the database made by the Reporting server during the interval. For each write, this metric increases one.
Aggregate Statistics	
Start Time	The time the service started collecting statistics.
Duration Elapsed	The amount of time that has elapsed since the service start time.
VXML Events Received	The number of reporting events received from the VoiceXML Service since the service started. For each reporting event received from the VoiceXML Service, this metric increases by one.
SIP Events Received	The number of reporting events received from the SIP Service since the service started. For each reporting event received from the SIP Service, this metric increases by one.
IVR Events Received	The number of reporting events received from the IVR Service since the service started. For each event received, this metric increases by one.
Database Writes	The number of writes to the database made by the Reporting server during since startup. For each write, this metric increases by one.

Unified CVP SNMP-Raise/Clear Mappings

The following log messages are SNMP-enabled by default. Administrators can define a unique alarm within their SNMP management station for all SNMP Raise events emitted by a system. These alarms are usually cleared automatically using one or more corresponding SNMP Clear events when the condition is resolved. The tables below list a mapping of Unified CVP SNMP Raise events with their corresponding SNMP Clears.

Note Raises are listed first, with their corresponding clears below them.

Table 44: Messaging Layer

Raise ID	Clear ID	Event Name
7		ADAPTER_INITIALIZATION_FAILURE
	8	ADAPTER_INITIALIZATION_SUCCESS

Raise ID	Clear ID	Event Name
9		PLUGIN_INITIALIZATION_FAILURE
	10	PLUGIN_INITIALIZATION_SUCCESS
15		SEND_QUEUE_THRESHOLD_REACHED
	20	SEND_QUEUE_SIZE_CLEAR

Table 45: Infrastructure

Raise ID	Clear ID	Event Name
9005		LICENSING
	1003	[AUDIT] "The system has started up."
9007		PORT_THRESHOLD
	9008	PORT_THRESHOLD
9014		SHUTDOWN
	1003	[AUDIT] "The system has started up."
	1004	[AUDIT] "The system has completely shutdown."
9016		SERVER_SETUP - "CCBUSNMPAgent Server setup failed because XXX"
	9015	SERVER_SETUP - "CCBUSNMPAgent Server setup on port YYY"
1011		HEARTBEATS_STOPPED - "Heartbeats from XXX stopped..."
	1014	RECEIVED_STATE_MSG - "StateManager: Subsystem [XXX] reported change to..."
1012		STATE_MANAGER_STARTUP_FAILURE
	1003	[AUDIT] "The system has started up."
1020		STARTUP
	1003	[AUDIT] "The system has started up."
1024		SERVLET_STARTUP

Raise ID	Clear ID	Event Name
	1003	[AUDIT] "The system has started up."
1025		START - "Could not start XXX due to: YYY"
	1003	[AUDIT] "The system has started up."
1033		START - "No Subsystems have been started..."
	1026	START - "All Subsystems have been started."
1035		LICENSE_EXPIRATION
	1003	[AUDIT] "The system has started up."

Table 46: Unified ICME

Raise ID	Clear ID	Event Name
2001		LOGMSG_ICM_SS_MSGBUS_SHUTDOWN
	2003	LOGMSG_ICM_SS_MSGBUS_ACTIVE
2002		LOGMSG_ICM_SS_PIM_SHUTDOWN
	2004	LOGMSG_ICM_SS_PIM_ACTIVE
2005		LOGMSG_ICM_SS_HEARTBEAT_FAILURE
	2012	LOGMSG_ICM_SS_INSERVICE_STATE
2006		LOGMSG_ICM_SS_STATE
	2012	LOGMSG_ICM_SS_INSERVICE_STATE

Table 47: Reporting

Raise ID	Clear ID	Event Name
4005		REPORTING_SS_ERROR_RAISE
	1026	START - "All Subsystems have been started."
4006		REPORTING_DB_PURGE_FAILED
	4007	REPORTING_DB_PURGE_COMPLETED

Raise ID	Clear ID	Event Name
4010		REPORTING_DB_BACKUP_FAILED
	4011	REPORTING_DB_BACKUP_COMPLETED
4014		REPORTING_DB_ALERT_MSG
	N/A	Not applicable
4017		REPORTING_DB_STARTING_PURGE
	4007	REPORTING_DB_PURGE_COMPLETED
	4009	REPORTING_DB_EMERGENCY_PURGE_COMPLETED
4018		REPORTING_DB_REMAINDER_DATA
	4019	REPORTING_DB_NO_REMAINDER_DATA

Table 48: IVR

Raise ID	Clear ID	Event Name
3014		VBCLIENT_REMOVED
	3013	VBCLIENT_ADDED
3027		VBCLIENT_SHUT_DOWN
	3026	VBCLIENT_RESTARTED
3021		VBSERVLET_STATE_CHANGED
	3022	VBSERVLET_STATE_IN_SERVICE
3002		STATE_CHANGED
	3001	STATE_CHANGED_IN_SERVICE
3000		SHUTDOWN_NOTICE
	3001	STATE_CHANGED_IN_SERVICE

Table 49: SIP

Raise ID	Clear ID	Event Name
5001		SS_STATE; The SIP subsystem changed state to something other than the *in service* state.
	5002	SS_STATE; The SIP subsystem changed state to the *in service* state.

Table 50: VoiceXML

Raise ID	Clear ID	Event Name
6012		VXML_SERVER_APP_SHUTDOWN_ALERT
	6011	VXML_SERVER_APP_STARTUP_CLEAR
6013		VXML_SERVER_APPADMIN_ERROR
	1003	[AUDIT] "The system has started up."
	1004	[AUDIT] "The system has completely shutdown."
6014		VXML_SERVER_SYSTEM_ERROR
	1003	[AUDIT] "The system has started up."
	1004	[AUDIT] "The system has completely shutdown."
6024		VXML_LICENSE_ALERT
	6025	VXML_LICENSE_ALERT_CLEAR

Note VXML_LICENSE_ALERT is raised when the VXML Port license utilization exceeds 90% of the total deployed license ports and the VXML_LICENSE_ALERT_CLEAR is raised when the VXML port license utilization drops below 70% of the total deployed license ports.

PART III

Configuration Detail of Non-Unified CVP Components

- Cisco Unified ICME Warm Consult Transfer/Conference to Unified CVP, page 273
- Configure Unified CM, page 279
- Configure the SIP Devices, page 281
- Transfer and Queue Calls with Unified CVP, page 299
- Configure High Availability for Unified CVP, page 357
- Configure the Media Servers, page 377
- Java Runtime Environment Minor Update, page 405

CHAPTER 8

Cisco Unified ICME Warm Consult Transfer/Conference to Unified CVP

This chapter provides information about the minimal software component release requirements for the Unified ICME Warm Consult Transfer and Conference to Unified CVP feature for Type 2/7 VRUs. Resource sizing and configuration requirements are also included.

Note For information about using the Warm Consult Transfer feature with SIP and Type 10 VRUs, refer to "Warm Transfer Feature with SIP Calls." Also, refer to the configuration instructions for the Call Director and Comprehensive call flow models using SIP in "High-Level Configuration Instructions."

- About Unified ICME Warm Consult Transfer/Conference to Unified CVP Feature, page 273
- Configure Unified ICME Warm Consult Transfer/Conference to Unified CVP Feature, page 274
- Minimal Component Version Required, page 275
- Warm Transfer Feature with SIP Calls, page 275
- Set Up Unified ICME Warm Consult Transfer, page 276

About Unified ICME Warm Consult Transfer/Conference to Unified CVP Feature

When an agent attempts a warm consultative transfer/conference to another agent, but there is no agent available in the skill group to service the request, the first agent is placed in a queue to wait for the availability of an agent in the desired skill group. In order to place the first agent in queue, a call is initiated from Unified CM to Unified CVP (via a Translation Route to VRU) to provide queue music to the first agent. To Unified CVP, this appears as a new call from an IP phone.

Optionally, customer business call flows may require that IP phone users call Unified CVP directly. For example, you may have a corporate IP phone network that is serviced by a Unified CVP help desk call center. IP phone users with problems would call a Unified CVP number to open trouble tickets.

Configure Unified ICME Warm Consult Transfer/Conference to Unified CVP Feature

To configure the Unified ICME Warm Consult Transfer/Conference to Unified CVP feature, perform the following procedure:

Procedure

Step 1 Install a new Call Server (refer to Installation and Upgrade Guide for Cisco Unified Customer Voice Portal for detailed information). It can be configured identically to all other Unified CVP machines, with the exception that you must add each Translation Route DNIS.

- Define it as a Type 2 VRU in the Network VRU Explorer tool in Unified ICME.
- Network Transfer Preferred must be disabled for this peripheral.
- Add a new DNIS in the **Add DNIS** box on the ICM tab in the Operations Console. Be sure to add each translation route DNIS.

Step 2 If the Unified CVP machine resides in a different location from the Unified CM cluster initiating the calls, WAN bandwidth will be a consideration since the prompts are played G.711 from the Unified CVP machine. In this case, size and configure the network appropriately. Where possible, Unified CVP should be co-located with Unified CM to eliminate these bandwidth requirements.

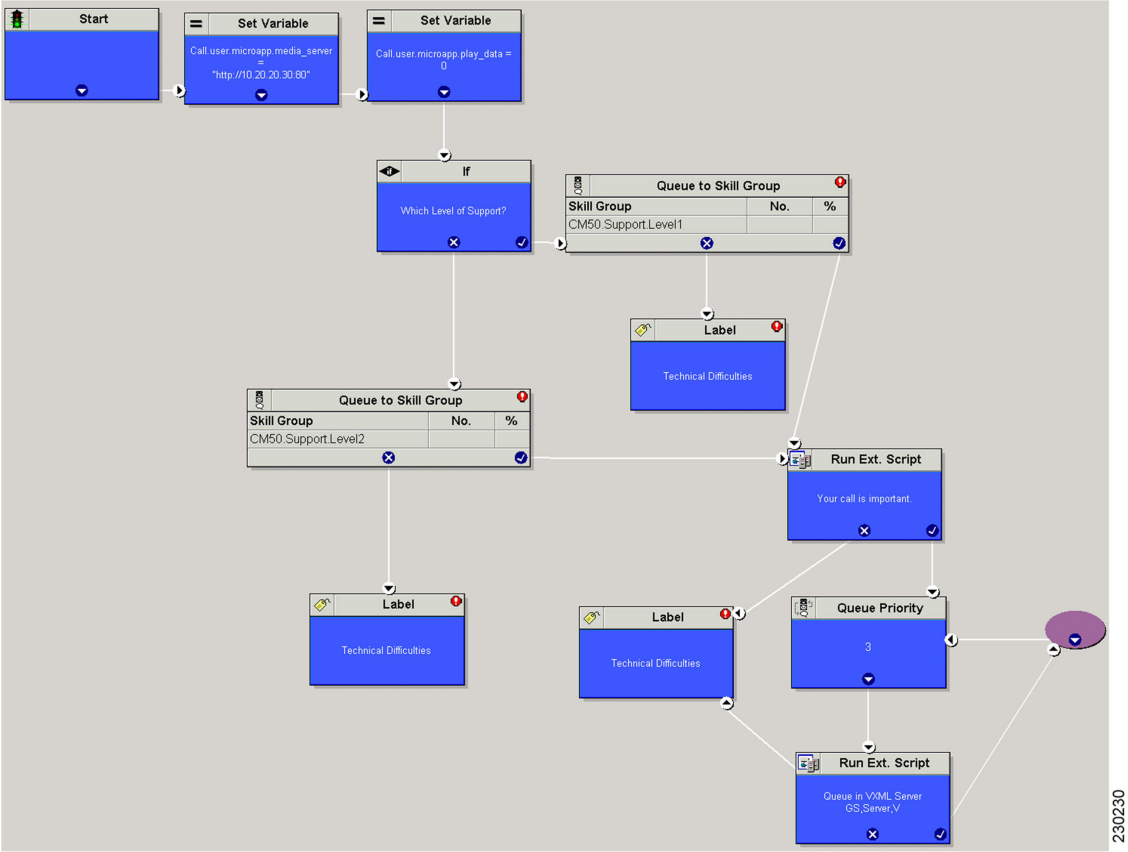
Step 3 Define a SIP trunk in the Unified CM, using the Unified CVP machine IP address as the Destination address in **Device** > **Trunk** > **SIP Information**.

Step 4 (Perform this step for IP-originated calls only). Determine if customer business call flows require that IP phone users call Unified CVP directly (as described in the second paragraph of About Unified ICME Warm Consult Transfer/Conference to Unified CVP Feature). In Unified CM administration under “Route Plan” using route groups/lists/patterns, route Unified CVP DNIS’s to the Unified CVP gateway installed in Step 1 above.
If you want to load-balance between two Unified CVP systems:

- Create a route group and put both of the Unified CVP gateways in the route group, both with order priority 1.
- Create a route list and put the route group in the route list.
- Create a route pattern and assign the route list to the route pattern.
- In Service Parameters for Unified CM, set “Reorder Route List” to **True** and the 'H225 TCP timer' to **5**.

Note The Reorder Route List setting applies only for Unified CM 3.3 and earlier.

Step 5 Create a Unified ICME script similar to the script below. (Refer to Unified ICME documentation for information.) This script should be tied to the Dialed number and call type that the agent invokes to do a warm consultative transfer/conference. This dialed number’s Routing Client should be associated with a Unified CM peripheral from which the agent will be invoking the transfer or conference.



Minimal Component Version Required

Refer to Hardware and System Software Specification for Cisco Unified Customer Voice Portal (Unified CVP) Software for the list of component versions required to use the Unified ICME Warm Consult Transfer and Conference to Unified CVP feature.

Warm Transfer Feature with SIP Calls

If an agent performs a warm transfer to another agent and then that agent is queued, or a SendToVRU label returns to Unified CM using jtapi on the Unified CM PG connection, then you must associate a Route Pattern for that label with a SIP TRUNK to send to Unified CVP or the Proxy Server to ensure the call returns to Unified CVP. Unified CVP then sends the **request instruction** message back to Unified ICME on the Unified CVP routing client and starts the queuing.

Note These SIP calls do not require MTP enablement on the SIP trunks.

When using the Warm Transfer feature for SIP Calls with queuing, and the agent completes a consult transfer to ther caller while the call is still in the queue (VXML Gateway), then the call flow does not require MTP enabled on the SIP trunk that is associated with the VRU label route pattern.

Set Up Unified ICME Warm Consult Transfer

Note Unified CVP with a Type 10 VRU does not support multiple Network VRUs on the same Unified CVP peripheral device. Multiple customer instances can be used in order to address multiple Network VRUs, but they must then address different physical Unified CVP Call Servers as well. Calls that originate from an ACD or Unified CM, such as Warm Transfer/Conference, Helpdesk, or Outbound calls, are also limited to one Network VRU on any given Unified CVP Call Server. Note that the reverse is supported: multiple Unified CVP Call Servers can share the same Network VRU.

In this scenario, an agent transfers a call to another agent by dialing that agent's ID. If the agent is not available, the originating agent will be placed in a queue to wait for the second agent to pick up the call.

In order for the first agent to be queued while waiting for another agent, set up the following configuration:

Procedure

Step 1 In the ICM Configuration Manager's PG Explorer tool Routing Client tabs, uncheck the NetworkTransferPreferred check box for Unified CM and Unified CVP routing clients.

Step 2 On the Advanced tab for the Unified CM routing client, select **None** for the Network VRU and your Type 10 VRU for the Unified CVP routing client.

Step 3 For your Type 10 VRU, in the ICM Configuration Manager's Network VRU Explorer tool, define a label for the Unified CM routing client as well as the Unified CVP routing client, and associate them with a customer instance.

Step 4 In the ICM Configuration Manager's Dialed Number List Tool, associate the dialed numbers for the incoming call as well as the transfer dialed number with the same customer instance.
When the second call is placed for the warm transfer and no agent is available, the label defined on the Unified CM RC plus the correlation ID will be sent back via EAPIM/JGW to Unified CM. For example, if the label is 7777777777, with a correlation ID it could be 777777777712345 since the call originated from the Unified CM RC and since the NetworkTransferPreferred check box is not checked.

Step 5 In Unified CM, select **Call Routing** > **Route/Hunt** > **Route Pattern** > **Add New**. Add a new route pattern to route the call to Unified CVP using the SIP trunk if you are adding from the Device Management menu (for example, 777! where ! allows label plus arbitrary length correlation ID).

When Unified CVP sees this call, it perceives it as a pre-routed call with a correlation ID and sends it back to Unified ICME to continue the script.

Unified ICME sends a temporary connection back to Unified CVP, which queues the agent call while the caller hears music on hold (MoH) from Unified CM.

Note When customized CTI clients are used, consult transfer mechanism is utilized to check if the second agent is really answering the call before the call is being finally transferred automatically by the customized CTI client. In this scenario, it is not required for the agents transferring the call to complete the transfer manually as customized CTI client automatically transfers the calls. However, this is applicable only when the second agent (called agent) answers the call and not before.

Customized clients should wait for five seconds before completing the automatic consult transfer to avoid race conditions.

CHAPTER 9

Configure Unified CM

This chapter describes how to configure Unified CM. Once Unified CM has been configured, you can add a pre-configured Unified CM Server to the Operations Console network control panel. Once added, you can add the Unified CM Server to a device pool and access a Unified CM administration web page.

- Configure Unified CM for Use with Unified CVP, page 279
- Configure a Unified CM Server, page 279

Configure Unified CM for Use with Unified CVP

Prerequisite Configuration

Before configuring Unified CM for use with Unified CVP, you must perform the following steps:

1. Configure a Unified CM Server. (Refer to the instructions in the Unified CM documentation.)
2. To hear a ring tone on a blind transfer call, select **Service** > **Service Parameters** in Cisco Unified Communications Manager. Then select your server and the Cisco Unified Communications Manager Service.
3. Set the Send H225 User Info Message setting to **H225 Info for Call Progress Tone**.
4. Save your settings.

Configure a Unified CM Server

From the **Communications Manager** option in the Device Management menu, you can add a Unified CM Server to the Operations Console. Once added, you can add the Unified CM Server to a device pool and access a Unified CM administration web page, from which you can configure the Unified CM Server.

Unified CM manages and switches VoIP calls among IP phones. When combined with Unified ICME, Unified CM becomes the IPCC product. Unified CVP interacts with Unified CM primarily as a means for sending PSTN-originated calls to IPCC agents. However, several applications require that calls be originated by IPCC agents instead. Specifically, IPCC Outbound Option and calls that are being handled using the warm-consultative-transfer feature from one agent to another are originated in this way. Help desk calls, in

which an agent or other IP phone user calls Unified CVP (or calls a skill group and gets queued on Unified CVP), also fall into this category. A single Unified CM can originate and receive calls from a SIP device.

Refer to the Operations Console online help topics under *Managing Devices > Configuring a Unified Cm Server* for the following topics:

- Adding a Unified CM Server
- Editing a Unified CM Server
- Unified CM Configuration Settings
- Deleting a Unified CM Server
- Finding a Unified CM Server

Before You Begin

Configure the following SIP-specific actions on Unified CM. (Refer to the Unified CM documentation for more information.)

Complete the following tasks:

- Configure the SIP Proxy Server.
- Configure the SIP Trunk to the Proxy Server or the Unified CVP Call Server, if you are not using a SIP Proxy.
- Add call routing (route patterns) to send the call from Unified CM; for example, ring tone, playback dial patterns, and ICM route table calls.

CHAPTER 10

Configure the SIP Devices

This chapter describes how to configure SIP calls on the gateway, Call Server, and Proxy Server.

- Configure a SIP Proxy Server, page 281
- Load-Balancing SIP Calls, page 282
- Set Up the Ingress Gateway to Use Redundant Proxy Servers, page 283
- Set Up the Unified CVP Call Server with Redundant Proxy Servers, page 283
- Cisco Unified SIP Proxy (CUSP) Configuration, page 283
- REFER Transfers, page 286
- Configure Custom Streaming Ringtones, page 287
- sendtooriginator Setting in the SIP Service, page 289
- 100rel and SIP Outbound Dialer, page 290
- Expiration Timeout Setting Using Dialed Number Patterns in SIP, page 290
- Critical Error Message Playback on Abnormal Disconnects with SIP, page 290
- Delivering SIP Incoming UUI to Unified ICME Routing Script, page 291
- SIP Hookflash, page 292
- SIP Hookflash Configuration, page 293
- Pass User-to-User Information, page 295
- Pass Information with SIP Headers, page 296
- Configure SIP Headers, page 297
- Example Dial-Peer Configuration for Connecting the Cisco IOS Gateway to Video Media Server, page 298

Configure a SIP Proxy Server

From the **SIP Proxy Server** option on the Operations Console Device Management menu, you can add a pre-configured SIP Proxy Server. You must configure the SIP Proxy server before adding it to the Operations

Console. Once added, you can add the SIP Proxy Server to a device pool. You can also configure a link to the administration web page for the SIP Proxy Server so that you can access that page from the Operations Console.

A SIP Proxy Server is a device that routes individual SIP transport messages among SIP endpoints. It plays a key role in a Unified CVP deployment for call switching. It is designed to support multiple SIP endpoints of various types, and implements load balancing and failover among those endpoints. SIP Proxy Servers can be deployed alone or as a pair. Also, smaller Unified CVP deployments can run without a SIP Proxy Server. In such cases, the Unified CVP SIP service assumes some of those functions because it provides the ability to configure a static table to look up destinations.

Before You Begin

Configure the following characteristics of the SIP Proxy Server for use with Unified CVP:

- A static route to the Unified CVP Call Server, Unified CM SIP trunks, VoiceXML gateway, and Ingress gateway for the transfer to the ringtone playback dialed number and error playback dialed number.

  Note You must configure a Unified CM SIP trunk on Unified CM to point to the SIP Proxy Server. In a cluster with multi-subscribers and device pools, adding only one SIP trunk per proxy in the default device pool will be sufficient.

- Incoming Access Control Lists (ACLs) for Unified CVP calls.

Unified CVP works with RFC-3261-compliant SIP Proxy Servers and has been qualified with the following:

- Unified CVP works with the CUSP server.

Refer to the Operations Console online help topics under **Managing Devices** > **Configuring a SIP Proxy Server** for details about performing the following tasks:

- Adding a SIP Proxy Server

  This topic also describes the SIP Proxy Server configuration settings.

- Deleting a SIP Proxy Server
- Editing a SIP Proxy Server
- Finding a SIP Proxy Server

Load-Balancing SIP Calls

SIP calls can be load balanced across destinations in several different ways as outlined below:

- Using the CUSP server, define several static routes with the same route pattern, priorities, and weights.
- Using DNS, configure SRV records with priorities and weights. Both the DNS client and the server settings must be configured and operating successfully for DNS "A" and "SRV" type queries to work. Configure SRV queries to be used wherever outbound SIP calls are made, such as on the IOS Ingress gateway, on the Call Server itself, and on Unified CM.

Note Refer to DNS Zone File Configuration for Call Director Call Flow Model for information about load balancing and failover without a Proxy Server. Only the DNS SRV method is supported for load balancing and failover without a Proxy Server.

Set Up the Ingress Gateway to Use Redundant Proxy Servers

Procedure

Configure the gateway with the following to send calls to redundant proxy servers as resolved using DNS SRV lookup:

```
ip domain name <your domain name>
ip name-server <your DNS server>
sip-ua
sip-server dns:<your SRV cluster domain name>
dial-peer voice 1000 voip
session target sip-server
```

Set Up the Unified CVP Call Server with Redundant Proxy Servers

You may use redundant proxy servers for CVP outbound calls by using a DNS-based SRV cluster name or a non-DNS SRV cluster name (also known as Server Group Name).
Refer to the **Operations Console online help** topic **Using SIP Server Groups** > **Configuring Server Groups** for information on how to configure local based SRV records.

Cisco Unified SIP Proxy (CUSP) Configuration

The following configuration shows a CUSP proxy in Unified CVP. The highlighted lines are specific to a Unified CVP solution. For additional configuration details, refer to the Configuring Cisco Unified SIP Proxy Server guide.

Configuration Example:

```
server-group sip global-load-balance call-id
                           server-group sip retry-after 0
                           server-group sip element-retries udp 1
                           server-group sip element-retries tls 1
                           server-group sip element-retries tcp 1
                           sip dns-srv
                           no enable
                           no naptr
                           end dns
                        !
                        no sip header-compaction
                        no sip logging
                        !
                        sip max-forwards 70
                        sip network netA noicmp
                        non-invite-provisional 200
                        allow-connections
```

```
retransmit-count invite-server-transaction 9
retransmit-count non-invite-client-transaction 9
retransmit-count invite-client-transaction 2
retransmit-timer T4 5000
retransmit-timer T2 4000
retransmit-timer T1 500
retransmit-timer TU2 32000
retransmit-timer TU1 5000
retransmit-timer clientTn 64000
retransmit-timer serverTn 64000
end network
!
no sip peg-counting
!
sip privacy service
sip queue message
drop-policy head
low-threshold 80
size 2000
thread-count 20
end queue
!
sip queue radius
drop-policy head
low-threshold 80
size 2000
thread-count 20
end queue
!
sip queue request
drop-policy head
low-threshold 80
size 2000
thread-count 20
end queue
!
sip queue response
drop-policy head
low-threshold 80
size 2000
thread-count 20
end queue
!
sip queue st-callback
drop-policy head
low-threshold 80
size 2000
thread-count 10
end queue
!
sip queue timer
drop-policy none
low-threshold 80
size 2500
thread-count 8
end queue
!
sip queue xcl
drop-policy head
low-threshold 80
size 2000
thread-count 2
end queue
!
route recursion
!
sip tcp connection-timeout 240
sip tcp max-connections 256
!
no sip tls
!
trigger condition in-netA
    sequence 1
```

```
    in-network netA
    end sequence
    end trigger condition
    !
    trigger condition mid-dialog
    sequence 1
    mid-dialog
    end sequence
    end trigger condition
    !
    trigger condition out-netA
    sequence 1
    out-network netA
    end sequence
    end trigger condition
!
accounting
no enable
no client-side
no server-side
end accounting
!
server-group sip group cucm-cluster.cisco.com netA
 element ip-address 10.86.129.219 5060 udp q-value 1.0 weight 10
 element ip-address 10.86.129.62 5060 udp q-value 1.0 weight 10
 element ip-address 10.86.129.63 5060 udp q-value 1.0 weight 10
 failover-resp-codes 503
 lbtype global
 ping
 end server-group
 !
 server-group sip group cvp-call-servers.cisco.com netA
 element ip-address 10.86.129.220 5060 udp q-value 1.0 weight 10
 element ip-address 10.86.129.224 5060 udp q-value 0.9 weight 10
 failover-resp-codes 503
 lbtype global
 ping
 end server-group
 !
 server-group sip group vxml-gws.cisco.com netA
 element ip-address 10.86.129.229 5060 udp q-value 1.0 weight 10
 element ip-address 10.86.129.228 5060 udp q-value 1.0 weight 10
 failover-resp-codes 503
 lbtype global
 ping
 end server-group
 !
 route table cvp-route-table
 key 9 target-destination vxml-gws.cisco.com netA
 key 8 target-destination cvp-call-servers.cisco.com netA
 key 7 target-destination vxml-gws.cisco.com netA
 key 700699 target-destination cvp-call-servers.cisco.com netA
 key 2 target-destination cucm-cluster.cisco.com netA
 key 1 target-destination cucm-cluster.cisco.com netA
 key 7000 target-destination 172.19.151.41 netA
 key 777333 target-destination cvp-call-servers.cisco.com netA
 key 1004 target-destination 10.86.139.84 netA
 key 7105 target-destination dialer-gws netA
 end route table
 !
 policy lookup cvp-policy
 sequence 1 cvp-route-table request-uri uri-component user
 rule prefix
 end sequence
 end policy
 !
 trigger routing sequence 1 by-pass condition mid-dialog
 trigger routing sequence 10 policy cvp-policy condition in-netA
 !
 server-group sip ping-options netA 10.86.129.200 5038
 method OPTIONS
 ping-type adaptive 5000 10000
 timeout 500
```

```
 end ping
 !
 server-group sip global-ping
 sip listen netA udp 10.86.129.200 5060
!
end
```

REFER Transfers

Unified CVP SIP Service can perform a SIP REFER transfer instead of using SIP re-invites, which allows Unified CVP to remove itself from the call, thus freeing up licensed Unified CVP ports. (Unified CVP cannot execute further call control operations after this kind of label has been executed; for example, it cannot perform subsequent transfers back to Unified CVP for self service or queuing to another agent.)

However, if the transfer fails, configure survivability to transfer the call elsewhere. This process is not the same as an ICM router requery; for example, it will appear as a new call to Unified ICME, but it is a way to take an alternate action, if the transfer fails.

Note This feature can be used in Comprehensive (SIP only), Call Director, and Standalone call flow models.

Note Router requery can be performed with a REFER transfer *only* if the NOTIFY messages are sent back to Unified CVP with the result of the REFER operation. Since Unified CVP does not hang up the call after sending REFER, it is possible to requery Unified ICM, get another label, and send another REFER.

Note The use of the survivability tcl service on the ingress gateway cannot currently support sending the NOTIFY messages with a failed transfer result, so router requery cannot be used with REFER when it is handled by the survivability service. Survivability service can handle REFER, except that it will always report a successful transfer to Unified CVP, even when the transfer failed. This is a known limitation of the TCL IVR API for REFER handling in IOS, including ingress and CUBE gateways.

Using this feature, the call can be queued at the VoiceXML gateway and then sent to an agent with a Unified ICME label that begins with the letters "rf." Otherwise, standard Unified ICME agent labels enable Unified CVP to remain in the signaling path for the duration of the call, and the licensed Unified CVP resource will not be freed until the end of the call. REFER transfers can be made to Unified CM or other SIP endpoints in the SIP cloud, such as an ACD. The ECC variable "user.sip.refertransfer" can also be set in Unified ICME scripts. (When using this ECC variable in a Unified ICME script, it must be set to the value of the single character "y" and Unified CVP will use REFERs when transferring to the agents.)

When using REFER transfers, including the REFER used to play back critical_error.wav for abnormal disconnects, the Ingress gateway must include an outbound voip dial peer. This outbound dial peer is necessary because when the REFER message enters the gateway from the Unified CVP Call Server, it needs to match an outbound dial peer in order for the call to succeed; otherwise, a 503 rejection occurs if no dial peers match the REFER-TO header URI. Dial peer destination targets must match the labels in the REFER-TO SIP URI; meaning that <errorDN>@<sip-server> and other labels that might be used in the Unified ICME routing label. For example:

```
dial-peer voice 1050 voip destination-pattern
1... voice-class codec 1 session protocol sipv2 session target <your
sip-server destination> dtmf-relay rtp-nte
```

```
no vad
```

When configuring Route Patterns on Unified CM for REFERs to destinations outside of the cluster, such as to the CUSP Server or the gateways directly, you must add **SIP Route Pattern** for the SIP Trunk associated with that endpoint. For example, if you use REFER to Error DN to the IP Originated caller on Unified CM, and the host of the REFER To header SIP URL is the CUSP Server, you must create a SIP Route Pattern with that IP address or domain name and associate it with your SIP Trunk for the CUSP Server.

Note When a TDM gateway handles REFER, and not Cisco Unified Border Element (CUBE), a REFER triggered INVITE is sent out. The REFER triggered INVITE requires a dial peer with a session target and typical codec information. The REFER-TO header URI host that was formulated by the CVP routing algorithm configuration is ignored.

Note When CUBE receives a CVP initiated REFER, it does not send it transparently through to the originator. A dial peer is required to match the DN (user portion of the REFER-TO header URI) and the host portion of the URI is rewritten to match the session target of the dial peer. The REFER is passed to the originator using cli "supplementary-service sip refer"; otherwise, CUBE will handle the REFER and send the triggered invite to the refer DN on its own as a back to back user agent.

Configure Custom Streaming Ringtones

You can configure custom ringtone patterns that enable you to play an audio stream to a caller in place of the usual ringtone. Customized streaming ringtones are based on the dialed number destination and, when configured, play an in-progress broadcast stream to the caller while the call is transferred an agent.

Procedure

To customize ringtone patterns for streaming audio, complete the following procedure:

Procedure

Step 1 Configure Helix for streaming audio.
The default installation and configuration of the Helix server is all that is required for use with Unified CVP. See the *Helix Server Administration Guide* for information about installing and configuring the Helix Server.

Step 2 In the Operations Console, perform the following steps to configure custom streaming ringtones:

a) Select **System** > **Dialed Number Pattern** .
b) Click **Add New**.
c) Complete the following fields to assoicate a dialed number pattern with a custom ringtone.

Table 51: Dialed Number Pattern Configuration Settings

Property	Description	Default	Value
General Configuration			

Property	Description	Default	Value
Dialed Number Pattern	The actual Dialed Number Pattern.	None	Must be unique Maximum length of 24 characters Can contain alphanumeric characters, wildcard characters such as exclamation point (!) or asterisk (*), single digit matches such as the letter X or period (.) Can end with an optional greater than (>) wildcard character
Description	Information about the Dialed Number Pattern.	None	Maximum length of 1024 characters
Enable Custom Ringtone	Enables customized ring tone. • **Ringtone media filename -** Enter the name of the file that is to be played for the respective dialed number pattern. Provide the URL for the stream name in the following format: `rtsp://<streaming server IP address>:<port>/<directory>/<filename>.rm`	Disabled none	Maximum length of 256 characters Cannot contain whitespace characters

d) Click **Save** to save the Dialed Number Pattern.
You are returned to the **Dialed Number Pattern** page. To deploy the Dialed Number Pattern configuration, click **Deploy** to deploy the configuration to all Unified CVP Call Server devices.

e) Access the IOS device in global configuration mode and add the following commands on your VXML Gateway:

```
rtsp client timeout 10
```

```
rtsp message timeout 10
```

The range is 1 to 20; the recommended value is 10 seconds.

Step 3 Add a Send to VRU node in your ICM script before any Queue node.
The explicit Send to VRU node is used to establish the VRU leg before the transfer to the agent; this is required to play streaming audio ringtones to a caller.

Related Topics

Configure Play Media Micro-Application to Use Streaming Audio, on page 117
Using the Helix Server, on page 407

sendtooriginator Setting in the SIP Service

Note The information in this section is applicable to *all* Unified CVP call flow models.

Note The setting on the IOS gateway for "signaling forward unconditional" is only required if ISDN call variables need to be available in the Unified ICME scripting environment. If these call variables are not required, then omitting this setting is acceptable. The setting makes the SIP INVITE message larger in terms of bytes due to the extra payload in the message body for GTD variables. If the packet size is significantly greater than 1300 bytes, then TCP transport might be recommended over UDP transport due to the possibility of a network "fragmentation" of messages. Refer to the Operations Console online help for more information.

Note If the pattern matches the label returned from ICM, the call is routed to the originating host derived from the incoming calls remote party ID header or contact header.

Note The call will be sent to the origination gateway if the following statements are true: the remote party ID header is present on the incoming SIP invite; the user agent header of the INVITE indicates an IOS gateway; and the pattern matcher on the label is configured for send-to-origin.

For the Unified CVP Branch call flow model, incoming calls into the Unified CVP Call Server from the gateway might be automatically routed back to the originating gateway at the branch using the"sendtooriginator" setting in the SIP Service of the Call Server. This setting overrides sending the call to the outbound proxy or to any locally configured static routes on Unified CVP. If the label returned from Unified ICME for the Unified CVP transfer matches one of the configured patterns under the Unified CVP "sendtoorginator" settings, then the call will be routed to the SIP URL "sip:<label>@<host portion from header of incoming invite>."

100rel and SIP Outbound Dialer

100rel is not supported in Unified CVP, but it is required for the SIP Outbound Dialer product. If the gateway is being used for calls on both Unified CVP and the SIP Outbound Dialer, then the rel1xx setting under the global `voice service voip` must not be set to `rel1xx required` , which is not supported in Unified CVP, or `rel1xx disabled`, which is not supported with SIP Outbound Dialer. The rel1xx settings must be set on the level of the dial peers, not at the global level. All the examples in this document show the rel1xx setting at the dial-peer level.

Expiration Timeout Setting Using Dialed Number Patterns in SIP

The Unified CVP SIP Service can be configured with Dialed Number patterns to set the expiration timeout of an outbound call for a Unified ICME label. Longer or shorter times can be configured before the call is cancelled with no pickup (ring no answer timeout). Use the **DN Pattern Outbound Invite Timeout** option in the Operation Console's SIP Service configuration tab to add the expiration timeout for a particular dialed number pattern. For example, you can configure Unified ICME labels to Unified CM agents with shorter timeouts to perform a router requery, if desired.

Note It is possible IOS will timeout before receiving INVITE number from Unified CVP (for example, to Unified CM or a gateway). As a practical setup, set the gateway sip-ua expires timer to higher than the highest setting on the Unified CVP configuration. The *minimum* ring timeout is 60 seconds on the IOS side. Unified CVP can set a minimum ring timeout on its outbound calls as low as necessary.

Critical Error Message Playback on Abnormal Disconnects with SIP

Unified CVP is designed to play a critical error message to the caller when a call rejection occurs. The SIP Service in the Call Server uses the SIP REFER method to refer the calls to the IOS gateway "cvp-error" TCL scripted service for message playback. The purpose of this referral is to define the ERROR DN (default 92929292) on the SIP Service of the Call Server in the Operations Console.

Note Critical Error messages will not be played if REFER blind transfers fail because the Unified CVP SIP Service is no longer holding onto the call; it cannot take the call back and refer to the error DN. (This also includes 302 redirect responses.)

Note Anything other than a code 200 (SIP 200 OK) or a 16 (Q.850;cause=16) is considered an abnormal disconnect. Error message playback only occurs if Unified CVP disconnects the caller abnormally. If the disconnect arrives on Unified CVP on the caller leg from the ingress gateway, it appears that the caller is disconnecting. Unified CVP does not count this as an abnormal disconnect even though the Unified CVP survivability script on the ingress gateway might detect this event.

