THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

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

© 2012-2013 Cisco Systems, Inc. All rights reserved.
# CONTENTS

## CHAPTER 1

**About Cisco Packaged Contact Center Enterprise**  1  
Packaged CCE Components  1  
System Deployment  3  

## CHAPTER 2

**Basics**  5  
  
Sign In  5  
  
User Access  6  
  
User Interface and Navigation  6  
  
Supervisor Interface and Navigation  6  
  
Languages  7  
  
Menus  7  
  
Home  7  
  
Manage  8  
  
Settings  10  
  
Common Tools and Tasks  10  
  
Lists  11  
  
Add Objects  11  
  
Search a List  11  
  
Sort a List  12  
  
Popups  12  
  
Keyboard Shortcuts  12  
  
Native Character Sets  13  
  
Configuration in Packaged CCE  13  

## CHAPTER 3

**Manage Agents**  15  
  
Agents  15  
  
Add and Maintain Agents  16  
  
Update Agents  19
Delete Agents  19
Reporting on Agents  19
Supervisors  20
   Add and Maintain Supervisors  20
   Update Supervisors  21
   Delete Supervisors  21
Desk Settings  22
   Add and Maintain Desk Settings  22
   Update Desk Settings  23
   Delete Desk Settings  24
Agent Desktops  24
   Agent Desktop Comparison  24
Reason Codes  29
   Add and Maintain Reason Codes  29
   Update Reason Codes  30
   Delete Reason Codes  30
Teams  30
   Add and Maintain Teams  31
   Update Teams  32
   Delete Teams  32

CHAPTER 4

Manage Queues  33
   Attributes  33
      Add and Maintain Attributes  34
      Edit Attributes  34
      Delete Attributes  35
   Precision Queues  35
      Add and Maintain Precision Queues  36
      Build Precision Queue Steps  39
      Consider If Formula for Precision Queue  41
      Edit Precision Queues  42
      Delete Precision Queues  42
      Skill groups or Precision Queues?  43
   Skill Groups  44
      Add and Maintain Skill Groups  44
CHAPTER 5

Manage Calls  49
  Bucket Intervals  49
    Add and Maintain Bucket Intervals  50
  Call Types  51
    Add and Maintain Call Types  51
  Dialed Numbers  53
    Add and Maintain Dialed Numbers  53
  Expanded Call Variables  55
    Add and Maintain Expanded Call Variables  56
    Sizing Expanded Call Variables  58
  Network VRU Scripts  58
    Add and Maintain Network VRU Scripts  59
    VRU Script Name Parameters  60
    Sample VRU Script Names  62
    Configuration Parameters  62
    Sample Configuration Values  66

CHAPTER 6

Bulk Jobs  67
  Manage Bulk Jobs  67
    Bulk Job Content Files  68
      Content File Rules  68
    Bulk Agent Content Plan  69
    Bulk Dialed Number Content Plan  71
  Add and Maintain Bulk Jobs  72
  Review Bulk Job Details  73

CHAPTER 7

Settings  75
  Configure Deployment  75
    Deployment Types  76
  Capacity Info  76
  System Validation  77
  Agent Trace  77
Packaged CCE Logs 78

CHAPTER 8

Configuration Manager 81
   Access Configuration Manager 81
   Configuration Manager Tools for Call Flow 82
      PG Explorer 82
      Change Default Desk Settings 83
   Agent Targeting Rule 84
   Configuration Manager Tools for Global Values and Actions 85
      System Information 85
      Permanent Deletion 87
   Configuration Manager Tools for Multichannel 87
      Media Class List 88
      Media Routing Domain List 88
      Application Instance List 89
      Application Path List 90
   Configuration Manager Tools for Scripting 90
      Script Reference 90
      User Variable List 91
   Configuration Manager Tools to Control User Access 91
      Feature Control Set List 91
      User List Tool 92

CHAPTER 9

Scripting with Packaged CCE 95
   Common Tasks 95
      The Palette 95
      General Tab 96
      Routing Tab 96
      Targets Tab 96
      Queue Tab 96
   Create Routing Script 97
   Add Comments to a Node 98
   Specify a Connection Label Location for a Node 98
   Validate Scripts 99
   Open Script Explorer 99
Schedule Routing Script 100
Viewing Modes 102
Making Packaged CCE Work with Unified CVP 103
Call Types, Contact Data, and Scripting 104
  Call Types 104
  Default Call Types 104
  Relation between Call Types and Scripts 104
  Call Type Qualifiers 104
    Dialed Number (DN) 105
  Association of Contacts with Call Types in Packaged CCE 105
  Determination of Call Type for Voice Contact 105
  Determination of Call Type for E-mail or Web Request 105
Contact Categorization 106
  Categorization and Call Type 106
    Categorization through Scheduling Scripts by Call Type 106
    Change Call Type to Static 106
    Change Call Type to Dynamic 107
    Change Call Type and Execute a New Script 109
  Categorization by Call Type Qualifiers 110
    Categorize Contact by Dialed Number 110
  Categorization by Time and Date 111
    Categorize Contact by Date and Time 112
    Categorize Contact by the Day of Week 114
  Categorization by Branching 115
    Execute a Different Script 115
    Direct Script Execution to Different Branches by Percentage 116
    Categorize Contact Based on a Condition 117
    Categorize a Contact Based on its Media Routing Domain 118
Selection of Routing Targets 118
  Routing Targets 118
  Agent Routing Nodes 119
  Transfer Calls from Agents to Agents 119
  Nodes Used to Stop Script Processing 121
    End Node 121
    Release Call Node 121
Target Requery   122
  Target Requery Functionality   122
  Test of the Requery Status Variable   122
  Nodes that Support Target Requery   123
  Use Target Requery   123

Network VRUs   124
  VRU Functionality   124
  Access to VRU Scripts in Unified ICM Scripts   124
  Send Call to a VRU with Send to VRU Node   124
  Run External Scripts   125
  VRU Errors   126

Call Queuing at VRUs   127
  Place a Call in Queue   128

Precision Queue Script Node   130
  Precision Queue Properties Dialog Box - Static Precision Queue   131
  Precision Queue Properties Dialog Box - Dynamic Precision Queue   132
  Queuing Behavior of Precision Queue Node   133

Adjust Priority of a Call in a Queue   133
Remove Call from a Queue   134
Temporarily Halt Script Execution   135

Multichannel Routing   136
  Overview of Multichannel Services   136
    Supported Route Requests   136
    Application Request Routing   136
    Synchronized Agents and Skill Groups   136
    Independent Media Queues   137

Universal Queue   137
  Universal Queue Functionality   137
  Universal Queue Configuration Overview   137

Multichannel scripts and media routing domains   137
  Media Routing Domains   137
  Media Routing Domains and Interruptibility   137
  Use Media Routing Domains to Categorize Contacts   138

Queue to Agent Node   139
Change Queue to Agent Type   140
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify an Agent Directly</td>
<td>141</td>
</tr>
<tr>
<td>Use of Formulas</td>
<td>142</td>
</tr>
<tr>
<td>Formula Usage</td>
<td>142</td>
</tr>
<tr>
<td>Formula Example</td>
<td>142</td>
</tr>
<tr>
<td>Variables</td>
<td>142</td>
</tr>
<tr>
<td>Variable Usage</td>
<td>142</td>
</tr>
<tr>
<td>Variable Syntax</td>
<td>142</td>
</tr>
<tr>
<td>Single-Target Variables</td>
<td>143</td>
</tr>
<tr>
<td>Multiple-Target Variables</td>
<td>143</td>
</tr>
<tr>
<td>Call Control Variables</td>
<td>143</td>
</tr>
<tr>
<td>Expanded Call Variables</td>
<td>145</td>
</tr>
<tr>
<td>Persistent vs. Non-Persistent Call Variables</td>
<td>145</td>
</tr>
<tr>
<td>User Variables</td>
<td>145</td>
</tr>
<tr>
<td>Set Variable Node Usage</td>
<td>146</td>
</tr>
<tr>
<td>SkillGroup.Avail and SkillGroup.ICMAvailable Variables</td>
<td>147</td>
</tr>
<tr>
<td>SkillGroup.ICMAvailable Variable</td>
<td>147</td>
</tr>
<tr>
<td>SkillGroup.Avail Variable</td>
<td>147</td>
</tr>
<tr>
<td>Closed Variables</td>
<td>149</td>
</tr>
<tr>
<td>Operator Precedence</td>
<td>149</td>
</tr>
<tr>
<td>Operators</td>
<td>150</td>
</tr>
<tr>
<td>Prefix Operators</td>
<td>150</td>
</tr>
<tr>
<td>Arithmetic Operators</td>
<td>150</td>
</tr>
<tr>
<td>Equality Operators</td>
<td>151</td>
</tr>
<tr>
<td>Relational Operators</td>
<td>151</td>
</tr>
<tr>
<td>Logical Operators</td>
<td>152</td>
</tr>
<tr>
<td>Bitwise Operators</td>
<td>152</td>
</tr>
<tr>
<td>Miscellaneous Operators</td>
<td>153</td>
</tr>
<tr>
<td>Built-in functions</td>
<td>154</td>
</tr>
<tr>
<td>Date and Time Functions</td>
<td>154</td>
</tr>
<tr>
<td>Mathematical Functions</td>
<td>157</td>
</tr>
<tr>
<td>Miscellaneous Functions</td>
<td>157</td>
</tr>
<tr>
<td>Custom Functions</td>
<td>159</td>
</tr>
<tr>
<td>Add Custom Functions</td>
<td>159</td>
</tr>
<tr>
<td>Import Custom Functions</td>
<td>160</td>
</tr>
<tr>
<td>Export Custom Functions</td>
<td>161</td>
</tr>
</tbody>
</table>
Scripts in a Packaged CCE Environment 161

Call Priority 161
Check for Available Agents 162
Queue to Skill Group Node 162
Cancel Queuing Node 162
End Node 162
Agent to Agent Node 162
CVP as a Queue Point 162

Interruptible vs. Non-Interruptible 163

Utility nodes 163
Start Node 163
Comment Node 164
Line Connector Node 164

CVP Scripting 165

Writing Scripts for Unified CVP 165

Before You Begin 166

Scripts to Access Unified CVP from Packaged CCE 166

Packaged CCE Scripting 166

Unified CVP Call Studio Scripting 167

Scripting for Unified CVP with Packaged CCE 167

Micro-Applications 167

Simple Example Script: Welcome to XYZ Corporation 168
Packaged CCE Unified CVP Micro-App Connection 170
Information Exchange Between Packaged CCE and Unified CVP 172
Packaged CCE Data Handling 172
Unified CVP Script Error Checking 173

Writing Packaged CCE Applications for Unified CVP 177

Configure a Unified CVP Network VRU Script 178
Run External Script Node That Accesses a Unified CVP Micro-Application 179

Unified CVP Micro-Applications 179

Dynamic Audio File Support for Micro-Applications 179
Default Media Server for Micro-Applications 180
Capture Micro-Application 181
Play Media Micro-Application 182

Configure Network VRU Script for Play Media 182
Contents

How Unified CVP Handles Barge-In 183
Configure Play Media Micro-Application to Use Streaming Audio 184
Configure Custom Streaming Ringtones 186
Play Media Examples: Play Welcome Message 187

Play Data Micro-Application 188
Play Data and Data Storage 188
Configure Network VRU Script Settings for Play Data Micro-Application 188
Play Back Types for Voice Data 190
System Media Files 193
Play Data Configuration Examples 210

Get Digits Micro-Application 211
Configure Network VRU Script Settings for Get Digits Micro-Application 211
Get Digits Configuration Examples 213
Get Speech and External Voice XML 214
Get Digits and Digit Entry Completion 215

Menu Micro-Application 216
Configure Network VRU Script Settings for the Menu Micro-Application 216
Menu Configuration Examples 218
Menu and Digit Entry Completion 219

Get Speech Micro-Application 220
Configure Network VRU Script Settings for the Get Speech Micro-Application 220
Passing Information to the Call Studio Scripts Executing on VXML Server 221
Passing Data Back to Packaged CCE from the VXML Server 222

Scripting for Unified CVP with Call Studio 222
About Call Studio 223
High-Level Configuration Instructions 224
Call Studio ReqICMLabel Element to Pass Data 225
Integrate Call Studio Scripts with Unified CCE Scripts - Traditional Method 228
Integrate Call Studio Scripts with Packaged CCE Scripts 228

Outbound Option Scripting 228
Outbound Option Administrative Scripts 229
Outbound Option Agent Routing Scripts 229
Call Type Reporting and Outbound Option Campaigns 229

Call Studio Scripts in Unified CVP 229
Deploy Call Studio Scripts Using Call Studio 230
CHAPTER 10 Reporting

- Reporting Data 231
  - Real Time Data Collection 231
  - Historical and Interval Data Collection 232
- Configuration Data 233
- Call Detail Data 234
- Event Data 235
- Database Tables that Hold Reporting Data 235
  - Real Time and Historical Reports 235
  - Interval Boundaries 236
- Reporting entities and concepts 236
  - Agent States 236
    - How Agent States Are Calculated in Reports 238
    - Agent States, Skill Groups, and Precision Queues 239
    - Agent State and Task State Relationship 239
    - Reports that Show Agent States 240
  - Agent Logout Reason Codes 240
    - Reports that Show Agent Logout Reason Codes 241
  - Agent Not Ready Reason Codes 241
    - Reports that Show Agent Not Ready Reason Codes 243
  - Agent Task Handling 243
    - Types of Tasks 243
    - Task Times 246
    - Reports that Show Agent Task Handling 246
  - Agent Utilization: Full-Time Equivalents and Percent Utilization 247
    - Reports that Show Percent Utilization and FTE Metrics 247
    - Reports that Show Agent States 248
    - Reports that Show Agent Logout Reason Codes 248
    - Reporting on Agents 248
    - Skill Groups 248
      - Default Skill Group 249
        - Role of Default Skill Group in Reporting 249
How New Calls Increment Default Skill Group Statistics 249
How Agent-to-Agent Dialing Increments Default Skill Group Statistics 250
How Transferred and Conferenced Calls Increment Default Skill Group Statistics 250
Reports on Skill Group Operations 250
How Calls Offered is Calculated for Skill Group 250
Reports that Show Skill Group Operations 251
Agent Teams and Supervisors 251
Agent Teams and Supervisors 251
Agent Team Reports 251
Supervisor Action Reports 251
Supervisor Assist and Emergency Assist for Existing Call 251
Barge-In 252
Intercept 253
Reports that Show Information on Agent Teams 253
Reporting on Supervisors 253
Reports that Show Information on Agent Teams 253
Average Speed of Answer 254
How ASA is Calculated 254
ASA for Agent and Skill group 255
Reports That Show ASA for Agents 255
ASA for Call Type 255
Reports That Show ASA for Call Type 255
Requery 256
Call Types 256
Call Type Reports 256
Call Types 257
Best Practices for Call Types 257
Changing Call Types 258
Call Type Reporting 259
How Calls Offered is Calculated for Call Type 260
How Call Errors Affect Call Type Reporting 260
How Calls that Experience CVP Ring No Answer Affect Call Type Reporting 261
How Calls that Terminate Label Node and Route to Non-Monitored Devices Affect Reporting 262
Data flow for Reporting 279
Recovery and Replication 279
Possible Points of Delay or Inconsistency 279
Methods to Prevent Data Loss from Logger and HDS Failure 280
Data Loss from PIM Failure and Reporting 281
Other Possible Points of Failover 281
Peripheral Gateway / CTI Manager Service Failover 281
Agent Desktop / CTI OS Server Failover 282
Call type and Skill Group Metrics 282
Call Type Real Time Table Metrics 282
Network VRU and Skill Group Metrics 284
Reporting on Skill Groups 285
Reporting on Precision Queues 285
Preface

• Audience, page xvii
• Purpose, page xvii
• Context, page xvii

Audience

This guide is prepared for:

• Contact center administrators who configure and run the contact center, provide tools for the agents and supervisors, run reports that show contact center metrics, address operational issues, and maintain call routing scripts.