How Critical Error Message Playback Works

Critical Error Message playback works as follows:

- *For typical TDM calls using the survivability service on the IOS Gateway* , the error message is played using the survivability script when the script detects the non-normal disconnect code after Unified CVP has disconnected the call (refer to SIP BYE message). The display name of "--CVP" is appended in a Remote Party ID header of the SIP call. The SIP Service can identify whether or not the incoming call uses survivability. For example, IP-originated Unified CM callers do not have this setting. All TDM callers must have survivability service.
- *For IP-originated calls that the SIP Service detects is not using the survivability service* , when the call is disconnected and the reason code is abnormal, Unified CVP sends a REFER message with the SIP URL of the error DN. The error DN must be configured in the dial plan to point to an IOS gateway that has the cvp-error TCL service configured. When the caller gets the REFER TO destination, it attempts to blind transfer the call to the Unified CVP Error service on the gateway. Once Unified CVP sends a REFER for error message playback, it cannot retrieve the call, so it is important to confirm this mechanism is configured correctly during deployment. For any call that Unified CVP sees is not using the survivability service, it will attempt the REFER method to the error DN for critical error message playback for abnormal disconnects.
- For third party SIP gateways, where the survivability service is not running, and to avoid the use of the SIP REFER to error service, the survivability service may be turned off in the SIP section of the Call Server configuration. Turning off this property allows Unified CVP to pass through the rejection response as-is to the caller, or else the BYE with reason header will be passed, depending on the call state. It may be that the SIP gateway does not implement SIP REFER and you desire to turn this setting off.

Delivering SIP Incoming UUI to Unified ICME Routing Script

For SIP Calls, ISDN call parameters such as "user-to-user information" (also known as "UUI" or "UUS") can be passed to the Unified ICME script. (ISDN call data is only forwarded in SIP calls from the TDM calls using a gateway.) On the gateway, the signaling forward unconditional setting is required on the voice service voip section of the IOS configuration. This allows forwarding of the Generic Type Descriptor (GTD) data of the call in the SIP INVITE to Unified CVP in a multipart mime content type media format.

The Unified CVP SIP Service extracts the GTD section of the call and parses out the parameters that are configured to be passed to Unified ICME in the payload field of the ICM Service's "NEW CALL" message "usertouserinfo." ISDN call parameters, which are forwarded in the GTD payload, as set in the Unified ICME script, cannot be sent in outbound SIP calls. In addition to sending ISDN data inbound to Unified ICME, the GTD section of the inbound call leg will be passed along untouched in the outbound call leg by the Unified CVP SIP Service.

Note The SIP Service only reads the data and sends it to Unified ICME (it does not modify the data).

Note **Session Timer RFC 4028 support in Unified CVP.** Unified CVP, acting as a Back-to-Back User Agent (b2bua) will transparently pass thru the headers for Supported, Session Expires, and Min-SE on the call legs. However, since the IOS gateway does not set a Session Expires header by itself, it will not request session refreshing. By default, Unified CVP adds a *Session-Expires: 1800;refresher=uac* value to the 200OK response on the initial invite sent into Unified CVP by the caller, if the Session Expires header is not already present in the 200OK of the outbound leg. This value will cause the gateway, as UAC, to send the refesher invites in mid call every 1800 seconds. This configuration allows the gateway to clean up a zombie call, due to a failed refresher reinvite, if the Unified Call Server is taken down in mid call. The setting on Unified CVP to disable adding a Session Expires header is configurable.

UUI Delivery

Assume the following ISDN data is sent in the call to Unified CVP:

```
PRN,isdn*,,ATT5*,
USI,rate,c,s,c,1
USI,lay1,ulaw
TMR,00
CPN,00,,u,5900
CPC,09
FCI,,,,,,,y,
UUS,3,3132333435
GCI,87c0c79d91dd11daa9c4000bfda207f2
```

Note By default, UUS field data is converted from 2-byte hex representation to 1-byte ASCII; for example, the UUS value would be"303132333530" in 2-byte hex representation; in 1-byte ASCII, the value would be "012350."

The **Generic Type Descriptor (GTD) Parameter Forwarding** configuration setting in the Operations Console is used for passing GTD (UUI) data to Unified ICME in a new call. (The default is UUS.) Additionally, other parameters in the GTD can also be extracted and sent to Unified ICME. UUS, PRN, and GCI use comma-separated values. Any parameter contained in the NSS IAM message can be extracted. (Refer to the ITU-T Narrowband Signaling Syntax spec (Q.1980.1).)

Configure the SIP Service by selecting **Device Management** > **CVP Call Server** > **SIP tab** in the Operations Console. Make an entry for the **Generic Type Descriptor (GTD) Parameter Forwarding** with the string to forward using the parameter names delimited by commas. For example, configure the SIP Service with the string **UUS,PRN,GCI** to forward these three parameters in a concatenated string to Unified ICME. The Unified ICME script needs to retrieve the call variable on the route request message called "UsertoUserInfo," and parse out the needed information.

SIP Hookflash

Hookflash is a Signaling mechanism typically associated with a TDM PBX or ACD. Hookflash sends a quick off-hook/on-hook/off-hook signal to the PBX or ACD. This signaling causes the voice gateway to send a string of routing digits to the PBX or ACD. Upon collection of the routing digits, the PBX or ACD transfers the caller to the new termination, such as another agent or an ACD queue or service on that same PBX or ACD.

Unified CVP supports using hookflash with SIP. Using this feature you can transfer SIP calls using a hookflash followed by the DTMF destination. This is useful for deployments in which a PBX "front-ends" the Unified CVP ingress gateway, and in which the PBX provides non-VOIP connectivity to agents.

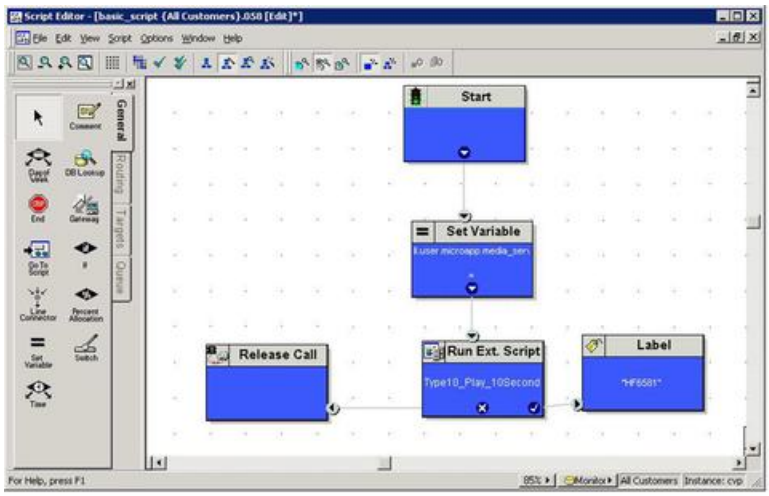
In a typical scenario, a caller calls into a system and is transferred to an agent who is associated with a non-Cisco ACD. Unified ICM returns a label to Unified CVP to perform a hookflash transfer to the PSTN so that the caller can be routed to the correct agent (the label returned must have "HF" prepended to the hookflash routing digits). The caller is then transferred to the agent and Unified CVP is no longer in control of the call.

SIP Hookflash Configuration

To configure your system to use SIP hookflash, the gateway T1 controller must be configured for hook flash relay. seeConfigure Hookflash Relay, on page 334 for more information.

Procedure

Step 1 On Unified ICM, in your script, use a label node such as "HF,5551212", or, as in the following example, HF6581.



Unified CVP automatically disconnects the call two seconds after sending the digits. If the switch needs more time to complete the hook flash sequence, increase the delay by appending more commas after the transfer number to extend the two-second timeout.

If the switch requires a pause after the hookflash, insert commas between the HF and the transfer number. Each comma represents 100 milliseconds. For example, to use the hook flash feature to transfer the call to the number “5551212” with a 500- millisecond pause after the hookflash, configure the Unified ICM label as "HF,,,,,5551212".

Step 2 On your ingress and VXML gateways, configure the survivability service on your POTS dial-peers. See Call Survivability for information.
For example:

```
service cvp-survivability flash:survivability.tcl
param icm-hf 1
!

dial-peer voice 800555.... pots
destination-pattern 800555....
service cvp-survivability
! etc...
```

Step 3 Configure your gateway T1 controller for hookflash relay. See Configure Hookflash Relay, on page 334 for information.
For example, for an analog configuration (assume incoming number is 6708):

```
voice-port 1/0/1
signal groundStart
connection plar 6708


dial-peer voice 6708 pots
service cvp-survivability
incoming called-number 6708
direct-inward-dial
port 1/0/1
forward-digits 0
```

For a digital configuration (assume incoming number is 6708):

```
controller T1 2/0
framing esf
linecode b8zs
ds0-group 1 timeslots 1-24 type fxo-loop-start
voice-port 2/0:1
output attenuation 0
connection plar 6708

dial-peer voice 6708 pots
service cvp-survivability
incoming called-number 6708
direct-inward-dial
port 2/0:1
forward-digits 0
```

Pass User-to-User Information

You can pass generic data to and from Unified ICM through Unified CVP using User-to-User Information (UUI). For example, it is sometimes desirable to capture data from an external system (such as caller-entered digits from a third-party IVR) and pass that data to Unified ICM on a new call.

This is accomplished by populating the UUS parameter (often known as the UUI) in the Initial Address Message (IAM) of the Generic Transparency Descriptor (GTD) data that is sent to the gateway from the network in the Q.931 setup message. The gateway sends this data to Unified CVP through SIP messages. Unified CVP can then send the data to Unified ICM on a new call.

UUI can also be set by ICM scripts and extracted by Unified CVP to be resent in SIP messages.

UUI processing scenarios:

- When GTD data is present in the inbound call leg of the SIP INVITE message, Unified CVP saves the GTD data as *inbound GTD* and the UUI portion (if present) is passed to Unified ICM.

  If Unified ICM modifies the data, it sends the modified UUI back to Unified CVP. Unified CVP converts the UUI data it receives from Unified ICM into Hex and modifies the UUS (if it is present) and overwrites the *inbound GTD* value. Only the UUS portion will be modified, using the format:

  ```
  UUS,3,<converted Hex value of data from ICM>
  ```

  The rest of the GTD parameter values are preserved, keeping the values as they arrived from caller GTD.

- When GTD is not present in the inbound call leg, Unified CVP prints an informational message on the trace stating *No GTD Body present in Caller Body* and the call continues as a regular call.

  The modified UUI from Unified ICM is passed using the ICM variable `Call.UsertoUserInfo`.

- REFER and 302 Redirects and UUI.

  If UUI is set in the Unified ICME script, and a refer call flow is being used, then the UUI will be placed in a mime body and hex encoded according to an ATT IP Toll Free NSS format. This applies to 302 redirect responses as well.

```
VER,1.00
PRN,t1113,*,att**,1993
FAC,
UUS,0,(hex encoded UUI string here)
```

When sending GTD, if a regular label is received from Unified ICM, then the modified GTD from the scenarios above is passed on the outbound INVITE from Unified CVP.

If a DTMF label for outpulse transfer is received on a connected call, then the BYE is sent with the following GTD only if UUI is passed by Unified ICM. The BYE is sent immediately after the SIP INFO with DTMF.

```
Content-Type: application/gtd
Content-Disposition: signal;handling=optional
REL,
PRN,isdn*,,NI***,
UUS,3,<converted Hex value of UUI from ICM>
```

Note Unified CVP does not send BYE on the DTMF label if UUI is not received from Unified ICM. If a BYE message is received, then the GTD from the received BYE is used to send it on the other leg.

Note The Ingress gateway must be configured with " `signaling forward unconditional` ", for example:

```
!
voice service voip
signaling forward unconditional
```

Pass Information with SIP Headers

Unified CVP enables the passing of one or more SIP headers to Unified ICM for use within ICM scripts. You configure which headers are passed to Unified CVP through the Operations Console, on the Call Server SIP tab. You can also modify SIP headers in the Unified ICM script and pass these back to Unified CVP.

When sending SIP headers to Unified ICM, in the Operations Console, you can specify a specific header, or a header and specific parameters within that header. For example:

- **Header:** *Supported*
- **Header:** *Remote-Party-ID* **Parameter:** *privacy*
- **Header:** *Call-Info* **Parameter:** *purpose*

These SIP headers are passed to Unified ICM in the **SIPHeaderInfo** field of New Call and Request Instruction messages. To access the SIP Header variable in the ICM script, read the **Call.SIPHeader** call variable.

The amount of space available to send header data to Unified ICM is limited to 255 bytes. The SIP protocol RFC provides a mechanism to represent common header field names in an abbreviated form. The compact header format, as defined in RFC 3261 and other RFCs for newly defined headers, is used for the header titles before passing the header to Unified ICM.

Note Not all headers have a compact format. For example, P-Headers (private headers, such as **P-Asserted-Identity**) may not have a compact form. In this case, Unified CVP passes the full header name to Unified ICM.

You can also write to headers using the ICM script and pass the modified headers back to Unified CVP. This feature provides a scriptable option to modify SIP headers on the outgoing Unified CVP transfer. You can specify SIP header values in outgoing SIP INVITES (only the initial INVITE, not reinvites). You can add, modify or remove header values. Changes to the INVITE are applied just before it is sent out.

Writing to **Call.SIPHeader** using the Unified ICM script causes Unified CVP to use that data in outbound SIP calls to IVR or Agents, or in REFER or 302 redirect messages.

Unified CVP uses the following format to send headers to ICM:

```
"f:Name <sip:from@127.0.0.1:6666>;param1;param2|e:tar|v:SIP/2.0/UDP viaHost"
```

Note In the example above, each header field is separated by the vertical var character (|).

Unified CVP provides the flexibility to add/modify/remove outgoing SIP header in the INVITE message only. This enables you to deploy Unified CVP in many scenarios to facilitate inter-op with third party devices.

For example, if there are inter-op problems in your call flows, you may not be able to enable inter-working with the default SIP signaling. Passing information using SIP headers enables Unified CVP to provide a work-around for inter-op issues.

Warning The SIP header modification feature is a powerful tool which can tweak SIP headers as needed. You should exercise caution when applying SIP Profiles and ensure that the profiles do not "create" interoperability issues, rather than solving them.

Warning The outgoing header modification feature enables you to remove, modify, or add any SIP header. These include SIP headers such as To, From, Via, CSeq, Call-Id, and Max-Forwards. However, there is no validation for your changes in the ICM script editor interface. If your changes are not configured correctly, they may cause unexpected errors and a call failure. You must test your modifications before using them in your production environment. A header modification that does not comply with the RFC specification for that header ABNF format will encounter a Java SIP stack parser exception, and will not be performed.

If there is a problem updating or adding a header with the string given from ICM script, then a WARN type message displays in the Unified CVP Call Server logs if there is a **DsSipParserException**, or Unified CVP sends the INVITE as is with possible unexpected results on the receiver end.

Configure SIP Headers

There is no syntax checking while adding or modifying headers. You must be careful that headers retain their correct syntax.

Note When modifying the headers, do not use the semicolon and the comma characters. These two characters are used internally to store the configuration in the sip.properties file and are reserved.

Refer to the examples that follow the header modification steps.

Procedure

Step 1 To send SIP Headers from Unified CVP to Unified ICM:

a) In the Unified CVP Operations Console, select **Device Management** > **CVP Call Server**.
b) On the Edit CVP Call Server Configuration window, select the Call Server that contains the SIP headers you want to configure.
c) Click the **SIP** tab and scroll down to the section **SIP Header Passing (to ICM)** .
d) Provide the header name, and, optionally, a parameter, and then click **Add**.
In your Unified ICM scripts, the call variable **Call.SIPHeader** now contains the SIP Headers you specified in the Operations Console.

Step 2 If you are going to send SIP headers back to Unified CVP, then you can modify the **Call.SIPHeader** variable using a Set node in the Unified ICM script.

Note The header length (including header name) after modification should not exceed 200.

Note In the Unified ICM script, you delimit the header, operation, and value with a tilde character, and use the bar character to concatenate operations.

Examples of modifying headers

Adds a Call-Info header with the proper call info syntax as per RFC3261:

```
"Call-Info~add~<sip:x@y>;parm1=value1"
```

Short Form notation, plus concatenated operations. Adds a VIA header and modifies the From header:

```
"Via~add~SIP/2.0/UDP viaHost"  Adds a Via header to the message.
"v~add~SIP/2.0/UDP viaHost|f~mod~<sip:123@host>;parm1=value1"
```

The following operation will fail due to incorrect syntax of Call-Info header per RFC 3261. You will see a WARN message in the CVP log. This is enforced in the stack.

```
"Call-Info~add~parm1=value1"
```

From header add and modify will do the same thing, since only one From header is allowed in a message per RFC 3261. This is enforced in the stack.

```
"From~add~<sip:x@y>;parm1=value1"

"Call-ID~add~12345@xyz"  Same as From header, only one allowed.

"Call-ID~mod~12345@abc" Same as From header, only one allowed.
```

Example Dial-Peer Configuration for Connecting the Cisco IOS Gateway to Video Media Server

This example shows the configuration you need to connect a dial-peer to a Video Media Server. Add the following lines of code sample to configure video capabilities on the gateway.

```
 application
  service cvp_videoconnect flash:cvp_videoconnect.tcl

voice service voip
  allow-connections sip to sip

dial-peer voice 6000 voip
  destination-pattern  600T #Match the Video Media Server Destination Number in VideoConnect
 element
  session protocol sipv2
  session target ipv4:<IP address of video media server>

voice-class sip midcall-signaling block
  dtmf-relay rtp-nte
  codec g711ulaw
  no vad
```

CHAPTER 11

Transfer and Queue Calls with Unified CVP

- IVRs From Unified ICME's Perspective, page 299
- Transfer Calls Using Unified CVP in Comprehensive Mode, page 300
- Transfer a Call From Agent to Agent, page 305
- IP Transfer Example, page 306
- CLI Field on Outgoing Transfers, page 307
- Configure Unified CCE Re-Route on No Answer for Unified CVP, page 308
- Call Survivability, page 313
- Enhanced Location Call Admission Control, page 319
- Configure Locations-Based Call Admission Control, page 326
- UUI to Be Used as the Correlation ID, page 329
- External Transfers in Unified ICME, page 330
- Courtesy Callback, page 335
- Courtesy Callback Configuration, page 337
- Multicast Music on Hold (MMoH), page 353
- Post Call Survey for SIP, page 354

IVRs From Unified ICME's Perspective

Essentially, Unified ICME categorizes IVRs into one of two types:

- **Intelligent Peripheral IVRs**: under Unified ICME's control - the carrier network routes calls to the IVR and then removes calls from the IVR for delivery to agents. With Intelligent Peripheral IVRs, once the IVR's prompting or queuing treatment has been completed, the IVR typically has no further role to play for that call.
- **Service Node IVRs**: following prompting/queuing treatment - the IVR initiates call delivery to agents, who are under Unified ICME's control. When functioning as a Service Node IVR, Unified CVP can stay involved with a call even after it has been transferred to another VoIP endpoint.

Unified CVP can act as either IVR type.

Note For complete information about the call flow models available for Unified CVP, refer to High-Level Configuration Instructions, on page 15.

Transfer Calls Using Unified CVP in Comprehensive Mode

This section provides examples of Unified CVP call transfer scripts.

Note The Script Editor Busy and Ring nodes are not supported.

Transfer Calls Using SIP Service

The SIP Service can be configured to operate in two modes to perform Unified CVP transfers. Typically, Unified CVP remains in the signaling path for the duration of the call, and in this normal mode it uses SIP re-INVITE messages to perform the transfers. This will also cause Unified CVP to hold the port license for the call duration.

To operate in standard re-INVITE mode, you do not need to modify the Unified ICME script; however, to send a REFER transfer, send a dynamic label with the letters "rf" prepended to it. Or, when using a Queue node in the Unified ICME script, define an ECC variable called "user.sip.refertransfer" and set it to the value of the lowercase "y." Unified CVP then uses the REFER method to blind transfer to agent labels.

Alternatively, Unified CVP might perform a SIP REFER type transfer where it moves out of the signaling path after sending a referral to the caller to the label provided by Unified ICME. This allows Unified CVP to release the port license after the REFER is sent. Unified CVP will receive notification of the outcome of the call using SIP NOTIFY messages, and this will be included in the reporting database as well.

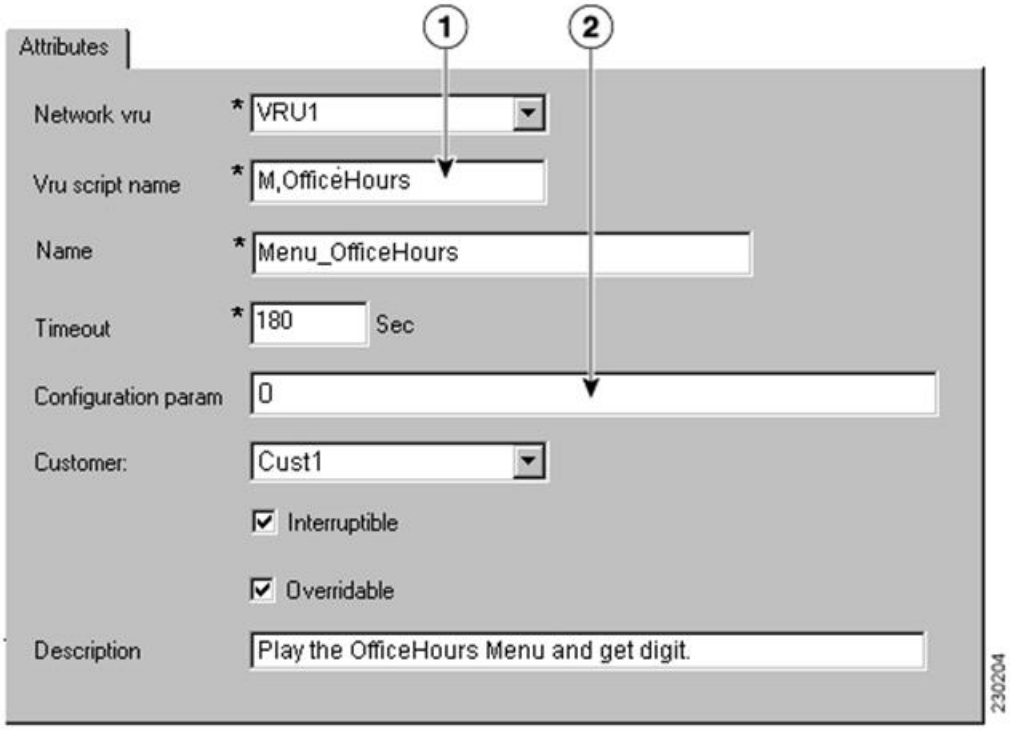
Caution When using REFER, do not apply the survivability script for TDM callers on the Ingress gateway. Also, SIP transfers to VoiceXML gateways for micro-applications will not use the REFER method. It will only be used for non-"SEND TO VRU" type transfers. When using REFERs, note that the survivability script does not currently support REFER messaging events, so when using REFER with TDM calls on the IOS gateway, the survivability service *must* be removed from the pots dial peer for those calls. REFER is used as a "blind refer" operation and can typically be used when sending calls to third-party ACD agents, but it can also be used to send calls to the Cisco Unified Communications Manager extensions as well, if desired.

Example: Transfer to a Label

This example shows sample ICM Configuration Manager and Script Editor screen captures for a Menu application that plays a prompt presenting a menu ("Our office hours are between 8 AM and 6 PM. If you would like to talk to a customer service representative, press 0 at any time.") and then performs one of the following actions:

- If the caller presses 0, the system collects the digit, and then routes and queues the call.

- If the caller does not press 0, the system releases the call.



The Network VRU Script List tool's Attribute tab in the figure above shows:

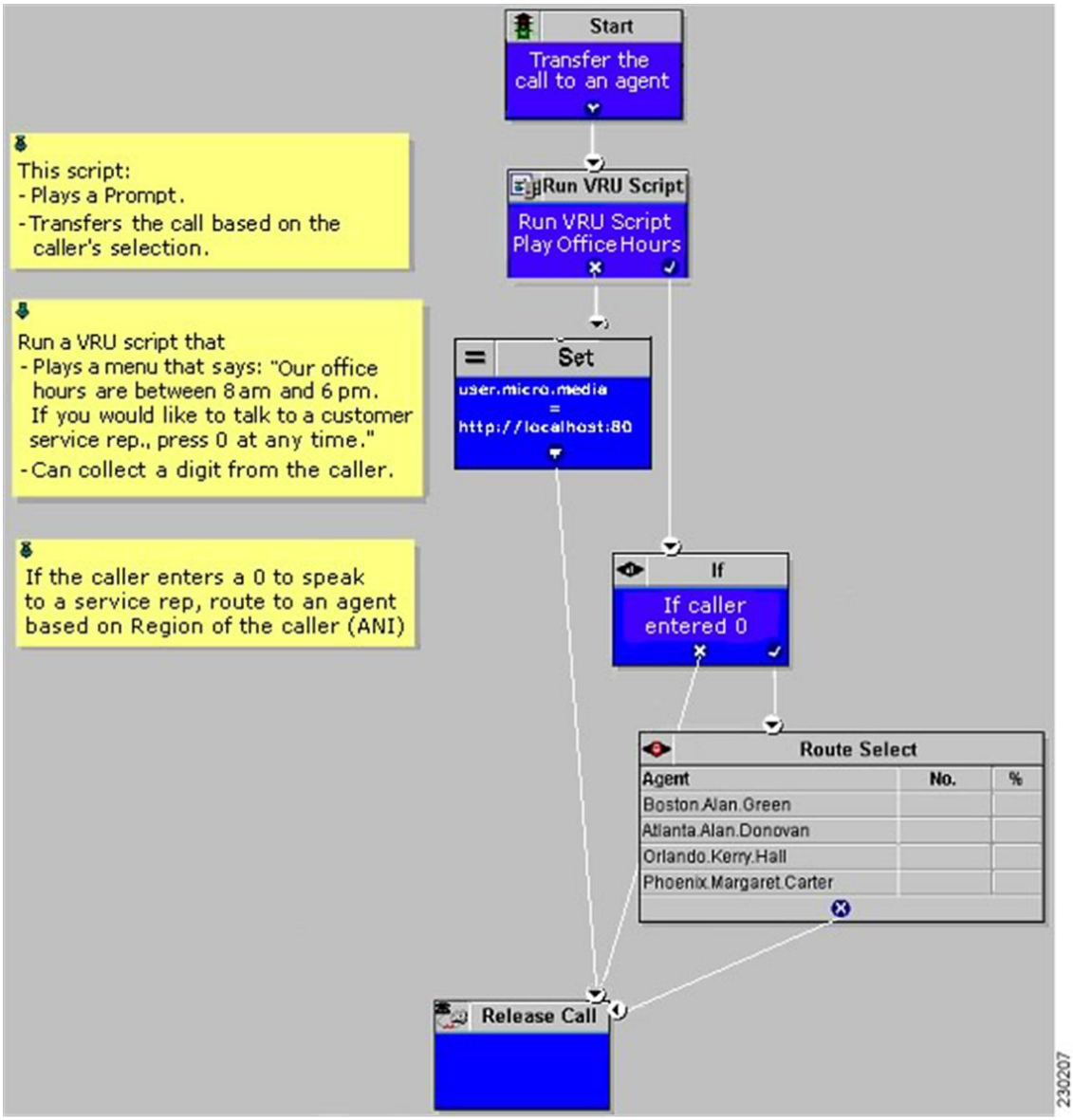
1 The VRU Script Name field contains two Unified CVP parameters:

   M: Menu

   OfficeHours: Media File name

2 The Config Params field contains the following Unified CVP parameter:

   0: The number 0 is the only valid option.

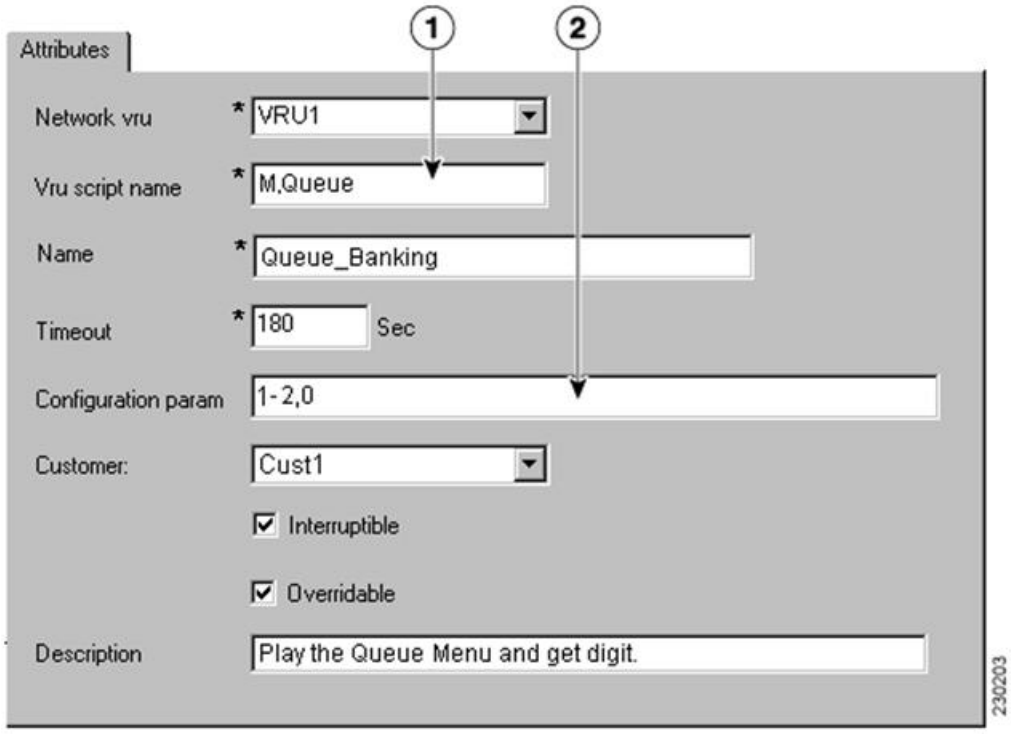


Example: Queue and Transfer to a Skill Group

Unified ICME can queue a call to an agent group and instruct Unified CVP to entertain the caller with IVR scripting using the Run VRU Script and other nodes. When the resource becomes available,

1. Unified ICME tells Unified CVP to cancel the original request.
2. Unified CVP then confirms the cancel request.
3. Unified ICME sends the label for the destination.
4. Unified CVP or the network will transfer the call to a freed-up agent.

This example shows sample ICM Configuration Manager and Script Editor screen captures for a Menu application that plays a prompt presenting a menu ("For Checking, press 1. For Savings, press 2. To speak to a customer service representative, press 0."), retrieves any caller-entered digits, and then routes and queues the call.



The Network VRU Script List tool's Attribute tab in the figure above shows:

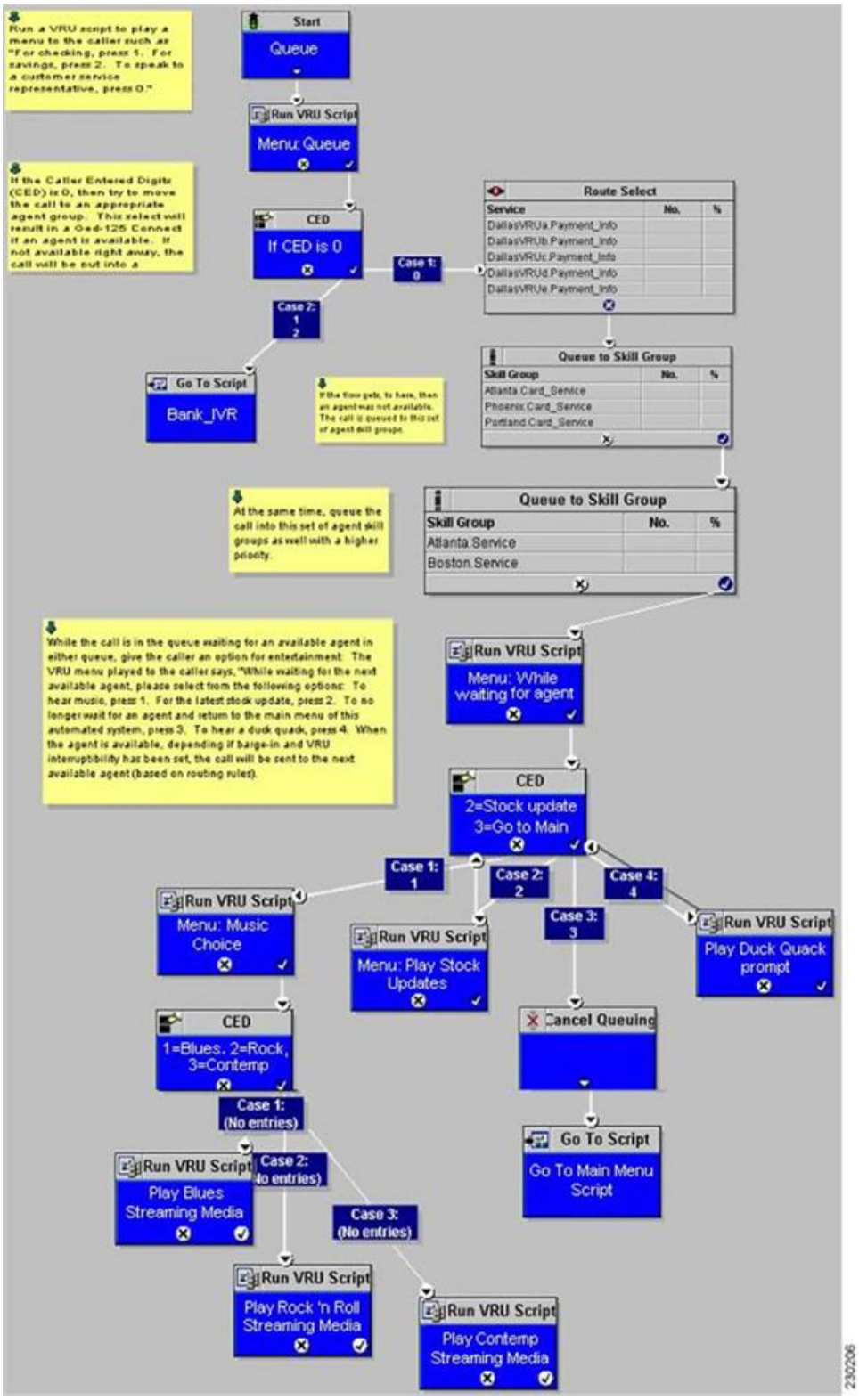
1. The VRU Script Name field containing two Unified CVP parameters:

   M: Menu

   Queue: Media File name

2. The Configuration Param field containing the following Unified CVP parameters:

   1-2,0: The numbers. 1, 2, and 0 are valid options

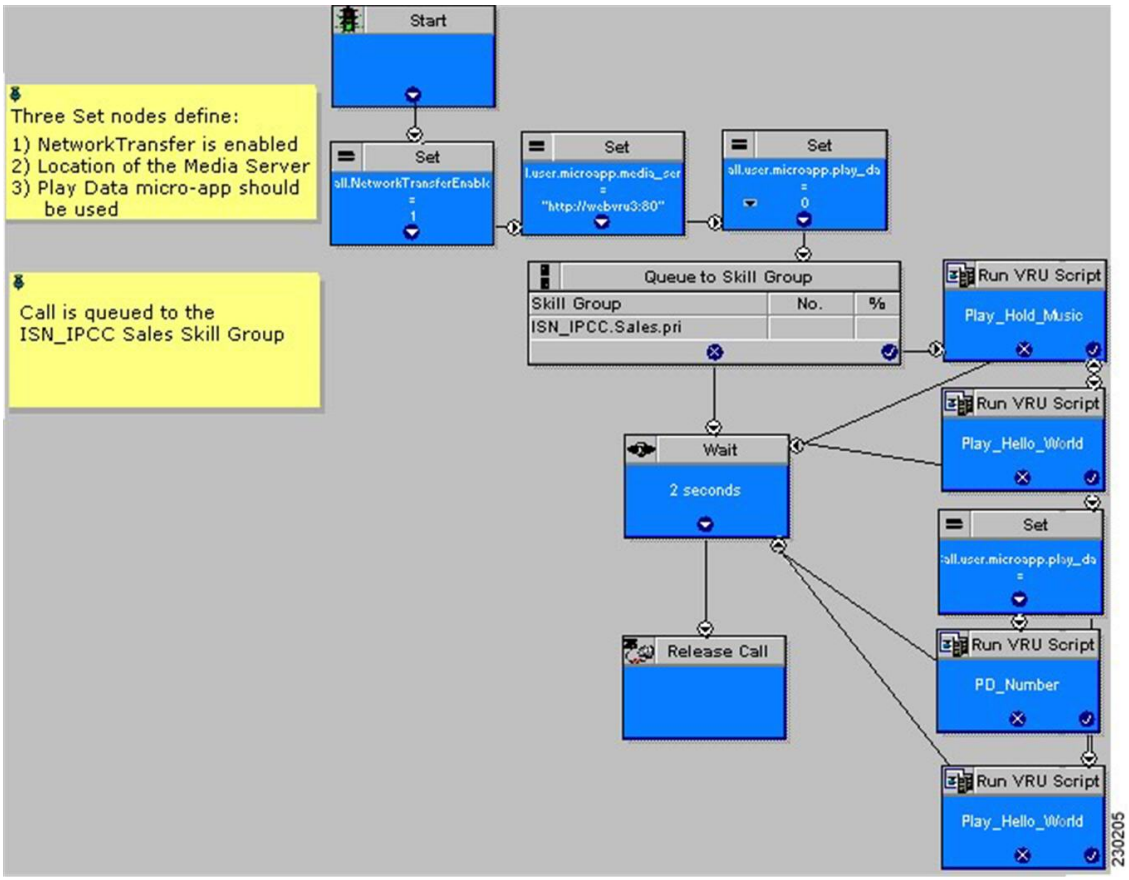

Run a VRU script to play a menu to the caller such as "For checking, press 1. For savings, press 2. To speak to a customer service representative, press 0."
Start
Queue
Run VRU Script
Menu: Queue
If the Caller Entered Digits (CED) is 0, then try to move the call to an appropriate agent group. This select will result in a Ced-125 Connect if an agent is available. If not available right away, the call will be put into a
CED
If CED is 0
Case 1: 0
Case 2: 1 2
Route Select
Service
No.
%
DallasVRUa.Payment_Info
DallasVRUb.Payment_Info
DallasVRUc.Payment_Info
DallasVRUd.Payment_Info
DallasVRUe.Payment_Info
Go To Script
Bank_IVR
If the flow gets to here, then an agent was not available. The call is queued to this set of agent skill groups.
Queue to Skill Group
Skill Group
Atlanta.Card_Service
Phoenix.Card_Service
Portland.Card_Service
At the same time, queue the call into this set of agent skill groups as well with a higher priority.
Queue to Skill Group
Skill Group
Atlanta.Service
Boston.Service
While the call is in the queue waiting for an available agent in either queue, give the caller an option for entertainment. The VRU menu played to the caller says, "While waiting for the next available agent, please select from the following options: To hear music, press 1. For the latest stock update, press 2. To no longer wait for an agent and return to the main menu of this automated system, press 3. To hear a duck quack, press 4. When the agent is available, depending if barge-in and VRU interruptibility has been set, the call will be sent to the next available agent (based on routing rules).
Run VRU Script
Menu: While waiting for agent
CED
2=Stock update 3=Go to Main
Case 1: 1
Case 2: 2
Case 3: 3
Case 4: 4
Run VRU Script
Menu: Music Choice
Run VRU Script
Menu: Play Stock Updates
Run VRU Script
Play Duck Quack prompt
Cancel Queuing
CED
1=Blues, 2=Rock, 3=Contemp
Case 1: (No entries)
Case 2: (No entries)
Case 3: (No entries)
Go To Script
Go To Main Menu Script
Run VRU Script
Play Blues Streaming Media
Run VRU Script
Play Rock 'n Roll Streaming Media
Run VRU Script
Play Contemp Streaming Media
230206

Example: Network Transfer Script

Unified CVP provides capabilities to transfer calls to another destination after they have been answered by an agent. These capabilities are referred to as Network Transfer. The Network Transfer feature does not require any special installation on the part of Unified CVP. The feature is disabled by default for all PG types except Enterprise Agent (EA).