• Contact center supervisors, who lead agent teams and are responsible for team performance. Supervisors can re-skill and can change attributes for agents on their teams and can add and remove agents from teams. Supervisors can view supervisory reports to ensure that Service Level Agreement is being met and can review agent metrics such as login time and time spent on calls.

Purpose

This guide introduces the Packaged CCE web administration interface and explains each tool on that interface. There is additional information on functionality—reporting, scripting, and legacy Configuration Manager—that is relevant and available to Packaged CCE.

Context

This guide is written with the understanding that Packaged CCE has been deployed by a partner or service provider who has validated its deployment type and verified that your contact center can receive and send calls and has functional virtual machines and a fully operational database.

There is no assumption that administrators or supervisors have expertise in VMWare or with Cisco contact center applications.

Useful links for Packaged CCE:
<table>
<thead>
<tr>
<th>Context</th>
<th>Link</th>
</tr>
</thead>
</table>
About Cisco Packaged Contact Center Enterprise

Cisco Packaged Contact Center Enterprise (Packaged CCE) delivers Cisco Contact Center Enterprise applications as a pre-defined, bounded, and virtualized solution.

Core components are configured in a web interface, and contact center functionality—such as call processing, prompts and rich VXML scripting, voice response collection, agent selection, queuing, and reporting—occur in a controlled environment with well-defined configuration and deployment boundaries.

- Packaged CCE Components, page 1
- System Deployment, page 3

Packaged CCE Components

Cisco Systems, Inc. delivers Packaged CCE on a pair of core duplexed Unified Computing System (UCS) C260-Series servers, referred to as Side A and Side B. Five Cisco Systems contact center components run on these servers as 17 virtual machines (VMs). These components are summarized below.

**Cisco Unified Contact Center Enterprise (Unified CCE)**

Unified CCE is installed as four VMs: two CCE Call Servers (Side A and Side B) and two CCE Data Servers (Side A and Side B).

The CCE Call Server routes calls, hosts the two Peripheral Gateways, and acts as the CTI Object server that interfaces with the agent desktops.

The CCE Data Server functions as the Logger, which is the database server that stores contact center configuration data and historical reporting data for distribution to the data servers. It also functions as the Administration and RealTime Data Server. You can access Configuration Manager from the console of this VM.

**Cisco Unified Customer Voice Portal (Unified CVP)—Call/VXML, OAMP, and Reporting servers**

Unified CVP is an intelligent voice and video self-service product that enables efficient retrieval of contact center information. It provides prompting, collecting, queuing, and call control. CVP is installed as follows:

- Four Unified CVP Call/VXML servers (two on Side A and two on Side B), which provide the call routing and ACD components.
• One Unified CVP OAMP server on Side A, which provides a web-based operations console for maintaining the Unified CVP cluster.

• One or two optional Unified CVP Reporting servers that collect reporting data from Unified CVP operations and is accessed by Unified Intelligence Center reporting templates.

**Cisco Unified Communications Manager (Unified Communications Manager)**

Unified Communications Manager is an enterprise-class IP communications processing system. In addition to traditional telephony features, it provides advanced capabilities, such as mobility, and full-featured conferencing services.

Unified Communications Manager is installed as three VMs—a Publisher on Side A which stores the read/write database and a Subscriber on both Side A and Side B which register phones and gateways.

**Cisco Unified Intelligence Center (Unified Intelligence Center)**

Unified Intelligence Center is an advanced reporting platform that provides real-time and historical reporting. Unified Intelligence Center offers a dashboard-based canvas for grouping multiple reporting objects together, offering a comprehensive view of contact center statistics at a glance.

Unified Intelligence Center is installed as two VMs—a Publisher on Side A and a Subscriber on Side B.

**Cisco Finesse (Finesse)**

Finesse is a next-generation agent and supervisor desktop designed to provide a collaborative experience for the various communities that interact with your customer service organization. It helps improve customer experience as well as customer care representative satisfaction.

Finesse is installed as two VMs—a Primary node on Side A and a Secondary node on Side B.
System Deployment

All installed and optional features depicted in the illustration above have been fully qualified for Packaged CCE and are documented in the Optional Features guide at http://www.cisco.com/en/US/products/ps12586/prod_maintenance_guides_list.html.
System Deployment

About Cisco Packaged Contact Center Enterprise
Basics

- Sign In, page 5
- Menus, page 7
- Common Tools and Tasks, page 10
- Configuration in Packaged CCE, page 13

Sign In

To sign in to Unified CCE Web Administration, point to https://<IP Address>/cceadmin, where <IP Address> is the address of either CCE data server (the one on Side A or the one on Side B) or the address of the optional, additional AW-HDS-DDS, if you have deployed one.

The Sign-In window has fields for:

<table>
<thead>
<tr>
<th>Role</th>
<th>From the drop-down list, select Administrator or Supervisor.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>The domain drop-down list shows all domains that are configured, as well as any Alternate User Principal Name (UPN) Suffixes that have been defined.</td>
</tr>
<tr>
<td>Username</td>
<td>Administrators sign in with their Active Directory UserName. Supervisors sign in with the agent username configured in the General tab on the Agent tool.</td>
</tr>
<tr>
<td>Password</td>
<td>Administrators sign in with their Active Directory password. Supervisors sign in with the password configured in the General tab on the Agent tool.</td>
</tr>
</tbody>
</table>

The administrator's initial sign-in opens to the Home menu.
The supervisor's initial sign-in opens to the Manage menu.
If the Language Pack has been installed, the Sign-In window has a drop-down menu for language selection.
User Access

Administrators who are newly created in the Packaged CCE Unified CCE Administration have both agent credentials and supervisor Active Directory credentials but are not in an AD security group. They must use their supervisor AD credentials to run reports on the teams they supervise. They must use their agent credentials to sign in to, and they have limited access to, tools on the Manage menu.

Agents have no AD credentials and therefore no access to the Packaged CCE administration tools or to Unified CCE Configuration Manager.

User Interface and Navigation

What you see after a successful sign-in depends on your user role (Administrator or Supervisor), on the deployment type, and also on whether any features have been made unavailable or display-only for your credentials.

Administrators see:

- A title bar above the menu that shows your username and offers a Sign Out option. The system automatically signs you out after 30 minutes of inactivity.
- A menu bar with three menus: Home, Manage, and Settings.

Supervisors see only one menu—Manage—and have restricted access to the tools on that menu.

Supervisor Interface and Navigation

Supervisors sign in to an interface with one menu, Manage.

Supervisors are restricted as follows for the Manage tools:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agents</td>
<td>On the Agent List window, supervisors can see and edit agents who are on teams they supervise. General tab: Supervisors can edit the Password and Re-enter password fields. Other fields are protected. Attributes tab: Supervisors can add, modify, and remove attributes for agents on teams they supervise. Skill Groups tab: Supervisors can add and remove the agent's membership in skill groups and can change the agent's default skill group. Supervisor tab: Not available for supervisors.</td>
</tr>
<tr>
<td>Attributes</td>
<td>Read-only</td>
</tr>
<tr>
<td>Precision Queues</td>
<td>Read-only</td>
</tr>
</tbody>
</table>
### Permissions

<table>
<thead>
<tr>
<th>Tool</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill Groups</td>
<td>On the Skill Group List window, supervisors can see and edit all skill groups. Supervisors cannot add or delete skill groups. General tab: fields are protected. Members tab: supervisors can add and remove skill groups for agents who are on teams they supervise.</td>
</tr>
<tr>
<td>Teams</td>
<td>Read-only</td>
</tr>
</tbody>
</table>

**Related Topics**

Common Tools and Tasks, on page 10

---

### Languages

After the Language Pack is installed, the Sign-In window includes a Language drop-down menu. The drop-down includes more than a dozen languages.

English is the initial and the default language. Pick any other language to see the user interface and the online help in that language. The system retains your choice for subsequent sign-ins until you change it again.

### Menus

Your access to menus and to the options within menus is based on whether you are a Supervisor or an Administrator. Menus are:

**Home**

The Home window shows the deployment type that was selected at the initial sign-in and the software version of Unified CCE that is currently in use.

**Manage**

The tools you use most frequently are organized under the Manage menu, which opens to three submenus: *Home*, *Manage*, and *Other*.

**System**

The System menu has these submenus: *Congestion Control* and *Agent Trace*.

---

### Home

The Home menu has three tabs: *General*, *Capacity Info*, and *System Validation*. Signing in puts the administrator on the General tab, which shows the current deployment type and software version for your system.
The Capacity Info and System Validation tabs track metrics and validation rules and show how your system meets, exceeds, or violates them. These tabs provide important diagnostic and monitoring information.

Manage

The tools you use most frequently are organized under the Manage menu, which opens to three submenus—Agent, Call, and Other. Each submenu has links that open tools where you create and maintain essential configuration for call center operation. Click a tool to open its interface.

Depending on your deployment type and the feature control set assigned to your credentials, you may not see all tools, or some tools may be display-only for you. There is an API for each tool.

Agent

Use these tools to configure your workforce and its areas of expertise.
Agents, on page 15
Use this tool to create and maintain agents, assign attributes to them, add them to skill groups, and establish them as supervisors.

Attributes, on page 33
Open this tool to create and maintain attributes that can be assigned to agents.

Desk Settings, on page 22
Open this tool to create and maintain desk settings that can be assigned to agents.

Precision Queues, on page 35
Open this tool to create and maintain precision queues for which agents can become eligible. Calls are routed to an agent in a precision queue, provided that agent's attributes qualifies her for that queue.

Reason Codes, on page 29
Open this tool to create and maintain reason codes that agents enter on their desktops when they go idle (or "Not Ready").

Skill Groups, on page 44
Open this tool to create and maintain skill groups and to add agents as members of a skill group.

Teams, on page 30
Open this tool to create teams, to add agents to a team, and to assign supervisors for a team.

Bucket Intervals, on page 49
Open this tool to define "buckets" or time intervals that segment call-handling activity. Bucket intervals can be associated with call types, skill groups, and precision queues and are used in a number of Cisco Unified Intelligence Center reports.

Add and Maintain Call Types, on page 51
Open this tool to create and maintain categories ("types") for calls. For example, you might want a call type for calls that require supervisor assistance.

Dialed Numbers, on page 53
Open this tool to create and maintain dialed number scripts that will be used to select routing scripts to deliver calls to agents.

Expanded Call Variables, on page 55
Open this tool to create and maintain call variable data that is delivered with a call and available to the agents who receive the call.

Call
Use these tools to configure the way calls are organized, delivered, and reported on.
<table>
<thead>
<tr>
<th>Tool</th>
<th>Use</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network VRU Scripts</td>
<td>Open this tool to create and maintain scripts that direct Unified Customer Voice Portal on the call delivery.</td>
<td>Network VRU Scripts, on page 58</td>
</tr>
</tbody>
</table>

**Other**

<table>
<thead>
<tr>
<th>Tool</th>
<th>Use</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulk Jobs</td>
<td>This is the only tool under <em>Other</em>. Open this tool to create bulk jobs that will create and edit multiple agent and dialed number records in a single operation.</td>
<td>Bulk Jobs, on page 67</td>
</tr>
</tbody>
</table>

**Settings**

This menu has two selections: *Configure Deployment* and *Agent Trace*.

**Common Tools and Tasks**

Unified CCE Web Administration windows share a consistent interface for common actions:
Lists

List Windows
All tools open to a List window that has rows for all currently-configured objects. The Teams tool has a list with a row for each team. The Call Types tool has a list for each call type. And so forth. List windows have a standardized interface. You can search, sort, edit, and delete from the List.

Permissions on list windows vary for administrators and supervisors and are noted in the topic for each tool.

Add Objects

Click New on a list window to open an Add window where you can complete fields to create and save a new object.

Search a List

Use the Search field at the top right of the window to locate objects in a list. Enter all or part of the Name (or name-equivalent, such as DialedNumberString or Text) or Description field to find matches. Clear the search by deleting text from the Search field.

The Search field in the Agents tool offers an advanced and flexible search. Click + to open at the far right of the Search field to open a pop-up window where you can:

- Select to search for agents only, supervisors only, or both.
- Enter a Username, Agent ID, First or Last name, or a Description to search for that string.
- Enter one or more team names separated by spaces. (Team is an OR search - the agent or supervisor must be a member of one of the teams.)
- Enter one or more attribute names separated by spaces. (Attributes is an AND search - the agent or supervisor must have all attributes.)
- Enter one or more skill group names separated by spaces. (Skill Groups is an AND search.)

This illustration shows a search for agents who are on either Team xyz or Team robots and who are members of Skill Group sg1. Three agents meet those criteria.
Sort a List

If a column on a list window has an arrow icon in the column header, click the arrow to sort in ascending or descending order.

Popups

Popup Selection

Many Add and Edit windows have popups for searching and choosing items relevant to that tool.