To change the Network Transfer setting, perform the following steps:

1. Use the Script Editor's Set node to specify the **Call.NetworkTransferEnabled** variable. If you set this variable to 1, Network Transfer is enabled; if you set it to 0, Network Transfer is not enabled.
2. In EA PG setups where the EA is behind a PBX, use the **Network Transfer Preferred** check box on the PG Explorer's Routing Client tab. If this box is "checked," Network Transfer is enabled; if "unchecked," Network Transfer is not enabled.



Transfer a Call From Agent to Agent

When a call is transferred from Unified CVP to an agent, and that agent wants to transfer the call to another agent, the agent can make that transfer using either the agent IP phone or agent desktop. Transfers from the

IP phone are made using CTI route points that point to a Unified ICME script. Transfers from the agent desktop are made using the Dialed Number Plan.

For network transfer from either the IP phone or CTI OS Agent Desktop, you must Queue the call to skill group in the first Unified ICME script, for example "NetXfer1", to create the call context. In this script, the "networkTransferEnabled" flag must be set to "1".

Note The NetworkTransferEnabled setting must explicitly be set to 1 in all post-route scripts.

Configure Network Transfer From IP Phone

Procedure

Step 1 In Unified CM, define a CTI Route Point, for example "9999." Associate it with the JTAPI user that is connected to Unified CCE PIM in Unified ICME.

Step 2 In the ICM Admin Workstation, define a Dialed Number with a call Type for Unified CCE PIM. This call type can then be associated with a Unified ICME Script; for example, "NetXfer2".

Note Do not define the labels of agents for the Unified CCE PIM. Instead, define the labels for VRU PIM so that the route result will be returned to VRU instead of Unified CCE PIM. If you define the agent labels for the Unified CCE PIM, the Unified ICME router returns the route result to the VRU PIM if "Network Transfer Preferred" is enabled on the Unified CCE PIM and VRU PIM and returns the route result to the Unified CCE PIM if "Network Transfer Preferred" is disabled on the Unified CCE PIM and VRU PIM.

Step 3 When the call is delivered to Agent 1 using the Unified ICME Script "NetXfer1", the agent can dial the number 9999 to send the call to another script, "NetXfer2."

Configure Network Transfer From CTI OS Agent Desktop

Procedure

Step 1 Define a Dialed Number Plan in Unified ICME. The routing client is the Unified CCE PIM and dialed number will be the one defined before for the Unified CCE PIM; that is, IPCC_PIM.9999.

Step 2 Set Post Route to **Yes** and Plan to be **international**.

Step 3 In the Agent Desk Settings, check all the **Outbound access** check boxes.

IP Transfer Example

An IP transfer to an Unified CCE agent is very similar to an IP transfer to an ACD (TDM) agent with the following exceptions:

- The egress Gateway for this case is Unified CM.
- When Unified CM receives the new call, it uses the “Skinny protocol” to connect to the agent at an IP phone. The voice channels are then connected from the Ingress Gateway to the IP phone.

CLI Field on Outgoing Transfers

Calling Line Identification (CLI) is a set of digits and related indicators (type of number, numbering, plan identification, screening indicator, presentation indicator) that provide numbering information related to the calling party. This feature allows customers to override the CLI field on outgoing transfers, using either a Label node or an ECC variable in the Unified ICME routing script. This feature is required for transfers into Unity, which uses both ANI and DNIS to determine the appropriate mailbox to access. CLI is passed through most networks and into most call-handling devices, so this feature provides a back-door method to transmit arbitrary data during transfers when translation routing is not feasible.

The following section describes how to enable the **call.user.microapp.override_cli** ECC variable, which you must configure to enable this feature.

Configure CLI Override

CLI override is controlled from the Unified ICME routing script.

Note For IP originated calls, the "Asserted-Identity" check box will need to be unchecked on the CUCM SIP Trunk configuration.

Note For SIP calls, the CLI Override feature is only supported using the ECC variable as shown in Method #2 below. Using a dynamic label as in Method #1 with "CLI" prepended is not supported.

You can configure CLI override one of two ways:

Procedure

Step 1 **(First Method)** Append **CLI=NNNNNNNN** to the label in a LABEL node. Setting NNNNNNNN to the word **null** will blank out the CLI on the transfer.
Example: Setting a label node to **1111;CLI=9876543** results in a transfer to 1111 using a CLI of 9876543.

Example: Setting a label node to **1111;CLI=null** results in a transfer to 1111 using an empty CLI.

Step 2 **(Second Method)** Set the **call.user.microapp.override_cli** ECC variable *before* invoking a transfer using Queue to Skill Group, Label node, and so on. For the **call.user.microapp.override_cli** Expanded Call Variable List, set the maximum length to the maximum length of the data that will be used for CLI override. The Unified CVP Call Server must be restarted after adding this variable to Unified ICME. Setting the variable to the word **null** will blank out the CLI on the transfer.
Example: Setting **call.user.microapp.override_cli** ECC variable to **9876543** prior to a Queue to SkillGroup where agent **1111** becomes available, results in a transfer to 1111 using a CLI of 9876543.

Example: Setting **call.user.microapp.override_cli=null** ECC variable *prior* to a Queue to Skill Group where agent **1111** becomes available, results in a transfer to 1111 using an empty CLI.

If both of the above methods are used in one routing script, the LABEL node CLI value takes precedence over the ECC variable.

CLI override takes precedence over the SetSetupCallingNum command in VBAdmin. That is, the new CLI will always be propagated to the transfer call leg regardless of the value of ShowSetupCallingNum.

CLI override also forces the presentationIndicator to presentationAllowed on the transfer call leg.

Note For *SIP* calls, the CLI Override feature is only supported using the ECC variable. Using a dynamic label with "CLI" prepended is not supported.

Configure Unified CCE Re-Route on No Answer for Unified CVP

This section describes how to use the Re-route On No Answer function when using Unified CVP as a queue point for Unified CCE.

Summary

When using Unified CCE with Unified CVP as a queuing point and routing client, the Re-route On No Answer function needs to be configured differently than when using it with Unified IP IVR. The difference is when using Unified IP IVR the call control is with Unified CM, whereas with Unified CVP, the call control is with Unified CVP.

Re-Route on No Answer Operation for Unified CCE with Unified IP IVR

The Re-route On No Answer function ensures that when an agent does not answer a call - for example, because the walked away from his desk without making himself Not Ready - the call is taken away after ringing for a configurable number of seconds, it is presented to another agent or put back in queue, and the agent is put in Not Ready state.

This function is implemented by setting a Re-route On No Answer timeout in the agent desk settings. When the call has been ringing for the configured number of seconds, the Unified CM PG will make the agent unavailable and send a post-route request to Unified ICME using a dialed number that is also configured in the Agent Desk Settings. A routing script is executed that determines a new destination for the call. This can be another agent, or the script can put the call back in a queue. When using Re-route On No Answer with Unified IP IVR, Unified ICME software responds back to Unified CM with the new destination for the call. Unified CM is responsible for sending the call to the right destination (IP IVR for queuing or new agent).

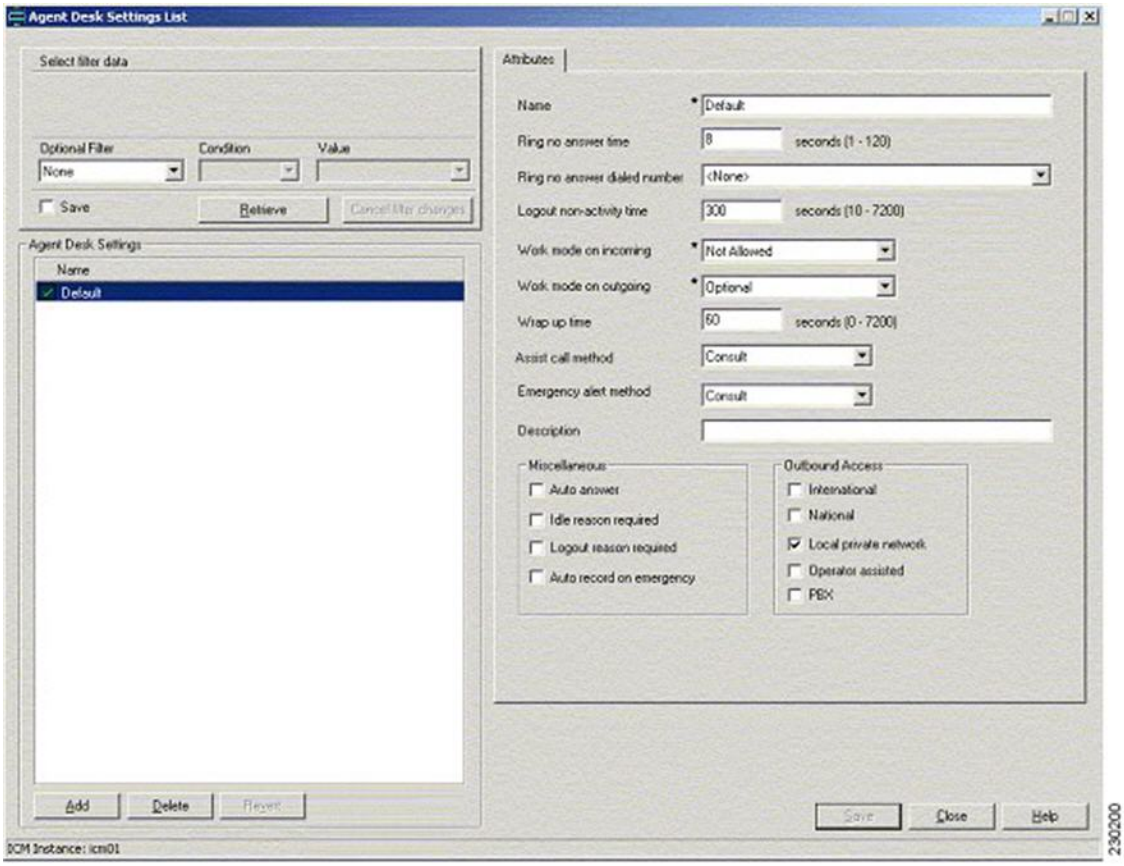
Re-Route on No Answer Operation with Unified CVP

When using Unified CCE with Unified CVP, Unified CM does not control the queuing platform (Unified CVP) and can therefore not send the call back to Unified CVP for re-queuing. Instead, Unified CVP controls the call and needs to take action.

The solution is to use the Re-route On No Answer function only to make the agent unavailable when the agent does not answer the call, and to use the ICM Router Requery function to take the call away from the non-answering agent.

Re-Route on No Answer Agent Desk Settings Configuration

Set a "Ring no answer time" in the Agent Desk Settings configuration, but do *not* set a "Ring no answer dialed number." Set the timeout to the maximum time you want to allow the agent to answer a call; for example, 2 rings = 8 seconds. Set this timer shorter than the no answer timeout for router requery (refer to the following example).



This will cause the agent to be made unavailable after the Re-route On No Answer timer expires, but will not invoke the Re-route On No Answer mechanism to re-route the call.

Router Requery Configuration

Router Requery is triggered by the routing client (Unified CVP) when a No Answer timer expires (a different timer than the Re-route On No Answer timer).

- The No Answer timer for Router Requery is not controlled by Unified ICME, but by the switching fabric, which is Unified CVP in this case. CVP 1.0 has a fixed No Answer timer of 15 seconds. The Unified CVP SIP has a configurable No Answer timer (RNATimeout) with a default value of 15 seconds.

  When using Unified CVP, set RNATimeout to the desired number of seconds that the agent phone should ring before being taken away; for example, less than 15 seconds (4 rings) such as 10 seconds. In any case, this timeout **must be longer than the Re-route On No Answer timeout** set in the Agent Desk Settings.

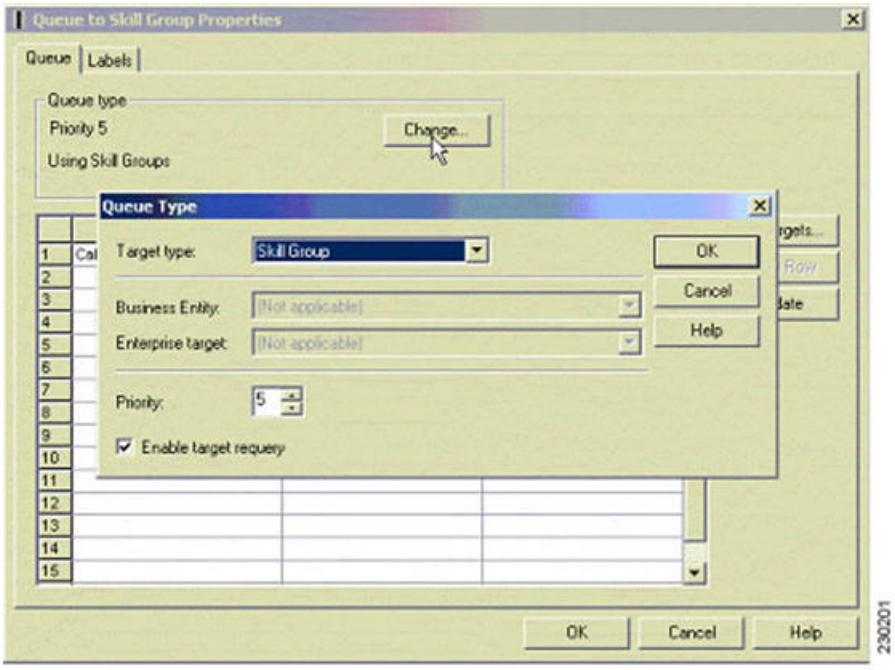
After the Unified CVP VB RNATimeout expires, the VB/AS/PG sends an **EventReport=NoAnswer** to the router. The router picks another target according to the routing script and sends the Connect message to Unified CVP. The target might be another agent or it might be a VRU label to requeue the call. When the call disappears from the first agent, this agent is put in Not Ready state due to the expiration of the No Answer Timeout in the desk setting.

Note Do not set the No Answer DN in the desk setting, because this is a global Unified ICME setting for all scripts. The No Answer DN may not be suitable for all scripts depending on the complexity of the deployment. Instead, each script should have the X path of the queue node set appropriately for each script.

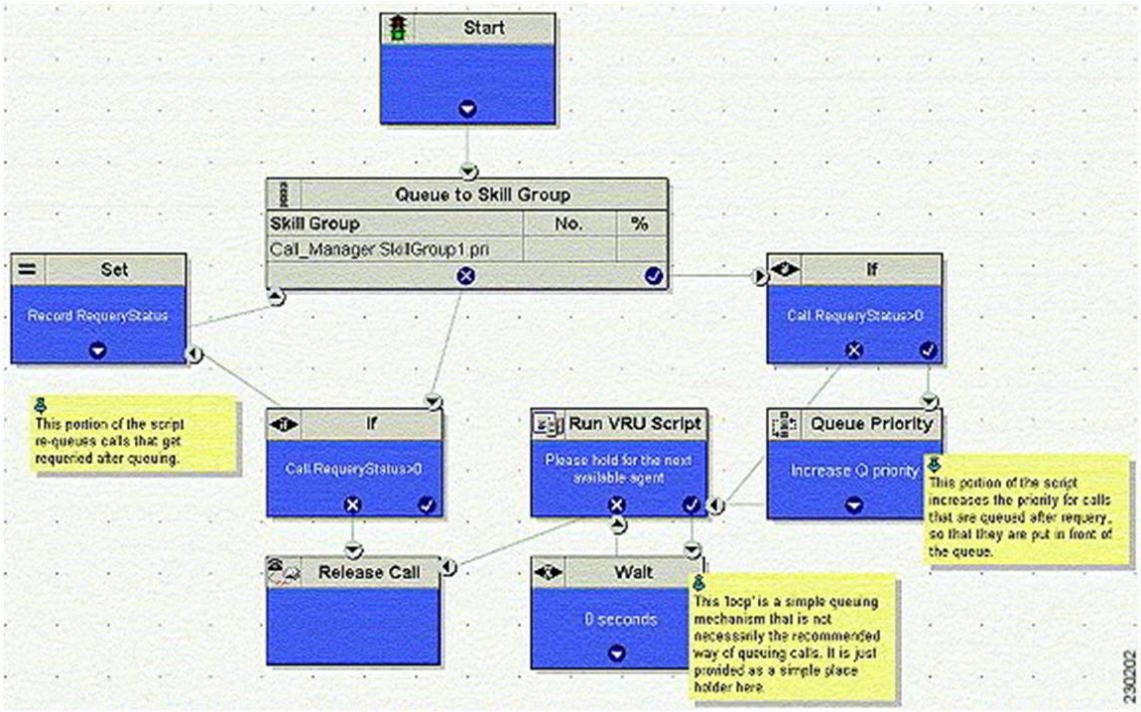
- Enable Requery on the node in the script that selects the first agent. Depending on the type of node used, the Requery mechanism will select a new target from the available agents or will require additional scripting. The Scripting and Media Routing Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted describes how Requery works for the different nodes.

In most cases Unified CCE will use the Queue node. The Queue node requires additional scripting to handle the requeuing of the call in front of the queue. The script example below provides a standard way of handling this.

If there is an available agent, the Queue node selects the longest available agent from the skill groups. If there is no available agent, it queues the call with a priority set in the node (refer to the following screen shot) and continues down the success exit of the node. When an agent becomes available, Unified ICME always selects the longest queued call from the ones with the highest priority. When the Queue node connects the call to an agent and the agent does not answer the call, Unified CVP Ring-No-Answer timeout will expire, causing the Requery mechanism to kick in.



When this happens, the script immediately continues through the failure exit of the Queue node with the Requery Status variable set to 'No Answer' (= 3). The typical treatment is to put the call back into the same queue but with a higher priority than all other calls, since the call needs to go in the front of the queue, not the back.



In this script, when the Queue node selects an agent who does not answer the call, the script exits through the failure exit (X) of the Queue node. The If node tests the RequeryStatus variable. If it has value of greater than zero, this is a requery call, and the script re-queues the call. In the preceding example, it also sets a flag using a call variable for reporting purposes. Assuming that there are no agents available, the Queue node immediately exits through the success exit (Checkmark). The If node checks to see if this is a requeried call. If so, it increases the Queue Priority of the call so that it will be handled before any other calls in queue. It then enters the normal wait loop with RunScripts.

The call flow is as follows:

- Script connects call to agent by sending connect message to Unified CVP (with requery enabled).
- Agent phone rings.
- After the Re-route On No Answer timeout expires, Unified ICME makes the agent unavailable. The agent state does not actually change until the call gets taken away from the agent. The agent phone continues to ring and the agent can still pick up the phone (if he does pick up the phone, he will be left in Ready state after the call, even if it was after the Re-route On No Answer timer expired).
- After the Unified CVP VB RNATimeout expires, the VB/AS/PG sends an EventReport=No Answer to the router. The router picks another target according to the routing script and sends the Connect message to Unified CVP. The target might be another agent or it might be a VRU label to requeue the call.
- When the call disappears from the first agent, this agent is put in Not Ready state.

Limitations

The configuration described in this section has the following limitation:

- Each call that is redirected by this mechanism is counted twice in the Skill Group: Once as redirected, and once as handled (if the call is finally handled). But the Call Type will only count this call once. Although it will be counted Handled and Requeried, Requeried is not used to balance CallsOffered in the Call Type. If you want to see this call counted twice in the Call Types, you can address this by changing the call type in the error path before the second queue to skill group node.

Configure Re-Route on No Answer for Unified CM with Unified CVP

In the case of an agent transfer, when calls are originated from Unified CM to a CTI Route Point, routing client responsibilities should be passed back to Unified CVP as soon as possible upon entering the Unified ICM script. To ensure that Unified ICM Router directs calls to Unified CVP, you need to include a SendtoVRU node in the Unified ICM script before any Runscript or SkillGroup node executes. When the routing script executes the SendToVRU node, the ICM Router instructs Unified CVP to become the routing client to handle for any subsequent transfers or VRU call processing.

RONA Operation to a script CTI Route Point Transfer

The Go to Script node is used as a RONA destination when "enable target requery" is configured on the Queue to Skill Group node and the agent does not answer. When the ICM script executes the Go To Script node, script execution proceeds to the specified script. For example, when an agent does not answer, the X-path out of the Queue to Skill Group Node will target a Go To Script node with the "CTI_Route_Point_Transfer" script specified. Script processing then continues from the beginning of the CTI_Route_Point_Transfer"script and proceeds normally.

Valid destinations out of the X-path of Queue to Skill Group node are:

- Another skill group
- A prompt
- GoTo node (do not use "Dynamic Label")

Related Topics

Router Requery Configuration, on page 309

Limitations

The configuration described in this section has the following limitations:

- The disposition of the requeried call is not correctly reported. The Redirect No Answer field in the agent and skill group reports do not show calls that are redirected by this mechanism. Each call that is redirected by this mechanism is counted twice: Once as abandoned, and once as handled (if the call is finally handled). There are two Unified CCE TerminationCallDetail records for this call: One for the rerouted call (with CallDisposition ‘Abandoned while Ringing’, code 3), and one for the handled call with a CallDisposition depending on how the call was finally handled. The scripting example above shows how a Peripheral Call Variable can be used to mark and count calls Requeried because of no answer. A custom reporting template can be written to report on this data.

Call Survivability

This section describes how to install and configure Unified CVP with a script that allows the gateway to transfer a call in the event of a critical Unified CVP application error or WAN failure. This application should be placed on the incoming pots dial-peer or the incoming VOIP dial-peer that is destined for Unified CVP. Call survivability is supported in all Unified CVP call flow models except the VRU-only call flow model. In the Unified CVP Standalone call flow model, survivability will be invoked if the gateway encounters an error from the CVP Voice Server, the "param survive" parameter has been included and a survivability service has been defined.

In the event of critical Unified CVP application errors or a WAN failure that would normally disconnect the caller, this script allows the gateway to attempt a transfer to some alternate location after the failure occurs instead of disconnecting the caller. In the event that the call cannot be transferred to an alternate agent, the script will play a "call-back-later" message and disconnect.

This script provides the following capabilities:

- Ability to perform multiple types of transfer in call failure conditions:
  - *8 transfer connect (outpulse)
  - Hairpin
  - SRST
  - Hookflash Relay
  - Two B-Channel Transfer (TBCT)
- Ability to differentiate call recovery behavior by incoming DNIS.
- Ability to differentiate call recovery behavior by incoming DNIS and how long the call had been in Unified CVP prior to failure.
- Ability to differentiate call recovery behavior based on time of day and date.
- Ability to hand off to an auto-attendant type application in the event of some downstream failure (for example, WAN failure, Unified ICME failure, Unified CVP failure). This auto-attendant functionality could be CME's BACD, a Unified CVP Standalone call flow model, a VXML Server application, or a custom-written VXML application.

Caution This script is a component of the Unified CVP software. Do not make any modifications to this script. Modifications to this script not made as part of an official Unified CVP release nullify Cisco support responsibility for this script.

Install Call Survivability Script

To install the Call Survivability script, perform the following steps:

Procedure

Step 1 Using the Operations Console, copy all script/prompt files to the gateway.

Step 2 On the gateway, perform the following:
For a **Unified CVP Comprehensive** call flow model, first define two services:

```
application
service survive flash:survivability.tcl
paramspace callfeature med-inact-det enable
service handoff flash:handoff.tcl
```

Then add the following parameters:

```
ip rtcp report interval 2000
gateway
timer receive-rtcp 4
```

Note This will cause survivability to be invoked between 8 and 16 seconds ((2000 ms *4) * 2) for an active call after a WAN failure. If IOS detects the absence of both RTP and RTCP packets after 8 to 16 seconds, it will raise an error event and survivability will be invoked. (The **factor of 2** is a built-in IOS factor that cannot be configured. *Do not* adjust these values lower as this could cause the survivability event to be prematurely invoked.)

Note The timer receive-rtcp command configures a media activity timer for SIP calls.

For a **Unified CVP Standalone** call flow model, first define one service:

```
application
service my-survivability-service flash:survivability.tcl
```

Note You can replace "my-survivability-service" with any name you choose.

Then associate the my-survivability-service that you just created as a parameter on the CVPSelfService.tcl service associated with the incoming pots dial-peer. Note that the text "param survive" must be entered exactly as shown, but the my-survivability-service can be renamed to the service name of your choice. For example:

```
dial-peer voice XXXX pots
service my-CVP-service
incoming called-number NNNNN
service my-CVP-service flash:CVPSelfService.tcl
param CVPPrimaryVXMLServer my-VXML-server-IP
param CVPBackupVXMLServer my-backup-VXML-server-IP
param CVPSelfService-app my-VXML-server-app
param keepalive my-CVP-service
param survive my-survivability-service
service my-survivability-service flash:survivability.tcl
```

Optionally, start a background keepalive service to the VXML Server. For example, for a service name of "my-standalone-service":

```
service my-standalone-service
param keepalive my-standalone-service
```

Note This service prevents the caller from hearing a period of silence at the start of each call if the VXML Server is down, as the gateway will know the current status of the VXML Server.

Step 3 On the gateway, perform a "call appl voice load my-survivability-service" and "call appl voice load handoff."

Step 4 Perform the following:
On a **Unified CVP Comprehensive** call flow model:

- Create a Unified CVP pots dial-peer on the gateway, placing the Unified CVP called number on an incoming-called-number parameter.
- Assign the my-survivability-service service to this dial-peer.

On a **Unified CVP Standalone** call flow model, no special survivability dial-peer needs to be created. However, the parameter "param survive my-survivability-service" must be included on the CVPSelfService.tcl service. This parameter indicates which service to run in the event of a system failure. In this way, different survivability services can be invoked depending on the incoming pots dial-peer invoked.

Configure the Gateway for Call Survivability

You can configure the following parameters on the gateway for call survivability:

- **open-hours-agent** - The destination recovery target DNIS to be used when the current time matches any open-hours-time parameter. The script will cycle through all agents sequentially until one answers. If no one answers, (or in the case of a takeback transfer, the PSTN does not take back the call), the script cycles through all after-hours-agent's (maximum of 50 agents).
  a) **Syntax:** open-hours-agentX DNIS
  b) **Arguments:** X = a number from 0 to 49, DNIS = target destination for the recovery transfer.
  c) **Example 1:**DTMF*8,9875551212 (When PSTN *8 takeback is desired), where **DTMF** - Indicates takeback and transfer via DTMF tones ***8** - The sequence the switch recognizes to perform the takeback. **Zero or more commas** - Each comma represents a pause of 100 ms. Some switches require a pause between the takeback sequence and the DNIS. **9875551212** - The actual DNIS to which the PSTN should transfer the call.
  d) **Example 2:** HF,,,,,9875551212 (when hookflash transfer is desired) where: **HF** - Indicates takeback and transfer via hookflash relay **Zero or more commas** - Each comma represents a pause of 100 ms. Some switches require a pause between the hookflash and the DNIS. **9875551212** - The actual DNIS to which the switch should transfer the call. **Note:** When using either DTMF or hookflash takeback, you need to configure the following additional parameters on the gateway voice ports: voice-port 7/1:0no echo-cancel enable no non-linear no vadplayout-delay maximum 250playout-delay nominal 200playout-delay minimum highplayout-delay mode fixed
  e) **Example 3:** 9875551212 (when hairpin or SRST transfer is desired)
  f) **Example 4:**TBCT9875551212 (when TBCT is desired)

g) **Example 5:<retry>** (when a retry to the original CVP DNIS is desired) - Assuming the original Unified CVP DNIS was *4444:*, **<retry>** will send the call to CVP using DNIS. 4444 **56<retry>78** will send the call to CVP using DNIS 56444478.

- **after-hours-agent** - The destination recovery target DNIS to be used when the current time matches any after-hours-time parameter or as a default destination if transfers to the open-hours-agent's fail. The script will cycle through all agents sequentially until one answers (maximum of 50 agents). If no one answers, a call-back-later message will be played to the caller and then disconnected.
  a) Syntax: identical to open-hours-agent
- **open-hours-time** - A string representing the date or days of week and time of day that open-hours-agent's will be used for the recovery transfer (maximum of 20 values). Month/day has higher selection priority than days of the week.
  a) **Syntax:** open-hours-timeX {month/day | days-of-week}[:HHMM-HHMM]
  b) **Arguments:X** = a number from 0 to 19, **month/day** = month of year and day of month (no year), **days-of-week** = a string of up to seven digits representing the days of the week (Sunday = 0, Saturday = 6), **HHMM-HHMM** = the starting and ending time of the period, expressed in 24-hour clock notation.
- **after-hours-time** - A string representing the date or days of week and time of day that after-hours-agents use for the transfer. These do not explicitly need to be listed. If the current date/time does not fall in an open-hours-time slot, it will default to an after-hours agent. A typical use would be to specify holidays that would normally fall on working weekdays. A maximum of 20 values are allowed.
  a) **Syntax:** identical to open-hours-time
- **open-hours-cvptime** - You may want to choose a particular recovery agent based on how long the call had been in Unified CVP before the failure occurred. If no open-hours-cvptime is specified, the associated open-hours-agent will be used regardless.
  a) **Syntax:** number-of-seconds
  b) **Arguments: X** = a number from 0 to 19, corresponding to the associated open-hours-agent **number-of-seconds**55 would use open-hours-agent0 only when the call had been in Unified CVP less than 55 secs.
- **after-hours-cvptime** - Same as open-hours-cvptime, but applies instead to after-hours-agents.
- **alert-timeout** - A numeric value indicating the maximum number of seconds the destination phone should ring before aborting the call attempt.
  a) **Syntax:** alert-timeout 20
- **setup-timeout** - A numeric value indicating the maximum number of seconds that the tcl script will wait in establishing a tcp connection to Unified CVP before aborting the call attempt. This value should be greater than the "h225 timeout tcp establish" parameter under the voice class h323 configuration on the gateway.
  a) **Syntax:** setup-timeout 7
- **aa-name** - If non-blank, indicates that when a failure occurs, the Unified CVP survivability script hands off the caller to the BACD auto-attendant application. Enter the following:

```
service <survivability-servicename>
param aa-name <BACD-servicename>
service <BACD-servicename>
param isn-name <survivability-servicename>
```

Where servicename is the service name of the BACD auto-attendant script to which control should be passed.

- **standalone** - If non-blank, indicates that when a failure occurs, this Unified CVP survivability script passes control to the service name specified. Typically, that service would reference the CVPSelfService.tcl script to invoke a Call Studio application to provide IVR treatment to the caller; for example:

```
service survivability flash:survivability.tcl
param standalone vxmlapp
service vxmlapp flash:CVPSelfService.tcl
```

- **standalone-isntime** - Select the standalone option depending on how long the call had been in Unified CVP before the failure occurred. If no standalone-isntime is specified, the standalone option is invoked if it is *non-blank*.
  a) **Syntax**: standalone-isntime {> OR <}number-of-seconds
  b) **Arguments**: **number-of-seconds** = number of seconds the call was in Unified CVP before the call failed, prefixed with > or <. For example, standalone-isntime <2 would use standalone only when the call had been in Unified CVP less than 2 seconds.
- **icm-tbct** - A numeric boolean value (0 or 1) indicating whether or not Unified ICME scripts will issue TBCT transfers. Default is 0 (by default, Unified ICME does not handle TBCT transfers). Set this value to 1 to enable TBCT transfers issued from a TBCT label in an Unified ICME script.
  a) **Syntax Example:** icm-tbct 1
- **disableDnisStrip** - By default survivability.tcl will strip of all leading zeros from the dialed number. To disable this , you can set the disableDnisStrip parameter to a value of 1.
  a) **Syntax Example:** disableDnisStrip 1

Call Survivability Examples

In the first Call Survivability example, the following configurations are used:

```
service survivability flash:survivability.tcl

param open-hours-agent0 9777123400
param open-hours-agent1 4444888
param open-hours-time0 12345:0900-1730
param open-hours-time1 12/18:0600-2300

param after-hours-agent0 7777008
param after-hours-agent1 8766008
param after-hours-time0 7/21:0700-0800
param after-hours-time1 11/25

param setup-timeout 7
param alert-timeout
dial-peer voice 800232 pots
application survivability
incoming called-number 8002321765
direct-inward-dial
```

Using the above survivability configurations, review the following cases:

- Case 1: Assume today is a holiday, Thursday, 11/25 at 1300 hours. Since 11/25 is defined as a specific after-hours-time, it will be selected before the 12345:0900-1730 open-hours-time which also falls on a Thursday. If the WAN fails, this script will first try a transfer to 7777008 and then to 8766008.

- Case 2: Assume today is Saturday, 12/18 at 0900 hours, peak of the holiday shopping season. Since 12/18 is defined as a specific open-hours-time, it will be selected for an open-hours-agent even though it falls on a Saturday which would normally be an after hours time. If the WAN fails, this script will first try a transfer to 9777123400, then try 4444888, 7777008, 8766008.
- Case 3: If time-of-day routing is not important, but you need a last-resort transfer mechanism, put one or more DNIS in the after-hours-agent slots and do not define any times. Any failed call will always be directed to the list of after-hours-agents.

The next example illustrates how to organize call survivability functionality by incoming DNIS, create a separate application for each DNIS and apply desired call recovery properties to each application. For example:

- Assume billing callers dial 45XX and sales callers dial 55XX to access Unified CVP.
- Assume that a billing call fails somewhere in the course of the call:
  - If the call fails and the call had been in Unified CVP less than 30 seconds (this would also include the case where the call had *never* made it to Unified CVP; for example, 0 seconds), send the caller back through the PSTN via a *8 takeback to 8005556666.
  - If the call fails and the call had been in Unified CVP greater than or equal to 30 seconds, send the caller back through the PSTN via a *8 takeback to 8007778888.
- Assume that a sales call fails somewhere in the course of the call:
  - If the call fails (in this case, the amount of time the call had been in Unified CVP is irrelevant), send the caller back through the PSTN via a hairpin transfer to 8009990000.
- Assume the PSTN switch is sending ANI and DNIS in such a way that the ANI and DNIS are concatenated together in the DNIS field. Assume that ANI length is 10 and DNIS length is 4. Also assume that ANI can be blank; for example, blocked callerID.

The IOS configuration elements necessary to accomplish these cases are shown below.

Note Dial-peers 2 and 4 are necessary in the event of no ANI (blocked caller ID). The lower preferences of dial-peers 2 and 4 is to protect against the case where a caller's ANI begins with 45, for example. For example, assume caller with ANI 4521111111 dials the sales DNIS. Without lower preferences, the caller would have matched dial-peer 2 and gone to the billing application instead of sales (you wanted it to match dial-peer 3).

The following are the configuration elements for the second example:

```
dial-peer voice 1 pots
preference 1
application billing
incoming called-number 45..
#-----------------------------------------
dial-peer voice 2 pots
preference 2
application billing
incoming called-number 45..
#-----------------------------------------
dial-peer voice 3 pots
preference 1
application sales
incoming called-number 55..
#-----------------------------------------
```

```
dial-peer voice 4 pots
preference 2
application sales
incoming called-number 55..
#-----------------------------------------
dial-peer voice 5 pots
destination-pattern 8009990000
port 7/0:D (or whatever port is desired)
#-----------------------------------------
dial-peer voice 6 voip
incoming called-number 8009990000
codec g711ulaw (To force the call to g711ulaw on the outgoing
hairpin)
#-----------------------------------------
service billing flash:survivability.tcl
param after-hours-agent0 DTMF*8,,,8005556666
param after-hours-cvptime0 <30
param after-hours-agent1 DTMF*8,,,8007778888
param after-hours-cvptime1 >29
param ani-dnis-split 10:4
#-----------------------------------------
service sales flash:survivability.tcl
param after-hours-agent0 8009990000
param ani-dnis-split 10:4
```

Enhanced Location Call Admission Control

ELCAC is used to maximize local branch resources, keeping a call within the branch whenever possible and limiting the number of calls that go over the WAN. Unified CVP Revision 9.0(1) supports *queue-at-the-edge*, a simpler and more effective configuration of ELCAC than the Transfer and Queue Calls with Unified CVP. Using the queue-at-the-edge functionality, the call originating from a specific branch office is deterministically routed to a local VXML Gateway based on priority, which means that ELCAC always selects a local branch agent if possible.