Some popups are "search-and-select," allowing you to chose one item. Other popups are "search-and-add," allowing you to select multiple items. For example, because an agent can be on only one team, the search-and-add popup for adding an agent to a team allows only one selection. The search-and-select Skill Group Members popup allows you to select one or more agents to add to the skill group.

Click the magnifying glass icon to open the popup, where you can locate and select items that have been configured.

Keyboard Shortcuts

Press the question mark (?) key to open a window that shows keyboard shortcuts that are appropriate for that tool and for your status (Supervisor or Administrator).

Tip

The keyboard shortcuts window does not open when you press the (?) key in a text field.
Native Character Sets

If you have installed the Language Pack, the Sign-In window includes a Language drop-down menu. The drop-down includes more than a dozen languages. Pick any one of them to see the Unified CCE Web Administration interface and online help in that language.

Regardless of the language you select, for the optional Description field in all tools and the required First Name and Last Name fields in the Agent tool, you must enter characters that are recognized by the database. If you do not, you see an error stating “The system does not support these characters”.

Configuration in Packaged CCE

Although the tools in the Unified CCE Administration Manage menu are sufficient for basic configuration for your contact center, you need to access the tools in Unified CCE Configuration Manager on the CCE Data Server for some configuration. See Configuration Manager, on page 81
Basics

Configuration in Packaged CCE
CHAPTER 3

Manage Agents

Administrators have full access to all tools documented in this chapter.
Supervisors have limited access to Agents, display-only access to Teams, and no access to Supervisors, Desk Settings, or Reason Codes.

- Agents, page 15
- Supervisors, page 20
- Desk Settings, page 22
- Agent Desktops, page 24
- Reason Codes, page 29
- Teams, page 30

Agents

Agents respond to contacts from customers. These contacts are often telephone calls, but a customer might have selected a "Click to Chat" button on a web page and need to interact with an agent by typing in the chat window. A customer might also have sent an email that needs a reply from an agent. These requests to answer a phone call, join a chat, or reply to an email are delivered to the first available agent who has the expertise to help the customer.

You can configure the types of contacts that are routed to an agent. For example, if an agent is a member of a skill group that is set up for the Cisco_Voice routing domain only, that agent is a "voice-only" agent for that skill group. And if an agent is a member of a skill group that is set up for a non-voice routing domain, that agent is a multichannel agent for that skill group.

Agents:

- Can be located at a call center site or designated as mobile agents who work elsewhere—perhaps from a home office.

Can be tagged as supervisors. An agent with supervisor status can oversee as many as 10 agent teams, can view reports that monitor activities of the agents on those teams, and can join and participate in agent/customer calls. Supervisors work from a supervisor desktop.

Can be assigned to one team. Teams are organizational units that reflect the reporting structure in a contact center.

Are each associated with one Desk Settings, either the current default Desk Settings or another Desk Settings. Desk settings are a set of permissions or characteristics that control the features agents can see and use while they are interacting with customers.

Can be assigned attributes that indicate their proficiency—perhaps expertise in a certain language or technology.

Can be assigned to skill groups.

Work from an agent desktop. Packaged CCE supports either (or both) Finesse or CTI OS toolkit as the agent desktop tool.

Navigate to **Unified CCE Administrator Manage > Agent > Agents** interface to view the Agent list. Administrators can see and maintain all agents. Supervisors see a list of agents who are on teams they supervise.

### Related Topics

- Attributes, on page 33
- Desk Settings, on page 22
- Skill Groups, on page 44
- Teams, on page 30
- Common Tools and Tasks, on page 10

### Add and Maintain Agents

**Tip**
You can add agents one at a time from the *List of Agents* window. You can also run *Manage Bulk Jobs*, on page 67 to add or edit multiple agent records.

**Procedure**

**Step 1** Navigate to **Unified CCE Administrator Manage > Agent > Agents**.

**Step 2** Click **New** to open the *New Agent* window.

This window has four tabs: General, Attributes, Skill Groups, and Supervisor. You cannot save the agent until you have entered all required fields on the General tab. You can complete other tabs as needed and in any order.

**Step 3** Complete the fields on the General tab:
<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>yes</td>
<td>Enter up to 32 alphanumeric characters as the username for this agent.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Remember</strong> An agent who is designated as a supervisor signs in to Unified CCE Web Administration with this username. If username was selected as the login option during the CTI OS Agent Desktop installation, this field is used for signing in to CTI OS Agent Desktop.</td>
</tr>
<tr>
<td>First Name</td>
<td>yes</td>
<td>See Native Character Sets, on page 13.</td>
</tr>
<tr>
<td>Last Name</td>
<td>yes</td>
<td>See Native Character Sets, on page 13.</td>
</tr>
<tr>
<td>Agent ID</td>
<td>yes</td>
<td>Enter a unique string of 11 digits. If you leave this field blank, Packaged CCE auto-generates a 7-digit agent ID, which you can later edit. The agent uses the Agent ID to sign in to the CTI OS Agent Desktop or to Finesse.</td>
</tr>
<tr>
<td>Description</td>
<td>no</td>
<td>Enter a Description of the agent. See Native Character Sets, on page 13.</td>
</tr>
<tr>
<td>Desk Settings</td>
<td>yes</td>
<td>The Desk Settings field defaults to show the current system-default. To change it, click the magnifying glass to display the Select Desk Settings list where you can select a different desk settings.</td>
</tr>
<tr>
<td>Team</td>
<td>no</td>
<td>The Team field defaults to <em>None</em> or to the current team for this agent. To change it, click the magnifying glass to display the Select Team list. Select a team.</td>
</tr>
<tr>
<td>Login Enabled</td>
<td>no</td>
<td>Checked by default. Uncheck it only if you do not want this agent to be able to sign in.</td>
</tr>
<tr>
<td>Set Password</td>
<td>no</td>
<td>Checked by default. Uncheck it if you do not want to create a password for this agent.</td>
</tr>
<tr>
<td>Enter Password</td>
<td>no</td>
<td>Enter and re-enter a maximum of 32 characters to establish and confirm a password for this agent</td>
</tr>
<tr>
<td>Re-enter Password</td>
<td>no</td>
<td>Tip An agent who is designated as a supervisor signs in to Unified CCE Web Administration with this password.</td>
</tr>
</tbody>
</table>
Step 4 When you complete General tab information, click **Save** to return to the List window, where a message confirms the successful creation of the agent. To enter fields in the other tabs, click those tabs.

Step 5 Complete the Attributes tab:
This tab shows the attributes associated with this agent and their current values. If the agent has no attributes, the **Name** field shows *No Items Found* and *No Items*.

Click the magnifying glass to open a popup list of all attributes, showing the name and current default value for each. You can:

a) Click the attributes you want to add for this agent, up to a maximum of 50.
b) Set the attribute value as appropriate for this agent.
c) Click **Save** to return to the List window, where a message confirms the successful creation of the agent.

To enter or change fields in the other tabs, click those tabs.

Step 6 Complete the Skill Groups tab:
This tab shows the skill group membership for this agent. If the agent is not a member of any skill groups, the List of Skill Groups shows *No Items Found* and *No Items*.

a) Click the magnifying glass to open a popup list of all configured skill groups. Then click the skill groups you want to add for this agent or supervisor.
b) Click **Save** to return to the List window, where a message confirms the successful creation of the agent.

To enter or change fields in the other tabs, click those tabs.

Step 7 Complete the Supervisor tab:

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is Supervisor</td>
<td>no</td>
<td>Check to configure this agent as a Supervisor.</td>
</tr>
<tr>
<td>Domain</td>
<td>no</td>
<td>Enabled only when Is Supervisor is checked. From the drop-down, select the domain for this supervisor.</td>
</tr>
<tr>
<td>Username</td>
<td>yes</td>
<td>Enabled only when Is Supervisor is checked and required when that is the case. Enter the Active Directory username for this supervisor.</td>
</tr>
<tr>
<td>List of Supervised Teams</td>
<td>no</td>
<td>Enabled only when Is Supervisor is checked. Shows the team associated with this supervisor. To select a team, click the magnifying glass to display the Add Supervised Teams list. Then click a link to select a team and close the list.</td>
</tr>
</tbody>
</table>
Step 8  Click **Save** to return to the List, where a message confirms the successful creation of the agent.

---

### Update Agents

Administrators have full permission to edit agents. Supervisors can edit the password fields on the General tab, all fields on the Attributes tab, and all fields on the Skill Groups tab.

To edit an agent, click the username on the *List of Agents* window to open the *Edit Agent <Name>* window.

**Procedure**

**Step 1**  Edit fields as needed.

**Step 2**  Check **Change Password** to clear the Password and Re-enter Password fields to reset the password.

**Step 3**  Click **Cancel** to retain existing agent configuration, or click **Save** to make the changes and return to the *List of Agents*.

---

### Delete Agents

Administrators can delete agents. Supervisors cannot.

You cannot delete an agent who is referenced by a script or who is configured as a supervisor and assigned to an agent team.

**Procedure**

**Step 1**  To delete an agent, open the List of Agents window and click the trash can icon for that Agent ID.

**Step 2**  Respond to the confirmation message.

This action marks the agent for deletion. You must permanently delete the agent from Configuration Manager.

---

**Related Topics**

- Permanent Deletion, on page 87

---

### Reporting on Agents

These are some Cisco Unified Intelligence Center reports relevant to agents.

**Related Topics**

- Agent Team Reports, on page 251
- Reports that Show Agent States, on page 240
Supervisors

Agents can be configured to have supervisor status for voice contacts.

**Remember**

An agent to whom you confer supervisor status must already exist in Active Directory.

Supervisors have both agent credentials and supervisor Active Directory credentials. They enter their agent credentials to sign in to Unified CCE Administration, where they have limited access to tools on the Manage menu. They must use their supervisor AD credentials to run reports on the teams they supervise.

A supervisor can supervise multiple teams and can be both a supervisor and a member of a team. Supervisors are also reporting users.

With Supervisor status, agents can:

- Generate reports and view data for the teams they supervise and the agents on those teams.
- Use a Supervisor desktop to barge-in, intercept, silently monitor, and log out agents Packaged CCE supports both the CTI OS and Finesse desktops, both of which have a supervisor desktop. (Not all supervisor features are supported on both desktops.)
- Join an agent/customer call to assist on a consultative or emergency basis
  The agent's ability to request supervisor assistance is a setting on the Desk Settings.
- Reskill agents who are on teams they supervise.

To configure supervisors in Unified CCE Administration, go to **Unified CCE Administrator Manage > Agent > Agents**. Select an agent and click the Supervisor tab.

### Add and Maintain Supervisors

In Unified CCE Administration, go to **Unified CCE Administrator Manage > Agent > Agents**.

**Procedure**

1. **Step 1** Create a new agent or edit an existing agent. See [Add and Maintain Agents](#), on page 16.
2. **Step 2** Click the Supervisor tab.
3. **Step 3** Check **Is Supervisor** to configure this agent as a Supervisor.
4. **Step 4** From the Supervisor Domain drop-down, select the domain for this supervisor.
5. **Step 5** Enter the unique active directory username for this supervisor.
6. **Step 6** Select the teams for this supervisor:
   a) Click the magnifying glass next to List of Supervised Teams to open **Add Supervised Teams**.
b) Click the team name link to add the team.

**Step 7**  
Click **Cancel** to retain the existing configuration, or click **Save** to create the supervisor.  
You return to the *List of Agents*.

---

## Update Supervisors

To edit a supervisor, click the username on the *List of Agents* window to open the Edit window. Then click the Supervisor tab.

All fields on this window are editable—with the exception of Supervisor Domain.

### Procedure

1. **Step 1**  
   Click the username on the *List of Agents* window to open the Edit window.

2. **Step 2**  
   Then click the Supervisor tab.

3. **Step 3**  
   Modify fields as needed.

4. **Step 4**  
   Check **Change Password** to clear the *Password* and *Re-enter Password* fields to reset the password.

5. **Step 5**  
   Click **Cancel** to retain existing agent configuration, or click **Save** to make the changes and return to the *List of Agents*.

---

## Delete Supervisors

You cannot delete a supervisor who is referenced by a script or who is assigned to an agent team.

### Procedure

1. **Step 1**  
   Open the *List of Agents* window and click the trash can icon for the supervisor's username.

2. **Step 2**  
   Respond to the confirmation message.  
   This action marks the supervisor for deletion. You must permanently delete from Configuration Manager.

---

### Related Topics

- Permanent Deletion, on page 87
Desk Settings

All agents have a desk settings configuration. A desk setting is a collection of permissions or characteristics for the agent, such as how and when calls to the agent are redirected, how and when the agent enters various work states, and how requests to the supervisor are handled. The agent desktop application you have deployed (Finesse or CTI Agent Desktop) reads some of these settings.

To configure Desk Settings, go to Unified CCE Administrator Manage > Agent > Desk Settings.

Administrators have unlimited access to Desk Settings configuration. Supervisors cannot access Desk Settings.

Add and Maintain Desk Settings

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Navigate to Unified CCE Administrator Manage &gt; Agent &gt; Desk Settings.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Click New to open the New Desk Settings window.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Enter fields:</td>
</tr>
<tr>
<td></td>
<td><strong>Field</strong></td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>Logout inactivity time</td>
</tr>
<tr>
<td></td>
<td>Wrapup on incoming</td>
</tr>
<tr>
<td></td>
<td>Wrapup on outgoing</td>
</tr>
<tr>
<td></td>
<td>Wrapup timer</td>
</tr>
</tbody>
</table>
From the drop-down, select Consultative Call or Blind Conference to indicate whether to create a consultative call or a blind conference call when the agent requests supervisor assistance. If you select Consultative Call, a caller will be aware when the supervisor joins. This setting is currently supported by the CTI OS desktop only.

From the drop-down, select Consultative Call or Blind Conference to indicate whether to create a consultative call or a blind conference call when the agent makes an emergency call to the supervisor. The default is Consultative Call. A caller will be aware when the supervisor joins in a consultative call. This setting is currently supported by the CTI OS desktop only.

Unchecked by default. When checked, indicates that calls sent to this agent will be answered automatically.

Check this to identify the agent as a Mobile Agent who can sign in remotely and take calls from any phone. With this selected, the agent can also sign in as a normal agent. The default is unchecked—meaning this is not a mobile agent.

Check this box if the agent must enter a reason before entering the Idle state. The default is unchecked—the agent is not required to enter a reason.

Check this box if the agent must enter a reason before logging out. The default is unchecked—the agent is not required to enter a reason.

There is no RONA field on the the Packaged CCE Desk Settings tool. The Unified CVP RONA timer controls the agent desk settings for Packaged CCE.

Save the desk settings to return to the List window, where a message confirms the successful creation.

**Update Desk Settings**

To edit a desk settings, click the hyperlinked name on the list window. This opens the *Edit Desk Settings* window.

All fields are editable.

Save the changes to return to the list window, where a message confirms that the changes are saved successfully.
Delete Desk Settings

To delete a desk settings, open the list of Desk Settings window and click the trash can icon for that desk settings name.

A message confirms the successful deletion or indicates reasons why you cannot delete.

Note

You cannot delete:

- A desk settings that is associated with an agent
- The system default desk settings.

Agent Desktops

Packaged CCE supports CTI OS and Cisco Finesse desktops. You can use either or both.

Packaged CCE imposes a limit of 1000 agents who can be logged in to the agent desktop. If you deploy both the CTI OS and Cisco Finesse desktops, when the 1001st agent logs in to either, both desktops display a warning that the login request failed because the limit has been reached.

Agent Desktop Comparison

CTI OS or Finesse

The table below summarizes and compares the features supported in these desktops.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cisco Finesse desktop 9.0</th>
<th>CTI OS desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESSIBILITY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>509 Compliance</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Keyboard shortcuts</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Large toolbar buttons</td>
<td>Yes / custom code</td>
<td>No</td>
</tr>
<tr>
<td>Scroll team messages</td>
<td>Yes / custom code</td>
<td>No</td>
</tr>
<tr>
<td>AGENT STATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log in and Log out</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Log out forced on application close</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Log out forced on duplicate agent ID</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature</td>
<td>Cisco Finesse desktop 9.0</td>
<td>CTI OS desktop</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Ready and Not Ready</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Reason codes: Log out Ready and Not Ready, global</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Wrap-up codes, editable by agent</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Wrap-up codes, global</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Wrap-up codes, optional</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Wrap-up for outbound calls</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>CHAT MESSAGING AND PRESENCE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agent chat to agent within team</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Agent chat to any agent while conferencing</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Agent chat to supervisor</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>CISCO OUTBOUND OPTION SUPPORT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predictive Outbound campaigns</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Progressive Outbound campaigns</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Scheduled and personal callbacks</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Outbound wrap-up code support</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>CONFIGURABLE GUI</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Desktop modes: Normal, open, on top, stealth, and agent selectable</td>
<td>Yes / custom code</td>
<td>No</td>
</tr>
<tr>
<td>Integrated browser</td>
<td>Yes / custom code</td>
<td>No</td>
</tr>
<tr>
<td><strong>DEPLOYMENT MODES</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard (Cisco Finesse server pair and Cisco Unified Contact Center Enterprise (CCE) peripheral gateway pair over the LAN)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Standard over the WAN (sides A and B separated by WAN)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature</td>
<td>Cisco Finesse desktop 9.0</td>
<td>CTI OS desktop</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Hosted (multiple Cisco Finesse instances per virtual machine</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>connecting to multiple peripheral gateways on a single box)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-instance (multiple Cisco Finesse instances per virtual machine</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>connecting to a single peripheral gateway running multiple peripherals)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-tenant (single Cisco Finesse instance and peripheral gateways</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>- logical partitioning of data per customer)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parent/Child</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Parent/Child clustered over WAN</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ENTERPRISE DATA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display caller information (Automatic Number Identification [ANI]</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>and Dialed Number Identification Service [DNIS])</td>
<td></td>
<td></td>
</tr>
<tr>
<td>View CallVars and Expanded Call Context (ECC) in user interface</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Set CallVars and ECC in user interface</td>
<td>Yes / custom code</td>
<td>Yes</td>
</tr>
<tr>
<td>Get and Set CallVars 1-10 and ECC Vars API</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MOBILITY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hot seating</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Extension mobility</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Mobile agent (PSTN)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Cisco IP Communicator support</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>VPN support</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>VMWare View VDI</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Citrix XenDesktop</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>PLATFORM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Localization</td>
<td>English only</td>
<td>Yes</td>
</tr>
<tr>
<td>Quality of Service (QoS)</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature</td>
<td>Cisco Finesse desktop 9.0</td>
<td>CTI OS desktop</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Secure HTTP</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**RECORDING**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cisco Finesse desktop 9.0</th>
<th>CTI OS desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent-initiated record call button</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**REAL-TIME DISPLAYS**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cisco Finesse desktop 9.0</th>
<th>CTI OS desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent state log</td>
<td>Implemented through Cisco Unified Intelligence Center. Cisco Finesse software does not currently have a native user interface or API</td>
<td>Yes</td>
</tr>
<tr>
<td>Agent statistics</td>
<td>Implemented only through Cisco Unified Intelligence Center.</td>
<td>Yes</td>
</tr>
<tr>
<td>Call log</td>
<td>Implemented only through Cisco Unified Intelligence Center.</td>
<td>Yes</td>
</tr>
<tr>
<td>Queue (skill) statistics</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Auto-refresh real-time data</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**SDK**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cisco Finesse desktop 9.0</th>
<th>CTI OS desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production SDK</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**SUPERVISOR**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cisco Finesse desktop 9.0</th>
<th>CTI OS desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barge intercept</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Change agent state</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Cisco Unified Communications Manager Silent Monitor</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Desktop Silent Monitor</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Emergency and assist</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature</td>
<td>Cisco Finesse desktop 9.0</td>
<td>CTI OS desktop</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Real-time queue information</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Switch Port Analyzer (SPAN) monitoring</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Team performance - see agent state by team</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Team performance - see time in state for each agent</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Team performance - view skill group associations</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Team performance - view call details for agent</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>THIRD-PARTY CALL CONTROL</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Login by ID</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Login by agent name</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Blank passwords</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Answer or release</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Hold or retrieve</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Alternate (while transferring)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Transfer - blind</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Transfer - consultation</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Conference - consultation</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Conference support for four participants</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Send digits (DTMF) while connected</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Pass call variables during transfer or conference</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Make call - dial pad</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Multiple call appearances</td>
<td>Yes</td>
<td>Yers</td>
</tr>
<tr>
<td>Multiline</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Recent call list</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>WEB ARCHITECTURE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Cisco Finesse desktop 9.0</td>
<td>CTI OS desktop</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Browser-based agent desktop</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Browser-based supervisor desktop</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Enterprise mash-up architecture</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Open Web 2.0 APIs</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Web 2.0 Software Development Kit (SDK)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Reason Codes**

Agents enter reason codes on their agent desktops when they go idle (or "Not Ready"). Reason codes appear in Unified Intelligence Center reports and help identify agent behavior. Are agents taking an inappropriate number of breaks? How long are agents at lunch or in meetings? What is the average time it takes an agent to enter wrap up data?

If you configure reason codes in the agent desktop software, it is a best practice to configure the same reason codes in Unified CCE Administration to ensure consistency in reporting.

Navigate to the **Unified CCE Administrator Manage > Agent > Reason Codes** interface to configure reason codes.

Supervisors cannot access the Reason Codes tool.

**Add and Maintain Reason Codes**

**Procedure**

1. **Step 1** Navigate to **Unified CCE Administrator Manage > Agent > Reason Codes**.
2. **Step 2** Click **New** to open the **New Reason Code** window.
3. **Step 3** Enter fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>yes</td>
<td>Enter up to 40 characters to name the reason code.</td>
</tr>
<tr>
<td>Code</td>
<td>yes</td>
<td>Enter a unique positive integer between 0 and 65,535.</td>
</tr>
</tbody>
</table>
| Description| no        | Enter up to 255 characters to describe the reason code. There is no restriction of characters. See [Native Character Sets](#) on page 13.
Step 4  Save the reason code to return to the List window, where a message confirms the successful creation.

Update Reason Codes

To edit a reason code, click the Text field on the List of Reason Codes to open the Edit Reason Code window. You can edit the Text and Description fields. Save the changes to return to the List window, where a message confirms that the changes are saved successfully.

Delete Reason Codes

Procedure

Step 1  To delete a reason code, open the List of Reason Codes window and click the delete icon for that reason code.

Step 2  Respond to the confirmation message. This action marks the reason code for deletion. You must permanently delete it from Configuration Manager.

Related Topics

Permanent Deletion, on page 87

Teams

Create teams to associate a set of agents with a specific supervisor who can run reports on that team and can receive Supervisor Assist requests from its members.

Note

Supervisor assist must be indicated in the desk settings tool and must be supported by the agent desktop.

An agent can be a member of one team.

Administrators can see and maintain teams.

Supervisors have display-only access to the Teams tool.

To configure teams, go to Unified CCE Administrator Manage > Agent > Teams.

Related Topics

Agents, on page 15
Supervisors, on page 20
Add and Maintain Teams

Procedure

Step 1
Navigate to Unified CCE Administrator Manage > Agent > Teams.

Step 2
Click New to open the New Team window. This window has three tabs: General, Members, and Supervisors. You cannot save the team until you have entered all required fields on the General tab. You can complete other tabs as needed and in any order.

Step 3
Complete the fields on the General tab:

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>yes</td>
<td>Enter up to 32 alphanumeric characters.</td>
</tr>
<tr>
<td>Description</td>
<td>no</td>
<td>Enter up to 255 characters to describe the team. See Native Character Sets, on page 13.</td>
</tr>
<tr>
<td>Supervisor Script Dialed Number</td>
<td>no</td>
<td>Click the magnifying glass to display the Select Supervisor Script Dialed Number list. These are all the internal dialed numbers that have been configured to be used by routing scripts to direct calls to a supervisor for assistance. They are created on the Dialed Number tool with a routing type of Internal Voice. Click a link to select a dialed number and close the list. You can select only one.</td>
</tr>
</tbody>
</table>

Step 4
Complete the Members tab:
This tab shows a list of agents on the team.
Click the magnifying glass to open the Add Agents popup. The row for each agent has four columns: a column that is either blank or that shows a "i" icon, a column with the agent's username, a column with the agent's last name, and a column with his first name.

- If an agent already has an "i" icon indicating that he is a member of a team, hover over the icon to see the name of that team. Clicking an agent who already has a team removes him from that team and reassigns him to this team.
- If an agent has no icon and has a blue row, that agent is already a member of this team.
- If an agent has no icon and has a white row, that agent is on no team. Clicking an agent who has no team membership adds him to this team.

Step 5
Complete the Supervisors tab:
This tab shows the supervisors for the team and allows you to add supervisors.

a) Click the magnifying glass to add supervisors to the team.
b) Click one or more links to select supervisors. The supervisors are now in the List of Supervisors.
c) Continue to another tab, or click **Save** to return to the List window, where a message confirms the successful creation of the team.

---

### Update Teams

To edit a team, click the team name on the List of Teams to open the Edit <Name> Team page. All fields on this page are editable.

Save the changes to return to the List page, where a message confirms that the changes are saved successfully.

### Delete Teams

**Procedure**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>To delete a team, open the List of Teams window and click the trash can icon for that team.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Respond to the confirmation message. A message either confirms the successful deletion or indicates reasons why you cannot delete.</td>
</tr>
</tbody>
</table>
Manage Queues

Calls are queued to agents based on their membership in skill groups or their qualification in precision queues. Administrators have access to all tools documented in this chapter. Supervisors have limited access to Skill Groups and display-only access to Attributes and Precision Queues.

- Attributes, page 33
- Precision Queues, page 35
- Skill Groups, page 44

Attributes

Attributes identify a call routing requirement, such as language, location, or agent expertise. You can create two types of attributes: Boolean or Proficiency.

- Use Boolean attributes to identify an agent attribute value as true or false. For example, you can create a Boston attribute that specifies that the agent assigned to this attribute must be located in Boston. An agent in Boston would have the attribute *Boston = True*. *Boston = True* as the term for that attribute.

- Use Proficiency attributes to establish a level of expertise in a range from 1 to 10, with 10 being the highest level of expertise. For a Spanish language attribute, for example, a native speaker would ideally have the attribute *Proficiency = 10*.

When you create a precision queue, you identify which attributes are part of that queue and then implement the queue in a script. When you assign a new attribute to an agent and the attribute value matches the precision queue criteria, the agent is automatically associated with the precision queue.

Navigate to the *Unified CCE Administrator Manage > Attributes* interface to configure attributes. Administrators can see and manage attributes. Supervisors can configure attributes for their supervised agents on the Attributes tab of the Agents tool.

Related Topics

- Precision Queues, on page 35
Add and Maintain Attributes

Procedure

Step 1 Navigate to Unified CCE Administrator Manage > Attributes.
Step 2 In the List of Attributes window, click New.
Step 3 Complete fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>yes</td>
<td>Type a unique attribute name. For example, to create an attribute for mortgage insurance, type mortgage.</td>
</tr>
<tr>
<td>Description</td>
<td>no</td>
<td>Enter a maximum of 255 characters to describe the Dialed Number String. See Native Character Sets, on page 13.</td>
</tr>
<tr>
<td>Type</td>
<td>yes</td>
<td>Select the type of Boolean or Proficiency.</td>
</tr>
<tr>
<td>Default</td>
<td>yes</td>
<td>Select the default of True or False for Boolean or a number from 1 to 10 for Proficiency.</td>
</tr>
</tbody>
</table>

Step 4 Click Save.

Edit Attributes

Procedure

Step 1 In the List of Attributes window, click the link for attribute name you want to edit. This opens the Edit Attribute window.
Step 2 Modify the Name, Description or Default. Note: You cannot change the attribute Type.
Step 3 Click Save.
Delete Attributes

Procedure

Step 1
From the List of Attributes window, click the trash can icon for the attribute you want to delete.

Step 2
Click Yes on the confirmation message.

Precision Queues

Precision routing offers a multidimensional alternative to skill group routing: using Unified CCE scripting, you can dynamically map the precision queues to direct a call to the agent who best matches the precise needs of the caller. Precision queues are the key components of precision routing.

To configure Precision Routing, you must:

1. Create attributes. Attributes are characteristics that can be assigned a True | False value or a Proficiency rating from 1 to 10.
2. Assign attributes to agents.
3. Create precision queues.
4. Create routing scripts.

There is no need to add an agent to a precision queue; agents become members of precision queues automatically based on their attributes. If a precision queue requires an agent who lives in Boston, who speaks fluent Spanish, and who is proficient in troubleshooting a specific piece of equipment, an agent with the attributes Boston = True, Spanish = True, and Repair = 10 is automatically part of the precision queue. A Spanish caller in Boston who needs help with equipment is routed to that agent.