Note For design discussion and design considerations when using ELCAC, see the Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND) guide.

ELCAC Topic Definitions

The following definitions are used in the configuration of ELCAC:

- **Phantom Location**: A default location with unlimited bandwidth used when calculating calls that are hairpinned over a SIP trunk or when the SIP call is queued at the local branch, to enable correct bandwidth calculations. The Phantom location should be assigned to the gateway or trunk for CVP.
- **SiteID**: The SiteID is a string of numbers that is appended to the label from Unified ICM so that the dial plan can be configured to route the call to a specific destination, such as the branch VXML gateway or egress gateway, or Unified CM node. The SiteID can be appended at the front of the label, at the end, or not at all. This configuration is separate from the Unified CM location configuration, and is specific to Unified CVP. The SiteID is used to indicate the real location of the call and allow the bandwidth to be deducted from the correct location.
- **Shadow Location**: This new location is used for inter-cluster trunks between two Cisco Unified Communications Manager clusters. This location is not used as inter-cluster ELCAC is not supported in Unified CVP 9.0(1).

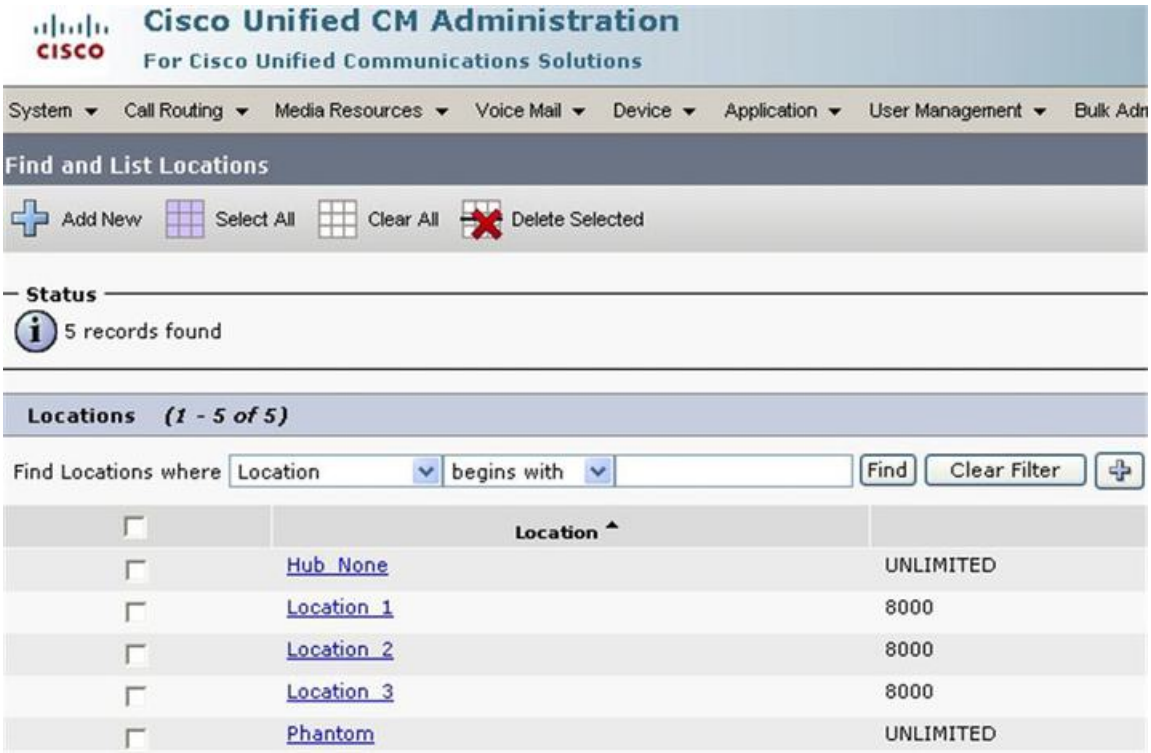
Configuration of ELCAC Queue-at-the-Edge

The following steps provide an example configuration for ELCAC with queue-at-the-edge functionality.

Using the Unified CM, configure all branches so that Location and Bandwidth are defined:

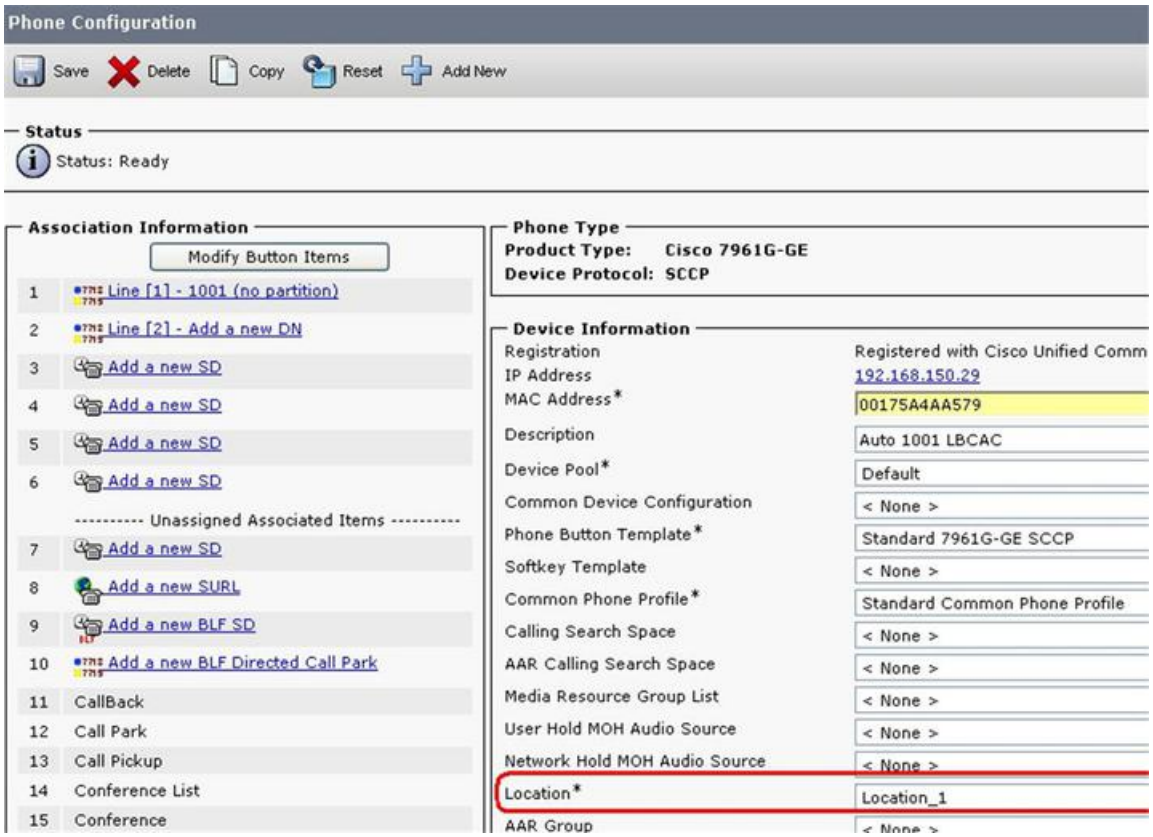
1 From Unified CM Administration, select **System** > **Location**. Click **Find** to list the locations and add new ones as appropriate.

Note **Unlimited** must be *unchecked* for each branch (the box to the left of the location name); otherwise bandwidth will not be deducted for that branch. (The Phantom location will still have unlimited bandwidth even when unchecked.)



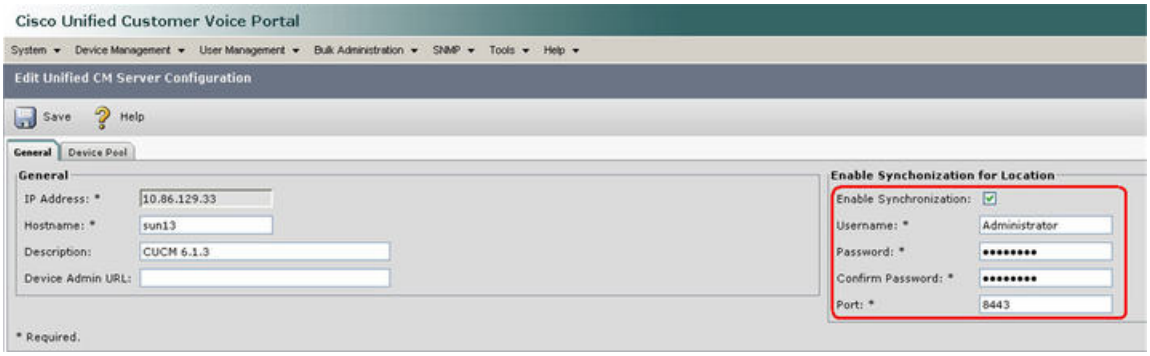
2 For the branch phones, configure each phone so that it is assigned the branch location for that phone.

- Select **Device** > **Phone**. Click **Find** to list the phones.
- Select a phone and set the **Location** field.



3 Verify that the Cisco AXL Web Service is started and that an Application User is defined and has a role of *Standard AXL API Access*.

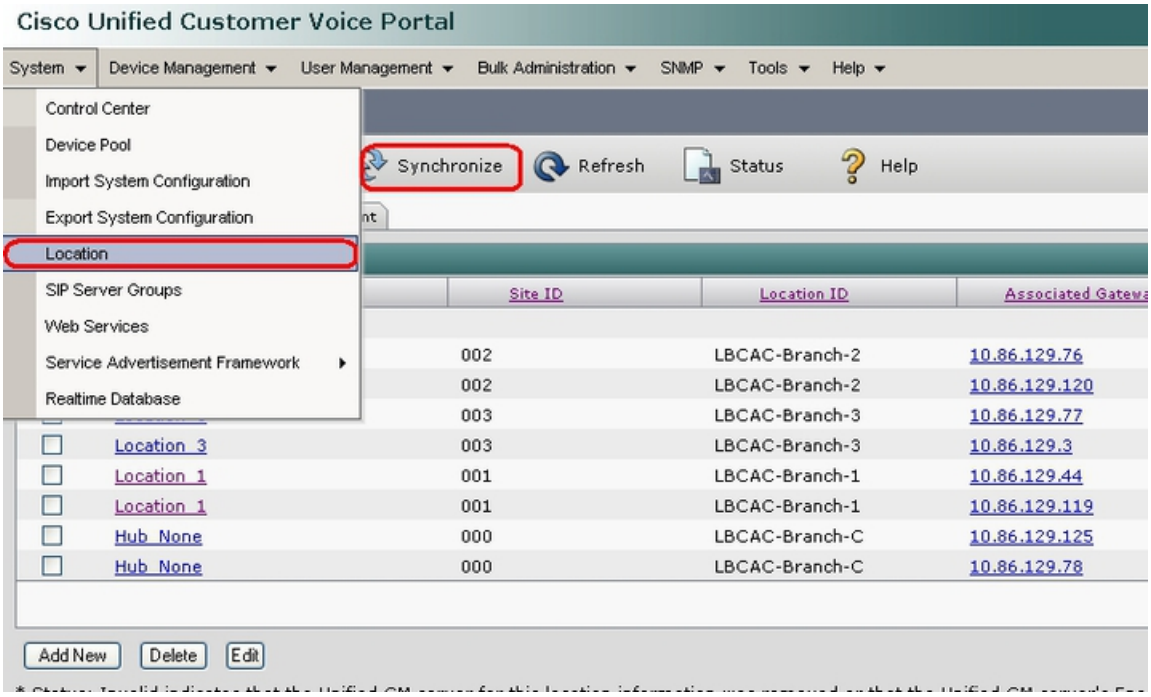
- From Cisco Unified Servicability, select **Tools** > **Control Center** > **Feature Services**
- Start the Cisco AXL Web Service, if it is not started.
- From Cisco Unified CM Administration, select **User Management** > **Application User**. Verify you have a user with the role of *Standard AXL API Access*, or create a new one and add that user to a group that has the role of *Standard AXL API Access* .



On Unified CVP, perform the following steps using the Operations Console:

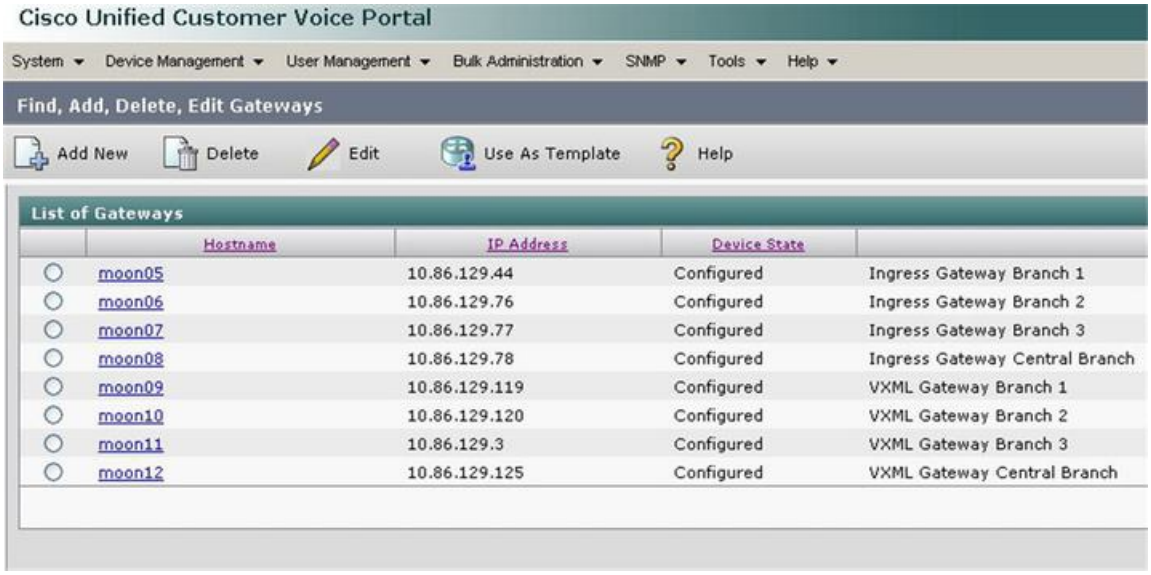
1 In **Device Management** > **Unified CM**, in the section **Enable Synchronization for Location** , enable synchronization and provide the credentials required to log in.

2 In **System** > **Location**, click **Synchronize** to retrieve the locations defined on Unified CM.



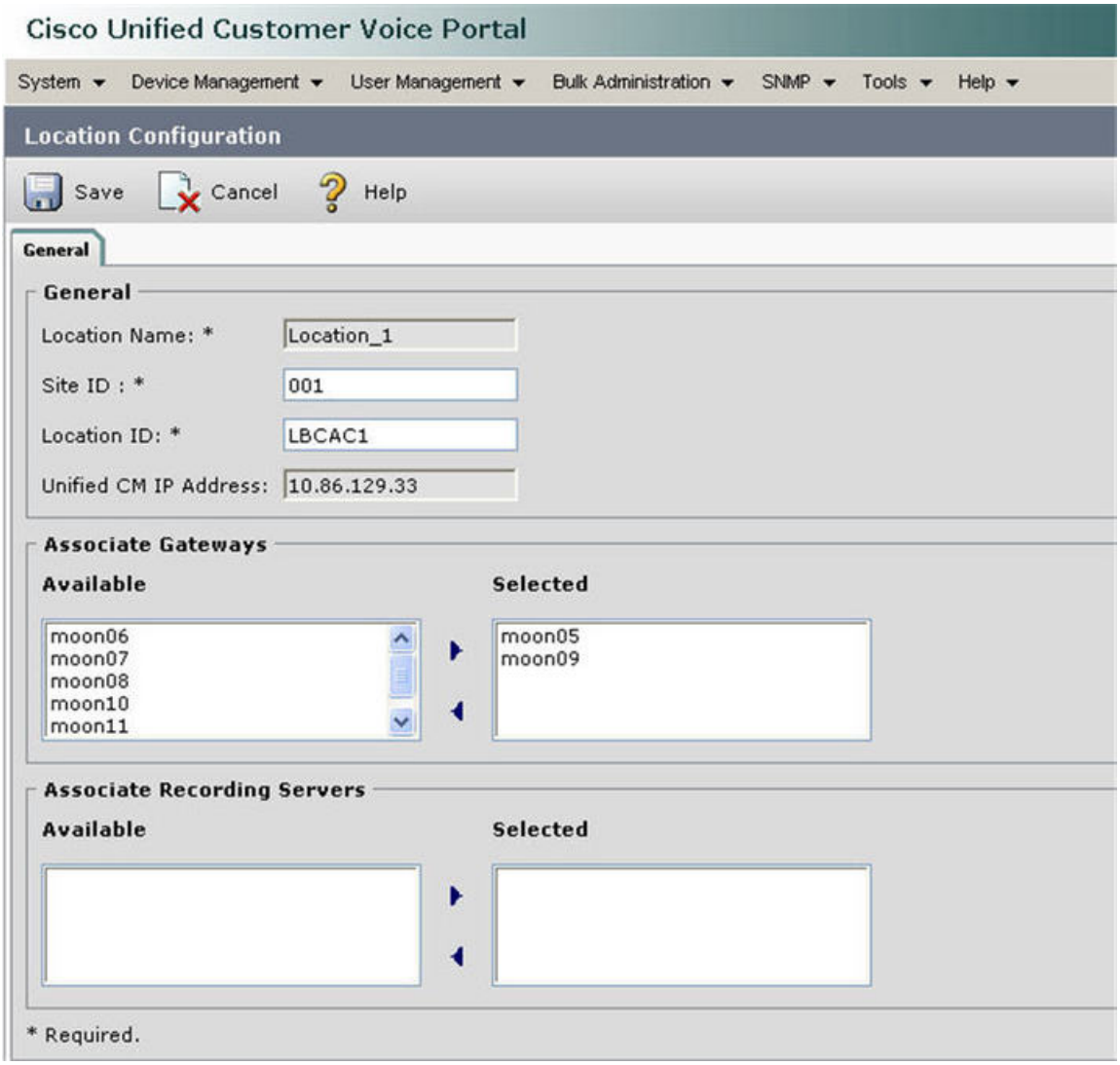
Select **System** > **Location** and verify that the locations have been synchronized from Unified CCM.

3 In **Device Management** > **Gateway**, define the Ingress and VXML gateways.

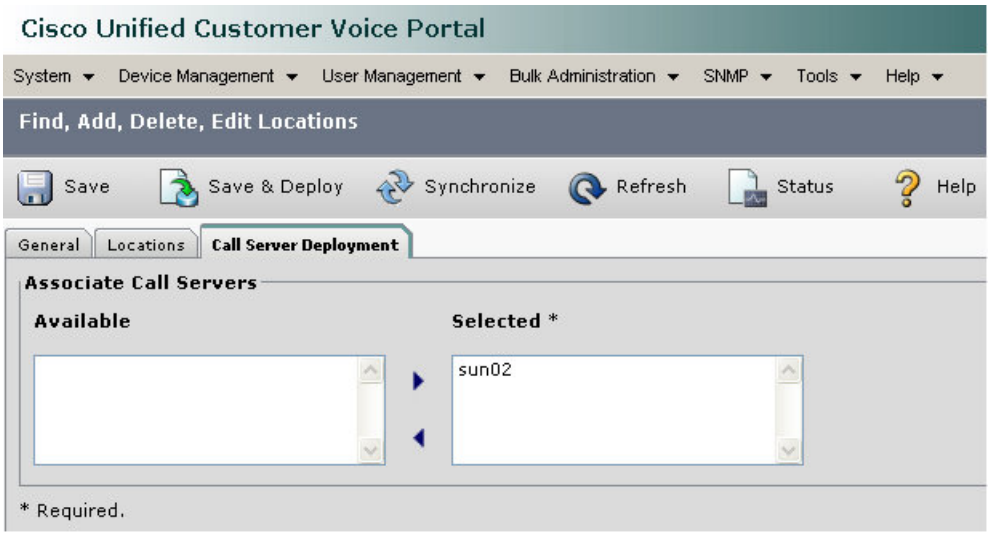


4 **Assign IDs.** In **System** > **Location**, select a location.

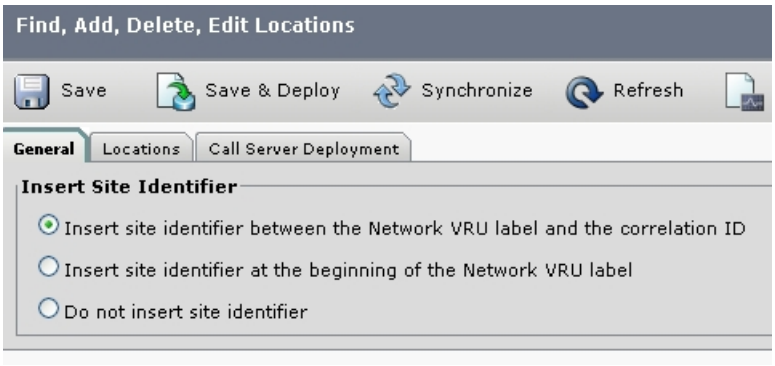
- Assign a Site ID and Location ID to the location, then add the associated gateways to the location.
- Repeat for each of the locations.



5 In **System** > **Location**, navigate to **Call Server Deployment** and select the Call Servers where the configuration is to be deployed. Click **Save and Deploy**.

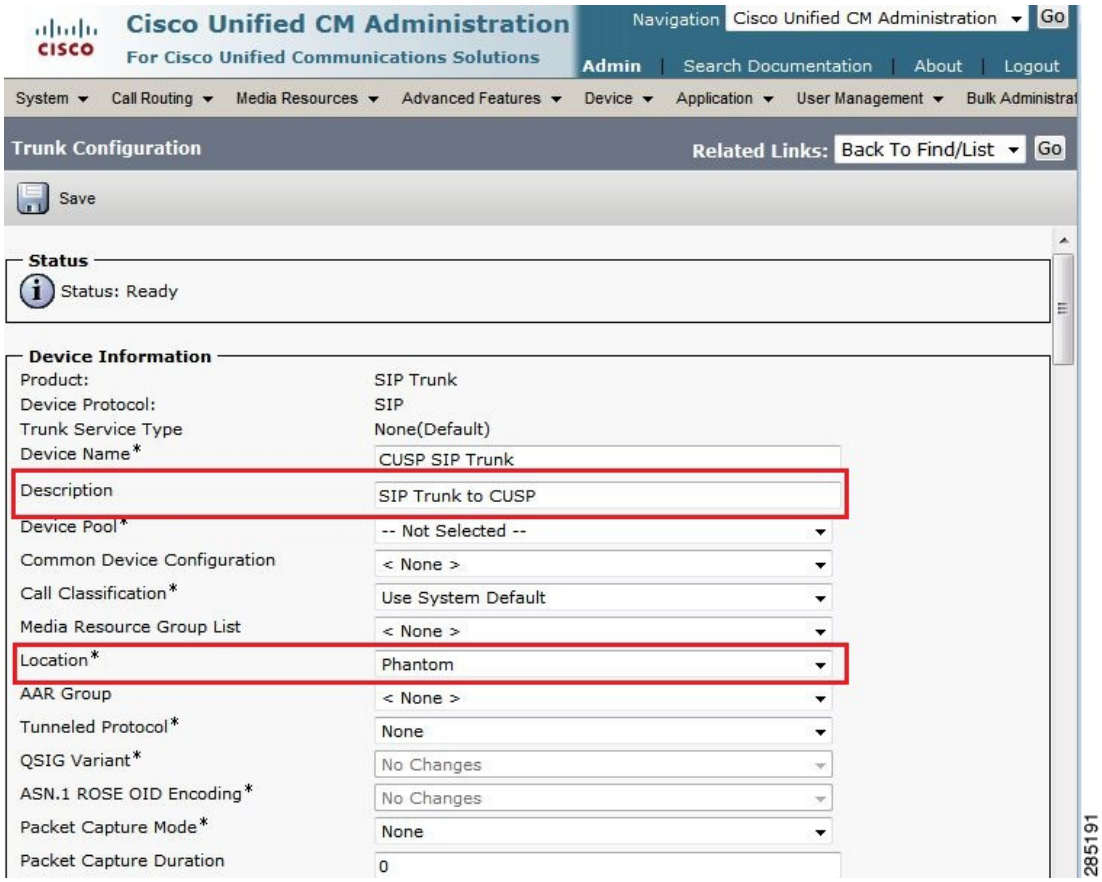


6 For the insertion point of the SiteID, use the default location *between the Network VRU label and the correlation ID* as shown in the following screenshot.

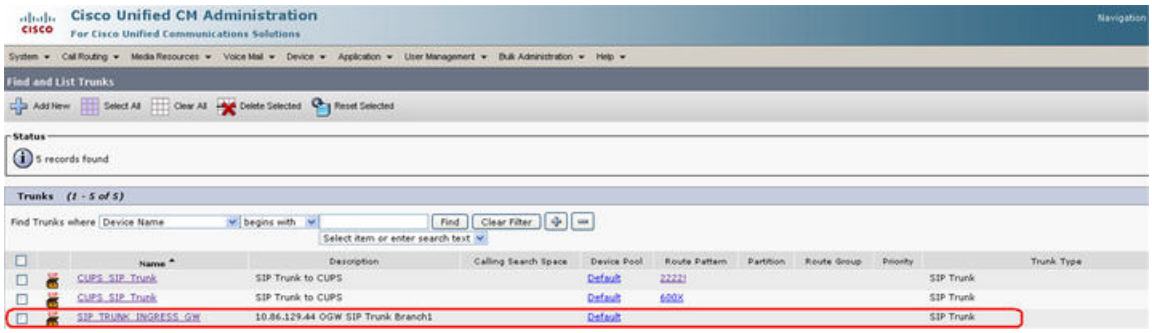


SIP Deployments: Unified CM Steps

1 Using Unified CM, create a SIP trunk toward the SIP proxy server and select the *Phantom* location.



2 Create a SIP trunk for each ingress gateway and make the location of these ingress TDM-IP gateways the actual branch location.



3 Create a route pattern pointing the Network VRU Label of the CCM routing client to the SIP trunk toward the SIP proxy you created in Step 1.

The SIP proxy should route the Network RRU label of CCM routing client to the farm of CVP Call Servers.

4 For any IP-originated calls, the CCM route pattern should be associated with the SIP trunk created in Step 1.

5 Using Unified CM Administration, select **Device** > **Device Settings** > **SIP Profile** > **Trunk Specific Configuration** > **Reroute Incoming Request to new Trunk based on** > **Call-Info header with the purpose equal to x-cisco-origIP**.

6 Associate the new SIP profile from Step 3 with the trunk defined in Step 1 and each Ingress gateway defined in Step 2.

Configure Locations-Based Call Admission Control

Locations-based call admission control (CAC) is used in the Unified CCE branch-office call flow model (also known as the Centralized Model). This means that all servers (Unified CVP, Unified ICME, Unified Communications Manager, SIP Proxy server, and media servers) are centralized in one or two data centers, and each branch office (of which there can hundreds or thousands) contains only a gateway and IP phones.

This section provides an overview of how to configure Unified CVP to:

- Accommodate Unified CM locations-based CAC
- Minimize bandwidth usage on the WAN

This section also describes other call flow and bandwidth usage issues to consider.

The following sections do not include detailed installation and configuration instructions. Rather, they are intended to provide you with guidance as you set up the Unified CVP solutions in your network. For additional information about how to install, set up, run, and administer Unified CVP, refer to Installation and Upgrade Guide for Cisco Unified Customer Voice Portal

Configure Unified CM Service Configuration Settings

Set the following configuration parameters to make Unified CM use the Ingress gateway instead of Unified CVP as the originating location of the call.

- Set "Accept Unknown TCP connection" in Unified CM Service parameters.
- Set the Unified CM Service parameter "GK controlled trunk that will listen to 1720" to “None”.
- *Do not* define Unified CVP as a gateway device in Unified CM.
- Define the Ingress gateways as gateway devices in Unified CM. Assign the correct location to the devices.

These settings ensure that CAC can be properly adjusted based on the locations of the calling endpoint and the phone.

Unified CVP Bandwidth Utilization

The following factors contribute to WAN bandwidth usage by Unified CVP in a Centralized Call Control with Distributed Queuing call flow model:

- VoiceXML Documents
- Prompt Retrieval

The following sections describe the bandwidth requirements of these factors in an example Centralized Call Control with Distributed Queuing call flow model. The examples in these sections are based on data that Cisco obtained from testing.

In these examples, assume that:

- Each call begins with some IVR treatment followed by a transfer to an agent.
- Each branch has 20 agents and each agent handles 30 calls per hour. Thus, the total number of calls is:

  20 * 30 = 600 calls per hour = 0.166 calls per second (CPS).

VoiceXML Documents

A VoiceXML document corresponds approximately to a Run External node in a Unified ICME script.

A round trip of a VoiceXML document between Unified CVP and the gateway consumes an average of 7 KB (7,000 bytes). If each call includes approximately 20 VoiceXML documents, the WAN bandwidth consumed by VoiceXML documents can be calculated as follows:

- 7,000 bytes * 20 VoiceXML documents * 8 bits = 1,120,000 bits per call
- 0.166 CPS * 1,120,000 bits per call = 185.9 Kbps per remote site

Prompt Retrieval

Voice prompts can be stored in the following locations:

- In flash memory on each local site gateway - In this way, gateways do not need to retrieve .wav files for prompts and WAN bandwidth is not affected. However, if a prompt needs to change, you must change it on every gateway.
- On an HTTP media server - In this way, each local site gateway (if properly configured) can cache many or all prompts, depending on the number and size of the prompts.

When prompts are stored on an HTTP media server, the refresh period for the prompts is defined on that server. The bandwidth consumed by prompts consists of the initial loading of the prompts at each gateway and of the periodic updates at the expiration of the refresh interval.

As an example of determining the bandwidth consumed by prompts, assume that a call flow has 50 prompts with an average size of 50 KB (50,000 bytes) each. Also, assume that the refresh period for the prompts is defined as 15 minutes (900 seconds) on the HTTP media server.

The WAN bandwidth required for prompts in this call flow can be calculated as follows:

- 50 prompts * 50,000 bytes * 8 bits = 20,000,000 bits
- 20,000,000 bits / 900 seconds = 22.2 Kbps per branch

Gateway Prompt Caching Considerations

When audio prompts are stored on an HTTP media server, proper gateway prompt caching methods are necessary to optimize both the performance of the gateway and network bandwidth consumption. Gateway performance decreases by approximately 35-40% if caching is disabled entirely.

Configure Caching on the Gateway

To configure caching on the gateway, perform the following procedure:

Procedure

Step 1 Set the following settings on the gateway:

a) ivr prompt memory 15000

b) http client cache memory file 500

c) http client cache memory pool 15000

Note The 'http client cache memory file' represents the largest size prompt file (in Kbytes) that can be cached. In general, break up customer prompts larger than 500K (about a minute in length) into smaller, more manageable pieces to facilitate loading and caching. For example, queue music could be a repetitive loop of a 30 second prompt. Note also that because the prompts are streamed, the prompt will not be cached unless the whole prompt is played. Therefore, you must make prompts a manageable size.

Step 2 Synchronize the datetime between the gateway and the HTTP media server.

Note Synchronization does not have to be exact, but at least within a minute or two. Times that are not synchronized can cause prompts to never refresh or they will refresh with every call, both of which are undesirable behaviors.

Step 3 On the media server, set the content expiration (for example 15 minutes).

What to Do Next

Note In IIS, this is done under the "HTTP Header" tab. The gateway prompt will be refreshed after this time period. The period chosen reflects how often the record prompts and how long you are willing to wait to have the new prompt load after modification.

Determine If the Gateway Is Caching Properly

To determine if you have properly configured gateway caching, perform one of the following actions:

- The IIS log on the media server records every time a client requests a prompt. If caching is set up correctly, these requests appear approximately every X minutes (X is whatever was defined as the refresh interval for any particular prompt. The log is located at: **C:\WINNT\system32\LogFiles\W3SVC1\ex***.
- Run 'show http client cache' on the gateway. The 'Fresh Time' column equals the refresh time period set on the HTTP media server. For example, if the refresh period was set to 15 minutes, it says 900 seconds. The 'Age' column shows how many seconds have passed since the prompt was last refreshed. In general, this number will be less than the 'Fresh Time'. However, if no call has ever accessed the prompt recently, this number could be greater than the fresh time. Prompts are only refreshed when triggered by a call *and* the prompt 'Fresh Time' has expired. If the Fresh Time is a very high value, the only way to remove the prompt from cache is to reload the gateway.

UUI to Be Used as the Correlation ID

Unified CVP uses the User-to-User Information (UUI) from the incoming call as a Correlation ID in the VRU-Only call flow model. This feature allows customers to transfer Correlation IDs through their network; for example, using a Call Routing Service Protocol (CRSP) NIC for call control.

Note This feature applies only to the Unified CVP VRU-Only call flow model.

The network typically has no place to store a Correlation ID, so it must be "hidden" in the ISDN setup that arrives at the IOS gateway and then is extracted by the gateway. The UUS parameter (often known as the User-to-User Information (UUI) of the Generic Transparency Descriptor (GTD) data) can be used to "hide" the Correlation ID, provided the call control client has the capability of inserting a Correlation ID value into the GTD.

When the call arrives at the gateway from the network, the call control client extracts the value and appends it to the DNIS before sending an HTTP request to the Type3 Unified CVP Call Server.

How It Works

The call control client (such as the CRSP NIC) inserts the desired Correlation ID value into the dat field of the UUS parameter of the NSS IAM message. These NSS messages are used as the basis of building the GTD data that ultimately arrives at the IOS gateway from the PSTN. Refer to the ITU-T Narrowband Signaling Syntax spec (Q.1980.1) for a detailed description of the IAM message and UUS parameter, included below for convenience. Note that the dat field contains pairs of hexadecimal digits, meaning that if the Correlation ID is "12345", the dat field must be populated as "3132333435". The gateway bootstrap.tcl script converts back to "12345" before appending to the DNIS and passing to the Unified CVP Call Server in the HTTP URL.

Configure the Gateway

Follow the instructions below to configure the gateway to enable incoming UUI to be used as the Correlation ID.

```
conf t
application
service <your-cvp-service-name>
param use-uui-as-corrid Y (Refer to Note 1)
param correlation-gtd-attribute XXX (Refer to Note 2)
param correlation-gtd-instance N (Refer to Note 2)
param correlation-gtd-field YYY (Refer to Note 2)
dial-peer voice 123 pots
service <your-cvp-service-name>
```

Note This is a mandatory parameter to enable this feature.

Note These parameters are optional. They need only be specified if the call control client placed the Correlation ID in a GTD parameter other than uus.dat.

Debugging Tips

The following two sections provide debugging tips.

Debug Trace Settings for the Gateway

On the gateway, enter the following:

```
debug voip application script
debug gtd
```

GTD Values in the Gateway Log

In the gateway log, look for the GTD values:

```
6616806: *Jan 31 17:12:41.220: cdapi_find_tsm:
Found a gtdmsg of length144:6616807: *Jan 31 17:12:41.220:
gtd msg = "IAM,PRN,isdn*,,
ATT5*,USI,rate,c,s,c,1USI,lay1,ulawTMR,00CPN,00
,,u,5900CPC,09FCI,,,,,,,y,UUS,3,3132333435

---> This is the UUI that will become the Correlation
ID12345GCI, 87c0c79d91dd11daa9c4000bfda207f2"
```

External Transfers in Unified ICME

Unified ICME Script Label for Outpulse Transfer

Labels in Unified ICME scripts for Unified CVP calls that require outpulse transfer mode must be prepended with the characters DTMF followed by *8 and some number of commas, where each comma represents a pause of 100 milliseconds. By configuring the target label with the form DTMFnnnnn (where nnnnn are the digits to outpulse), Unified CVP sends the digits out-of-band using H.245 signaling to the Ingress gateway for outpulsing.

To use the AT&T Transfer Connect feature to transfer the call to the number “4441234”, configure the label as DTMF*8,,,4441234.

Note Usually the PSTN switch expects a delay between the *8 and the phone number. Each comma represents 100ms by default. It can be changed with the SetTakebackDelay command in VBAdmin.

Note In outpulse transfer mode, Unified CVP sends whatever digits are in the label to the Gateway for outpulsing. It is the customer’s responsibility to confirm interoperability with the target switch.

Note In your Unified ICME script, when using outpulse transfers with SIP calls, digits can only be outpulsed on a call that has already been established. This means that it is necessary to transfer the call to the VXML gateway with a run external script node *before* you can send the DTMF*8 label. The Unified ICME script cannot send the DTMF*8 label back to Unified CVP for the first connect message in the call because the call has not been answered at this point. The Unified CVP Call Server uses SIP INFO messages to send the digits to the gateway for outpulsing.

Note When using outpulse transfers with SIP, you can also use the comma duration as the default interdigit pause duration.

For example, with the default 100 msec comma duration, a label such as "DTMF*8,,,8009785001" will have 300 msecs between the first 8 and the second 8. The interdigit pause will also be 100 msecs. The tone duration is also configurable and defaults to 100 msecs.

Note Outpulse transfer with SIP uses SIP INFO messages being sent to the TDM gateway, where the outpulsing of digits occurs. If the agent using the CTI desktop performs a blind transfer (single step transfer), and the scheduled script for the transfer DN returns a DTMF type label, the CUCM SIP Trunk can loop the CVP DTMF label through the bridged call using an UPDATE message. Unified CVP can get the label back and convert the digits to SIP INFO messages to forward to the ingress gateway. This only works on blind transfers, and is not supported on consult transfers.

Unified ICME Script Label for Two B-Channel Transfer

For Unified CVP calls that require Two B-Channel Transfer (TBCT) mode, add a label node to your Unified ICME script with the following syntax:

TBCT99#8005551212#

Replace "8005551212" with your transfer destination target; TBCT99 and the # sign are mandatory.

By configuring the target label in this form, Unified CVP sends the digits to the Ingress endpoint for Two B-Channel transfer.