A precision queue includes:

• Terms - A term compares an attribute against a value. For example, you can create the following term: Spanish == 10. The term of the attribute is the highest proficiency in Spanish.

• Expressions - An expression is a collection of one or more terms. The terms in an expression must share the same operator—they must all be AND or must all be OR relationships.

• Steps - A precision queue step is a time-based routing point within the precision queue. You must have at least one step and can have up to ten steps. A step is a collection of at least one or more expressions.
A step may also include wait time and a Consider If formula. Use wait time to assign a maximum amount of time to wait for an available agent. Use a Consider If formula to evaluate the step against predefined criteria, for example, another queue.

Navigate to Unified CCE Administrator Manage > Agent > Precision Queues to configure precision queues.

Administrators have full permission to configure precision queues. Supervisors have display-only access to the Precision Queues tool.

Related Topics
  - Build Precision Queue Steps, on page 39
  - Consider If Formula for Precision Queue, on page 41
  - Attributes, on page 33
  - Skill groups or Precision Queues?, on page 43

Add and Maintain Precision Queues

Before You Begin
Before you can create precision queues, you must:

- Create agents
- Create attributes

Procedure

<table>
<thead>
<tr>
<th>Name</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>no</td>
<td>Enter up to 255 characters to describe the precision queue. See Native Character Sets, on page 13.</td>
</tr>
</tbody>
</table>

Step 1  Navigate to Unified CCE Administrator Manage > Agent > Precision Queues. This opens a List of Precision Queues page showing all precision queues currently configured.

Step 2  Click New to open the New Precision Queue screen. Complete the fields.
<table>
<thead>
<tr>
<th>Name</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
</table>
| Service Level Type    | yes       | Select the service level type used for reporting on your service-level agreement. Service level type indicates how calls that are abandoned before the service level threshold affect the Service Level calculation. This is a drop-down menu that defaults to Ignore Abandoned Calls and includes these options:  
  - **Ignore Abandoned Calls**: Select this option if you want abandoned calls to be excluded from the service level calculation.  
  - **Abandoned Calls have Negative Impact**: Select this if you want only those calls that are answered within the Service level threshold time to be counted as treated calls. The Service Level will be negatively affected by calls that abandon within the Service Level Threshold time.  
  - **Abandoned Calls have Positive Impact**: Select this if you consider a call abandoned within the Service level threshold time as a treated call. With this configuration, abandoned calls will have a positive impact on the Service Level. |
| Service Level Threshold | yes       | Enter the time in seconds that calls are to be answered based on your service level agreement. The time entered in this field is used to report on service level agreements and does not impact how long a call remains in a precision queue. The length of time a call remains in a step is determined by each individual step wait time. |
Add and Maintain Precision Queues

### Agent Order

<table>
<thead>
<tr>
<th>Name</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
</table>
| Agent Order   | yes       | Select an option to determine which agents receive calls from this queue:

The ordering of agents does not dictate the agents who are selected into a Precision Queue step. Agents are included or excluded based on the conditions specified for the step.

- **Longest Available Agent** (the default) - The default method of agent ordering for a precision queue. The call is delivered to the agent who has been in the available (or ready) state the longest.

- **Most Skilled Agent** - The call is delivered to the agent who has the highest competency sum from all the attributes pertinent to the Precision Queue Step. In an agent-rich environment, this can mean that more competent agents would be utilized more than less competent agents.

- **Least Skilled Agent** - The call is delivered to the agent who has the lowest competency sum from all the attributes pertinent to the Precision Queue step.

### Bucket Intervals

<table>
<thead>
<tr>
<th>Name</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
</table>
| Bucket Intervals| no        | Select the Bucket Interval whose bounds are to be used to measure the time slot in which calls are answered. The field defaults to the System Default configured in System Settings. To select a different bucket interval:

1. Click the magnifying glass to display *Select Bucket Intervals*.
2. Click a link to make a selection and close the list.

Click *x* to clear the selection.

---

**Step 3**  Click the numbered step builder link (‘Step 1’, ‘Step 2’, and so on) to open the Step Builder pop-up.

**Step 4**  When you have added steps, click **Save**.

---

Packaged CCE Administration Guide for Release 9.0(x)
Build Precision Queue Steps

Every precision queue must have a step, and every step must have an Expression. An Expression is a collection of attribute terms.

Procedure

Step 1  Enter all required fields for the new precision queue.

Step 2  Click the numbered step link in the Steps panel (Step 1, Step 2, and so forth). This opens a Step # screen.

Step 3  Build the first step as follows.

a) Bypass the Consider if and Wait for fields. They are display-only on the first and last step of a precision queue. As soon as you create a second step, you can return to the first step and enter values for Consider if and Wait for.

b) Click the magnifying glass to the right of the Select Attribute field in the Expression 1 panel.

c) Select an attribute from the list.

d) Use the two Select fields to establish the terms of the attribute. Click the first Select... field to choose an operator.

   • For Boolean attributes, choices are the operators for equal and not equal.

   • For Proficiency attributes, choices are the operators for True, False, Less Than, Less Than or Equal To, Greater Than, and Greater Than or Equal To.

e) Click the second Select field to choose a value.

   • For Boolean attributes, values are True and False.

   • For Proficiency attributes, values are numbers from 1 to 10.
Your selection creates an attribute term for the Expression. At this point, the term will appear in the precision queue similar to this: [(Spanish == 10)]. The term is a requirement that the agent must be fluent in Spanish.

**Step 4** To add a second attribute to the first Expression, click *Add Attribute* in the *Expression 1* row.

a) Select **AND** or **OR** to establish the relationship between the first and second attributes.

b) Repeat steps 3b, 3c, and 3d.

**Step 5** Continue to add attributes to Expression 1. All attributes within an Expression must be joined by the same logical operator. They must all be ANDs, or they must all be ORs.

**Step 6** To add a second Expression, click the *Add Attribute* drop-down in the *Expression 1* row and select *Add Expression*.

**Step 7** Select **AND** or **OR** to establish the relationship between the first and second Expressions.

**Step 8** Add attributes to Expression 2.

**Step 9** Continue to add Expressions as needed.
In this example, a Spanish caller located in the Boston area needs an onsite visit from a technician to repair his ServerXYZ. An ideal agent should be fluent in Spanish and have the highest proficiency in ServerXYZ. This can be seen in Expression 1. Expression 2 allows us to specify that the selected agent must also be from either Boston or the New England area.

**Step 10** When you have completed the step, click **OK** to add it to the precision queue.

**Step 11** To build the next step, click the numbered step link in the Steps panel (Step 1). Each successive step is pre-populated with the Expressions and attributes of its predecessor. The expectation is that you will decrease the attribute qualifications and competencies—to lower the bar such that the pool of acceptable agents increases.

**Step 12** When you have created all steps, you can open any step except the last and enter values in the **Consider If** and **Wait for** fields.

- **Consider If** is a formula that evaluates a call within a step against additional criteria.
- **Wait for** is a value in seconds to wait for an available agent. A call will queue at a particular step looking for an available agent matching that step criteria until the number of seconds specified. A blank wait time indicates that the call will proceed immediately to the next step if there are no available agents matching the step criteria. Wait time defaults to 0 and can take a value up to 2147483647.

---

**Consider If Formula for Precision Queue**

If you are not on the last Step of the Precision Queue, then you can enter a **Consider If** formula for that step. A **Consider If** formula evaluates a call (within a step) against additional criteria. Each time a call reaches a step with a **Consider If** expression, the expression is evaluated. If the value for the expression returns as true, the call is considered for the step. If the value returns as false, the call moves to the next step. If no expression is provided for a step, the step is always considered for calls.
To add a Consider If formula, type the formula into the Consider If box. Alternatively, you can use the Script Editor to build the formula and then copy and paste it into the Consider If box. Objects used in Consider If formulas are case-sensitive. All Consider If formulas that you add to a precision queue must be valid. If you add an invalid formula, you cannot save the precision queue. To ensure that the formula is valid, use Script Editor to build and validate the formula.

Only the following scripting objects are valid in a Consider If formula:

- Call
- PQ
- Skillgroup
- ECC
- PQ Step
- Call Type
- Custom Functions (you can create custom functions in Script Editor)

It is possible that a valid Consider If formula can become invalid. For example, if you delete an object used in the formula after you create or update the precision queue, the formula is no longer valid.

Consider If Formula Examples

- PQ.PQ1.LoggedOn > 1 - Evaluates whether there is more than one agent logged into this queue
- CallType.CallType1.CallsRoutedToday > 100 - Evaluates whether more than 100 calls of this call type were routed today.
- PQStep.PQ1.1.RouterAgentsLoggedIn > 1 - Evaluates whether there is more than one router agent logged into this queue for step 1.
- CustomFunction(Call.PeripheralVariable1) > 10 - Evaluates whether this formula using a custom function returns a value greater than ten.

Edit Precision Queues

Procedure

Step 1  On the List page, navigate or search for the hyperlink name of the precision queue to edit.
Step 2  Click the Name to open the Edit page.
Step 3  Make changes and Save.

Delete Precision Queues

You cannot delete a precision queue that is referenced statically in any version of a saved script. Before you can delete a precision queue that is referenced statically in a script, you must remove the precision queue from every saved version of the script.
If you reference a precision queue dynamically in a script and there are calls queued against the precision queue, you can delete the precision queue. However, the router keeps the precision queue operational until all calls are no longer in the queue. Once the precision queue is deleted, no new calls are queued against it.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>On the List of Precision Queues page, select the precision queue to delete.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Click the X icon. You receive a prompt to confirm that you want to delete the precision queue.</td>
</tr>
<tr>
<td>Step 3</td>
<td>To delete the queue, click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>

**Skill groups or Precision Queues?**

Should you use skill groups or precision queues for the routing needs of your organization? This section distinguishes the two methods.

**Use a skill group**

A skill group is a collection of the attribute definitions of your proficiencies. For example, it could be a predefined collection of traits, such as salespeople who are in charge of selling to England. The skill group could be called "English sales." If you wanted to divide the agents in this group into two types of proficiencies (perhaps based on experience), you would need to set up two separate skill groups; for example, English Sales 1 and English Sales 2. You would then associate an agent with one of them, based on his or her proficiency. Do this by accessing the skill group and locating the agent that you want to add to it (or add that skill group to the agent). To summarize, creating a skill group involves first building a concept of what combinations of attributes you want for each agent, like English Sales 2.

**Use a precision queue**

In contrast to skill groups, a precision queues breaks down attribute definitions to form a collection of agents at an attribute level. The agents that match the attribute level of the precision queue become associated with that precision queue.

With precision queues, the above English sales example involves defining the attributes English and Sales, and associating agents that have those traits to them. The precision queue English Sales would dynamically map all those agents that had those traits to the precision queue. In addition, you can define more complex proficiency attributes to associate with those agents. This would allow you to build, in a single precision queue, multiple proficiency searches like English language proficiency 10 and sales proficiency 5.

To break down the precision queue example into skill groups, you would need to set up two separate skill groups: English language proficiency 10 and sales proficiency 5. With precision queues you can refine agents by attributes. With skill groups you must define a single skill groups as a mixture that you then attach to the relevant agents.

**Decide on skill groups or a precision queue**

Precision Routing enhances and can replace traditional routing. Traditional routing looks at all of the skill groups to which an agent belongs and defines the hierarchy of skills to map business needs. However, traditional routing is restricted by its single dimensional nature.
Precision Routing provides multidimensional routing with simple configuration, scripting, and reporting. Agents are represented through multiple attributes with proficiencies so that the capabilities of each agent are accurately exposed, bringing more value to the business.

If your routing needs are not too complex, consider using one or two skill groups. However, if you want to conduct a search involving as many as ten different proficiency levels in one easily-managed queue, use precision queues.

**Skill Groups**

A skill group is a collection of agents who share a common set of competencies that equip them to handle the same types of requests. Some examples of skill groups are a collection of agents who speak a specific language or who can assist callers with billing questions.

An agent can be a member of up to fifteen skill groups. Each skill group is associated with a specific media routing domain (MRD) such as voice, chat, or email.

An agent's skill group membership can determine the types of contacts that are routed to her. For example, if an agent is a member of a skill group that is set up for the Cisco_Voice routing domain only, that agent is a "voice-only" agent for that skill group. If an agent is a member of a skill group that is set up for a non-voice routing domain, that agent is a multichannel agent for that skill group.

Use Cisco Unified Intelligence Center (CUIC) reports to view agent activity in skill groups, to monitor call distribution among skill groups, or to see how one skill group is performing compared with others.

Navigate to the **Unified CCE Administrator Manage > Agent > Skill Groups** interface to configure skill groups.

Administrators have full permission to configure skill groups. Supervisors can see all skill groups but cannot add or delete skill groups. Supervisors have permission to add and remove their supervised agents on the Skill Groups Members tab.

### Related Topics

- Agents, on page 15
- Skill groups or Precision Queues?, on page 43
- Supervisors, on page 20

**Add and Maintain Skill Groups**

**Procedure**

**Step 1** Navigate to **Unified CCE Administrator Manage > Agent > Skill Groups**.

**Step 2** Click **New** to open **New Skill Group**. This window has two tabs: General and Members. You can complete the tabs in any order, but you cannot save the skill group until you have entered all required fields on the General tab.

**Step 3** Complete the fields on the General tab:
<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>yes</td>
<td>Enter a name using up to 32 alphanumeric characters.</td>
</tr>
<tr>
<td>Description</td>
<td>no</td>
<td>Enter up to 255 characters to describe the skill group. See <a href="#">Native Character Sets</a>, on page 13.</td>
</tr>
<tr>
<td>Media Routing Domain</td>
<td>yes</td>
<td>MRDs are Unified CCE entities that organize how requests for media are routed. The system routes calls to skill groups that are associated with a particular communication medium; for example, voice or email. This field defaults to Cisco_Voice. To select a different Media Routing Domain: 1 Click the magnifying glass to display Select Media Routing Domain. Selections are all MRDs that have been created in Unified CCE Configuration Manager. 2 Click a link to make a selection and close the list. Click x to clear the selection.</td>
</tr>
<tr>
<td>Bucket Intervals</td>
<td>no</td>
<td>Select the Bucket Interval whose bounds are to be used to measure the time slot in which calls are answered. The field defaults to the System Default. To select a different bucket interval: 1 Click the magnifying glass to display Select Bucket Intervals. 2 Click a link to make a selection and close the list. Click x to clear the selection.</td>
</tr>
<tr>
<td>Service Level Threshold</td>
<td>no</td>
<td>Enter a value in seconds that you set as a goal for connecting a call with an agent. The field defaults to the threshold configured in Unified CCE Configuration Manager for this Media Routing Domain. Leave this field blank to use the service level threshold value for the Media Routing Domain. Enter a value of 0 seconds if you do not want a service level event to be set for the calls. They will not be treated as a service-level calls.</td>
</tr>
<tr>
<td>Field</td>
<td>Required?</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Service Level Type</td>
<td></td>
<td>Service level type indicates how calls that are abandoned before the service level threshold affect the Service Level calculation. This is a drop-down menu that defaults to Use Media Routing Domain and includes these options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Use Media Routing Domain Value:</strong> Select this option to use the value that is currently defined for the MRD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Ignore Abandoned Calls:</strong> Select this option if you want abandoned calls to be excluded from the service level calculation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Abandoned Calls have Negative Impact:</strong> Select this if you want only calls that are answered within the Service level threshold time as to be counted as treated calls. The Service Level will be negatively affected by calls that abandon within the Service Level time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Abandoned Calls have Positive Impact:</strong> Select this if you consider a call abandoned within the Service level threshold time as a treated call. With this configuration, abandoned calls will have a positive impact on the Service Level.</td>
</tr>
</tbody>
</table>

**Step 4**  
Complete the Members tab:  
This tab shows the list of agents for this skill group. If the skill group has no agents, the Agent field shows *No Items Found* and *No Items*.

a) Click the magnifying glass to open *Add Agents*.

b) Click the agents you want to add to this skill group. There is no limit to the number of agents in a skill group.

c) Close *Add Agents*. The agents you chose appear on the List of Agents, as well as the total agents (number of entries) for this skill group.

d) **Save** this tab to return to the List window, where a message confirms the successful creation of the skill group.
Edit Skill Groups

Procedure

Step 1
To edit a skill group, click the skill group name on the List of Skill Groups to open Edit Skill Group.
This window shows all fields from the Add Skill Group page and an additional Peripheral Number field, which is autogenerated when you add and save a new skill group. This field shows the number of the skill group, as known by the peripheral.

Media Routing Domain and Peripheral Number are protected. You can edit all other fields.

Step 2
Save the changes to return to the List window, where a message confirms that the changes are saved successfully.

Delete Skill Groups

Administrators can delete agents. Supervisors cannot.

Procedure

Step 1
Display the List of Skill Groups.
Step 2
Click the trash can icon next to the name of the skill group that you want to delete.
Step 3
Respond Yes to the message that you want to delete. This marks the skill group for deletion. You must delete it permanently from Configuration Manager.
A message either confirms the successful deletion or indicates reasons why you cannot delete.

Related Topics
Permanent Deletion, on page 87
CHAPTER 5

Manage Calls

This chapter explains the Unified CCE Administration tools relevant for calls. Administrators have access to all tools documented in this chapter. Supervisors do not have access to these tools.

- Bucket Intervals, page 49
- Call Types, page 51
- Dialed Numbers, page 53
- Expanded Call Variables, page 55
- Network VRU Scripts, page 58

Bucket Intervals

Configure bucket intervals to report on how many calls are handled or abandoned during specific, incremental time slots. Each bucket interval has a maximum of nine configurable time slots, called Upper Bounds. Upper Bounds are ranges measured in seconds to segment and capture call-handling activity. You can run reports that show calls answered and calls abandoned for these intervals.

For example, if your goal is to have calls handled within 1 minute, you might set up Upper Bounds for intervals that show how many calls are handled in less than or more than 1 minute. Intervals might be for 30 seconds, 60 seconds, 80 seconds, 120 seconds, 150 seconds, 180 seconds, and 240 seconds. Using these intervals, you can see if calls are being answered within 1 minute or if callers are waiting longer. The intervals also give you insight into how long callers are willing to wait before abandoning a call. Perhaps many callers do not abandon a call until they have waited for two minutes. This might indicate that you can modify your goal.

You can associate bucket intervals with Call Types, with Skill Groups, and with Precision Queues.

To avoid reporting inconsistencies, modify Bucket Interval settings only at specific time boundaries (that is, end of day, week, or month). Ensure that no one is running reports for the intervals that you are changing when you modify the boundaries.

Unified CCE ships with a single System default Bucket Interval whose boundaries (increments) are: 8, 30, 60, 90, 120, 180, 300, 600, and 1200 (in seconds).
Add and Maintain Bucket Intervals

Procedure

Step 1 Navigate to Unified CCE Administrator Manage > Call > Bucket Intervals.
Step 2 Click New to open the New Bucket Interval window.
Step 3 Complete fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>yes</td>
<td>Enter a name for the call type using a maximum of 32 characters.</td>
</tr>
<tr>
<td>Upper Bound 1</td>
<td>yes</td>
<td>Enter a value in the Upper Bound 1 field, using a number greater than 0 and less than 2147483647. This value is interpreted as seconds, and the beginning value for Upper Bound 1 is 0. For example, your entry of 10 in this field creates an Upper Bound 1 interval of 0 to 10 seconds.</td>
</tr>
<tr>
<td>Upper Bound 2 to Upper Bound 9</td>
<td>no</td>
<td>If used, the beginning value for an Upper Bound must be higher than the ending value of the previous Upper Bound. If you leave an Upper Bound field blank, all remaining fields must be blank. For Example: To configure three intervals that span 10 seconds each and then have all other calls grouped into an interval that extends beyond your third defined interval, enter the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Upper Bound 1 interval: 10 This time slot is 0 to 10 seconds. Reports will show the total number of calls answered and calls abandoned from 0 to 10 seconds.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Upper Bound 2 interval: 20 This time slot is any time greater than 10 seconds and less than 20 seconds. Reports will show the total number of calls answered and calls abandoned between 10 and 20 seconds.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Upper Bound 3 interval: 30 This time slot is any time greater than 20 seconds and less than 30 seconds. Reports will show the total number of calls answered and calls abandoned between 20 and 30 seconds.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• All other blank (Upper Bound Intervals 4 to 9 are not set). Reports will show the total number of calls answered and calls abandoned after 30 seconds.</td>
</tr>
</tbody>
</table>
Step 4 Save the bucket interval to return to the List screen, where a message confirms the successful creation of the bucket interval.

---

Call Types

Call types categorize calls. Based on call type, the system maps a dialed number to a routing script that ultimately sends the call to the appropriate destination. Consider the call types you need to create to meet your reporting needs, and configure a separate call type for each type of call treatment that you want to offer.

For example, you might create call types for:

- Calls answered by agents
- Calls abandoned at the VRU
- Calls that reroute when the agent does not answer
- Calls that are transferred and conferenced
- Outbound Option calls
- Calls that require Supervisor assistance

Add and Maintain Call Types

Procedure

| Step 1 | Navigate to **Unified CCE Administrator Manage** > **Call** > **Call Types**.
| Step 2 | Click **New** to open the **New Call Type** window.
| Step 3 | Enter fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>yes</td>
<td>Enter a name for the call type using a maximum of 32 characters. This name must be unique among call types in the system.</td>
</tr>
<tr>
<td>Description</td>
<td>no</td>
<td>Enter a maximum of 255 characters to describe the call type. See Native Character Sets, on page 13.</td>
</tr>
<tr>
<td>Service Level Threshold</td>
<td>no</td>
<td>This value is used in reports to identify the percentage of calls that are answered within that time threshold, enabling you to see whether agents are meeting the target goal. The field defaults to the System Default. To select a different service level threshold, enter a value in seconds, greater than 1 and less than 86,400.</td>
</tr>
</tbody>
</table>
### Service Level Type

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Level Type</td>
<td>no</td>
<td>Indicates how the system software calculates the service level. The field defaults to the System Default. To override the system default for this call type, select one of these other options from the drop down list:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Ignore Abandoned Calls:</strong> This selection excludes abandoned calls from the service level calculation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Abandoned Calls have Negative Impact:</strong> Select this if you want only calls that are answered within the Service level threshold time to be counted as treated calls. The Service Level will be negatively affected by calls that abandon within the Service Level time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Abandoned Calls have Positive Impact:</strong> Select this if you consider a call abandoned within the Service level threshold time as a treated call. Abandoned calls will have a positive impact on the Service Level.</td>
</tr>
</tbody>
</table>

### Bucket Intervals

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket Intervals</td>
<td>no</td>
<td>Bucket intervals appear in call type reports and display the number of calls answered and abandoned for different time intervals.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configure the Bucket Interval associated with this call type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The field defaults to the System Default configured in.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, the field might show System Default (New_Bucket_Interval). For example, the field might show System Default (New_Bucket_Interval).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>To select a different bucket interval:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Click the magnifying glass to display Select Bucket Interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Click the link to select a bucket interval and close the List.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Click <strong>Save</strong>.</td>
</tr>
</tbody>
</table>

**Step 4** Save the call type and return to the List window, where a message confirms the successful creation.
Dial numbers are string values used to select the appropriate routing script so that a voice call or a non-voice task (such as an email or a request for a web chat) can be delivered to an agent. Each dialed number string is configured with a Routing Type and a Media Routing Domain and can be mapped to a Call Type.

A typical call center requires multiple dialed number strings. In addition to creating dialed number strings for each telephone number that customers can use to reach you, you must set up dialed number strings:

- So that an agent can transfer to, or conference in, another agent
- For requery on no answer (RONA)
- For supervisor/emergency assist calls

**Related Topics**

Call Types, on page 51

### Add and Maintain Dialed Numbers

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Navigate to <strong>Unified CCE Administrator Manage &gt; Call &gt; Dialed Numbers</strong>.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Click <strong>New</strong> to open the <em>New Dialed Numbers</em> window.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Complete fields:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialed Number String</td>
<td>yes</td>
<td>This is the value used to route the call. Enter a string value that is unique for the routing type, using a maximum of 25 characters.</td>
</tr>
<tr>
<td>Description</td>
<td>no</td>
<td>Enter a maximum of 255 characters to describe the Dialed Number String. See Native Character Sets, on page 13.</td>
</tr>
</tbody>
</table>
From the drop-down menu, select one of the following:

- **External Voice**: Select this for Dialed Number Strings that apply to voice calls coming from Cisco Customer Voice Portal (CVP). These calls are referred to as external because they typically come from outside of the enterprise through a gateway. External Voice is the selection for calls that come in from customers and need to be answered by agents or sent to the IVR.

- **Internal Voice**: Select this for Dialed Number Strings that can be called from a Cisco Unified Communications Manager phone. These calls must have a Route Point on Unified Communications Manager that corresponds to the internally dialed number. They are referred to as internal because they can only be accessed by Unified Communications Manager.

  Each internally-dialed number creates a single dialed number record tied to the Unified Communications Manager routing client. Internal Voice is used for dialed numbers that agents use to transfer calls to other agents, to enable the system to redirect calls internally when the agent does not answer, and to direct a call from an agent to a supervisor for assistance.

  Dialed numbers with the routing type Internal Voice appear on the Supervisor Script Dialed Number list when you create or edit a team. See Add and Maintain Teams, on page 31.

- **Outbound Voice**: Select this for Dialed Number Strings that are used by the Cisco Outbound Option Dialer. These Dialed Number Strings are referenced and used to route calls to agents or to IVR scripts in the Campaign Skill Group Selection.

- **Multichannel**: Select this for Dialed Number Strings that are routed to an agent who must interact with a customer by email or by web chat.

### Table: Manage Calls

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing Type</td>
<td>yes</td>
<td>From the drop-down menu, select one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>External Voice</strong>: Select this for Dialed Number Strings that apply to voice calls coming from Cisco Customer Voice Portal (CVP). These calls are referred to as external because they typically come from outside of the enterprise through a gateway. External Voice is the selection for calls that come in from customers and need to be answered by agents or sent to the IVR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Internal Voice</strong>: Select this for Dialed Number Strings that can be called from a Cisco Unified Communications Manager phone. These calls must have a Route Point on Unified Communications Manager that corresponds to the internally dialed number. They are referred to as internal because they can only be accessed by Unified Communications Manager.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each internally-dialed number creates a single dialed number record tied to the Unified Communications Manager routing client. Internal Voice is used for dialed numbers that agents use to transfer calls to other agents, to enable the system to redirect calls internally when the agent does not answer, and to direct a call from an agent to a supervisor for assistance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dialed numbers with the routing type Internal Voice appear on the Supervisor Script Dialed Number list when you create or edit a team. See Add and Maintain Teams, on page 31.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Outbound Voice</strong>: Select this for Dialed Number Strings that are used by the Cisco Outbound Option Dialer. These Dialed Number Strings are referenced and used to route calls to agents or to IVR scripts in the Campaign Skill Group Selection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Multichannel</strong>: Select this for Dialed Number Strings that are routed to an agent who must interact with a customer by email or by web chat.</td>
</tr>
</tbody>
</table>
The Media Routing Domain associated with the dialed number. Media Routing Domains (MRDs) are Unified CCE entities that organize how requests for media are routed. The system routes calls to agents who are associated with a particular communication medium; for example, voice or email. The selection of Routing Type determines what appears in this field.

- If the Routing Type is External Voice, Internal Voice, or Outbound Voice, the Media Routing Domain is Cisco_Voice and you cannot change it.
- If the Routing Type is Multichannel, click the magnifying glass to display Select Media Routing Domain. You see all MRDs that have been created in Unified CCE Configuration Manager.

**Call Type**

To select a different Call Type, click the magnifying glass to display the Select Call Type pop-up window. Click a link to make a selection and close the list. Click **x** to clear the selection and reapply None.

### Step 4

**Save** the dialed number to return to the List screen, where a message confirms the successful creation.

---

## Expanded Call Variables

Calls can carry data with them as they move through the system. This data, called expanded call variable data, is embedded with the call and is visible to the agent on the Finesse and CTI OS agent desktops. Seeing expanded call variable data can assist the agent in working with the caller.

The call context data that is stored in an expanded call variable can be set or updated by Customer Voice Portal (CVP), by Unified CCE scripting, or by an agent who is transferring the call.

- If the call is at Unified CVP for IVR treatment, the call context is exchanged between Unified CVP and Unified CCE.
- If the call is at an agent, the call context is exchanged between the desktop (CTI OS or Finesse) and Unified CCE.

Note that this is a two-way exchange: in some cases the expanded call variable data is sent to Unified CCE from Unified CVP or the agent desktop, and in some cases the data is sent by Unified CCE based on script configuration to Unified CVP or the agent desktop.