Unified ICME Script Label for Hookflash Transfer

Labels in Unified ICME scripts for Unified CVP calls that require hookflash transfer mode must be prepended with the characters HF. By configuring the target label with the form HFnnnnn (where nnnnn are the digits to call), Unified CVP sends the digits to the Ingress endpoint for hookflash transfer.

If the switch requires a pause after the hookflash, insert commas between the HF and the transfer number. (Each comma represents 100 milliseconds.)

For example, to use the hookflash feature to transfer the call to the number “4441234” with a 500- millisecond pause after the hookflash, configure the Unified ICME label as "HF,,,,,4441234."

Takeback and Transfer in VoiceXML Scripts

Unified CVP provides the following takeback and transfer methods that you can invoke from a VoiceXML script:

- **Two B-Channel Transfer (TBCT)** - A call transfer standard for ISDN interfaces. This feature enables a Cisco voice gateway to request an NI-2 switch to directly connect two independent calls. The two calls can be served by the same PRI or by two different PRIs on the gateway.
- **Hookflash Relay** - A brief interruption in the loop current that the originating call entity (PBX or Public Switch Telephone Network switch) does not interpret as a call disconnect. Instead, once the PBX or Public Switch Telephone Network switch senses the hookflash, it puts the current call on hold and provides a secondary dial tone, which allows Unified CVP to transfer the caller to another destination.
- **SIP Refer** - VoiceXML applications can use a SIP REFER transfer instead of a blind or bridged transfer. This allows Unified CVP to remove itself from the call, thus freeing up licensed Unified CVP ports. Unified CVP cannot execute further call control or IVR operations after this kind of label has been executed.

Configure Two B-Channel Transfer

To configure Two B-Channel Transfer (TBCT) for use with Unified CVP from a VoiceXML script, perform the following procedure:

Procedure

Step 1 Configure the originating gateway for TBCT call transfer.

Step 2 Locate the following files on the VXML Server and copy them to flash memory on the gateway, using the tftp command:
en_holdmusic.wav

en_pleasewait.wav

survivability.tcl

CVPSelfService.tcl

CVPSelfServiceBootstrap.vxml

Step 3 Add the following lines to the gateway:

```
service takeback flash:survivability.tcl
param icm-tbct 1
```

Step 4 Configure the CVPSelfService application, as follows:

```
service [gateway application name] flash:CVPSelfService.tcl
param CVPBackupVXMLServer 12.34.567.890
param CVPSelfService-port 7000
param CVPSelfService-app [name of application on the VXML Server, exactly how it appears]
param CVPPrimaryVXMLServer 12.34.567.891
```

Note CVPSelfService is required. Backup server is optional. For the Tomcat Application Server, set the port to 7000.

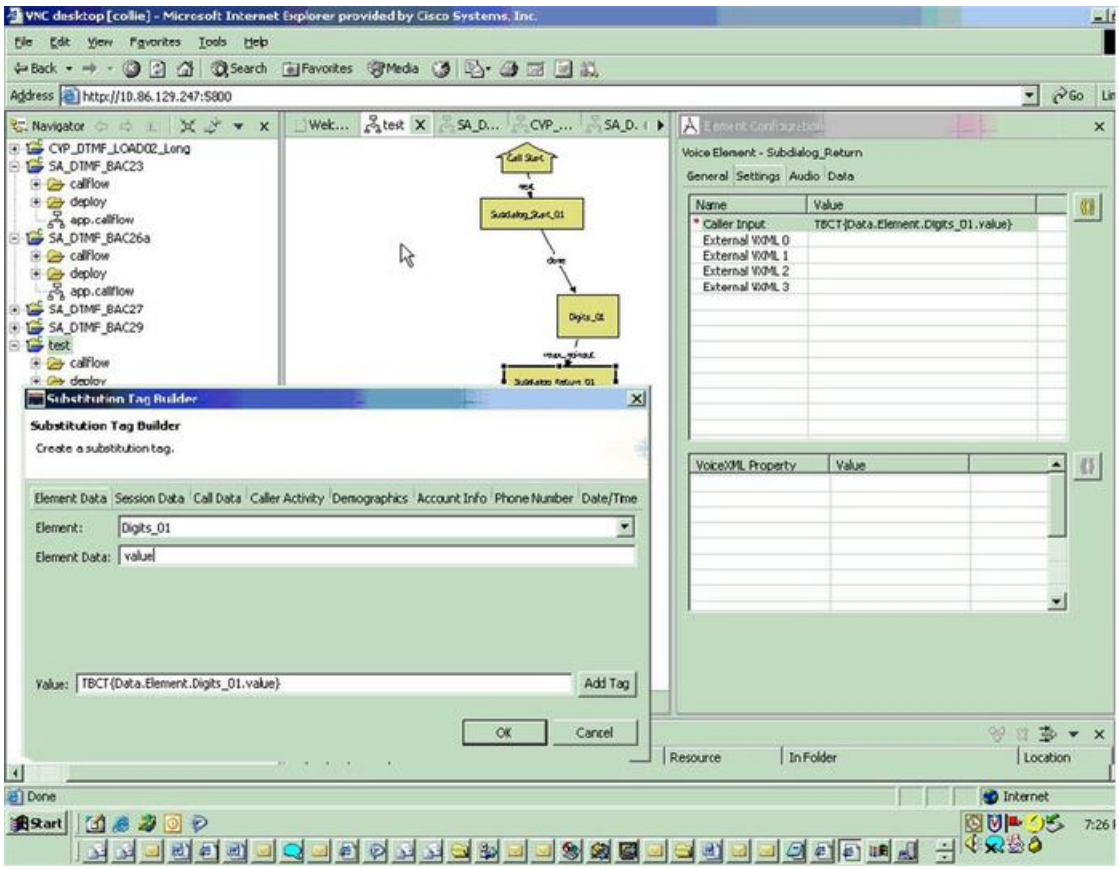
Step 5 From command line mode, enter:

```
call application voice load takeback
call application voice load CVPSelfService
```

Step 6 Specify the target destination for the TBCT transfer either by entering the number manually, or dynamically by using caller input.

a) **Manually**. In the SubdialogReturn node in the VXML Server application, next to Caller Input in the Settings Tab, enter TBCT<target_destination_number>, where *target_destination_number* is the target destination of the TBCT transfer. For example:
TBCT8005551212

b) **Dynamically**. The target destination is created dynamically using input entered by the caller during the call. Click the **Substitution** icon next to the Caller Input variable and select substitution values. For example:



Step 7 In the event that the TBCT fails (for example, bad number, bad gateway config, etc.), specify alternate transfer targets under the survivability service according to survivability rules as defined in the "Call Survivability" section.

Configure Hookflash Relay

To configure Hookflash Relay for use with Unified CVP from VoiceXML scripts, perform the following procedure:

Procedure

Step 1 Configure the originating gateway for Hookflash Relay call transfer.

Step 2 Locate the following files on the VXML Server and copy them to flash memory on the gateway:
en_error.wav

en_holdmusic.wav

en_pleasewait.wav

survivability.tcl

en_0.wav en_1.wav

en_2.wav en_3.wav

en_4.wav

en_5.wav

en_6.wav

en_7.wav

en_8.wav

en_9.wav

en_pound.wav

en_star.wav

Step 3 Add the following lines to the gateway:

```
service hookflash flash:survivability.tcl
```

Step 4 If you have not already done so, configure the CVPSelfService application as follows:

```
service [gateway application name] flash:CVPSelfService.tcl
param CVPBackupVXMLServer 12.34.567.890
param CVPSelfService-port 7000
param CVPSelfService-app [name of application on the VXML Server, exactly how it appears]
param CVPPrimaryVXMLServer 12.34.567.891
```

Note CVPSelfService is required. Backup server is optional. For the Tomcat Application Server, set the port to 7000.

Step 5 From command line mode, enter:

```
call application voice load hookflash
call application voice load CVPSelfService
```

Step 6 In the SubdialogReturn node in your VXML Server application, next to Caller Input in the Settings tab, enter "HF8005551212," replacing "8005551212" with the target destination of the hookflash transfer.
The label can also be defined dynamically using digits entered by the caller in conjunction with the VXML Server substitution tags. If the switch requires a pause after the hookflash, commas can be inserted between the HF and the transfer number. Each comma represents 100ms.

Configure SIP REFER

To configure SIP REFER for use with Unified CVP from a VoiceXML script, follow this procedure:

Procedure

Step 1 Configure the gateway according to either "Configure the Unified CVP VXML Server (Standalone) Call Flow Model (Without ICM Lookup)" or "Configure the Unified CVP VXML Server (Standalone) with ICM Lookup Call Flow Model," whichever is appropriate for the implementation.

Note The incoming dial-peer running the CVPSelfService application must be a voip dial-peer, not a pots dial-peer. Additionally, the IOS image that is used must contain the IPIPGW feature. SIP REFER will not work with the IPIPGW feature.

Step 2 Specify the target destination for the REFER transfer in the Call Studio application either by entering the number manually, or dynamically using caller input.

a) **Manually**. In the SubdialogReturn node in the VXML Server application, next to CallerInput in the Settings tab, enter **RF<target_destination_number>**, where target_destination_number is the target destination of the REFER transfer; for example, RF8005551212.

b) **Dynamically**. The target destination is created dynamically using input entered by the caller during the call. To do this, click the **Substitution** icon next to the Caller Input variable and select the substitution values.

Courtesy Callback

Courtesy Callback reduces the time callers have to physically wait on hold or in a queue. The feature enables your system to offer callers (who meet your criteria) the option to receive a courtesy callback by the system instead of waiting on the phone for an agent. The caller who has been queued by Unified CVP can hang up and subsequently be called back when an agent is close to becoming available (preemptive callback).

Preemptive callback does not change the time a customer must wait to be connected to an agent, but rather enables the caller to hang up and not be required to remain in queue listening to music. Callers who have remained in queue or have undergone the callback treatment appears the same to agents answering the call.

If the caller decides to be called back by the system, they leave their name and phone number. Their request remains in the system and when the system determines that an agent will be available soon (or is available), then the system places a call back to the caller. The caller answers the call and confirms that they are the original caller and the system connects the caller to the agent after a brief wait.

In the event that the caller cannot be reached after a configurable max number and frequency of retries, the callback is aborted and the database status is updated appropriately. You can run reports to determine if any manual callbacks are necessary based on your business rules.

Note that you cannot schedule a callback for a specific time.

Note There are a number of prerequisites and design considerations for using this feature. See the Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND) guide.

Note The Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND) guide contains a *typical use scenario* section that walks through a caller experience with the callback process, including alternate results for different caller decisions.

Callback Criteria

In your callback script, you can establish criteria for offering a caller a courtesy callback. Examples of callback criteria include:

- Number of minutes a customer is expected to be waiting *in queue* that exceeds a maximum number of minutes (based on your average call handling time per customer)

  Note The included example scripts use this method for determining callback eligibility.

- Assigned status of a customer (*gold* customers may be offered the opportunity to be called back instead of remaining on the line)
- The service a customer has requested (sales calls, or system upgrades, for example, may be established as callback criteria)

Modifiable Example Scripts and Sample Audio Files

The courtesy callback feature is implemented using Unified CCE scripts. Modifiable example scripts are provided. These scripts determine whether or not to offer the caller a callback, depending on the callback criteria (previously described). Sample audio files are also provided.

The example scripts and audio files are located on the CVP installation media in the `\CVP\Downloads and Samples\` folder.

The files provided are:

- `CourtesyCallback.ICMS`, the ICM script, in the `ICMDownloads` subfolder
- `CourtesyCallbackStudioScripts.zip`, a collection of Call Studio scripts, in the `helloStudio` Samples subfolder.

  The following example scripts are provided:

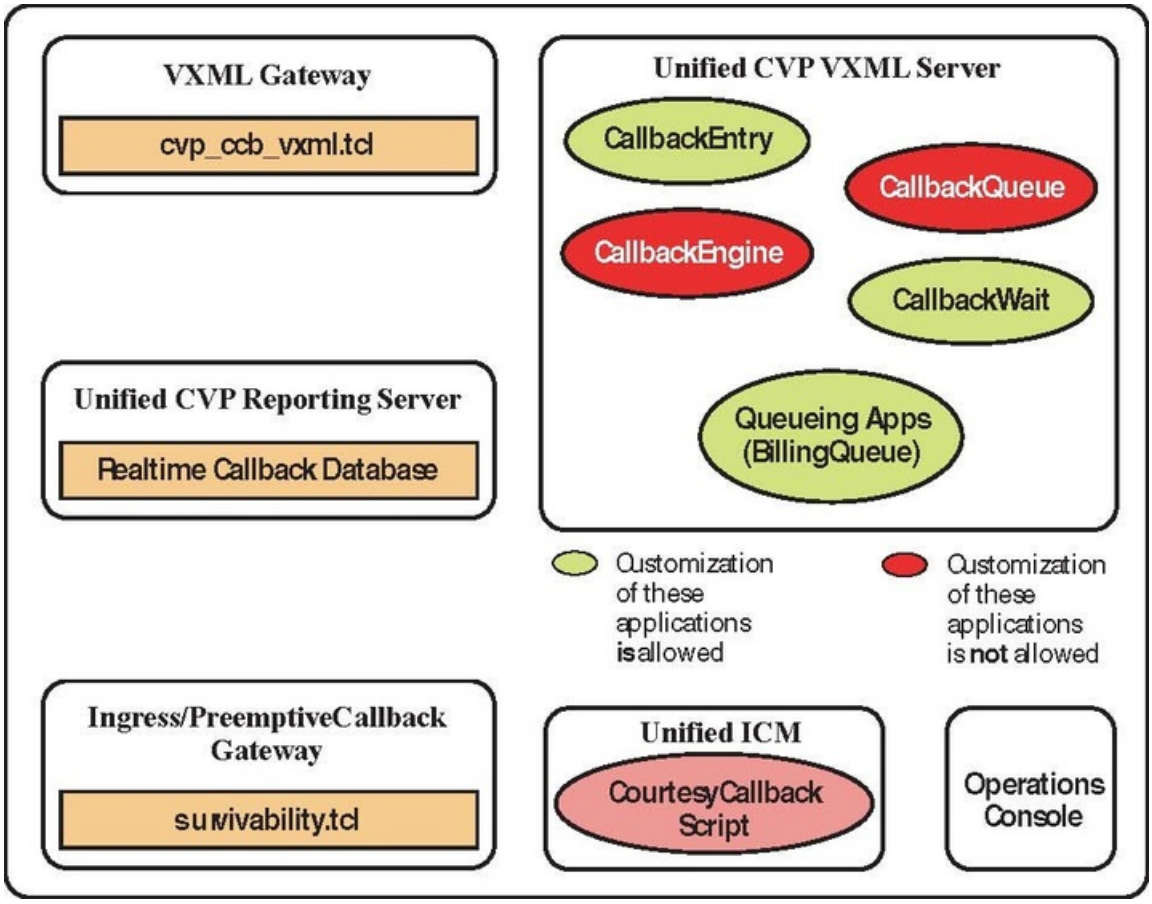
  - BillingQueue: Plays queue music to callers. Can be customized.
  - Callback Engine: Keeps the VoIP leg of the call alive when the caller elects to receive the callback (and hangs up) and when the caller actually receives the callback. *Do not* modify this script.
  - CallbackEntry: Initial IVR when caller enters the system and is presented with opportunity for a callback. Can be customized.

- **CallbackQueue:** Handles the keepalive mechanism for the call when callers are in queue and listening to the music played by BillingQueue. Do **not** modify this script.
- CallbackWait: Handles IVR portion of call when caller is called back. Can be customized.

- `CCBAudioFiles.zip`, in the `CCBDownloads` subfolder, contains sample audio files that accompany the sample studio scripts.

Courtesy Callback Configuration

The following diagram shows the components that must be configured for Courtesy Callback:



The Courtesy Callback feature must be configured on the following servers/gateways:

- Ingress Gateway (IOS configuration)
- VXML Gateway (IOS configuration)
- Reporting Server (through the Unified CVP Operations Console)
- Media Server (upload of Courtesy Callback media files)
- Unified CVP VXML Server (upload of Call Studio Scripts)

- Unified ICM (through the ICM script)

Configure the Ingress Gateway for Courtesy Callback

The ingress gateway where the call arrives is the gateway that processes the preemptive callback for the call, if the caller elects to receive a callback.

Note A sip-profile configuration is needed on ISR for the courtesy callback feature, only when deploying an IOS-XE version affected by CSCts00930. For more information on the defect, access the Bug Search Tool at https://sso.cisco.com/autho/forms/CDClogin.html.

For more information about sip-profile configuration, see *Design Guide for Cisco Unified Customer Voice Portal,* at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-implementation-design-guides-list.html.

Procedure

Step 1 Log in to the CVP OAMP Operations Console (from the CVP OAMP VM), using this syntax:
`https://<server_ip>:9443/oamp`.

Step 2 Copy survivability.tcl from the Operations Console to the flash memory of the gateway. Using the Operations Console, perform the following:

a) Select: **Bulk Administration** > **File Transfer** > **Scripts and Media**.
b) In Device Association, for **Select Device Type** select: **Gateway**.
c) Select all the Ingress gateways.
d) From the default gateway files, highlight: **survivability.tcl**.
e) Click **Transfer**.

Step 3 Log in to the ingress gateway.

Step 4 If survivability is not already configured, configure it as described in the "Call Survivability" section of the *Configuration Guide for Cisco Unified Customer Voice Portal* at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-installation-and-configuration-guides-list.html.

Step 5 To add services to the gateway, you must be in enabled-config application mode. Type these commands at the gateway console:

```
GW81#en
GW81#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
GW81(config)#application
GW81(config-app)#
```

Step 6 Add the following to the survivability service:

```
param ccb id:<host name or ip of this gateway>;loc:<location name>;trunks:<number of callback trunks>
```

Where the definitions of the preceding fields are:

- *id*: A unique identifier for this gateway and is logged to the database to show which gateway processed the original callback request.

- *loc*: An arbitrary location name specifying the location of this gateway.
- *trunks*: The number of DS0's reserved for callbacks on this gateway. Limit the number of T1/E1 trunks to enable the system to limit the resources allowed for callbacks.

The following example shows a basic configuration:

```
service cvp-survivability flash:survivability.tcl
param ccb id:10.86.132.177;loc:doclab;trunks:1
!
```

If you are updating the survivability service, or if this is the first time you created the survivability service, remember to load the application using the command:

```
call application voice load cvp-survivability
```

Step 7 Create the incoming dial peer, or verify that the survivability service is being used on your incoming dial peer. For example:

```
dial-peer voice 978555 pots
service cvp-survivability
incoming called-number 9785551234
direct-inward-dial
!
```

Note: We support both POTS and VoIP dial peers that point to a service provider.

Step 8 Create outgoing dial peers for the callbacks. These dial peers place the actual callback out to the PSTN. For example:

```
dial-peer voice 978554 pots
destination-pattern 978554....
no digit-strip
port 0/0/1:23
!
```

Step 9 Use the following configuration to ensure that SIP is set up to forward SIP INFO messaging:

```
voice service voip
signaling forward unconditional
```

Step 10 Save your changes.

Configure the VXML Gateway for Courtesy Callback

To configure the VXML gateway for Courtesy Callback:

Procedure

Step 1 Copy **cvp_ccb_vxml.tcl** from the CVP OAMP Operations Console to the flash memory of the gateway. Using the Operations Console:

a) Select: **Bulk Administration** > **File Transfer** > **Scripts and Media**.
b) In Device Association, for **Select Device Type** select: **Gateway**.
c) From the default gateway files, highlight: **cvp_ccb_vxml.tcl**.
d) Click **Transfer**.

Step 2 To add services to the gateway, you must be in enabled-config application mode. Type these commands at the gateway console:

```
GW81#en
GW81#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
GW81(config)#application
GW81(config-app)#
```

Step 3 Add the cvp_cc service to the configuration:

```
service cvp_cc flash:cvp_ccb_vxml.tcl
```

The service does not require any parameters.

Load the application with the command:

```
call application voice load cvp_cc
```

Note The media-inactivity detection feature must be turned off in the VXML Gateway to successfully call back the caller. With media-inactivity enabled on the VXML Gateway, the cvp_cc service will disconnect the waiting callback calls after 'ip rtcp report interval' * 1000-milliseconds interval. This configuration becomes important in a colocated Ingress/VXML setup where media inactivity timers are always enabled. In such scenarios, the 'ip rtcp report interval' must be increased to support the maximum allowable waiting for a callback call as defined by the solution requirements.

Step 4 On the VoIP dial-peer that defines the VRU leg from Unified ICM, verify that the codec can be used for recording. The following example shows that g711ulaw can be used for recording in Courtesy Callback:

```
dial-peer voice 123 voip
service bootstrap
incoming called-number 123T
dtmf-relay rtp-nte
codec g711ulaw
no vad
!
```

In other words, this example shows the g711ulaw codec set on the 123 voip dial-peer. The codec must be specified explicitly. A codec class cannot be used because recording will not work.

Step 5 Use the following configuration to ensure that SIP is set up to forward SIP INFO messaging:

```
voice service voip
signaling forward unconditional
```

Step 6 VXML 2.0 is required to play the beep to prompt the caller to record their name in the BillingQueue example script. Add the following text to the configuration so the VXML Server uses VXML 2.0:

```
vxml version 2.0
```

Note Whenever vxml version 2.0 is enabled on the gateway,vxml audioerror is off by default. When an audio file cannot be played, error.badfetch will *not* generate an audio error event. To have the gateway generate an error.badfetch event when a file cannot be played, enable vxml audioerror in your gateway configuration. The following example uses config terminal mode to add both commands:

```
config t
vxml version 2.0
vxml audioerror
exit
```

Step 7 Save your changes.

Configure the Reporting Server for Courtesy Callback

A Reporting Server is required for the Courtesy Callback feature. The Reporting Server must be installed and configured prior to completing the following task.

- **Installation:** If you have not **installed** a Reporting Server, refer to the *Installation and Upgrade Guide for Cisco Unified Customer Voice Portal* at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-installation-guides-list.html.
- **Configuration:** If you have not configured a Reporting Server in the Operations Console, refer to the Operations Console Online Help, **Managing Devices** > **Configuring a Reporting Server**.

Once you have added the Reporting Server, configure the Reporting Server for courtesy callback using the following procedure:

Procedure

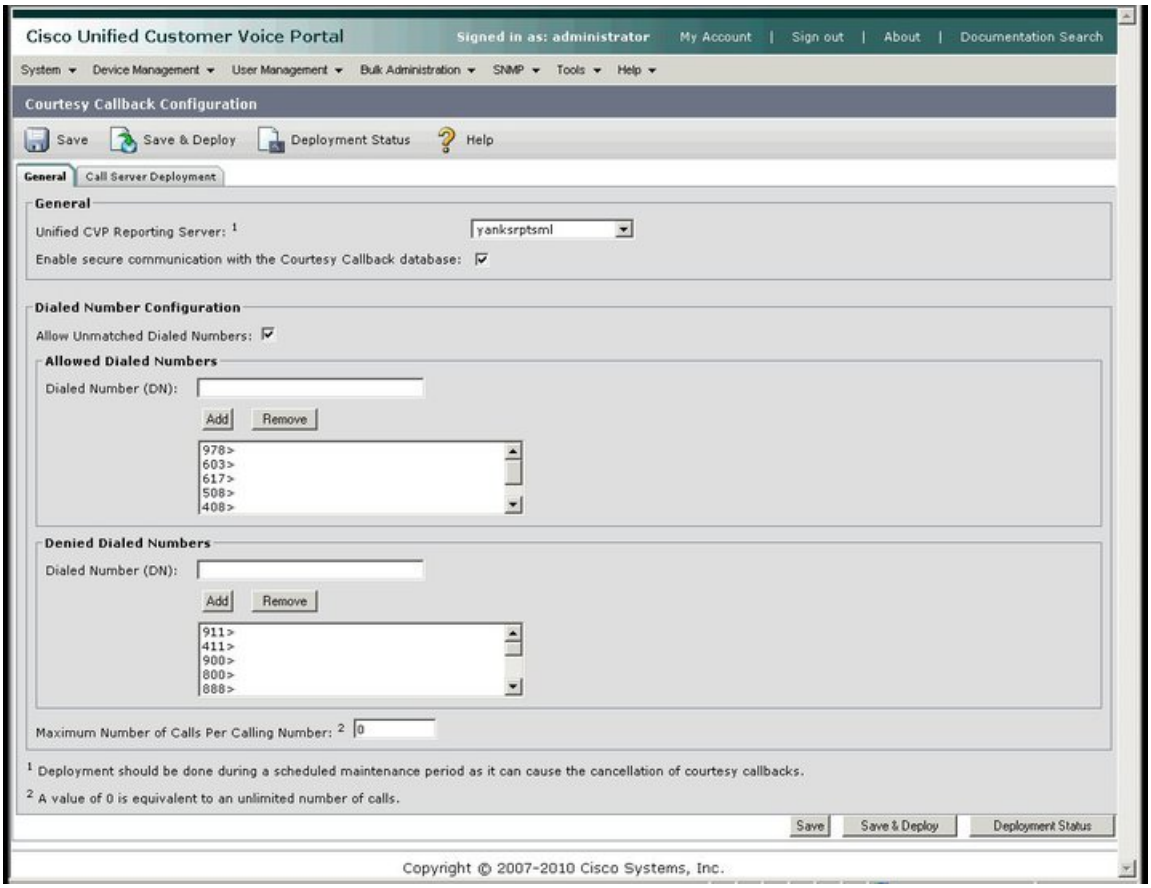
Step 1 Login to the CVP OAMP Operations Console, using this syntax: `https://<server_ip>:9443/oamp`.

Step 2 In the Operations Console, select **System** > **Courtesy Callback**. The **Courtesy Callback Configuration** page opens.
From this window, on the General tab you can:

- Select the Reporting Server for Courtesy Callback
- Enable secure communication with the Courtesy Callback database
- Configure allowed and disallowed dialed numbers

These operations are described in the following steps.

Figure 2: Courtesy CallBack Configuration



Step 3 On the Courtesy Callback Configuration page, select the **Unified CVP Reporting Server** drop-down list, and select the Reporting Server to use for storing Courtesy Callback data.

Note If you leave the selection blank, then there will be no Reporting Server associated with the Courtesy Callback deployment.

If you do not have a Reporting Server configured, refer to the notes at the beginning of this procedure to configure one.

Step 4 If desired, enable secure communication with the callback reporting database. Check **Enable secure communication with the Courtesy Callback database**.

Step 5 Configure allowed and denied dialed numbers. These are the numbers that the system *should and should not* call when it is making a courtesy callback to a caller. Also configure the Maximum Number of Calls Per Calling Number.

Use the following table to configure these fields:

Initially, there are **no** allowed dialed numbers for the Courtesy Callback feature. which means:

- **Allow Unmatched Dialed Numbers** is deselected.
- And, the **Allowed Dialed Numbers window** is empty.

This initial configuration is intentional; you must specifically enable the dialed numbers allowed for your deployment.

If you wish to allow **all** dialed numbers *except* those that are specifically listed in the Denied Dialed Numbers box, check **Allow Unmatched Dialed Numbers** .

Otherwise, add specific allowed number to the Allowed Dialed Numbers box. Refer to the Operations Console online help for detailson how to add specific allowed numbers, and for allowed valid dialed number shortcut patterns.

Note The Denied Dialed Numbers window is prepopulated if your local language is "en-us" (United States, English). Be sure to add any additional numbers you want to deny.

- Wildcarded DN patterns can contain "." and "X" in any position to match a single wildcard character.
- Any of the wildcard characters in the set ">*!T" will match multiple characters but can only be used for trailing values because they will always match all remaining characters in the string.
- The highest precedence of pattern matching is an exact match, followed by the most specific wildcard match.
- When the number of characters are matched equally by wildcarded patterns in both the Allowed Dialed Numbers and Denied Dialed Numbers lists, precedence is given to the one in the Denied Dialed Numbers list.

Step 6 Adjust the “Maximum Number of Calls per Calling Number” to the desired number. By default, this is set to 0 and the maximum limit is 1000.
This setting allows you to limit the number of calls, from the same calling number, that are eligible to receive a callback when there are outstanding callbacks already waiting for this number. If this field is set to a positive number (X), then the courtesy callback “Validate” element only allows X callbacks per calling number to go through the “preemptive” exit state at any time. If there are already X callbacks offered for a calling number, new calls go through the “none” exit state of the “Validate” element. In addition, if no calling number is available for a call, the call always goes through the “none” exit state of the “Validate” element."

Step 7 Click the Call Server Deployment tab and move the Call Server you want to use for courtesy callbacks from the **Available** box to the **Selected** box, as shown in the following screen shot :

Step 8 Click **Save & Deploy** to deploy the new Reporting Server configuration immediately.

If you click **Save**, the configuration is saved and becomes active (is deployed) the next time the Reporting Server restarts.

Note If you are updating the courtesy callback configuration (for example, changing to a different Reporting Server), perform deployment during a scheduled maintenance period. Otherwise, restarting the Reporting Server could cause the cancellation of currently scheduled courtesy callbacks.

Configure the Media Server for Courtesy Callback

Several Courtesy-Callback-specific media files are included with the sample scripts for Courtesy Callback. During the Unified CVP installation , these files are placed in the following directory:
`%CVP_HOME%\OPSConsoleServer\CCBDownloads\CCBAudioFiles.zip`

After CVP installation, the files are located on the CVP OAMP Server, in *%CVP_Home%*\OPSConsoleServer\. A typical value for *%CVP_Home%* is C:\Cisco\CVP.

CCBAudioFiles.zip has callback-specific application media files under `C:\inetpub\wwwroot\en-us\app` and media files for Say It Smart under `C:\inetpub\wwwroot\en-us\sys`.

The special audio files should be unzipped and copied to your media server.

Note If you selected the Media File installation option, during the Unified CVP install, the audio files were unzipped and copied to `C:\inetpub\wwwroot\en-us\app` on the installation server.

Note CCBAudioFiles.zip also contains media files for Say It Smart. During installation, these files are copied to `C:\inetpub\wwwroot\en-us\sys`. Copy these files to your media server, if you do not have them there already.

Note The sample scripts are set up to use the default location of `http://<server>:<port>/en-us/app` for the audio files. Later in this configuration process you will change the <server> and <port> parameters in the default location of the audio files in the example scripts to be your media server IP address and port number.

Configure Call Studio Scripts for Courtesy Callback

The Courtesy Callback feature is controlled by a combination of Call Studio scripts and ICM scripts. Refer to the *Design Guide for Cisco Unified Customer Voice Portal* (formerly the *Cisco Unified Customer Voice Portal Solution Reference Network Design*) at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-implementation-design-guides-list.html for a discussion of the script logic.

To configure the Call Studio scripts, perform the following procedure:

Note This example follows the BillingQueue example application.

Procedure

Step 1 Extract the example Call Studio Courtesy Callback scripts contained in CourtesyCallbackStudioScripts.zip to a folder of your choice on the computer running Call Studio.
You can access the .zip file from the following two locations:

- From the Unified CVP install media in \CVP\Downloads and Samples\Studio Samples\CourtesyCallbackStudioScripts
- From the Operations Console server in %CVP_HOME%\OPSConsoleServer\StudioDownloads.

Step 2 Each folder contains a Call Studio project having the same name as the folder. The five individual projects comprise the Courtesy Callback feature.
Do *not* modify the following scripts:

- CallbackEngine: Keeps the VoIP leg of the call alive when the caller elects to receive the callback (and hangs up) and when the caller actually receives the callback. Do **not** modify this script.
- CallbackQueue: Handles the keepalive mechanism for the call when callers are in queue and listening to the music played by BillingQueue.

Modify the following scripts to suit your business needs:

- BillingQueue: Determines the queue music played to callers.
- CallbackEntry: Modify the initial IVR treatment a caller receives when entering the system and is presented with an opportunity for a callback.
- CallbackWait: Modify the IVR treatment a caller receives when they respond to the callback.

Note Do not change the CCB application names.

Step 3 Start Call Studio by selecting **Start** > **Programs** > **Cisco** > **Cisco Unified Call Studio**.

Step 4 In Call Studio, select **File** > **Import**.

Step 5 In the Import dialog box, expand the Call Studio folder and select **Existing Call Studio Project Into Workspace** .

Step 6 Click **Next**.

Step 7 In the Import Call Studio Project From File System dialog, browse to the location where you extracted the call studio projects. For each of the folders that were unzipped, select the folder (for example BillingQueue) and click **Finish**.
The project is imported into Call Studio. Repeat this action for each of the five folders.

When you are finished importing the five folders, you should see five projects in the *Navigator* window in the upper left.

Step 8 Update the Default Audio Path URI field in Call Studio to contain the IP address and port value for your media server.
For each of the Call Studio projects previously unzipped, complete the following steps:

a) Select the project in the Navigator window of Call Studio.
b) Click **Project** > **Properties** > **Call Studio** > **Audio Settings**.
c) On the Audio Settings window, modify the Default Audio Path URI field by supplying your server IP address and port number for the <*Server*> and <*Port*> placeholders.
d) Click **Apply**, and then click **OK**.

Step 9 Billing Queue Project: If desired, change the music played to the caller while on hold.
You can also create multiple instances of this project if you want to have different hold music for different clients, for example, BillingQueue with music for people waiting for billing, and SalesQueue with music for people waiting for sales. You also need to point to the proper version (BillingQueue or SalesQueue) in the ICM script. In the ICM script, the parameter queueapp=BillingQueue would also have a counterpart, queueapp=SalesQueue.

The CallbackEntry Project (in the following step) contains a node called SetQueueDefaults. This node contains the value Keepalive Interval which must be *greater* than the length of the queue music you use. Refer to the Keepalive Interval in the next step for details.

Step 10 Callback Entry Project: If desired, in the CallbackEntry project, modify the caller interaction settings in the SetQueueDefaults node.
This step defines values for the default queue. You can insert multiple SetQueueDefaults elements here for each queue name, if it is necessary to customize configuration values for a particular queue. If you do not have a SetQueueDefaults element for a given queue, the configuration values in the default queue are used.

Note You can define a Callback_Set_Queue_Defaults node with **Queue Name** parameter set to default. Configuration defined in this default node will be picked whenever a queue type is encountered for which there are no explicitly defined values.

a) In the Call Studio Navigator panel, open the CallBackEntry project and double click **app.callflow** to display the application elements in the script window.
b) Open the Start of Call page of the script using the tab at the bottom of the script display window.
c) Select the SetQueueDefaults node.
d) In the **Element Configuration panel**, select the Setting tab and modify the following default settings as desired:
For the SetQueueDefaults element, the caller interaction values in the Start of Call and the Wants Callback elements, may be edited. For more information on the caller interaction values, see the Settings table in Chapter 10, Callback_Set_Queue_Defaults, in the *Element Specifications for Cisco Unified CVP VXML Server and Cisco Unified Call Studio* guide.

Step 11 Perform the following steps.

1. Set the path for the storage of recorded caller names.
2. Select app.callflow.
3. In the CallbackEntry project, on the Wants Callback page, highlight the Record Name node and click the **Settings** tab in the Element Configuration window of Call Studio.
4. In the Path setting, change the path to the location where you want to store the recorded names of the callers.

By default, Call Studio saves the path string in your VXML Server audio folder. If you are using the default path, you can create a new folder called recordings in the `%CVP_HOME%\VXMLServer\Tomcat\webapps\CVP\audio\` folder on the VXML Server. If you are using IIS as your media server, create a new folder called recordings under `C:\Inetpub\wwwroot\en-us\app` and set that as the path for recordings.

Step 12 Set the name of the Record name file.
From the CallbackEntry project on the Wants Callback page, highlight the **Add Callback to DB** node and select the **Settings** tab in the Element Configuration window of Call Studio.

Change the **Recorded name file** setting to match the location of the recording folder you created.

This setting references the URL of the recordings folder, whereas the Path setting references the file system path.

The AddCallback element setting in the CallbackEntry project is configured to do automatic recorded file deletions. If automatic recorded file deletion is not desired, then remove the value of the Recorded name path setting in the AddCallback element. This removal action assumes that you will be doing the deletion or management of the recorded file yourself.

Step 13 In the CallbackEntry project on the Callback_Set_Queue_Defaults node, be sure the keepalive value (in seconds) is greater than the length of the queue music being played. The default is 120 seconds.

Step 14 Save the **CallbackEntry** project.

Step 15 CallbackWait Project: Modifying values in the CallbackWait application.
In this application, you can change the IVR interaction that the caller receives at the time of the actual callback. The caller interaction elements in **CallbackWait** > **AskIfCallerReady (page)** may be modified. Save the project after you modify it. The WaitLoop retry count can also be modified from the default of six retries in the Check Retry element. This will allow a larger window of time to pass before the call is dropped from the application. It is used in a failure scenario when the CallbackServlet on the reporting server cannot be reached. For instance, in a reboot or a service restart, this allows more time for the reporting server to reload the entry from the database when it is initializing. If the reporting server is not online within the retry window, then the entry will not be called back.

Step 16 Validate each of the five projects associated with the Courtesy Callback feature by right-clicking each Courtesy Callback project in the Navigator window and selecting **Validate**.

Step 17 Validate each of the five projects associated with the Courtesy Callback feature and deploy them to your VXML Server.

a) Right-click each Courtesy Callback project in the Navigator window and select **Validate**.
b) Right click each of the projects and click **Deploy**, then click **Finish**.

Step 18 Using windows explorer, navigate to `%CVP_HOME%\VXMLServer\applications`.

Step 19 For each of the five Courtesy Callback applications, open the project's admin folder in `%CVP_Home%\VXMLServer\applications`, and double-click **deployApp.bat** to deploy the application to the VXML Server.

Step 20 Verify that all the applications are running by going into `%CVP_HOME%\VXMLServer\admin` and double-clicking **status.bat**. All five applications should be listed under Application Name, and the status for each one should be Running.