Cisco-defined expanded call variables are identified by the Cisco Provided check box on the Edit Expanded Call Variable screen. You can create new expanded call variables subject to certain sizing constraints.
Add and Maintain Expanded Call Variables

Procedure

**Step 1** Navigate to **Unified CCE Administrator Manage > Call > Expanded Call Variables** to open the List of Expanded Call Variables. The List of Expanded Call Variables is populated with Cisco-provided expanded call variables that carry such data as the account number for the contact and the caller’s time zone. These are:

- user.microapp.app_media_lib
- user.microapp.error_code
- user.microapp.FromExtVXML
- user.microapp.ToExtVXML
- user.microapp.metadata
- user.microapp.input_type
- user.microapp.play_data
- user.microapp.sys_media_lib
- user.microapp.UseVXMLParams

The top of the window tracks the number of bytes used by the current list of expanded call variables, measured against the system total and the CTI Server total.

**Step 2** Click **New** to open the *New Expanded Call Variable* window.

**Step 3** Complete fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>yes</td>
<td>The name of the expanded call variable, prepended by user. This field allows a maximum of 32 characters. (This maximum includes the four characters in user.) The name must be unique. To accomplish this, use a consistent naming syntax such as user.company.name, where company identifies the company and name is the expanded call variable.</td>
</tr>
<tr>
<td>Description</td>
<td>no</td>
<td>Enter up to 255 characters to describe the expanded call variable. There is no restriction of characters. See Native Character Sets, on page 13.</td>
</tr>
<tr>
<td>Field</td>
<td>Required?</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maximum Length</td>
<td>yes</td>
<td>Specifies the maximum number of characters allowed in the value that will be stored in the expanded call variable value (the size of the variable string). The range is from 1 to 210 characters.</td>
</tr>
<tr>
<td>Array</td>
<td>no</td>
<td>This field is unchecked by default to indicate that the expanded call variable is scalar. Select this to configure the expanded call variable as an array, not a scalar.</td>
</tr>
<tr>
<td>Maximum Array Size</td>
<td>no</td>
<td>This field appears when Array is checked. Use it to indicate the maximum number of elements (1-255) in the array.</td>
</tr>
<tr>
<td>Enabled</td>
<td>no</td>
<td>Visible only when Array is unchecked, checking this indicates that the scalar expanded call variable is currently enabled—it can be used in scripts and appears on the agent desktop.</td>
</tr>
<tr>
<td>Persistent</td>
<td>no</td>
<td>Visible only when Array is unchecked, checking this indicates that data for this scalar expanded call variable will be written to the historical database; specifically to the Termination Call Detail (TCD) and Route Call Detail (RCD) tables. Note that storing excessive call variable data can degrade historical database performance. Only persistent call variables are written to the historical database. Non-persistent variables can be used in routing scripts, but are not written to the database.</td>
</tr>
<tr>
<td>Cisco Provided</td>
<td>—</td>
<td>This check box is display-only and appears only for the expanded call variables that are provided with the system. For this reason, the New Expanded Call Variable window does not include this field.</td>
</tr>
<tr>
<td>Bytes Required (if enabled)</td>
<td>—</td>
<td>This display-only field indicates the number of bytes required to store the expanded call variable in the system.</td>
</tr>
<tr>
<td>Bytes Required in CTI Server (if enabled)</td>
<td>—</td>
<td>This display-only field is similar to Bytes Required, above, but applies to the CTI Server. In CTI Server, the number of bytes required includes the length of the expanded call variable name.</td>
</tr>
</tbody>
</table>
Sizing Expanded Call Variables

An expanded call variable name can be up to 32 characters long. The Maximum Length of an expanded call variable is 210 characters. If the variable is an array, the Maximum Array Size is 255. The number of characters or digits does not indicate the number of bytes. For example, a scalar expanded call variable of 10 characters could take up 15 bytes of space.

Expanded call variable usage impacts PG, Router, and Logger bandwidth. As a best practice and to control capacity impact, the Expanded Call Variables List, Add, and Edit windows track the space that your expanded call variables are consuming, as compared with the system maximums.

The maximum amount of space that all expanded call variables can take up in Unified Contact Center cannot exceed 2000 bytes. Each expanded call variable in Unified CCE is calculated using the following formula:

- For scalar: $5 + \text{Maximum\_Length}$
- For array: $5 + (1 + \text{Maximum\_Length}) \times \text{Maximum\_Array\_Size}$

The maximum amount of space that all expanded call variables can take up in CTI Server cannot exceed 2500 bytes. Each expanded call variable in CTI Server is calculated using the following formula:

- For a scalar variable, the size is length of Name + Maximum Length + 4.
- For an array variable, the size is (length of Name + Maximum Length + 5) * Maximum Array Size.

Network VRU Scripts

Not all calls are delivered directly to agents. Some are sent to a Voice Response Unit (VRU) instead of, or before, they are sent to an agent. In the Packaged CCE deployment, the VRU is Customer Voice Portal (Unified
CVP). You must configure Network VRU scripts to direct Unified CVP on how to handle the treatment of individual calls.

These directions are called CVP Micro-Application functions. There are six Unified CVP micro-application types:

- Play Media (PM) - Retrieves and plays a media file such as a welcome.wav or an agent greeting.
- Play Data (PD) - Retrieves and plays data of various types, such as numbers, characters, time of day, or currency.
- Get Digits (GD) - Plays a media file and retrieves digits from the caller.
- Menu (M) - Plays media menu file and retrieves a single telephone keypad entry from the caller.
- Get Speech (GS) - A "GS,Server,V" script is provided with Packaged CCE and appears in the List of Network VRU Scripts.
- Capture - Allows you to trigger the storage of current call data at various points.

Add and Maintain Network VRU Scripts

Procedure

**Step 1** Navigate to **Unified CCE Administrator Manage > Call > Network VRU Scripts** to open the List of Network VRU Scripts.

There is one default Network VRU Script (GS,Server,V), which is used to route calls to the VXML Server IVR application.

**Step 2** Click **New** to open the *New Network VRU Script* window.

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>yes</td>
<td>Enter a unique name that will identify the script, using a maximum of 32 alphanumeric characters. For example: <em>Play Welcome</em>.</td>
</tr>
<tr>
<td>Description</td>
<td>no</td>
<td>Enter additional information about the script. See <a href="#">Native Character Sets</a> on page 13.</td>
</tr>
<tr>
<td>Routing Type</td>
<td>yes</td>
<td>Retain the default (Voice) or select Multichannel from the drop-down. Voice routes the script to Unified CVP. Multichannel routes the script to Unified E-Mail Information Manager/Web Interaction Manager (EIM/WIM) to support customer-agent chat.</td>
</tr>
<tr>
<td>VRU Script Name</td>
<td>yes</td>
<td>Enter the name of the script as it is known on the Unified CVP.</td>
</tr>
</tbody>
</table>
VRU Script Name Parameters

VRU Script Name parameters are positional, and take the format Micro_app acronym, parameter, parameter.

- The micro-application acronym is case-insensitive (Enter PM or pm.).
- Use double commas (,,) to skip a parameter; Unified CVP will supply the default.

The Play Media position sequence is PM, media file name, media library type, Uniqueness value.
The Play Data position sequence is PD, Data Playback Type, Uniqueness value.
The Get Digits position sequence is GD, media file name, media library type, Uniqueness value.
The Menu position sequence is M, media file name, media library type, Uniqueness value.

<table>
<thead>
<tr>
<th>Field</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Param</td>
<td>no</td>
<td>A string used by Unified CVP to pass additional parameters to the IVR Service. The content of the string depends on the micro-application to be accessed. The format is parameter, parameter, parameter.</td>
</tr>
<tr>
<td>Timeout</td>
<td>yes</td>
<td>Enter a number to indicate the number seconds for the system to wait for a response from the routing client after directing it to run the script. The default value is 180 seconds. Valid values are 1 to 2147483647.</td>
</tr>
<tr>
<td>Interruptible</td>
<td>no</td>
<td>Checked by default, this check box indicates whether or not the script can be interrupted; for example, when an agent becomes available to handle the call.</td>
</tr>
</tbody>
</table>

Step 3  Save the network VRU script to return to the List window, where a message confirms the successful creation. After you add a network VRU script, it is visible in the Script Editor Run External Script node. Processing this script node sends the network VRU script parameters to Unified CVP. Once the system establishes that the call has been successfully delivered, the Run VRU Script node executes, instructing Unified CVP to run the network VRU script and apply the call treatment.
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Used For</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Media File Name</strong></td>
<td>Options are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A file name - (for instance, a .wav file)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• (number 1-10) - Unified CVP plays the file in the corresponding Call.PeripheralVariable file.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For example, a value of 2 instructs Unified CVP to look at Call.PeripheralVariable2.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If you use the (number 1-10) option and set the Media Library Type to &quot;V,&quot; Unified CVP plays the external VoiceXML file specified in the corresponding Call.PeripheralVariable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If you set the value to (no value) and set the Media Library Type to &quot;A&quot; or &quot;S&quot;, the IVR Service creates VoiceXML without a media prompt.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• a - Unified CVP will automatically generate 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.</td>
<td></td>
</tr>
<tr>
<td><strong>Data Playback Type</strong></td>
<td>options are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Number</td>
<td>Play Data</td>
</tr>
<tr>
<td></td>
<td>• Char (character)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Etime (elapsed time)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• TOD (Time of Day)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 24TOD (24-hour Time of Day)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DOW (Day of Week)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Currency (USD only)</td>
<td></td>
</tr>
<tr>
<td><strong>Media Library Type Flag</strong></td>
<td>indicating the location of the media files to be played. Options are:</td>
<td>V is an option for PlayMedia only.</td>
</tr>
<tr>
<td></td>
<td>• A - (default) Application</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• S - System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• V - External VoiceXML</td>
<td></td>
</tr>
</tbody>
</table>

Packaged CCE Administration Guide for Release 9.0(x)
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Used For</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniqueness value</td>
<td>Optional. A string identifying a VRU Script Name as unique.</td>
<td>Play Media, Play Data, Get Digits, Menu</td>
</tr>
</tbody>
</table>

### Sample VRU Script Names

<table>
<thead>
<tr>
<th>This VRU Script Name</th>
<th>Tells Unified CVP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM,July,S</td>
<td>To use the Play Media (PM) micro-application to play the &quot;July.wav&quot; media file, using the System (S) Media library.</td>
</tr>
<tr>
<td>PM,Website,,1</td>
<td>To use the Play Media (PM) micro-application to play the &quot;Website.wav&quot; media file, using the default Media Type (Application library), and setting 1 as the Uniqueness value.</td>
</tr>
<tr>
<td>GD,Password,A,O</td>
<td>To use the Get Digits micro-application to play the media file named password.wav, using the Application (A) media library and setting 0 as the Uniqueness value.</td>
</tr>
<tr>
<td>M,Main_Menu</td>
<td>To use the Menu micro-application to play the media file named Main_Menu.wav.</td>
</tr>
</tbody>
</table>

### Configuration Parameters

Configuration parameters are positional, and take the format parameter,parameter,parameter. Use double commas (,,) to skip a parameter; Unified CVP will supply the default.

The Play Media position sequence is **Barge-in allowed, RTSP Timeout, Type-ahead Buffer Flush**.

The Play Data position sequence is **Location of files to be played, Barge-in allowed, Time Format, Type-ahead Buffer Flush**.

The Get Digits position sequence is **Minimum Field Length, Minimum Field Length, Barge-in allowed, Inter-digit Timeout, No Entry Timeout, Number of Invalid Tries, Timeout Message Override, Invalid Entry Message Override, Dtmf Termination Key, Incomplete Timeout**.

The Menu position sequence is **List of Menu Choices, Barge-in allowed, No Entry Timeout, Number of No Entry Tries, Number of Invalid Tries, Timeout Message Override, Invalid Entry Message Override**.
### UnifiedCVPhandlesbarge-inas

• If barge-in is not allowed, the SIP/H.323 Service/Gateway continues prompt play when a caller starts entering digits, and the entered digits are discarded.

• If barge-in is allowed, the H.323Service/Gateway discontinues prompt play when the caller starts entering digits.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Used For</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Barge-in Allowed</strong></td>
<td>Play Media</td>
<td>Unified CVP handles barge-in as follows:</td>
</tr>
<tr>
<td></td>
<td>Play Data</td>
<td>• If barge-in is not allowed, the SIP/H.323 Service/Gateway continues</td>
</tr>
<tr>
<td></td>
<td>Get tDigits</td>
<td>prompt play when a caller starts entering digits, and the entered</td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td>digits are discarded.</td>
</tr>
<tr>
<td></td>
<td>Get Digits</td>
<td>• If barge-in is allowed, the H.323Service/Gateway discontinues</td>
</tr>
<tr>
<td></td>
<td></td>
<td>prompt play when the caller starts entering digits.</td>
</tr>
<tr>
<td><strong>DTMF Termination Key</strong></td>
<td>Get Digits</td>
<td>Valid options are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0 to 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• * (asterisk)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• # (pound sign, the default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• N (no termination key)</td>
</tr>
<tr>
<td><strong>Incomplete Timeout</strong></td>
<td>Get Digits</td>
<td>The amount of time after a caller stops speaking to generate an</td>
</tr>
<tr>
<td></td>
<td></td>
<td>invalid entry error because the caller input does not match the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>defined grammar. The valid options are 0 to 9. The default is 3.</td>
</tr>
<tr>
<td><strong>Inter-digit Timeout</strong></td>
<td>Get Digits</td>
<td>The number of seconds the caller is allowed between entering digits.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If exceeded, the system times out.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The valid options are 1-99. The default is 3.</td>
</tr>
<tr>
<td><strong>Invalid Entry Message Override</strong></td>
<td>Get Digits</td>
<td>The valid options are:</td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td>• Y - override the system default with a pre-recorded Application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Media Library file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• N - (default) do not override the system default</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Used For</td>
<td>Notes</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>List of Menu Choices</strong> Valid options are:</td>
<td></td>
<td>- 0-9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- * (asterisk)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- # pound sign</td>
</tr>
<tr>
<td><strong>Location of the data to be played</strong> Valid options are:</td>
<td></td>
<td>- null (the default) - If you leave this option empty, the system uses the expanded call variable named user.microapp.play_data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- a number representing a Call Peripheral Variable number (for example, a 1 to represent Call.PeripheralVariable1).</td>
</tr>
<tr>
<td><strong>Maximum Field Length</strong> Maximum number of digits expected from the caller. The valid options are 1 to 32. The default is 1.</td>
<td>Get Digits</td>
<td></td>
</tr>
<tr>
<td><strong>Minimum Field Length</strong> Minimum number of digits expected from the caller. The valid options are 1 to 32. The default is 1.</td>
<td>Get Digits</td>
<td></td>
</tr>
<tr>
<td><strong>No Entry Timeout</strong> The number of seconds a caller is allowed to begin entering digits. If exceeded, the system times-out. The valid options are 0 to 99. The default is 5.</td>
<td>Get Digits Menu</td>
<td></td>
</tr>
<tr>
<td><strong>Number of Invalid Tries</strong> Unified CVP repeats the &quot;Get digits&quot; cycle when the caller enters invalid data. (Total includes the first cycle.) The valid options are 1 to 9. (The default is 3.)</td>
<td>Get Digits Menu</td>
<td></td>
</tr>
<tr>
<td><strong>Number of No Entry Tries</strong> Unified CVP repeats the &quot;Get Digits&quot; 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 to 9. (The default is 3.)</td>
<td>Get Digits Menu</td>
<td></td>
</tr>
<tr>
<td><strong>RTSP Timeout</strong> Specifies the Real-time Streaming Protocol (RTSP) timeout—in seconds—when RTSP is used. The valid range is 0 to 43200 seconds. The default is 10 seconds. If the value is set to 0 or a timeout value is not provided, the stream will not end.</td>
<td>Play Media</td>
<td></td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Used For</td>
<td>Notes</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td><strong>Time format</strong></td>
<td>Valid only for the time Data Playback types Etime, TOD, and 24TOD.</td>
<td>Play Data</td>
</tr>
<tr>
<td>The available formats are:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• null - leave this option empty for non-time formats</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• HHMM - default for time formats</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• HHMMSS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• HHMMAP - includes am or pm; valid only for TOD</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Timeout Message Override.</strong> The valid options are:</td>
<td>Get Digits Menu</td>
<td></td>
</tr>
<tr>
<td>• Y - override the system default with a pre-recorded Application Media Library file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• N - (default) do not override the system default</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Type-ahead buffer flush</strong></td>
<td>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. Valid options are:</td>
<td>Play Media Play Data</td>
</tr>
<tr>
<td>• Y—flush the type-ahead buffer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• N—(default) do not flush the type-ahead buffer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Sample Configuration Values