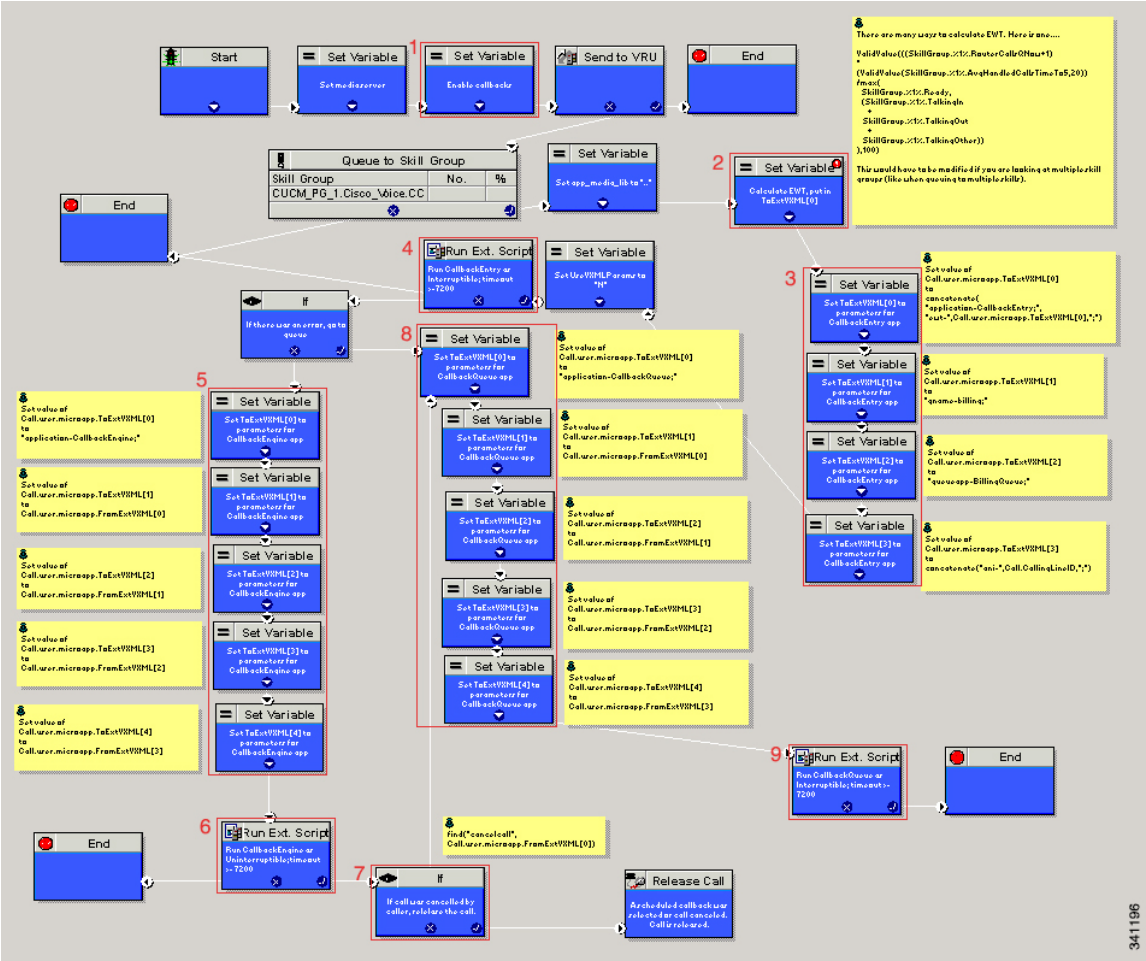
Note As an alternative to following steps 16-19 above, to deploy a VXML application to the VXML Server, you can also use the Bulk Administration VXML Applications feature. This way, you can deploy all the applications into a single archive, and then deploy them from OAMP in one click. This process is simpler and can save time. Bulk Administration deploys the application to the VXML Server, and then executes update-all-apps batch file, then executes deploy-all-new-apps batch file.

CCE Script for Courtesy Callback

The following discussion provides an overview of the scripts used for the courtesy callback feature. There are nine numbered blocks, or sets of blocks, identified in the following figure.

Note In the following example, the yellow comment blocks describe first the value being set and then the place where the value is being sent.

Figure 3: Setting Value for Courtesy Callback



The following bullets provide descriptions for the numbered blocks in the preceding graphic:

- Block 1: Enable callback or shut it off.
- Block 2: Compute average wait time. Once the caller is *in queue*, calculate the Estimated Wait Time (EWT) for that queue and place the value in ToExtVXML[0].

If there is poor statistical sampling because of sparse queues and the wait time cannot be calculated in the VXML Server, use the ICM-calculated estimated wait time.

One method of calculating EWT (the method used in this example) is:

```
ValidValue(((SkillGroup.%1%.RouterCallsQNow+1)
*
(ValidValue(SkillGroup.%1%.AvgHandledCallsTimeTo5,20))
/max(
SkillGroup.%1%.Ready,
(SkillGroup.%1%.TalkingIn
+
SkillGroup.%1%.TalkingOut
+
SkillGroup.%1%.TalkingOther))
),100)
```

Modify this method if you are looking at multiple skill groups (when queuing to multiple skills).

- Block 3: Set up parameters to be passed.
- Block 4: Run this block and prompt the caller. If the caller does not accept the offer for a callback, keep the caller in the queue and provide queue music.
- Block 5: Set up variables. Call flow returns to this block if the caller elects to receive a callback. Otherwise, the call remains queuing in the queuing application (BillingQueue in this example) on the VXML Server.
- Block 6: Run external to Callback engine to keep the call alive. If the agent becomes available and there is no caller, then agent can't interrupt (do not want an agent to pick up and have no one there).
- Block 7: Has the caller rejected the callback call? If no, then go to block 8.
- Block 8: Compute average wait time, as in block 2.
- Block 9: Set up variables.
- Block 10: Put caller briefly into queue (after caller accepts the actual callback call).

Overview of CCE Script Configuration for Courtesy Callback

The CCE script elements needed to enable Courtesy Callback are on the CVP Installation CD in *CVP\Downloads and Samples\ICM Downloads*. The script sample found there (CourtesyCallback) contains the necessary sample elements for the courtesy callback feature. However you must merge this script into your existing CCE scripts.

As a starting point and to run a simple test, import the script into the CCE script editor, validate it with the CCE script editor validation tool to locate nodes that need extra configuration (such as for Network VRU scripts and expanded call variables), and then modify the script according to your existing CCE environment.

The general process is as follows:

1. Locate each queue point in every CCE script. For example: Queue To Skill Group, Queue to Enterprise Skill Group, Queue to Scheduled Target or Queue to Agent.
2. Categorize each queue point according to the pool of resources that it is queuing for. Each unique pool of resources will ultimately require a queue in VXML Server if Courtesy Callback is going to be offered for that resource pool. For example, using the following example, QueueToSkill X and QueueToSkill Z are

queuing for the exact same resource pool (despite the different queuing order). Queue to Skill Y, however, is queuing to a different pool because it includes Skill Group D.

- QueueToSkillGroup X is queuing for Skill Group A, B, C in that order.
- QueueToSkillGroup Y is queuing for Skill Group A, C and D in that order.
- QueueToSkillGroup Z is queuing for Skill Group C, B, A in that order.

3. Assign a unique name to each unique resource pool. In the above example, we can use names ABC and ACD as example names.
4. For each resource pool, decide whether callbacks will be allowed in that resource pool. If yes, then every occurrence of that resource pool in all ICM scripts must be set up to use VXML Server for queuing. This is to ensure that the Courtesy Callback mechanism in the VXML Server gets a full, accurate picture of each resource pool's queue.
5. For any queue point where Courtesy Callback will be offered, modify all CCE scripts that contain this queue point according to the guidelines in the following CCE script examples.

Configure the CCE Script for Courtesy Callback

Many of the following configuration items relate to the numbered blocks in the diagram and provide understanding for CCE Script for Courtesy Callback (for more information, see CCE Script for Courtesy Callback, on page 348). Steps that refer to specific blocks are noted at the beginning of each step.

To configure CCE to use the sample Courtesy Callback CCE script, perform the following steps:

Procedure

Step 1 Copy the CCE example script, **CourtesyCallback.ICMS** to the CCE Admin Workstation.
The example CCE script is available in the following locations:

- On the CVP install media in `\CVP\Downloads and Samples\`.
- From the Operations Console in **%CVP_HOME%\OPSConsoleServer\ICMDownloads**

Step 2 Map the route and skill group to the route and skill group available for courtesy callback.

a) In Script Editor, select **File** > **Import Script...**.
b) In the script location dialog, select the **CourtesyCallback.ICMS** script and click **Open**.
c) In the **Import Script - Manual Object Mapping** window, map the route and skill group to the route and skill group available for courtesy callback (identified previously).

Step 3 Once the script is open in Script Editor, open the **Set media server** node and specify the URL for your VXML Server.
For example: **http://10.86.132.139:7000/CVP**

Step 4 **Refer to Block #1:** A new ECC variable is used when determining if a caller is in queue and can be offered a callback. Define the **user.CourtesyCallbackEnabled** ECC variable for courtesy callback.

a)
b) On the CCE Admin Workstation, in the ICM Configuration Manager, use the Expanded Call Variable List tool.

c) Create **user.CourtesyCallbackEnabled**.
d) Set **Maximum Length** to 1.
e) Check **Enabled**.
f) Check **Persistent**.

This step assumes that you have already created the standard ECC variables required for any Unified CVP installation. Refer to "Define Unified CVP ECC Variables, on page 86."

Step 5 **Block #2:** If you wish to use a different estimated wait time (EWT), modify the calculation in block #2. You must do this if you use a different method for calculating EWT or if you are queuing to multiple skill groups.

Step 6 **Block #3:** Set up the parameters to be passed to CallbackEntry (VXML application).

Note This step assumes that you have already configured the CCE and expanded call variables not related to Courtesy Callback.

Variable values specific to Courtesy callback include:

ToExtVXML[0] = concatenate("application=CallbackEntry",";ewt=",Call.user.microapp.ToExtVXML[0])

ToExtVXML[1] = "qname=billing";

ToExtVXML[2] = "queueapp=BillingQueue;"

ToExtVXML[3] = concatenate("ani=",Call.CallingLineID,";");

Definitions related to these variables are:

- CallbackEntry is the name of the VXML Server application that is executed.
- ewt is calculated in **Block #2**.
- qname is the name of the VXML Server queue into which the call is placed. There must be a unique qname for each unique resource pool queue.
- queueapp is the name of the VXML Server queuing application that is executed for this queue.
- ani is the caller's calling Line Identifier.

Step 7 Create Network VRU Scripts.

Using the ICM Configuration Manager, Network VRU Script List tool, create the following Network VRU Scripts:

Block #4: Interruptible Script (agent can interrupt the caller on hold):

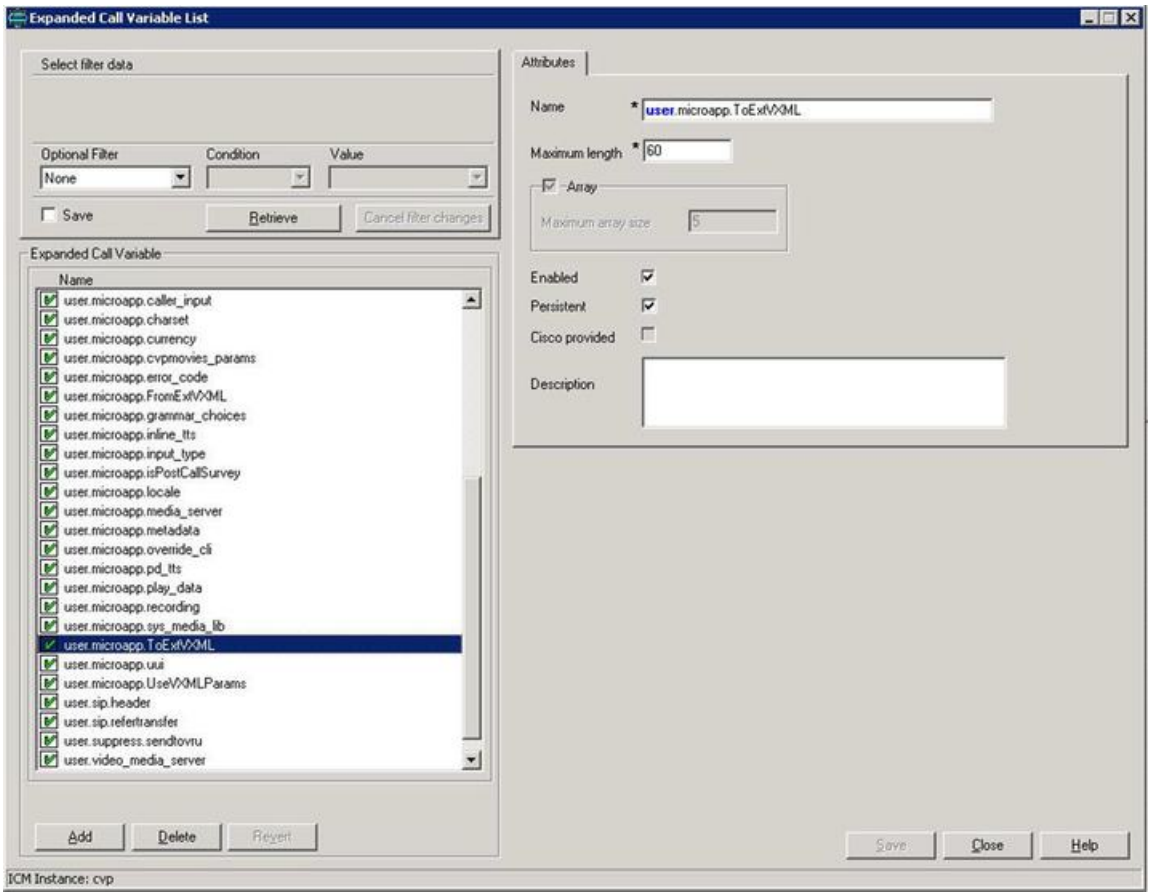
- Name: **VXML_Server_Interruptible**
- Network VRU: Select your Type 10 CVP VRU
- VRU Script Name: **GS,Server,V,interrupt**
- Timeout: **9000 seconds**
- Interruptible: **Checked**

Block #6: Noninteruptible Script (agent cannot interrupt because no caller is available):

- Name: **VXML_Server_Noninterruptible**
- Network VRU: Select your Type 10 CVP VRU
- VRU Script Name: **GS,Server,V,nointerrupt**
- Timeout: **9000 seconds** (must be greater than the maximum call life in Unified CVP)

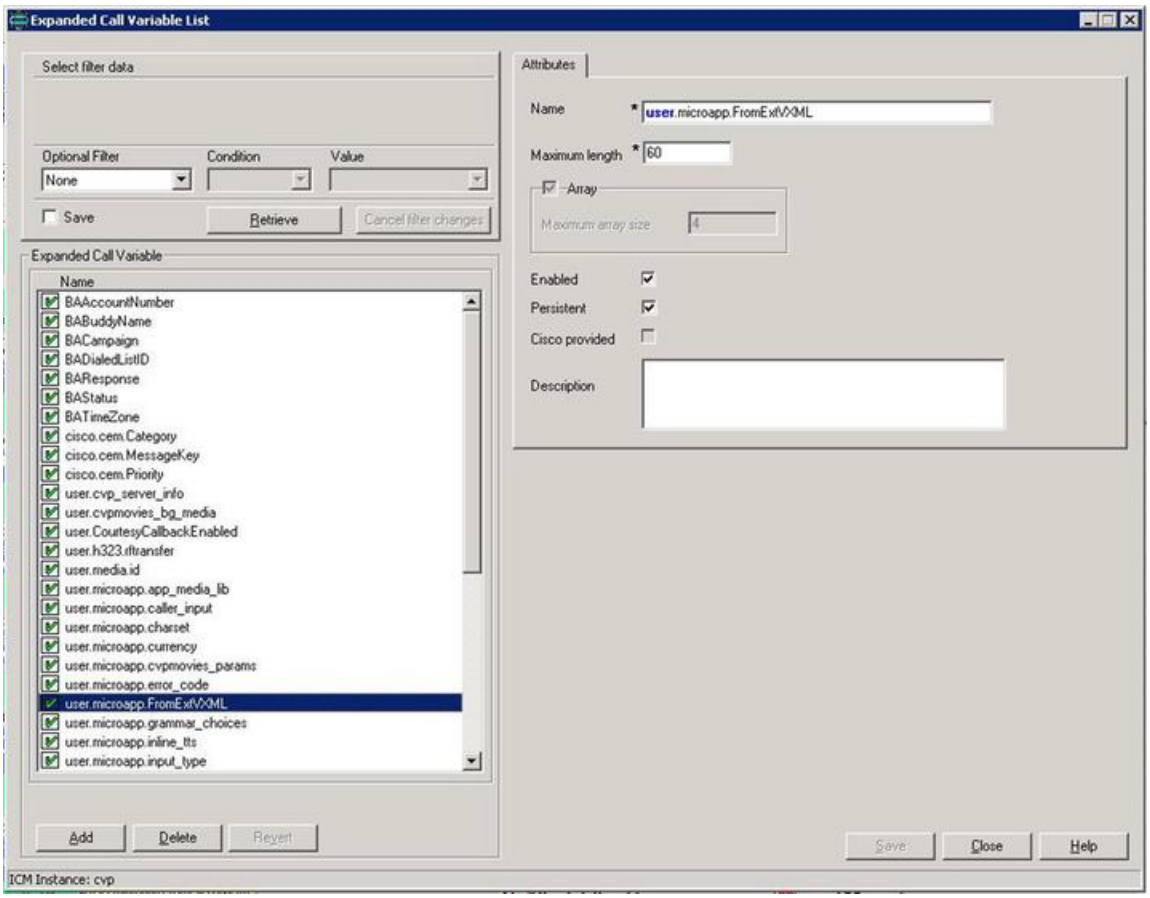
- Interruptible: **Not Checked**

Step 8 Verify that the **user.microapp.ToExtVXML ECC** variable is Enabled,Persistent, with a maximum field size of 60 (chars), set up as an array with a maximum array size of 5 elements.



Check **Array** and then a subfield for **Maximum array size** appears.

Step 9 Verify that the **user.microapp.FromExtVXML** variable is Enabled, Persistent, with a maximum field size of 60 (chars), set up as an array with a maximum array size of 4 elements.



Check **Array** and then a subfield for **Maximum array size** appears.

Step 10 Verify that you have at least one available route and skill group to map to the route and skill group in the example script.

Step 11 Save the script, then associate the call type and schedule the script.

Note For an example of scheduling the script refer to *Getting Started with Cisco Unified Customer Voice Portal* , the *Create a Call Type Manager Entity Routing Script and Call Schedule* topic.

Multicast Music on Hold (MMoH)

Multicasting may be used for Music On Hold with supplementary services on Unified CM as an alternative to the unicast MoH.

There are two ways to deploy with this feature:

- With the Unified CM multicasting the packets on the local LAN
- With the branch gateway(s) multicasting on their local LAN(s)

The latter is used when SRST (survivable remote site telephony) is configured on the gateway, and allows the deployment to utilize MOH locally and avoid MOH streaming over the WAN link.

For configuration details access the following links:

- Configuring Music on Hold
- Integrating Cisco CallManager and Cisco SRST to Use Cisco SRST as a Multicast MoH Resource

Note The SIP Trunk for Unified CVP (configured on Unified CM) must be associated with a Media Resource Group List (MRGL) that supports MMOH resources. Details on creating the MRGL are available from the links above.

Post Call Survey for SIP

A Post Call Survey takes place after normal call treatment. It is typically used to determine whether customers are satisfied with their call center experiences. This feature lets you configure a call flow that, after the agent disconnects from the caller, optionally sends the call to a Dialed Number configured for a Post Call Survey.

The Unified CCE script can enable and disable Post Call Survey on a per-call basis by testing for conditions and setting an expanded call variable that controls post call survey. For example, the script can invoke a prompt that asks callers whether they want to participate in a survey. Based on the caller's response, the script can set the expanded call variable that controls whether the call gets transferred to the Post Call Survey dialed number.

The Post Call Survey call works just like a regular call from the Unified CCE point of view. Scripts can be invoked and the customer can use the keypad on a touch tone phone and/or voice with ASR/TTS to respond to questions asked during the survey. During Post Call Survey, the call context information is retrieved from the original customer call.

Note For reporting purposes, the Post Call Survey call has the same CallGUID and call context as the original inbound call.

Note The call context for the post call survey includes all context up to the point where the call is transferred to the agent. Context that the agent creates after the transfer is not included in the post call survey context.

If you wish to use the Post Call Survey feature through Unified CVP, you must configure it on the Call Server. Also, you can configure the Unified ICM script to toggle the use of Post Call Survey off and on. The two configuration topics that follow, explain these methods.

Configure the Unified CVP Call Server for Post Call Survey

In the following procedure, you enter a dialed number pattern for the inbound call and a dialed number pattern for the post call survey. In both cases, the patterns can use alphanumeric characters and wildcard characters such as the exclamation point (!), asterisk (*), and single digit matches such as the letter X or period (.). The pattern can end with an optional greater than (>) wildcard character. The maximum length of the dialed number pattern is 24 characters.

Procedure

Step 1 Access the CVP Operations Console by typing **https://<OAMP_server_IP>:9443/oamp**.

Step 2 Log in to the Operations Console and select **Device Management** > **Unified CVP Call Server**.
The **Find, Add, Delete, Edit Call Servers** window opens.

Step 3 Click the Call Server for which you want to configure Post Call Survey.
The **Edit CVP Call Server Configuration** page displays.

Step 4 Click the **SIP** tab. Verify the **Override System Dialed Number Pattern Configuration** is not checked.

Step 5 Click **Save** and **Deploy** to deploy the Unified CVP Call Server device.

Step 6 Select **System** > **Dialed Number Pattern**.
The Dialed Number Pattern window opens.

Step 7 Click **Add New**.

Step 8 Enter a pattern in the **Dialed Number Pattern** field. This is the incoming Dialed Number for calls that you want to direct to a Post Call Survey. Make sure that dialed number patterns entered here are unique. (An incoming dialed number can not be associated with multiple survey numbers.)

Step 9 Check **Enable Post Call Survey for Incoming Calls**. This action enables post call surveys for all incoming calls with the specified dialed number pattern.
The **Survey Dialed Number Pattern** field appears.

Step 10 In the **Survey Dialed Number Pattern** field, enter a dialed number for the Post Call Survey. This is the dialed number to which the calls should be transferred to after the normal call flow completes.
Record the number you have entered. In the next task, you create this dialed number in CCE Administration and create a call type to associate with this dialed number.

Step 11 Click **Save** to save the Dialed Number Pattern.
You are returned to the **Dialed Number Pattern** page.

Step 12 Click **Deploy** to deploy the configuration to all Unified CVP Call Server devices.

Configure Unified CCE for Post Call Survey

You can enable and disable Post Call Survey within a CCE routing script by using the ECC variable **variableuser.microapp.isPostCallSurvey**. A value of n or y disables and enables the feature. (The value is case-insensitive.)

Configure the ECC variable to a value of n or y before either the label node or the Queue toSkillgroup node. This configuration sends the correct value to Unified CVP before the agent transfer. This ECC variable is not needed to initiate a Post Call Survey call, but you can use it to control the feature once Post Call Survey is configured using the Unified CVP Operations Console. As long as a DN is mapped in the Operations Console for Post Call Survey, the call is automatically transferred to the configured Post Call Survey DN.

Note

- The Post Call Survey DN is called if the Unified CVP has received at least one CONNECT message from CCE (either from the VRU leg or from the Agent leg). Use the END node in your CCE routing script if the Post Call Survey is not required for the calls disconnected from the IVR.
- If Router Requery is configured incorrectly and the Ring-No-Answer timeout expires, the caller is still transferred to the Post Call Survey DN. This can occur if a Queue node is used and Enable target requery is not checked.

Procedure

Step 1 In Unified CCE Administration, select **Manage** > **Expanded Call Variables**.

Step 2 Create a new ECC variable with **Name:**`user.microapp.isPostCallSurvey`.

Step 3 Set **Max Length:** to 1.

Step 4 Check the **Enabled** checkbox. Then click **Save**.
In your CCE routing scripts, remember that, at script start, the default behavior of Post Call Survey equals **enabled**, even if **user.microapp.isPostCallSurvey** has not yet been set in the script. You can turn **off** Post Call Survey in the script by setting **user.microapp.isPostCallSurvey** to *n*. You can later enable Post Call Survey in the same path of the script by setting this variable to *y*.

Step 5 Select **Manage** > **Call Types**.

Step 6 Add the call type for Post Call Survey, and click **Save**.

Step 7 Select **Manage** > **Dialed Numbers**.

Step 8 Create a dialed number for each of the Post Call Survey Dialed Number Patterns created in Unified CVP. Select the following for each dialed number:

- **Routing Type:** External Voice
- **Call Type:** Post Call Survey call type you created.

Step 9 Click **Save**.

Step 10 Restart the active generic PG (side A or B) to register the new ECC variable.
If the ECC variable already existed, you can skip this step.

Note The **user.microapp.isPostCallSurvey** setting takes effect on Unified CVP only when it receives a connect or temporary connect message. Therefore, if you do not want the survey to run, without first reaching an agent (such as 'after hours of treatment'), you must set the isPostCallSurvey to *n* before the initial 'Run script request'.

CHAPTER 12

Configure High Availability for Unified CVP

This chapter provides information about how to accommodate load balancing and redundancy in Unified CVP deployments.

- Server Groups, page 357
- Redundancy and Failover for Unified CVP, page 360
- Configure a Speech Server, page 362
- Application Control Engine for Load Balancing in Unified CVP, page 363

Server Groups

A Server group is a dynamic routing feature that enables the originating endpoint to have knowledge of the status of the destination address before attempting to send the SIP INVITE. Whether the destination is unreachable over the network, or is out of service at the application layer, the originating SIP user agent can have fore-knowledge of the status through a heartbeat mechanism.

The Server Groups add a heartbeat mechanism with endpoints for SIP. This feature enables faster failover on call control by eliminating delays due to failed endpoints.

The following list is a summary of important configuration items:

- Server Groups are not automatically added to your configuration. You must explicitly configure Server Groups for their deployment and turn on this feature.
- If you have already configured the **local SRV** feature and therefore created a srv.xml file, you must run the **srvimport.bat** command before you configure Server Groups using the Operations Console. Otherwise, your existing definitions will be overwritten. This process is explained in the configuration details that follow.
- You define Server Groups using the Operations Console. You must always configure at least one Call Server first, because you will not be able to save the Server Groups configuration without assigning it to at least one Call Server.

Configure Server Groups

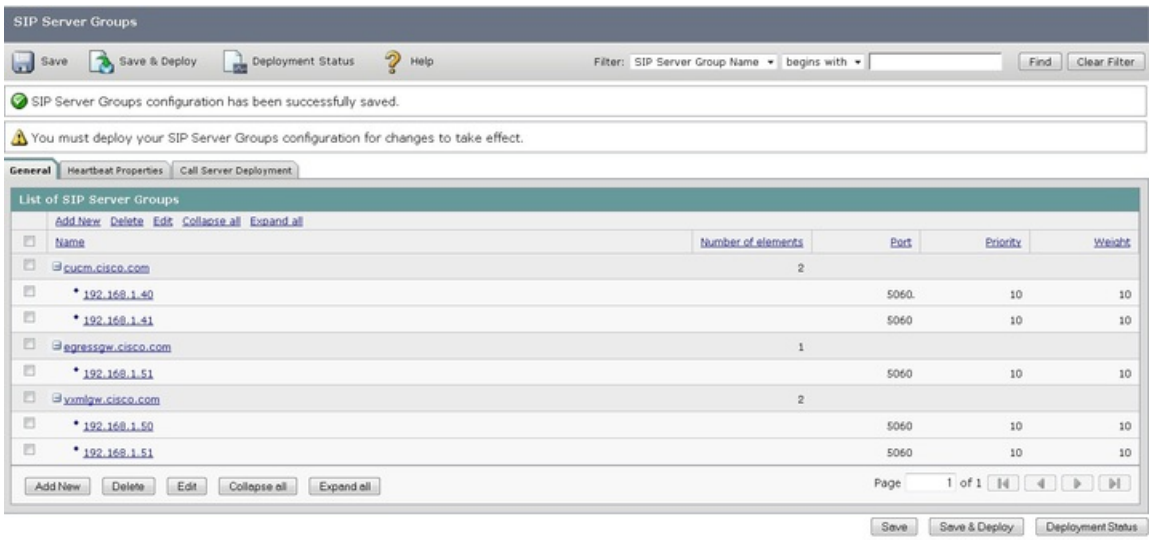
Complete the following steps to configure Server Groups:

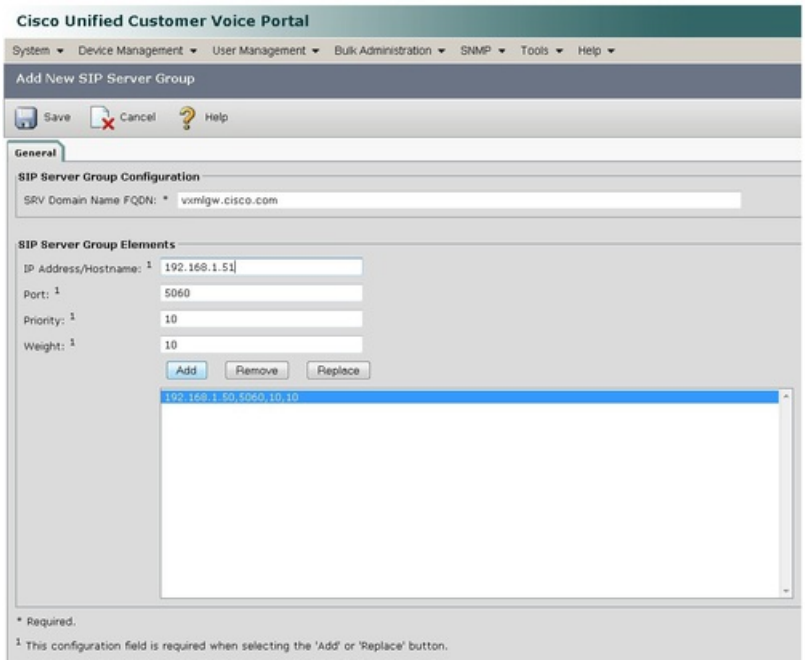
1. If you have previously created an srv.xml file, after you upgrade your Unified CVP installation, run the batch file **srvimport.bat** to transfer your prior configuration to the new Server Groups feature.

   The srvimport.bat file is located in the **CVP bin directory**. This batch file takes your **srv.xml** file as an argument. Copy this file from your Call Server configuration directory. Running srvimport.bat brings this configuration data into the Operations Console.

Note You must **stop** the OAMP (Operations Console) service before you run the .bat file.

2. If you have not defined a Call Server using the Operations Console, refer to *Configuring a Call Server* in the Operations Console online help.
3. Use the following window in the Operations Console to configure Server Groups. Access this window by selecting **System** > **SIP Server Groups**.





4 A Server Group consists of one or more destination addresses (endpoints) and is identified by a Server Group domain name. This domain name is also known as the SRV cluster name, or Fully Qualified Domain Name (FQDN). Define the FQDN and add it to the list. Refer to Configuring Server Groups in the Operations Console online help.

5 Refer to SIP Server Group Configuration Settings in the Operation Console online help to complete the Server Group configuration.

6 Click the **Call Server Deployment** tab and select the Call Server(s) that you want to associate with the Server Group(s). Then click **Save & Deploy** .

Note When you associate the Call Server(s) configuration, all the SIP Server Group configurations are applied to the Call Server(s), but individual deployment of SIP Server Group is not supported.

Server Groups Diagnostics

The CVP log file has traces which show endpoint status events. From the diagnostic servlet, click on the link for *dump SIP state machine* to display information as shown in the following example:

SIP Stack Local SRV Configuration

SRV key = proxy.cisco.com

record = host:10.10.10.10 port:5060 priority:10 weight:10 transport:1 enabled:true

record = host:10.86.129.239 port:5060 priority:20 weight:10 transport:1 enabled:true

Server Group Element Status (duplicates not shown)			inUnreachableTableUDP	inUnreachableTableTCP
proxy.cisco.com	10.10.10.10	5060	true	true
proxy.cisco.com	10.86.129.239	5060	false	false

Redundancy and Failover for Unified CVP

This section describes redundancy and failover mechanisms for ASR, TTS, Media, and VXML Servers in the Unified CVP solution.

Redundancy for VXML Server Applications

VXML Server applications rely on the gateway's configured default for ASR and TTS servers, which allow only a single host name or IP address to be specified for each. This differs from the Unified CVP micro-applications based applications, which support automatic retries to specifically named backup ASR and TTS servers.

Use the following configuration on the gateway if you are using Nuance or Scansoft ASR/TTS servers:

```
ip host asr-en-us 10.10.10.1
ip host tts-en-us 10.10.10.2
```

Use the following configuration on the gateway if you are using Nuance or Scansoft ASR/TTS servers:

```
mrcp client rtpsetup enable
ivr asr-server rtsp://asr-en-us/recognizer
ivr tts-server rtsp://tts-en-us/synthesizer
http client cache memory pool 15000
http client cache memory file 500
ivr prompt memory 15000
ivr prompt streamed none
mrcp client timeout connect 5
mrcp client timeout message 5
```

```
rtsp client timeout connect 10
rtsp client timeout message 10
vxml tree memory 500
http client connection idle timeout 10
no http client connection persistent
```

The URL configured by the above ivr commands defines the gateway's default target for ASR and TTS services, and is in effect for all calls handled by that gateway. You can override it dynamically in your VXML Server application by populating the Cisco-proprietary VoiceXML properties **com.cisco.asr-server** or **com.cisco.tts-server**.

Note For ASR/TTS failover to function when using Custom VXML Applications, you require either an Application Control Engine (ACE) or any other supported load balancer.

Redundancy for Micro-App-Based Applications

When ACE is used for ASR or TTS servers, the IVR Service plays a significant role in implementing a failover mechanism for Media Servers, ASR/TTS Servers and micro-app-based applications. Up to two of each such servers are supported, and the IVR Service orchestrates retries and failover between them.

Note This redundancy mechanism is only available for Unified CVP micro-applications.

Note For information about setting up the IVR Service to accommodate failover, see the *Operations Console online help*.

IVR Service Failover Mechanism

The IVR Service failover mechanism applies to:

- Connections between the IVR Service and the IOS Voice Browser, only.
- All communication between the IOS Voice Browser and an ASR Server, TTS Server, or Media Server.
- Media Server, when the ICM Script ECC variable, **user.microapp.media_server**, is set to mediaserver. When **user.microapp.media_server** is set to mediaserver, the IVR Service uses the IP Address defined on the gateway as:
  - ip host mediaserver 10.86.129.50
  - ip host mediaserver-backup 10.86.129.51

Note If **user.microapp.media_server** is configured as the hard-coded IP Address of the media server, then the IVR Service will not perform any failover for the media server.

If the IVR Service receives a Call Result error code value of **9** (MEDIA_FILE_NOT_FOUND), **33** (GENERAL_ASR_TTS), **31** (MEDIA_RESOURCE_ASR) or **32** (MEDIA_RESOURCE_TTS), it does the following:

- When attempting to connect to a *Media Server* , the IVR Service:
  - Resends the request the number of times defined in the IVR Service Configuration's **Media Server Retry Attempts** field.
  - If the connection is not successful after the specified number of attempts, and the IVR Service Configuration's **Use Backup Media Servers** field is set to **Yes** (the default), the IVR Service makes the same number of attempts to retrieve the media from a backup media server before failing and generating an error.

    Note The backup media server is defined on the gateway as <mediaserver>-backup.

  - Passes the error in a Call State Event to the ICM Service, which then passes it to Unified ICME.
- When attempting to connect to an *ASR/TTS Server*, the IVR Service:
  - Resends the request the number of times defined in the IVR Service Configuration's **ASR/TTS Server Retry Attempts** field.
  - If the connection is not successful after the specified number of attempts, and the IVR Service Configuration's **Use Backup ASR/TTS Servers** field is set to **Yes** (the default), the IVR Service makes the same number of attempts to connect to a backup ASR/TTS server before failing and generating an error.

    Note The backup ASR and TTS servers are defined on the gateway as asr-<locale>-backup and tts-<locale>-backup.

  - Passes the error in a Call State Event to the ICM Service, which then passes it to Unified ICME.

Each new call attempts to connect to the primary server. If failover occurs, the backup server is used for the duration of the call; the next new call will attempt to connect to the primary server.

Note This failover mechanism differs from that used in prior releases of Unified CVP software. Legacy releases used a *sticky* connection. In a sticky connection, if failover occurs to a backup server, subsequent new calls automatically connect to the backup server, rather than attempt to connect with the primary server.

Configure a Speech Server

A speech server provides speech recognition and synthesis services. You can add a pre-configured speech server to the Operations Console. Once added to the Operations Console, you can add a speech server to one or more device pools.

A speech server provides speech recognition services and text-to-speech services for a VoiceXML Gateway. For capacity and redundancy reasons, a ACE is usually used to mediate between a farm of such servers. If you do not use ACE, Unified CVP can support a maximum of two speech servers.

Caution The Operations Console can only manage speech servers installed on *Windows*, not on Linux. If the speech server is installed on Linux, the server cannot be managed.

The following list indicates some of the tasks you can perform on a speech server. See the *Operations Console online help* for details on these tasks.

- Add a Speech Server

  Note Install the Remote Operations component in the Speech Server before you add the Speech Server to the Operations console.

- Examine and change a Speech Server Configuration settings
- Editing a Speech Server Configuration
- Applying a License to a Speech Server
- Deleting a Speech Server
- Adding and Removing Speech Servers from a Device Pool
- Finding a Speech Server

Application Control Engine for Load Balancing in Unified CVP

For configuration details, refer to the Cisco ACE 4700 Series Appliance Server Load-Balancing Configuration Guide.

ACE provides load-balancing services for HTTP, MRCP and RTSP traffic, but not for call control signaling SIP messages. As a load-balancing device, ACE determines which server in a set of load-balanced servers, should receive the client request for service. Load balancing helps fulfill the client request without overloading either the server or the server farm as a whole. Also, by monitoring the state of each server and transferring a server's load to a working server during a server failure, ACE provides high availability support.

In this application of ACE, the engine is used primarily to direct initial session requests for a particular type of service. There are four types of services:

- http prompts
- ASR/TTS
- Unified CVP Call Server
- Unified CVP VXML Server

The following general approach applies to configuring each Unified CVP component type for use with ACE.

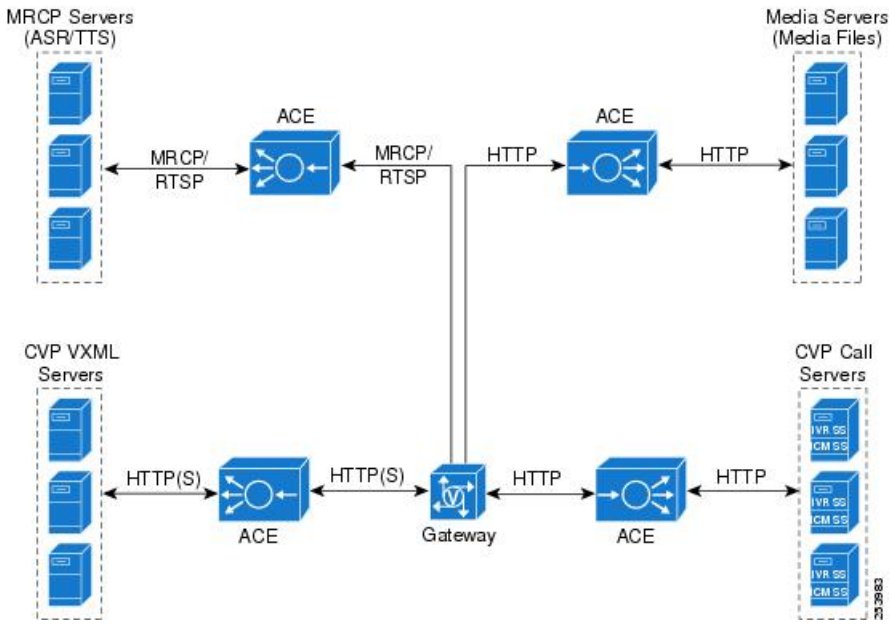
- **Real Server**s - One ACE real server is configured for each group of Unified CVP components (Call Servers, VXML Servers, etc.) that need ACE Load balancing. For general step-by-step guidelines for

configuring Real Servers, refer to the Cisco ACE 4700 Series Appliance Server Load-Balancing Configuration Guide

- **Server Farms** - Typically, in data centers, servers are organized into related groups called server farms. Servers within server farms often contain identical content (referred to as mirrored content) so that if one server becomes inoperative, another server can take its place immediately. After you create and name a server farm, you can add existing real servers to it and configure other server-farm parameters, such as the load-balancing predictor, server weight, backup server, health probe, and so on. For general step-by-step guidelines for configuring server farms, refer to the Cisco ACE 4700 Series Appliance Server Load-Balancing Configuration Guide
- **Health Monitoring** - You can instruct the ACE servers to check the health of servers and server farms by configuring health probes (sometimes referred to as keepalives). After you create a probe, you assign it to a real server or a server farm. A probe can be one of many types, including TCP, ICMP, Telnet, or HTTP. The ACE server sends out probes periodically to determine the status of a load-balanced server, verifies the server response, and checks for other network problems that may prevent a client from reaching a server. For general step-by-step guidelines for configuring probes, refer to the Cisco ACE 4700 Series Appliance Server Load-Balancing Configuration Guide
- **Class-Map and Policy Map** - The ACE server uses several configuration elements to filter traffic and then to perform various actions on that traffic before making the load-balancing decision. These filtering elements and subsequent actions form the basis of a traffic policy for server load balancing. For general step-by-step guidelines for configuring traffic policies, refer to the Cisco ACE 4700 Series Appliance Server Load-Balancing Configuration Guide

Specific component-type configuration is covered in the following sections.

In this section, you will configure the probes and other configuration needed for the ACE server to ensure that each server in each server farm is operating properly, so that the ACE server can load balance between all the servers of each type that are usable at any given moment.



General Probes

In your ACE unit's configuration, create an ICMP probe to check for server connectivity. In the subtopics that follow you associate this probe with each of your real servers.

```
probe icmp PROBE_SERVICE_ICMP
interval 5
receive 3
faildetect 1
passdetect interval 5
passdetect count 1
```

Unified CVP Media Servers

Media Servers are standard web servers that are responsible for serving Unified CVP prompt files to the voice gateway.

Create an HTTP Probe

The probe below is used to determine whether the Media Server is operating properly. A simple HTTP request is sent to the Media Server and the probe does a check for HTTP return code 200.

The Media Server probe sends an HTTP request to **/index.html**. The request is sent to the default HTTP port (80) and the IP address of the real server associated with the probe.

In the probe below, the following parameters are set. Set the actual values according to your own requirements. Refer to the Cisco ACE 4700 Series Appliance Server Load-Balancing Configuration Guide.

To create the HTTP probe for the media servers, place the following code in the configuration for the ACE server.

```
probe http PROBE_HTTP
interval 5
receive 3
faildetect 1
passdetect interval 5
passdetect count 1
request method get url /index.html
expect status 200 200
open 1
```

Configure the Physical Servers

Create a real server for every physical media server you would like to load balance. Associate the ICMP probe with each server by creating a section, as shown in the following example, for each media server in the server farm.

```
rserver host mediaServer1
ip address 10.1.1.1
probe PROBE_SERVICE_ICMP
inservice
rserver host mediaServer2
ip address 10.1.1.2
probe PROBE_SERVICE_ICMP
inservice
```

Group Your Physical (Media) Servers

In the ACE configuration file, create a server farm and associate servers with this farm. The following example applies the HTTP Probe to the server farm and the ACE server probes each media server in the server farm. However, you can also associate this probe with the physical server.

Note By specifying the port, only connections on this port will be accepted by this server farm.

```
serverfarm host media_server_FARM
description Media Server Farm
probe PROBE_HTTP
rserver mediaServer1 80
inservice
rserver mediaServer2 80
inservice
```

Class Map Configuration

The configuration below defines a Layer 3 and a Layer 7 class-map.

- The Layer 3 class-map is used to define a Virtual IP and the allowed traffic port. This class map gets applied to the Layer 3/4 policy-map. Traffic sent to the virtual IP is directed by the ACE server to real media servers based on the load balancing policy.
- The Layer 7 class-map is used to filter traffic based on the URL pattern specified. This class-map is associated with a Layer 7 policy-map, which contains information about which servers to load balance.

When traffic entering the ACE server matches the class-map L3_Media_Server_VIP, the ACE server applies the actions specified in Media_Server_L7SLB, which is defined below.

```
class-map match-all L3_Media_Server_VIP
2 match virtual-address 10.1.1.3 tcp eq www

class-map type http loadbalance match-all L7_HTTP_CLASS
2 match http url .*
```

Policy Map Configuration

Note In the code below, the layer 7 class map gets associated with the layer 7 policy map.

```
policy-map type loadbalance first-match Media_Server_L7SLB
class L7_HTTP_CLASS
serverfarm media_server_FARM

policy-map multi-match POLICY
class L3_Media_Server_VIP
loadbalance vip inservice
loadbalance policy Media_Server_L7SLB
loadbalance vip icmp-reply active
```

ASR/TTS Servers

Probe

The probe below is used to determine whether the MRCP ASR/ TTS Server is up. The ACE server makes a connection to the MRCP port to validate that the ASR/TTS server is running. In the configuration below, a TCP probe is used. The probe waits for the configured 3 seconds to receive information from the server. The ASR/TTS service is considered down if the ACE server is unable to connect to port 554 for MRCP traffic.

In the probe below, the parameters are set. Set the actual values according to your own requirements. Refer to the Cisco ACE 4700 Series Appliance Server Load-Balancing Configuration Guide.

The following configuration example is part of the ACE server's configuration.

```
probe tcp PROBE_ASR_TTS
port 554
interval 5
receive 3
faildetect 1
passdetect interval 5
passdetect count 1
open 1
```

Real Server and Server Farm Configuration

The following code defines your physical servers and associates them with the ICMP probe.

```
rserver host asrtts1
ip address 10.1.1.12
probe PROBE_SERVICE_ICMP
inservice
rserver host asrtts2
ip address 10.1.1.13
probe PROBE_SERVICE_ICMP
inservice
```

The following code defines your server farms and associates them with the PROBE_ASR_TTS probe. The servers in the server farm only accept connections on port 554.

```
serverfarm host ASR
description ASR Farm
probe PROBE_ASR_TTS
rserver asrtts1 554
inservice
rserver asrtts2 554
inservice
serverfarm host TTS
description TTS Farm
probe PROBE_ASR_TTS
rserver asrtts1 554
inservice
rserver asrtts2 554
inservice
```

Class-map Configuration

Create a class-map that accepts connections only on port 554. (By default, rtsp maps to port 554.)

```
class-map match-all ASR_CLASS_L3
2 match virtual-address 10.1.1.14 tcp eq rtsp


class-map match-all TTS_CLASS_L3
2 match virtual-address 10.1.1.18 tcp eq rtsp
```

Policy-map Configuration

```
policy-map type loadbalance first-match ASR_POLICY_L7
class class-default
serverfarm ASR

policy-map type loadbalance first-match TTS_POLICY_L7
class class-default
serverfarm TTS
policy-map multi-match POLICY
class ASR_CLASS_L3
loadbalance vip inservice
```

```
loadbalance policy ASR_POLICY_L7
loadbalance vip icmp-reply active
class TTS_CLASS_L3
loadbalance vip inservice
loadbalance policy TTS_POLICY_L7
loadbalance vip icmp-reply active
inspect rtsp
```

ASR and TTS Server Location Setup

There are two ways to specify an external media server for TTS and ASR operations:

- Specify ASR TTS Server Location Globally on the Gateway, on page 368
- Specify ASR and TTS Server Location with Individual VoiceXML Document, on page 369

Specify ASR TTS Server Location Globally on the Gateway

Media server sessions are created for each call to IVR applications, regardless of whether an application needs to communicate with the media server. Follow these steps to specify an ASR and TTS server location globally on the gateway.

Procedure

Step 1 Define the Hostname to IP Address mapping for the ASR and TTS servers.

```
ip host asr-en-us 10.78.26.31
ip host tts-en-us 10.78.26.31
```

Step 2 Define the Voice class URI that matches the SIP URI of the ASR Server in the dial-peer.

```
voice class uri TTS sip
pattern tts@10.78.26.31
```

Step 3 Define the Voice class URI that matches the SIP URI of TTS server in the dial-peer. Syntax - voice class uri tag sip.

```
voice class uri ASR sip
pattern asr@10.78.26.31
```

Step 4 Define the SIP URI of the ASR and TTS Server. Syntax -sip:server-name@host-name | ip-address.

```
ivr asr-server sip:asr@10.78.26.31
ivr tts-server sip:tts@10.78.26.31
```

Step 5 Set up a SIP VoIP dial-peer that is an outbound dial-peer when the Gateway initiates an MRCP over SIP session to the ASR server.

```
dial-peer voice 5 voip
session protocol sipv2
destination uri ASR
dtmf-relay rtp-nte
codec g711ulaw
no vad
```

Step 6 Set up a SIP VoIP dial-peer that is an outbound dial-peer when the Gateway initiates an MRCP over SIP session to the TTS server.

```
dial-peer voice 6 voip
session protocol sipv2
destination uri TTS
dtmf-relay rtp-nte
```

```
codec g711ulaw
no vad
```

Step 7 Specify the name or IP address of a SIP server; usually a proxy server. You can then configure the dial-peer session target as session target sip-server. Syntax - sip-server {dns:[host-name] |ipv4: ip-addr[:port-num]}.

```
sip-ua
sip-server ipv4:10.78.26.31
```

Specify ASR and TTS Server Location with Individual VoiceXML Document

Media server sessions occur for each call to that application. If only a small number of applications require TTS/ASR media sessions, use the <property> extensions within those applications to define the external media server URL in the VoiceXML script.

Note Specifying the URL of media servers in a VoiceXML document takes precedence over the gateway configuration. Any value that is configured on the gateway is ignored if the same attribute is configured with a VoiceXML property.

com.cisco.tts-server

The "com.cisco.tts-server" allows the document to specify an external media server for text-to-speech operations. The media server is specified in the form of an URI, and is used in all consecutive ASR operations until the next media server is specified. An external media server specified by a property in the script takes precedence over being specified by a command through the CLI.

It can be defined for:

- An entire application or document at the <vxml> level
- A specific dialog at the form or menu level
- A specific form item

You can format the media server URI for Media Resource Control Protocol version 1 (MRCP v1), which uses Real Time Streaming Protocol (RTSP); or MRCP v2, which uses Session Initiation Protocol (SIP), for example:

<property name="com.cisco.tts-server" value="rtsp://tts-server1/synthesizer" />

<property name="com.cisco.tts -server" value="sip:mresources@mediaserver.com" />

com.cisco.asr-server

The "com.cisco.asr-server" allows a document to specify an external media server for automatic speech recognition operations. The media server is specified in the form of an URI, and is used in all consecutive ASR operations until the next media server is specified. An external media server specified by a property in the script takes precedence over being specified by a command through the CLI.

The media server's URI can be formatted for Media Resource Control Protocol version 1 (MRCP v1) which uses RTSP or MRCP v2, which uses SIP, for example:

<property name="com.cisco.asr-server" value="rtsp://asr-server1/synthesizer" />

<property name="com.cisco.asr -server" value="sip:mresources@mediaserver.com" />

Set Up VoiceXML Document Properties

Procedure

Step 1 In Unified Call Studio, view the properties for the AgeIdentification.

Step 2 Specify the VoiceXML document properties at either the root or node level.

Step 3 Select **Properties** > **General Settings** > **Language**, and specify “en–us” as the language.
Certain third-party software and hardware are compatible only with US English.

Example Gateway Configuration for MRCPv2 with Failover

```
-----------------Primary Server-----
ip host asr-en-us 10.78.26.83
ip host tts-en-us 10.78.26.83
ivr asr-server sip:asr@asr-en-us
ivr tts-server sip:tts@tts-en-us

voice class uri ASR sip
pattern asr@asr-en-us*
voice class uri TTS sip
pattern tts@tts-en-us*

dial-peer voice 5 voip
destination uri ASR
session target ipv4:10.78.26.83
session protocol sipv2
dtmf-relay rtp-nte
codec g711ulaw
no vad

dial-peer voice 6 voip
destination uri TTS
session target ipv4:10.78.26.83
session protocol sipv2
dtmf-relay rtp-nte
codec g711ulaw
no vad

-------Backup --------------------
dial-peer voice 7 voip
destination uri ASR
session target ipv4:10.78.26.20
session protocol sipv2
dtmf-relay rtp-nte
codec g711ulaw
preference 2
no vad

dial-peer voice 8 voip
destination uri TTS
session target ipv4:10.78.26.20
session protocol sipv2
dtmf-relay rtp-nte
codec g711ulaw
preference 2
no vad
```

Unified CVP Call Servers

Note Call Server load balancing is only supported on *IVR only* deployments.

Probes

The probe below is used to determine whether the Call Server is up and in service. The probe passes only if the Call Server is *In Service*. This probe is an HTTP probe using the ACE server.

The ACE server Call Server probe sends an HTTP request to **/cvp/VBServlet?MSG_TYPE=HEARTBEAT&TIMEOUT=0**. This probe takes a little more than 4 seconds to send back a response. If the Call Server is *In Service*, the HTTP 200 OK response returns.

Refer to the Cisco ACE 4700 Series Appliance Server Load-Balancing Configuration Guide.

To create the Call Server HTTP probe, place the following lines in the configuration for the ACE server:

```
    probe http PROBE_CALLSERVER_HTTP
    port 8000
    interval 6
    faildetect 1
    passdetect interval 6
    passdetect count 1
    receive 5
request method get url /cvp/VBServlet?MSG_TYPE=HEARTBEAT&TIMEOUT=0

    open 1
    expect status 200 200
```

QoS for Unified CVP Call Server

Quality of Service (QoS) is the measure of transmission quality and service availability of a network (or internetworks).

Note For more information about defining QoS criteria, see the latest Enterprise QoS Solution Reference Network Design Guide.

For more information to create policy based QOS, see section Create Policy Based QoS

Unified CVP VXML Servers

Real Servers: Configure the Physical Servers

Create a real server for every physical VXML Server you would like to load balance. Associate the probe with each server by creating a section, as shown in the following example, for each VXML server in the server farm.

```
rserver host vxml1
probe PROBE_SERVICE_ICMP
ip address 10.1.1.15
inservice
rserver host vxml2
probe PROBE_SERVICE_ICMP
ip address 10.1.1.16
inservice
```

HTTP Probe Configuration

The probe below is used to determine whether the VXML Server is up and in service. The probe passes only if the VXML Server is *In Service* . To create the VXML Server HTTP probe, place the following lines in the configuration for the ACE server.

The VXML Server probe sends an HTTP request to /CVP/Server?probe=true. If the VXML Server is up and inservice, HTTP 200 OK is returned. In the HTTP probe below, the http request is made to the port specified in the probe and the IP of the real server that this probe is associated with.

```
probe http PROBE_VXMLSERVER_HTTP
port 7000
interval 5
receive 3
faildetect 1
passdetect interval 5
passdetect count 1
request method get url /CVP/Server?probe=true
expect status 200 200
open 1
```

Note In order to get the "?", press CTRL-V before pressing the question mark.

Server Farm Configuration

```
serverfarm host vxmlserver
probe PROBE_VXMLSERVER_HTTP
rserver vxml1 7000
inservice
rserver vxml2 7000
inservice
```

Sticky Server Farm

For a VXML Server to preserve HTTP session information, you must ensure that, once the ACE server has chosen a particular VXML Server from the list of servers in a server farm, the ACE server continues to send all traffic for that session to the same VXML Server. To accomplish this, use a *sticky group* .

Refer to the Cisco ACE 4700 Series Appliance Server Load-Balancing Configuration Guide.

The following definitions apply to the settings shown below:

- **http-cookie:** Sticky method being used. In this method, when the ACE server examines a request for content, and determines through policy matching that the content is sticky, the ACE server examines any cookie or URL present in the content request. The ACE server uses the information in the cookie or URL to direct the content request to the appropriate server.
- **Cookie insert:** The ACE server inserts the cookie on behalf of the VXML Server upon the return request, so that the ACE server can perform cookie stickiness even when the VXML servers are not configured to set cookies. The cookie contains information that the ACE server uses to ensure persistence to a specific real server.

The following ACE server configuration code accomplishes the *sticky* function.

```
sticky http-cookie ACE_COOKIE VXMLServer_HTTP_STICKY
cookie insert
serverfarm vxmlserver
```

Class map Configuration

```
class-map match-all vxmlserver_HTTP_CLASS_L3
2 match virtual-address 10.1.1.17 tcp eq 7000
```

Policy map Configuration

```
policy-map type loadbalance first-match vxmlserver_HTTP_POLICY_L7

class L7_HTTP_CLASS
sticky-serverfarm VXMLServer_HTTP_STICKY

policy-map multi-match POLICY
class vxmlserver_HTTP_CLASS_L3
loadbalance vip inservice
loadbalance policy vxmlserver_HTTP_POLICY_L7
loadbalance vip icmp-reply active
```

ACE Sample Configuration

The following sample configuration does require some additional ACE configuration. Some of the modifications include interface, resource, access-list, Admin and additional context configuration changes.

```
*************PROBE CONFIGURATION***************

probe icmp PROBE_SERVICE_ICMP
interval 5
receive 3
faildetect 1
passdetect interval 5
passdetect count 1

probe http PROBE_HTTP
interval 5
receive 3
faildetect 1
passdetect interval 5
passdetect count 1
request method get url /index.html
expect status 200 200
open 1

probe tcp PROBE_ASR_TTS
port 554
interval 5
receive 3
faildetect 1
passdetect interval 5
passdetect count 1
open 1

probe http PROBE_CALLSERVER_HTTP
port 8000
interval 6
faildetect 1
passdetect interval 6
passdetect count 1
receive 5
request method get url /cvp/VBServlet?MSG_TYPE=HEARTBEAT&TIMEOUT=0
open 1
expect status 200 200

probe http PROBE_VXMLSERVER_HTTP
port 7000
interval 5
receive 3
```

```
faildetect 1
passdetect interval 5
passdetect count 1
request method get url /CVP/Server?probe=true
expect status 200 200
open 1


*************REAL SERVER CONFIGURATION****************

rserver host mediaServer1
ip address 10.1.1.1
probe PROBE_SERVICE_ICMP
inservice

rserver host mediaServer2
ip address 10.1.1.2
probe PROBE_SERVICE_ICMP
inservice

rserver host asrtts1
ip address 10.1.1.12
probe PROBE_SERVICE_ICMP
inservice

rserver host asrtts2
ip address 10.1.1.13
probe PROBE_SERVICE_ICMP
inservice

rserver host callserver1
ip address 10.1.1.19
probe PROBE_SERVICE_ICMP
inservice

rserver host callserver2
ip address 10.1.1.20
probe PROBE_SERVICE_ICMP
inservice

rserver host vxml1
probe PROBE_SERVICE_ICMP
ip address 10.1.1.15
inservice

rserver host vxml2
probe PROBE_SERVICE_ICMP
ip address 10.1.1.16
inservice


*************SERVERFARM CONFIGURATION***************

serverfarm host media_server_FARM
description Media Server Farm
probe PROBE_HTTP
rserver mediaServer1 80
inservice
rserver mediaServer2 80
inservice

serverfarm host ASR
description ASR Farm
probe PROBE_ASR_TTS
rserver asrtts1 554
inservice
rserver asrtts2 554
inservice
serverfarm host TTS
description TTS Farm
probe PROBE_ASR_TTS
rserver asrtts1 554
inservice
```

```
rserver asrtts2 554
inservice

serverfarm host callserver_farm
description Call Server Farm
probe PROBE_CALLSERVER_HTTP
rserver callserver1 8000
inservice
rserver callserver2 8000
inservice

serverfarm host vxmlserver
probe PROBE_VXMLSERVER_HTTP
rserver vxml1 7000
inservice
rserver vxml2 7000
inservice


*************STICKY-SERVERFARM CONFIGURATION*****************

sticky http-cookie ACE_COOKIE VXMLServer_HTTP_STICKY
cookie insert
serverfarm vxmlserver


*************CLASS-MAP CONFIGURATION***************

class-map match-all L3_Media_Server_VIP
2 match virtual-address 10.1.1.3 tcp eq www

class-map type http loadbalance match-all L7_HTTP_CLASS
2 match http url .*

class-map match-all ASR_CLASS_L3
2 match virtual-address 10.1.1.14 tcp eq rtsp

class-map match-all TTS_CLASS_L3
2 match virtual-address 10.1.1.18 tcp eq rtsp

class-map match-all L3_CallServer_VIP
2 match virtual-address 10.1.1.21 tcp eq 8000

class-map match-all vxmlserver_HTTP_CLASS_L3
2 match virtual-address 10.1.1.17 tcp eq 7000


*************POLICY-MAP CONFIGURATION*************

policy-map type loadbalance first-match Media_Server_L7SLB
class L7_HTTP_CLASS
serverfarm media_server_FARM

policy-map type loadbalance first-match ASR_POLICY_L7
class class-default
serverfarm ASR

policy-map type loadbalance first-match TTS_POLICY_L7
class class-default
serverfarm TTS

policy-map type loadbalance first-match CallServer_L7SLB
class L7_HTTP_CLASS
serverfarm callserver_farm

policy-map type loadbalance first-match vxmlserver_HTTP_POLICY_L7
class L7_HTTP_CLASS
sticky-serverfarm VXMLServer_HTTP_STICKY

policy-map multi-match POLICY
class L3_Media_Server_VIP
loadbalance vip inservice
loadbalance policy Media_Server_L7SLB
```

```
loadbalance vip icmp-reply active
class ASR_CLASS_L3
loadbalance vip inservice
loadbalance policy ASR_POLICY_L7
loadbalance vip icmp-reply active
class TTS_CLASS_L3
loadbalance vip inservice
loadbalance policy TTS_POLICY_L7
loadbalance vip icmp-reply active
inspect rtsp
class L3_CallServer_VIP
loadbalance vip inservice
loadbalance policy CallServer_L7SLB
loadbalance vip icmp-reply active
class vxmlserver_HTTP_CLASS_L3
loadbalance vip inservice
loadbalance policy vxmlserver_HTTP_POLICY_L7
loadbalance vip icmp-reply active
```

CHAPTER 13

Configure the Media Servers

A Media Server administers the media files that contain messages and prompts callers hear. You can add a pre-configured Media Server to the Operations Console Control Center. Once added, you can add a Media Server to one or more device pools.

- Media File Overview, page 377
- System Media Files, page 383
- Configure a Media Server, page 403

Media File Overview

This section presents a brief overview of how Unified CVP performs media file handling.

It includes information about:

- What the Media Server is.
- The media file names and types Unified CVP supports.
- How to specify the address of a media file.
- Locale syntax backward compatibility.

Media Server

In Unified CVP, the Media Server is a computer or set of computers, which “serve” the media files that contain messages and prompts that callers hear. Media files can be installed on an individual server or colocated with Unified CVP Call Server or VXML Server.

There is no artificial limit on the number of prompts; prompts are limited only by system capacity.

Tools for prompt creation are off-the-shelf, such as Audition by Adobe (formerly known as Cool Edit Pro by Syntrillium Software Corporation), and Vox Studio (http://www.xentec.be).

Note It is the customer's responsibility to select the tool, select a voice talent, record the system and application media files in the supported locales, format and encode the media files, and contact the person who is responsible for the media files on the Media Servers.

Media File Names and Types

A *media file name* is specified through Unified ICME Network VRU Script Configuration and used in the Run VRU Script request for the Play Media, Play Data, Get Digits, Menu, and Get Speech (in non-TTS applications) micro-applications. The media file naming convention allows alpha-numeric characters with the underbar character as a separator. (Spaces or hyphens are not allowed.) This naming convention provides a mechanism for an "understandable" naming convention as opposed to numeric media file names typically used by stand-alone VRUs.

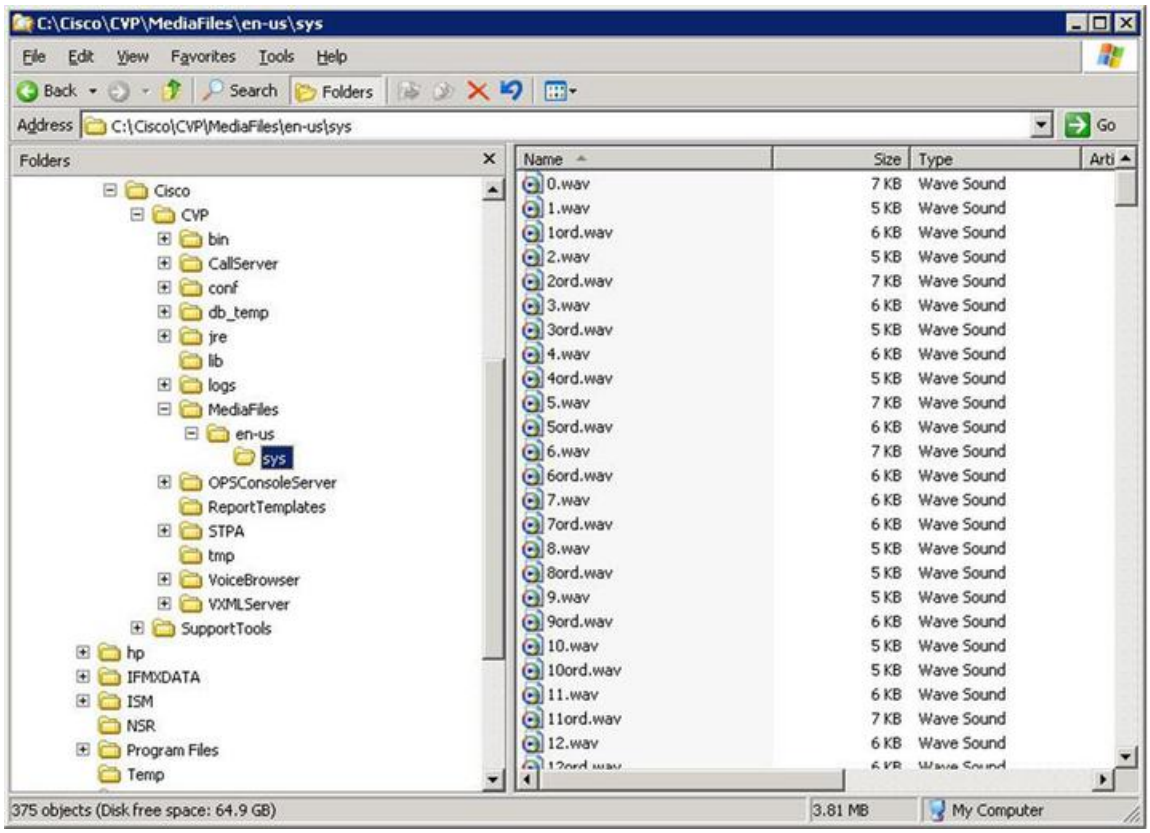
Caution The Unified Customer Voice Portal includes a library of media files/prompts for individual digits, months (referenced internally by Unified Customer Voice Portal software for a Play Data script type request), default error messages, and so on. **Creation of a full set of media/prompts for each locale referenced by the Unified CVP customer is the responsibility of the customer's Media Administrator.**

The *media file types* Unified CVP supports are µ-Law 8-bit .wav files and A-law 8-bit .wav files. Media files specified with an extension are used "as is," for example, hello.xxx. (The default file extension is .wav.)

Caution Any unexpected (and unsupported) type of media file encountered generates the logging of an error and a result code of False is returned to Unified ICME along with the ECC **user.microapp.error_code** set appropriately. From the caller's perspective, nothing was played, however it is the Script Editor developer's responsibility to write the script to handle this error condition.

Location of Media Files

The figure below displays the location of the media files if you choose to install System Media Files during Unified CVP installation.



Media File Address

The address for media files that reside on the Media Server(s) is generated by the Unified CVP. Unified ICME provides information about the file location or base URL address in the Unified ICME/IVR messages it passes when the Run VRU Script node is executed. The Unified ICME/IVR messages include ECC variables for: locale, media server set address, as well as optional system and application library name overrides. (For details about the Unified ICME/IVR messages passed to Unified CVP, refer to "Writing Scripts for Unified CVP."

The table below summarizes the data that combines to form the address of the media file:

Table 52: Media File Address Components

Parameter	Location of Data	Description	Examples
Media Server Set	ECC variable: user.microapp.media_server	File location or base URL for the Media Server. When the Media Server URL is the DNS name and the DNS Server is configured to return multiple IP addresses for a host name, the Unified CVP attempts to get the media files from each Media Server IP address in sequence with the priority given to those on the subnet. **Note** Unified CVP supports playing prompts from flash on the GW. To play these prompts, set the media_server to "flash:" instead of the hostname or IP address of the media server. When using the Media Server set for external grammars or external VXML, if the Media Server URL is the DNS name with multiple IP addresses for the hostname, it is the ASR Engine's responsibility to decide which machine to retrieve the grammar file from.	Base URL example: **http://www.machine1.com/dir1/dirs/cust1** **Note** By convention, the service provider may include their customer names at the end of the Media Server set.
Locale	ECC variable: user.microapp.locale **Default:** en-us	This field is a combination of language and country with a default of en-us for English spoken in the United States.	en-us

Parameter	Location of Data	Description	Examples
Note The Unified CVP supports the following locales: **en-us** (English, United States) and **en-gb** (English, United Kingdom), **es-es** (Spanish, Spain), and **es-mx** (Spanish, Mexico). The locale defines the grammar of a Play Data script type. If a date is to be played with a locale of **en-gb** (English, United Kingdom), the date would be played in the order of day, month, then year; for **en-us**, it is month, day, year.			
Media Library Type	The Media Library Type value passed from the VRU Script Name field. Valid options are: **A** - Application prompt library. **S** - System prompt library. **V** - External VXML. **Default:** A	The media library (directory) for the prompt is either the application prompt library defined by ECC variable user.microapp.app_media_lib (default "app") or the system prompt library defined by ECC variable user.microapp.sys_media_lib (default "sys"). **Note** When the Media Library Type is V (external VXML), the VXML file will reside in the Application Prompt Library. **Note** When the Media Library Type is A (Application prompt library), you must create the directory specified by this variable. For example, if you use the default "app" directory, you must create an app directory in ./MediaFiles/en-us	**A** (user.microapp.app_media_lib= app_banking)

Parameter	Location of Data	Description	Examples
Media File Name	The Media File Name value passed from the VRU Script Name field. Valid options are the name of the .wav file to be played, or external VXML file name, or <blank>, which translates to playing no media. This file name is ignored if TTS is being used (that is, if the **user.microapp.inline_tts** ECC variable contains a value.) **Default:** none	Name of media file or external VXML file to be played.	Main_menu
Note There are four possible reasons for using <blank> as the Media File Name: (1) For Get Digits, a prompt may not be necessary, (2) the customer may want to have a “placeholder” in the script for playing a prompt which may or may not be there (for example, an emergency conditions message), (3) change the value of barge-in to indicate a buffer flush, and (4) TTS is being used and this field is ignored.			
Media File Name Type	If not given as part of the Media File Name, the type is .wav	Type of media file to be played.	.wav

Based on the examples shown in the table above, a valid address for the Media File might be:

http://www.machine1.com/dir1/dirs/cust1/en-us/app_banking/main_menu.wav

Locale Backward Compatibility

The locale string values are compatible with current industry naming schemes:

- **en_US** has changed to **en-us**, which means that "**en** underscore **US**" (upper case) has changed to "**en** hyphen **u**s" (lower case).
- **en_GB** has changed to **en-gb**, which means that "**en** underscore **GB**" (upper case) has changed to "**en** hyphen **gb**" (lower case).

Existing scripts from previous versions of Unified CVP will continue to work with the current version of Unified CVP:

- **en_US** and **en-us** both map to U.S. English in the Application Server for use by the Application Server’s internal grammar
- **en_GB** and **en-gb** both map to U.K. English in the Application Server for use by the Application Server’s internal grammar.

- The base URL for media prompts uses the locale that is specified, without making modifications. For example, if the locale is set to **EN_US**, the base URL contains **EN_US**. If the locale is set to **XX**, the base URL contains **XX**.

To use the Unified CVP Version 1.1 default locale directory (for example, **en_US**), you must explicitly set it. When you upgrade to the current version of Unified CVP, only the new files are installed under the Unified CVP default locale directory, **en-us**. You want to have all your system prompts under one directory and all your application prompts and, optionally, external VXML in another directory. Use the **user.microapp.locale** ECC variable to set the locale directory to use, such as **en_US**.

Note Do not set the **user.microapp.locale** ECC variable if you used the default **en-us**. Also, remember that all locale values are case-sensitive.

System Media Files

The following tables describe the English System Media Files installed by Unified CVP. These system media files are intended as samples only. It is the Customer/Media Administrator's responsibility to record all the system prompts for all the locales.

The table that follows lists the System Media File information for cardinal numbers.

Table 53: System Media Files, Cardinal Numbers

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		point	point	Number
		minus	minus	Number
0	48	0	zero	All except DOW
1	49	1	one (masculine version), uno (es-mx and es-es)	All except DOW
2	50	2	two	All except DOW
3	51	3	three	All except DOW
4	52	4	four	All except DOW
5	53	5	five	All except DOW
6	54	6	six	All except DOW
7	55	7	seven	All except DOW
8	56	8	eight	All except DOW

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
9	57	9	nine	All except DOW
		10	ten	Same for the rest of all the numbers
		11	eleven	
		12	twelve	
		13	thirteen	
		14	fourteen	
		15	fifteen	
		16	sixteen	
		17	seventeen	
		18	eighteen	
		19	nineteen	
		20	twenty	
		21	twenty-one	
		22	twenty-two	
		23	twenty-three	
		24	twenty-four	
		25	twenty-five	
		26	twenty-six	
		27	twenty-seven	
		28	twenty-eight	
		29	twenty-nine	
		30	thirty	
		31	thirty-one	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		32	thirty-two	
		33	thirty-three	
		34	thirty-four	
		35	thirty-five	
		36	thirty-six	
		37	thirty-seven	
		38	thirty-eight	
		39	thirty-nine	
		40	forty	
		41	forty-one	
		42	forty-two	
		43	forty-three	
		44	forty-four	
		45	forty-five	
		46	forty-six	
		47	forty-seven	
		48	forty-eight	
		49	forty-nine	
		50	fifty	
		51	fifty-one	
		52	fifty-two	
		53	fifty-three	
		54	fifty-four	
		55	fifty-five	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		56	fifty-six	
		57	fifty-seven	
		58	fifty-eight	
		59	fifty-nine	
		60	sixty	
		61	sixty-one	
		62	sixty-two	
		63	sixty-three	
		64	sixty-four	
		65	sixty-five	
		66	sixty-six	
		67	sixty-seven	
		68	sixty-eight	
		69	sixty-nine	
		70	seventy	
		71	seventy-one	
		72	seventy-two	
		73	seventy-three	
		74	seventy-four	
		75	seventy-five	
		76	seventy-six	
		77	seventy-seven	
		78	seventy-eight	
		79	seventy-nine	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		80	eighty	
		81	eighty-one	
		82	eighty-two	
		83	eighty-three	
		84	eighty-four	
		85	eighty-five	
		86	eighty-six	
		87	eighty-seven	
		88	eighty-eight	
		89	eighty-nine	
		90	ninety	
		91	ninety-one	
		92	ninety-two	
		93	ninety-three	
		94	ninety-four	
		95	ninety-five	
		96	ninety-six	
		97	ninety-seven	
		98	ninety-eight	
		99	ninety-nine	
		oh	oh	24TOD, Date
		hundred	hundred	Number, 24TOD, Date, Currency
		thousand	thousand	Number, Date, Currency

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		million	million	Number, Currency
		billion	billion	Number, Date, Currency
		trillion	trillion	Number, Currency

The table that follows lists the System Media File information for ordinal numbers.

Note If ordinal system prompts are to be used in a script for a purpose other than dates, they should be recorded as application prompts with the true ordinal values.

Table 54: System Media Files, Ordinal Numbers

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		1ord	first	Date
		2ord	second	Date for all ordinal numbers
		3ord	third	
		4ord	fourth	
		5ord	fifth	
		6ord	sixth	
		7ord	seventh	
		8ord	eighth	
		9ord	nineth	
		10ord	tenth	
		11ord	eleventh	
		12ord	twelveth	
		13ord	thirteenth	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		14ord	fourteenth	
		15ord	fifteenth	
		16ord	sixteenth	
		17ord	seventeenth	
		18ord	eighteenth	
		19ord	nineteenth	
		20ord	twentieth	
		21ord	twenty-first	
		22ord	twenty-second	
		23ord	twenty-third	
		24ord	twenty-fourth	
		25ord	twenty-fifth	
		26ord	twenty-sixth	
		27ord	twenty-seventh	
		28ord	twenty-eight	
		29ord	twenty-nineth	
		30ord	thirtieth	
		31ord	thirty-first	

The table that follows lists the System Media File information for measurements.

Table 55: System Media Files, Measurements

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
½	189	one_half	one half	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
¼	188	one_quarter	one quarter	Char
¾	190	three_quarters	three quarters	Char
A, a	65,97	a	A	Char
B,b	66,98	b	B	Char
C, c	67,99	c	C	Char
D, d	68,100	d	D	Char
E, e	69,101	e	E	Char
F, f	70,102	f	F	Char
G, g	71,103	g	G	Char
H, h	72,104	h	H	Char
I, I	73,105	I	I	Char
J, j	74,106	j	J	Char
K, k	75,107	k	K	Char
L, l	76,108	l	L	Char
M, m	77,109	m	M	Char
N, n	78,110	n	N	Char
O, o	79,111	o	O	Char
P, p	80,112	p	P	Char
Q, q	81,113	q	Q	Char
R, r	82,114	r	R	Char
S, s	83,115	s	S	Char
T, t	84,116	t	T	Char
U, u	85,117	u	U	Char
V, v	86,118	v	V	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
W, w	87,119	w	W	Char
X, x	88,120	x	X	Char
Y, y	89,121	y	Y	Char
Z, z	90,122	z	Z	Char
Œ, œ	140,156	oe_140_156	Ligature OE	Char
À,à	192,224	a_192_224	A grave	Char
Á,á	193,225	a_193_225	A acute	Char
Â,â	194,226	a_194_226	A circumflex	Char
Ã,ã	195,227	a_195_227	A tilde	Char
Ä,ä	196,228	a_196_228	A umlaut	Char
Å,å	197,229	a_197_229	A with ring above	Char
Æ,æ	198,230	ae_198_230	Ligature AE	Char
È,è	200,232	e_200_232	E grave	Char
É,é	201,233	e_201_233	E acute	Char
Ê,ê	202,234	e_202_234	E circumflex	Char
Ë,ë	203,235	e_203_235	E umlaut	
Ì,ì	204,236	i_204_236	I grave	Char
Í, í	205,237	i_205	I acute	Char
Î,î	206,238	i_206	I circumflex	Char
Ï,ï	207,239	i_207	I umlaut	Char
Ð	208	char_208	character 208	Char
ð	240	char_240	character 240	
Ò,ò	210,242	o_210_242	O grave	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
Ó,ó	211,243	o_211_243	O acute	Char
Ô,ô	212,244	o_212_244	O circumflex	Char
Õ,õ	213,245	o_213_245	O tilde	Char
Ö,ö	214,246	o_214_246	O umlaut	Char
x	215	multiply	multiplication sign	Char
Ø,ø	216,248	o_216_248	oh stroke	Char
Ù,ù	217,249	u_217_249	U grave	Char
Ú,ú	218,250	u_218_250	U acute	Char
Û,û	219,251	u_219_251	U circumflex	Char
Ü,ü	220,252	u_220_252	U umlaut	Char
Ý,ý	221,253	y_221_253	Y acute	Char
Þ	222	char_222	character 222	Char
ß	223	ss	double s	Char
÷	247	divide	division sign	Char
þ	254	char_254	character 254	Char
Ÿ,ÿ	159,255	y_159_255	character 159 or 255	Char

The table that follows lists the System Media File information for month values.

Table 56: System Media Files, Months

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		January	January	Date
		February	February	Date
		March	March	Date

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		April	April	Date
		May	May	Date
		June	June	Date
		July	July	Date
		August	August	Date
		September	September	Date
		October	October	Date
		November	November	Date
		December	December	Date

The table that follows lists the System Media File information for month values.

Table 57: System Media Files, Days

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		Sunday	Sunday	DOW
		Monday	Monday	DOW
		Tuesday	Tuesday	DOW
		Wednesday	Wednesday	DOW
		Thursday	Thursday	DOW
		Friday	Friday	DOW
		Saturday	Saturday	DOW

The table that follows lists the System Media File information for month values.

Table 58: System Media Files, Time

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		hour	hour	Etime, 24TOD per locale, TOD per locale
		hours	hours	Etime,24TOD per locale,TOD per locale
		minute	minute	Etime
		minutes	minutes	Etime
		second	second	Etime,24TOD
		seconds	seconds	Etime,24TOD
		on	on	per locale(unused for en-us)
		at	at	per locale(unused for en-us)
		am	am	TOD
		pm	pm	TOD
		oclock	oclock	TOD

The table that follows lists the System Media File information for currency values.

Note The customer's Media Administrator may prefer to replace the contents of "currency_minus" (for the negative amount) and "currency_and" (the latter can even be changed to contain silence).

Table 59: System Media Files, Currency

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		currency_ minus	minus	Currency
		currency_and	and	Currency

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
$	36	USD_dollar	dollar	Currency
		USD_dollars	dollars	Currency
		Note Unified CVP uses the USD_dollar.wav and USD_dollars.wav media files; the dollar.wav and dollars.wav used by ISN Version 1.0 are no longer installed.		
$	36	CAD_dollar	dollar	Currency
		CAD_dollars	dollars	Currency
		HKD_dollar	dollar	Currency
		HKD_dollars	dollars	Currency
¢	162	cent	cent	Currency
		cents	cents	Currency
		euro	euro	Currency
£	163	GBP_pound	pound	Currency
		GBP_pounds	pounds	Currency
		penny	penny	Currency
		pence	pence	Currency
		MXN_peso	peso	Currency
		MXN_pesos	pesos	Currency
		centavo	centavo	Currency
		centavos	centavos	Currency

The table that follows lists the System Media File information for gaps of silence and miscellaneous phrases.

Table 60: System Media Files, Silence and Miscellaneous Phrases

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		silence_.1_ sec	(.1 second of silence)	Used for pauses where needed
		silence_.25_ sec	(.25 second of silence)	Used for pauses where needed
		silence_.5_ sec	(.5 second of silence)	Used for pauses where needed
		silence_1_sec	(1 second of silence)	Used for pauses where needed
		and	and	Etime,TOD,25TOD

The table that follows lists the System Media File information for ANSI characters.

Table 61: System Media Files, ANSI Characters

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
	32	space	space	Char
!	33	exclamation_ mark	exclamation mark	Char
"	34	double_ quote	double quote	Char
#	35	pound	pound	Char
%	37	percent	percent	Char
&	38	ampersand	ampersand	Char
'	39	apostrophe	apostrophe	Char
(	40	open_ parenthesis	open parenthesis	Char
)	41	close_ parenthesis	close parenthesis	Char
*	42	asterisk	asterisk	Char
+	43	plus	plus	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
,	44	comma	comma	Char
-	45	hyphen	hyphen	Char
.	46	period	period	Char
/	47	slash	slash	Char
:	58	colon	colon	Char
;	59	semicolon	semicolon	Char
<	60	less_than	less than	Char
=	61	equal	equal	Char
	62	greater_than	greater than	Char
?	63	question_ mark	question mark	Char
@	64	at_symbol	at	Char
[	91	left_square_bracket	left square bracket	Char
\	92	backslash	backslash	Char
]	93	right_square_bracket	right square bracket	Char
^	94	caret	caret	Char
_	95	underscore	underscore	Char
`	96	single_quote	single quote	Char
{	123	open_brace	open brace	Char
|	124	pipe	pipe	Char
}	125	close_brace	close brace	Char
~	126	tilde	tilde	Char
’	130	char_130	low single quote	Char
ƒ	131	char_131	F with hook	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
”	132	low double quote	low double quote	Char
…	133	ellipsis	ellipsis	Char
†	134	char_134	character 134	Char
‡	135	char_135	character 135	Char
ˆ	136	char_136	character 136	Char
‰	137	per_mille	per mile	Char
Š	138	char_138	character 138	
<	139	left_pointing _angle	left pointing angle	Char
‘	145	left_single_ quote	left single quote	Char
’	146	right_single_ quote	right single quote	Char
“	147	left_double_ quote	left double quote	Char
”	148	right_double _quote	right double quote	Char
·	149	bullet	bullet	Char
–	150	en_dash	en dash	Char
—	151	em_dash	em dash	
˜	152	small_tilde	small tilde	Char
™	153	trade_mark	trade mark	Char
š	154	char_154	character 154	Char
›	155	char_155	character 155	Char
¡	161	exclamation_ mark_ inverted	inverted exclamation mark	Char
¤	164	char_164	character 164	Char
¦	166	broken_pipe	broken pipe	Char
§	167	section	section	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
¨	168	char_168	character 168	Char
©	169	copyright	copyright	Char
ª	170	char_170	character 170	Char
«	171	left_double_ angle_ quote	left double angle quote	Char
¬	172	not	not	Char
-	173	char_173	character 173	Char
®	174	registered	registered	Char
¯	175	char_175	character 175	Char
°	176	degree	degree	Char
±	177	plus_minus	plus or minus	Char
²	178	superscript_ 2	superscript two	Char
³	179	superscript_ 3	superscript three	Char
´	180	acute_accent	acute accent	Char
µ	181	micro	micro	Char
¶	182	paragraph	paragraph	Char
·	183	middle_dot	middle dot	Char
¸	184	cedilla	cedilla	Char
¹	185	superscript_ 1	superscript one	Char
º	186	char_186	character 186	Char
»	187	right_double _angle_ quote	right double angle quote	Char
¿	191	question_ mark_ inverted	inverted question mark	Char

Miscellaneous Files

The table that follows lists files that are not used by Unified CVP micro-applications; these files are included for use in customer scripts.

Table 62: Miscellaneous Media Files

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
Error	v	invalid_entry_error	Your entry is invalid.	Error message
	v	no_entry_error	Please make a selection.	Error message
	v	system_error	We are currently experiencing technical difficulties with this site. Please try again later when we can service you much better.	Error message
	v	critical_error	We are currently experiencing technical difficulties with this site. Please try again later when we can service you much better.	Error message
	v	critical_error_ULaw .	We are currently experiencing technical difficulties with this site. Please try again later when we can service you much better	Error message
	v	critical_error_ALaw	We are currently experiencing technical difficulties with this site. Please try again later when we can service you much better.	Error message
	v	440beep	<single beep tone>	Unused
	v	busy_tone	<single busy tone>	Unused
	v	busy_tone30	<busy tone 1 per second for 30 seconds>	Unused

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
	v	central	Central	Unused
	v	credit_of	Credit Of	Unused
	v	dash	dash	Unused
	v	daylight	daylight	Unused
	v	dialtone	<4 seconds of dial tone>	Unused
	v	dialtone2fastbusy60	<9 seconds of dialtone> followed by <30 seconds of fast busy tone>	Unused
	v	dot	dot	Unused
	v	eastern	Eastern	Unused
	v	ENTER_PHONE_NUMBER	Please enter the phone number.	Unused
	v	fastbusy	<a single fastbusy tone + silence (total of 1 second)>	Unused
	v	fastbusy60	30 seconds of <fastbusy tone>	Unused
	v	FINISHED	When you have finished, press	Unused
	v	goodbye	Goodbye	Unused
	v	Mountain	Mountain	Unused
	v	negative	negative	Unused
	v	of	of	Unused
	v	pmgr_sys	pmgr_sys	Unused
	v	pacific	Pacific	Unused
	v	positive	positive	Unused

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
	v	ringback	<ring back tone for 1 second followed by 2 seconds of silence>	Unused
	v	savings	savings	Unused
	v	standard	Standard	Unused
	v	Star	star	Unused
	v	thankyou	Thank you	Unused
	v	the	the	Unused
	v	time	time	Unused
	v	try_again	Please try again	Unused

System Media File Error Messages

Three error messages are included with the System Media files:

- **Critical error.** Message played when system problem exists and the SIP Service cannot process the call. (Example content for en-us: "We are currently experiencing technical difficulties with the site, please try again later and we can serve you much better.")

  Note If you do not want an English spoken critical media, you need to copy the language specific files to the location specified in this section.

  Critical error messages are *not* located on the Media Server:

  - For **SIP Service**, the critical_error.wav media file is located in <*install path*> \OpsConsoleServer\GWDownloads (for example, C:\Cisco\CVP\OpsConsoleServer\GWDownloads).
  - For **non-Unified CVP SIP Service**, an error.wav media file is located in <*install path*>\CVP\audio (for example, C:\Cisco\VXMLServer\Tomcat\webapps\CVP\audio).

  Note You can record "override" prompts to replace the critical media files. However, you must save them with their original hard-coded names and place them in their original locations.

- **no_entry_error**. Message played when the caller does not respond to a menu prompt. (Example content for en-us: “Please make a selection.”) The original prompt is then repeated.
- **invalid_entry_error**. Message played when the caller enters an incorrect response to a menu prompt. (Example content for en-us: “Your entry is invalid.”) The original prompt is then repeated.

Note These files are shared by all applications.

If a dialogue needs to be altered for a specific Get Digits, Get Speech or Menu request in the Unified ICME script, override flags can be set in the Network VRU Script Configuration Parameters.

Note Override flags are available for the Get Digits, Get Speech, and Menu micro-applications, only. Refer to "Writing Scripts for Unified CVP."

You must record the “override” prompts, save them with the hard coded names <prompt name>_no_entry_error.wav and <prompt_name>_invalid_entry_error.wav, and place them with other application-specific media files in the Application Media library.

Note This override will not work when there is not a specific file name used (for instance, when Unified CVP is using the TTS feature).

Configure a Media Server

A Media Server administers the media files that contain messages and prompts callers hear. You can add a pre-configured Media Server to the Operations Console. Once added, you can add a media server to one or more device pools.

The Media Server is a simple web server with the sole purpose within Unified CVP is to store and serve .wav files to the VoiceXML gateway, as required in order to render VoiceXML pages. As with ASR/TTS Servers, media servers can be deployed alone, as a redundant pair, or with a ACE in a farm. The VoiceXML gateway caches the `.wav` files it retrieves from the Media Server. In most deployments, the Media Server encounters extremely low traffic from Unified CVP.

The Operations Console online help topic *Managing Devices > Configuring a Media Server* provides details for performing the following tasks:

- Adding a Media Server

  Collect the IP address from the media server before adding it to the Operations Console.
- Editing a Media Server Configuration
- Deleting a Media Server
- Adding and Removing a Media Servers from a Device Pool
- Finding a Media Server

CHAPTER 14

Java Runtime Environment Minor Update

Use the JREUpdate.bat script to install a minor update of Java Runtime Environment (JRE) version on your Unified CVP Server. For example, you can install a minor update of JRE version 1.6.0_24 to 1.6.0_81.

Download the JREUpdate.zip from the following location:

http://software.cisco.com/download/release.html?mdfid=270563413&softwareid=280840592&release=10.5%281%29&relind=AVAILABLE&rellifecycle=&reltype=latest

Note The script does not support a major upgrade of JRE versions. For example, the script does not allow a major upgrade of JRE Version 1.6.0_81 to 1.7.0_45.

Procedure

Step 1 Download and install the preferred Java Development Kit (JDK) version on your personal machine.

Step 2 Copy the JRE folder from the installed JDK to a known location on the Unified CVP Server. For example, `C:\JRE`.

Note The jre folder is available in the JDK root folder. For example: `C:\jdk1.7.0_67\jre`.

Step 3 Right-click the JREUpdate.zip file and extract the files to a known location on your Unified CVP Server. For example, `C:\Cisco\CVP\bin`.

Step 4 Run this script from the command prompt: **C:\Cisco\CVP\bin >JREUpdate.bat apply C:\JRE**. The script runs and Unified CVP JRE is updated to the new version.

Step 5 Ensure that the script output displays the updated JRE version.

The JREUpdate.bat script takes a backup of the old JRE to `C:\Cisco\CVP\jre.old` folder location. To revert to the previous backup version of JRE, run this script from the command prompt: **C:\Cisco\CVP\bin>JREUpdate.bat revert** .

APPENDIX A

Using the Helix Server

Starting with Unified CVP10.0(1), Cisco IOS gateway supports both µ-Law and A-law wav file format.

- Broadcast Stream Using the Helix Server, page 407

Broadcast Stream Using the Helix Server

Procedure

To create a broadcast stream using the Helix™ Server:

Procedure

Step 1 Install Helix Server.
The default installation and configuration of Helix server is all that is required for use with Unified CVP. Refer to *Helix Server Administration Guide* for information about installing and configuring Helix Server.

Step 2 Use Simulated Live Transport Agent (SLTA) of Helix Server to create a broadcast stream.
Helix provides many ways to generate broadcast streams. Use the SLTA utility to create a broadcast stream of wav files that are already encoded in the µ-Law/ A-law format.

Step 3 Create a text file playlist that contains the path and filename of each file you intend to broadcast.
Using a mix of other file formats in the playlist is not supported by Helix Server and may cause a broadcast stream to terminate.

```
Command Prompt
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>more playlist.txt
african\Track_1.wav
african\Track_2.wav
african\Track_3.wav
african\Track_4.wav
african\Track_5.wav
african\Track_6.wav
african\Track_7.wav
african\Track_8.wav
african\Track_9.wav
african\Track_10.wav
african\Track_11.wav
african\Track_12.wav

C:\broadcast_files>_
```

Step 4 Using the SLTA utility, run the **slta.bat script. slta.bat <ip-address> <port> <helix-username> <helix-password> <stream-name> <playlist>** to start the broadcast stream.
The helix userid and password will match the user name and password that you entered when you installed Helix Server.

In the following illustration you see the creation of the RTSP broadcast stream for african.rm in which the files specified in playlist.txt will be continuously looped.

```
Command Prompt - slta.bat 127.0.0.1 80 helix cvpcisco african.rm playlist.txt
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>
C:\broadcast_files>slta.bat 127.0.0.1 80 helix cvpcisco african.rm playlist.txt
SLTA.EXE - Live Broadcast Simulation Utility

Executing SLTA with
host=127.0.0.1. port=80. userid=helix. password=cvpcisco.
live_file=african.rm, playlist_file=playlist.txt

SLTA (c) 2001-2008 RealNetworks, Inc. All rights reserved.
Transmitting african\Track_1.wav...
0----1----2----3----4----5----6----7----8----9----10
*****_
```

Related Topics

Configure Custom Streaming Ringtones, on page 287

INDEX

A

accessing micro-application by specifying Run VRU Script Node 109
ACE for Load Balancing 363
    configuration overview 363
    major components listed 363
ACE sample configuration 373
application logging, VoiceXML 248
array formulas, ECC variable 156
ASR, type-ahead support 162
Automatic Speech Recognition for micro-applications 110

B

before you begin tasks 4, 6, 7
    installation 6
    routing calls through the network to the VRU 7
    setting speed and duplex for Unified CVP 7
broadcast stream 407
    Helix Server 407

C

CA-signed certificates 207, 208, 212
    adding for HTTPS web access 208
    adding to the keystore 207
    using Tomcat to present to inbound HTTPS clients 212
CAC 326
    configuring for locations-based 326
caching 327, 328
    configuring on the gateway 328
    considerations for gateway prompt 327
    properly, determining if gateway is 328
Call Director call flow model using SIP 27
    configuration instructions 27
Call Director with Unified ICME call flow model 25
    configuring 25
Call Director with Unified ICMH call flow model 26
    configuring 26
Call Director, Unified ICME, with SIP Refer Transfer featurel 286
call flow models 8, 17, 18, 25, 26, 80, 227, 286
    Call Director with Unified ICME 25
    Call Director with Unified ICMH 26
    Call Director, Unified ICME, with SIP Refer Transfer feature 286
    Comprehensive call flows for pre-routed calls 80
    NetworkVRU types 8
    Standalone with ICM Lookup 18
    VXML Server (Standalone) 17, 227
call flows 327
    WAN bandwidth required for prompts 327
Call Server 224, 283
    communicating with VXML Server 224
    setting up with redundant proxy servers 283
call statistics 253, 255, 258
    ICM Service 253
    IVR service 255
    SIP Service 258
Call Studio 162, 163, 166, 168, 170, 219, 221
    about 162, 221
    deploying scripts 170
    integrating scripts (simplified with Unified ICME scripts 168
    integrating scripts with Unified ICME scripts 166
    using ReqICMLabel 163
call survivability 313, 315, 317
    examples 317
    gateway configuration 315
    installing the script 313
    script, about 313
calls 80, 81, 83, 299, 300, 305, 313, 315
    arriving at Unified ICME through a pre-route-only NIC 80
    originated by an ACD or call routing interface based VRU using a PG 83
    originated by Cisco Unified Communications Manager 81
    survivability script, installing 313
    survivability, configuring 313
    survivability, configuring the gateway 315
    transferring and queuing 299
    transferring from agent to agent 305
    transferring using SIP Service 300
    transferring using Unified CVP in Comprehensive mode 300

Capture micro-application error code settings 103
Centralized Call Control with Distributed Queuing call flow model example 326
Cisco Developer Services Program 4
CLI (for WSM) 189, 190, 191, 194
  access the CLI 191
  create user 189
  overview 190
  various kinds of help for the CLI 194
CLI field, overriding on outgoing transfers 307
Communications Manager 279
  configuration tasks 279
communications, securing between VoiceXML Gateway and VXML Server 212
Comprehensive 80
  call flows for pre-routed calls 80
Comprehensive call flow model using SIP 42
  configuration instructions 42
configuration example, DNS zone file Linux NAMED for Ingress Gateway 36
configuration instructions 27, 42, 64, 73, 84
  Call Director call flow model using SIP 27
  Comprehensive call flow model using SIP 42
  Type 3 or 7 VRU-Only call flow model 73
  Type 8 VRU-Only call flow model 64
  Unified CVP switch leg 84
configuration parameters 108, 115
  Play Media micro-application 115
configuration prerequisites 3
configuration settings 326
  configuring Unified CM Service 326
configuration, DNS zone file Linux NAMED for VoiceXML Gateway 62
configuring 17, 18, 25, 26, 84, 86, 93, 108, 201, 210, 219, 239, 243, 246, 281, 306, 307, 308, 309, 313, 315, 326, 328
  caching on the gateway 328
  Call Director with Unified ICME call flow model 25
  Call Director with Unified ICMH call flow model 26
  CLI override 307
  common Unified ICMH:defining Unified CVP ECC variables 86
  determining if gateway is caching properly 328
  differentiating VRUs based on dialed number 93
  gateway for call survivability 315
  locations-based call admission control 326
  logging 246
  logging and event notifications 243
  network transfer from CTI OS Agent Desktop 306
  network transfer from IP phone 306
  Network VRU Script 108
  Re-route on No Answer agent desk settings 309
  Re-route on No Answer for Unified CVP 308
  Re-route on No Answer operation for Unified CCE with Unified IP IVR 308
configuring *(continued)*
  Re-route on No Answer operation with Unified CVP 308
  Router Requery 309
  secure access to the router using Protocol SSH Version 2 210
  security 201
  setting up VXML Server connection to Call Server 239
  SIP devices 281
  Standalone with ICM Lookup call flow model 18
  Unified CCE Re-route on No Answer for Unified CVP 308
  Unified CM service configuration settings 326
  Unified CVP switch leg, common configuration instructions 84
  Unified CVP with a script to transfer calls with critical application error or WAN failure 313
  VXML 219
  VXML Server (Standalone) call flow model 17
correlation ID 329
  how it works 329
  incoming UUI 329
Courtesy Callback 335, 336, 337, 344
  configuration topics 337
  descriptions of sample scripts 344
  files provided 335
  Location of example scripts and audio files 336
  overview 335
  useage scenerio 335
creating 108
  Unified ICME applications 108
CTI OS Agent Desktop, configuring network transfer 306
CVP devices 206
  enabling security 206
CVP scripting 95, 97, 98, 102, 103, 114, 115, 122, 132, 139, 145, 162
  about 97
  Capture micro-application 114
  considerations for ICM Enterprise 95
  data handling 102
  error checking 103
  Get Digits (GD) micro-application 132
  Get Speech (GS) micro-application 145
  information exchange 102
  Menu (M) micro-application 139
  micro-applications 98
  Play Data (PD) micro-application 122
  Play Media (PM) micro-application 115
  with Call Studio 162
  with ICM Enterprise 98
CVPLogMessages.properties file 244
CVPLogMessages.xml file 244
  editing 244
CVPSNMPLogger, viewing for VXML Server 241

D

data format, Get Speech micro-application 152
data handling in scripting 102
data storage for Play Data micro-application 122
debugging tips 330
    trace settings for the gateway 330
Default Media Server 113
    for Micro-applications 113
defining 330, 331
    Unified ICME script label for hookflash transfer 331
    Unified ICME script label for outpulse transfer 330
    Unified ICME script label for Two B-Channel Transfer 331
deploying scripts 170
    with Call Studio 170
    with Operations Console 170
Dialed Number Patterns, setting the expiration timeout of an outbound call 290
digit entry completion 138, 144
    Get Digits micro-application 138
    Menu micro-application 144
DNIS - Dialed Number Identification Service 7
documents 327
    VoiceXML, WAN bandwidth 327
DTMF input collection, Get Speech micro-application 151
duplex, settings for Ethernet Switch/Server NIC settings 7
dynamic audio file capability 111
dynamic prompt, using 112

E

ECC variables 86, 92
    defining for common Unified ICMH call flow models 86
    using metadata 92
enabling 307
    CLI override 307
entry completion 153
    Get Speech micro-application 153
error checking 103
    in scripting 103
error code settings 103
    Capture micro-application 103
    Get Digits micro-application 103
    Get Speech micro-application 103
    Menu micro-application 103
    Play Data micro-application 103
    Play Media micro-application 103
    user.microapp.error_code ECC variable for non-video 103
    user.microapp.error_code ECC variable for video 103
error codes 103, 230
    user.microapp.error_code ECC 103
    VXML Server 230
error message playback on abnormal disconnects with SIP 290
Ethernet Switch/Server NIC settings for speed and duplex 7
event statistics 249
    viewing 249
events 243, 245, 265
    notifications, configuring 243
    severity levels for 245
    SNMP raise/clear 265
    using logs to interpret 243
examples 36, 62, 111, 112, 120, 131, 134, 141, 148, 326, 327
    determining bandwidth consumed by prompts 327
    DNS zone file Linux NAMED configuration for Ingress Gateway 36
    DNS zone file Linux NAMED configuration for VoiceXML Gateway 62
    Get Digits micro-application configuration 134
    Get Speech micro-application configuration 148
    Menu micro-application configuration 141
    Play Data micro-application configuration 131
    Play Media micro-application configuration 120
    using a dynamic audio file 111
    using the dynamic prompt 112
    WAN bandwidth requirements in Centralized Call Control 326
exporting Operations Console certificate to a file 198, 209
external VoiceXML 111, 154, 157, 160
    error handling, Get Speech micro-application 154
    file contents 160
    for micro-applications 111
    Get Speech micro-application 154
    passing data back to Unified ICME 157
    passing information 154

F

failover mechanisms 361
    IVR Service 361
file configuration example for SIP messaging redundancy 93
file contents, external VoiceXML 160

G

gateway 260
    statistics 260
gateways 327, 328, 330
    configuring caching 328
    debug trace settings 330
    log, GTD values 330
    prompt caching methods 327
Get Digits micro-application 103, 134, 138
    configuration examples 134
    digit entry completion 138

Get Digits micro-application *(continued)*
error code settings 103
if no entry timeout occurs 138
Get Speech micro-application 103, 145, 148, 151, 152, 153, 154
configuration example 148
data format 152
DTMF input collection 151
entry completion 153
error code settings 103
external VoiceXML 154
grammar specification 145
if no entry timeout occurs 153
grammar specification, Get Speech micro-application 145
GTD, debugging 330

H

Helix Server 287, 407
broadcast stream 407
streaming ringtones 287
high-level configuration steps for call flow models 15
hookflash transfer, defining a Unified ICME script label 331
HTTPS 212, 213
securing between Unified CVP and IOS devices 213
using Tomcat to present CA-signed certificates 212

I

ICM Service 253
statistics 253
importing 210
Operations Console certificate to a gateway 210
incoming UUI 329
configuring the gateway 329
correlation ID 329
information exchange 102
in scripting 102
infrastructure statistics 250
descriptions 250
Ingress Gateway, setting up to use redundant proxy servers 283
installing 313
call survivability script 313
integrating scripts 168
Call Studio and ICME 168
integrating scripts - traditional 166
Call Studio and ICME 166
IOS devices 209
securing communications 209
IP phone, configuring network transfer 306
IP transfer example for Unified CCE routing 306
ISDN call parameters 291
IVR Service 255
statistics 255
IVR Service failover mechanism 361
IVRs, types of 299

J

JMX communications 201
securing 201

K

keystore 202
and trusted certificates 202
keytool 202
and trusted certificates 202

L

licensing 250
statistics 250
load balancing SIP calls 282
loadbalancing using ACE 363
also refer to ACE 363
locations-based call admission control, configuring 326
log messages 244
CVPLogMessages.xml 244
logging and event notifications, configuring 243
logs 243
using to interpret events 243
logs, VXML Server 247

M

media server 403
using with Unified CVP 403
Media Server 113
specify in Operations Console 113
Menu micro-application 103, 141, 144
configuration examples 141
digit entry completion 144
error code settings 103
if no entry timeout occurs 144
micro-applications 98, 108, 109, 110, 111, 114, 115, 122, 132, 139, 145
accessing 108
Capture 114
dynamic audio file support 111
Get Digits (GD) 132
Get Speech (GS) 145

micro-applications *(continued)*
Menu (M) 139
Play Data (PD) 122
Play Media (PM) 115
specifying Run External Script Node to access 109
using Automatic Speech Recognition and Text-to-Speech 110
using external VoiceXML 111
using for scripting 110
Micro-applications 113
Default media server 113

N

network transfer script example 305
Network VRU Script 108
configuring 108

O

Operations Console 170, 249
deploying Call Studio scripts 170
viewing event statistics 249
outpulse transfer, defining a Unified ICME script label 330

P

param elements, using 154
parameter element, URL 154
passing data 239
to Unified ICME 239
play back types for voice data 124
Play Data micro-application 124
Play Data micro-application 103, 122, 124, 131
configuration examples 131
data storage 122
error code settings 103
play back types for voice data 124
Play Media micro-application 103, 120
configuration example, play welcome message 120
error code settings 103
Post Call Survey 354, 355
configure CVP Call Server 354
configure ICM 355
overview 354
pre-routed calls 80
Comprehensive call flows 80
prompts 327
proper gateway caching methods 327
retrieval 327
Protocol SSH Version 2, configuring secure access to the router 210

Q

Quality of Service (QoS) 226
on VXML Server 226
queue and transfer to a skill group script example 302

R

Re-route on No Answer 308, 309
agent desk settings configuration 309
configuring Unified CCE for Unified CVP 308
operation for Unified CCE with Unified IP IVR 308
operation with Unified CVP 308
recording Unified CVP system-specific information 246
redundancy 93
local SRV file configuration example for SIP messaging 93
reporting 222
VXML Server 222
Reporting Server 224, 264
communicating with VXML Server 224
statistics 264
ReqICMLabel 163
using to pass data 163
requirements 157, 275
minimal component version for Unified ICME Warm Consult Transfer/Conference to Unified CVP feature 275
VoiceXML 157
resource properties file 244
Router Requery configuration 309

S

scripting for CVP 95, 97, 98, 102, 103, 114, 115, 122, 132, 139, 145, 162
about 97
Capture micro-application 114
considerations for ICM Enterprise 95
data handling 102
error checking 103
Get Digits (GD) micro-application 132
Get Speech (GS) micro-application 145
information exchange 102
Menu (M) micro-application 139
micro-applications 98
Play Data (PD) micro-application 122
Play Media (PM) micro-application 115
with Call Studio 162
with ICM Enterprise 98

scripts 228, 291
incoming UUI to be delivered to Unified ICME routing 291
integrating VXML and ICME scripts 228
securing 202, 210, 213
access to the router using Protocol SSH Version 2 210
HTTPS between Unified CVP and IOS devices 213
JMX communications between Unified CVP components 202
security 201, 206, 209, 212, 225
communications between Unified CVP and IOS devices 209
communications between VoiceXML Gateways and VXML Server 212
configuring 201
enabliing on Unified CVP XML Server 225
enabling on CVP devices 206
JMX communications 201
self-signed certificate, adding 204
service creation environments (script eidtors) 97
Call Studio 97
ICM Script Editor 97
services 4
developer 4
setting 7, 283
Call Server with redundant proxy servers 283
Ingress Gateway to use redundant proxy servers 283
speed and duplex for Ethernet Switch/Server NIC settings 7
settings, sendtooriginator in the SIP Service 289
severity 244, 245
log messages 244
of events 245
SIP 93, 290
critical error message playback on abnormal disconnects 290
local SRV file configuration example for messaging redundancy 93
using Dialed Number Patterns 290
SIP calls 275
using Warm Transfer feature 275
SIP calls, load balancing 282
SIP Headers 296, 297
configuration 297
overview, using to pass information 296
warnings about modifying 296
SIP Hookflash Support 292, 293
configuration 293
Overview 292
SIP Proxy Server 281
configuring for Unified CVP 281
qualified server 281
SIP Service call statistics 258
SIP Service sendtooriginator setting 289
SNMP 241, 265
raise/clear events 265
raise/clear mappings 265
viewing CVPSNMPLogger for VXML Server 241
specifying Run External Script Node to access Unified CVP micro-applications 109
speech server 362
configuring for Unified CVP 362
speed, settings for Ethernet Switch/Server NIC settings 7
SSL 201
CVP component communication 201
Standalone call flow model 239
Standalone with Request ICM Label variation 239
Standalone call flow models 17, 18
VXML Server 17
with ICM Lookup 18
Standalone with Request ICM Label variation 239
statistics 250, 253, 255, 258, 260, 262, 264
gateway 260
ICM service 253
infrastructure 250
IVR Service 255
licensing 250
Reporting Server 264
SIP Service 258
thread pool 250
VXML Server 262
streaming ringtones 287
Helix Server 287
switch leg, common configuration 84
syslog, using 243
system statistics 250
licensing 250
thread pool 250

T

tasks, setting up Unified ICME for scripting 108
Text-to-Speech 110
for micro-applications 110
thread pool statistics 250
transferring 300, 305
call from agent to agent 305
calls using SIP Service 300
calls using Unified CVP in Comprehensive mode 300
transferring and queuing calls 299
transfers 286, 300, 306, 307
configuring network from CTI OS Agent Desktop 306
configuring network from IP phone 306
label script example 300
overriding the CLI field on outgoing 307
using "RFXXXX" type Unified ICME labels with Call Director, Unified ICME 286
trusted certificate 204, 207, 208
adding for HTTPS web access 208
adding to the keystore 207

trusted certificate *(continued)*
self-signed 204
Two B-Channel Transfer, defining a Unified ICME script label 331
Type 3 or 7 VRU-Only call flow models 73
configuration instructions 73
overview 73
Type 8 VRU-Only call flow models 64, 65
configuration instructions 64
overview of Unified ICME version 64
overview of Unified ICMH version 65

U

Unified CVP XML Server 225
enabling security 225
Unified ICME 96, 108, 157, 239
creating applications to access Unified CVP micro-applications 108
passing data 239
passing data back with external VoiceXML 157
scripting 96
setting up for scripting 108
Unified ICME setup 108
for scripting 108
User-to-User Information (UUI) 295
DTMF label 295
overview, passing information 295
processing scenerios 295
using 110, 154, 243, 275, 276
Automatic Speech Recognition and Text-to-Speech for micro-applications 110
logs to interpret events 243
param elements 154
syslog 243
Warm Transfer feature with SIP calls 275
Warm Transfer feature with Type 10 VRU 276

V

videos 103
error code settings for Full Video 103
viewing SNMP for the VXML Server 241
voice data, play back types 124
voice prompts, storing 327
VoiceML Service 219
VoiceXML 96, 154, 157, 160
error handling, Get Speech micro-application 154
external, Get Speech micro-application 154
file contents, external 160
passing data back to Unified ICME with external 157
VoiceXML *(continued)*
passing information to the external 154
requirements 157
scripting 96
VoiceXML application logging 248
VoiceXML gateways 212
securing communications 212
VoiceXML logs 246
VoiceXML Service, about 220
VRU 109, 117
Run VRU Script node 109, 117
VXML 219, 223
configuring 219
configuring VXML Server 223
Server 219
VXML Server 212, 220, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 239, 241, 262
about 220
as a standalone component 220
as an add-on component 220
communicating with Reporting Server 224
configuring 223
configuring Standalone 227
correlating logs with CVP and ICME logs 229
error codes 230
filters for reporting 225
integrating VXML and ICME scripts 228
Quality of Service (QoS) 226
reporting 222
securing communications 212
setting up to connect to Call Server 239
Standalone solution configuration 230
Standalone solution, ICM Lookup 231
Standalone solution, withou ICM Lookup 232
statistics 262
viewing CVPSNMPLogger 241
VXML Server (Standalone) 227
configuring 227
VXML Server logs 246, 247
about 247
correlating Unified CVP/Unified ICME logs 246

W

WAN bandwidth 326, 327
consumed by VoiceXML documents 327
contributing factors 326
Warm Transfer feature 275, 276
using with SIP calls 275
using with Type 10 VRU 276
Web Services Manager 189, 190
create user 189

Web Services Manager *(continued)*
overview 190