<table>
<thead>
<tr>
<th>This Configuration sequence</th>
<th>Tells Unified CVP</th>
</tr>
</thead>
</table>
| (for a Menu micro-application) 
0-2/9,,4,2,2 | To accept numbers 0, 1, 2, and 9. 
To set the no entry timeout value to 4 seconds. 
To allow 2 not entry tries. 
To allow 2 invalid tries. 
To accept all other defaults. |
| (for a Get Digits micro-application) 
GD,Password,A,O | To use the Get Digits micro-application to play the media file named password.wav, using the Application (A) media library and setting 0 as the Uniqueness value |
| (for a Menu micro-application) 
M,Main_Menu | To use the Menu micro-application to play the media file named Main-Menu.wav. |
CHAPTER 6

Bulk Jobs

Use bulk jobs to create and edit multiple individual agent and dialed number records in a single operation, based on the data specified in a content file.

Bulk jobs are a fast and efficient way to enter data at initial setup and to incorporate large-scale changes, such as changing agent skill groups between shifts and incorporating a new contact center with multiple new agents.

Individual changes, such as updating the last name for a newly-married agent or updating the call type for one or two dialed numbers, are best handled in the Agents or Dialed Numbers tools.

Although bulk job content files create agents and dialed number records explicitly, they also implicitly create related records. For example, an agent bulk job content file contains cells for agent teams, skill groups, and attributes. Entering content in those cells creates those objects if they do not exist. Similarly, a dialed number bulk job content file contains cells for call types and departments; entering those cells creates those objects if they do not exist.

Important

- Run bulk jobs during off-peak hours. Do not run bulk jobs during heavy call load.
- Administrators have full access to Bulk Job configuration. Supervisors have no access to this tool.

Manage Bulk Jobs

Bulk jobs are a fast and efficient way to enter data at initial setup and to incorporate large-scale changes, such as changing agent skill groups between shifts and incorporating a new contact center with multiple new agents.

Individual changes, such as updating the last name for a newly-married agent or updating the call type for one or two dialed numbers, are best handled in the Agents or Dialed Numbers tools.

Although bulk job content files create agents and dialed number records explicitly, they also implicitly create related records. For example, an agent bulk job content file contains cells for agent teams, skill groups, and attributes. Entering content in those cells creates those objects if they do not exist. Similarly, a dialed number bulk job content file contains cells for call types and departments; entering those cells creates those objects if they do not exist.
Run bulk jobs during off-peak hours. Do not run bulk jobs during heavy call load.
Supervisors have no access to this tool.

**Bulk Job Content Files**

Bulk jobs apply changes entered in content file templates. Content file templates are in .csv format.
To implement a content file for a bulk job:

1. Click **Download** on the List of Bulk Jobs page.
2. Open the template you want in Microsoft Excel (the Agents Template or the Dialed Numbers Template).
3. Populate the file.
4. Save the populated file locally.

The content file is syntactically validated before the bulk job is created. Database related errors and conflicts are reported during execution of the job.

**Content File Rules**

### Content File Create Operations

The content file spreadsheets follow these CREATE rules:

- All columns in the spreadsheet must be present, but the cells for optional fields can be left blank.
- Rows in the file are processed sequentially. It is possible for a content file to fail at any point (at any row), in which case objects up to but not including that row are added or updated.
- All additions or updates before that row succeed, but all subsequent create and update operations fail.
- Creating an agent with these cells populated implicitly creates the objects if they do not exist — agent team, skill group, attributes, supervisor team, and department.
- Creating a dialed number with the call type and department populated implicitly those objects, if they do not already exist.

### Content File Update Operations

The Content file spreadsheets follow these UPDATE rules:

- Enter a value in a field to change the existing record.
- Leave a field blank to keep the existing record.
- Enter ~ in a field to flush the existing record.
## Bulk Agent Content Plan

The content file for the agent bulk job has these fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>operation</strong> (Required)</td>
<td>CREATE or UPDATE (case-insensitive)</td>
</tr>
<tr>
<td><strong>agentID</strong> (Optional)</td>
<td>Enter a unique string of up to 11 digits. AgentID is auto-generated if you leave the field blank.</td>
</tr>
<tr>
<td></td>
<td>In an UPDATE operation:</td>
</tr>
<tr>
<td></td>
<td>• you cannot change agentID</td>
</tr>
<tr>
<td></td>
<td>• if you leave the field blank, the loginName must reference an existing agent</td>
</tr>
<tr>
<td><strong>userName</strong> (Required)</td>
<td>Enter up to 32 alphanumeric characters as the username for this agent.</td>
</tr>
<tr>
<td></td>
<td>In an UPDATE operation:</td>
</tr>
<tr>
<td></td>
<td>• if you enter leave this field blank, the agentID must reference an existing agent</td>
</tr>
<tr>
<td></td>
<td>• if the agentID references an existing agent, you can change the userName</td>
</tr>
<tr>
<td><strong>firstName</strong> (Required)</td>
<td>Enter a maximum of 32 characters.</td>
</tr>
<tr>
<td><strong>lastName</strong> (Required)</td>
<td>Enter a maximum of 32 characters.</td>
</tr>
<tr>
<td><strong>password</strong> (Optional)</td>
<td>Enter a maximum of 32 characters. Password is case-sensitive. There are no size restrictions unless Minimum Password Length has been configured in Settings.</td>
</tr>
<tr>
<td><strong>loginEnabled</strong> (Optional)</td>
<td>Indicates whether the agent is able to log in to the agent desktop. If not specified, defaults to true.</td>
</tr>
<tr>
<td><strong>description</strong> (Optional)</td>
<td>Enter up to 255 characters to describe the agent. If description is left blank on a create, it is set to &quot;BulkJob ID ####&quot; where #### is the ID of the bulk job performing the creation.</td>
</tr>
<tr>
<td><strong>agentStateTrace</strong> (Optional)</td>
<td>Indicates whether agent state trace enabled for this agent. Defaults to False.</td>
</tr>
<tr>
<td><strong>agentDeskSettingsName</strong> (Optional)</td>
<td>Enter the desk settings associated with this agent. In a CREATE operation, your entry of agentDeskSettingsName generates an error when there is no desk settings with that name. Leaving this blank applies the System Default Desk Settings.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>agentTeamName (Optional)</td>
<td>Enter the team in which this agent is a member. In a CREATE operation, your entry of agentTeamName creates that team if it does not already exist. It appears in the List of Teams with the description BulkJob ID ####, where #### is the number of the bulk job.</td>
</tr>
<tr>
<td>skillgroup(s) (Optional)</td>
<td>Enter the skill groups with which this agent is associated, delimited by the &quot;;&quot; character. For example: sales;billing;support. In a CREATE operation, your entry of skillgroup creates that skill group if it does not already exist. It appears in the List of Skill Groups with the description BulkJob ID ####, where #### is the number of the bulk job. Leaving this blank in an UPDATE operation removes all skill group associations for the agent.</td>
</tr>
<tr>
<td>defaultSkillGroup (Optional)</td>
<td>Enter the default skill group associated with this agent. If the field is specified, it must reference a skill group defined for the agent. In an UPDATE operation, an error is generated if the value is no longer one of the agent's skill groups. Leaving this blank applies the System Default Skill Group.</td>
</tr>
<tr>
<td>attributes (Optional)</td>
<td>These fields are name = value pairs delimited by the &quot;,&quot; character, where = value is optional for existing attributes. For example, english=true;sales=7. Adding an attribute with a data type (Boolean or Proficiency) and a value (true or 9), either directly in the Attributes tool or with a bulk job, defines and protects the data type and establishes that value as the default. If an attribute does not yet exist in the Attributes tool, entering an attribute name without a value generates an error. For example if english is not yet an attribute, then english returns an error. You cannot change the data type, but you can change the value. If english was created as true, entering english retains the true value in a bulk update. You can also enter english=false, which sets the agent attribute value to false, leaving the attribute default value at true. You cannot enter english=10. To clear an agent's attribute value and reestablish the attribute default on a bulk update, just specify the attribute name; for example, english. In a CREATE operation, your entry of attribute creates that attribute if it does not already exist. It appears in the List of Attributes with the description BulkJob ID ####, where #### is the number of the bulk job.</td>
</tr>
<tr>
<td>supervisorUsername (Optional)</td>
<td>Enter an existing active directory username. When this field is valid, the agent's supervisor flag is set to true. An invalid entry or a blank field sets the agent's supervisor flag to false.</td>
</tr>
<tr>
<td>domainName (Optional)</td>
<td>Enter the active directory domain name in which the supervisorUsername exist. If this field is blank, the system populates it with the default domain name.</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
supervisorTeams (Optional) | Enter names of teams that will be supervised by this supervisor, delimited by the ";" character. For example: team1;team2;team3. Populating this field but leaving supervisorUserName blank generates an error.
In a CREATE operation, your entry of supervisorTeams creates that team if it does not already exist. It appears in the List of Teams with the description Bulk Job ID: ####, showing the number of the bulk job.
In an UPDATE operation, if this field is left blank, the agent is removed as a supervisor for any teams.

Bulk Dialed Number Content Plan

The content file for the dialed number bulk job has these fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>operation (Required)</td>
<td>CREATE or UPDATE (case-insensitive)</td>
</tr>
</tbody>
</table>
| dialedNumberString (Required) | The dialedNumberString for this dialed number. Enter a string value that is unique for the routing type, using a maximum of 25 characters. Valid values are alphanumeric, +, and @. You cannot edit a dialedNumberString in an UPDATE operation.
You cannot update dialedNumberString. |
| routingType (Required) | The routing type for this dialed number. Values are:
- 1 (External Voice) Select this for dialed number strings that apply to voice calls coming from Cisco Customer Voice Portal (CVP).
- 2 (Internal Voice) Select this for dialed number strings that can be called from a Cisco Unified Communications Manager (CUCM) phone.
- 3 (Outbound Voice) Select this for dialed number strings that are used by the Cisco Outbound Option Dialer.
- 4 (Multichannel) Select this for dialed number strings that are routed to an agent who must interact with a customer by email or by web chat.
You cannot update routingType. |
| description (Optional) | The description for this dialedNumberString. Enter a maximum of 255 characters. There is no restriction on characters. If the description field is left blank on a create, it is set to "BulkJob ID ####" where #### is the ID of the bulk job performing the creation. |
## Add and Maintain Bulk Jobs

Navigate to the **Unified CCE Administrator Manage > Other > Bulk Jobs** interface to maintain (Add, Review, and Delete) bulk jobs. This opens a *List of Bulk Jobs* window.

### Procedure

**Step 1** Click **New** to open the *New Bulk Job* window.

**Step 2** In the optional **Description** fields, enter up to 255 characters to describe the bulk job. See *Native Character Sets*, on page 13.

**Step 3** In the required **Content File** field, browse to the content file you have completed for this bulk job. The content file is validated before the bulk job is created.

**Step 4** Click **Save**.

### Related Topics

- **Agents**, on page 15
- **Dialed Numbers**, on page 53

### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callTypeName (Optional)</td>
<td>Enter a name for the call type using a maximum of 32 characters. Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric. In a CREATE operation, your entry of callTypeName creates that call type if it does not already exist. It appears in the List of Call Types with the description BulkJob ID ####, where #### is the number of the bulk job.</td>
</tr>
<tr>
<td>mediaRoutingDomainName</td>
<td>Optional for routingTypes 1, 2, and 3. If supplied, must be Cisco_Voice. Required for routingType 4 (Multichannel). Must be defined in the Configuration Manager tool (Media Routing Domain List). In a CREATE operation, your entry of mediaRoutingDomainName generates an error when there is no media routing domain with that name.</td>
</tr>
</tbody>
</table>
Review Bulk Job Details

To review the details for a bulk job, click the bulk job ID link on the List of Bulk Jobs page. With the exception of Content File Download and Log File Download, fields on the page are display-only.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID, Description, and Type</td>
<td>Show the ID, description entered and type of bulk job selected when the bulk job was created.</td>
</tr>
<tr>
<td>State</td>
<td>Shows one of:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Queued</strong>: The bulk job has been queued and will process when any jobs submitted ahead of it have completed. When multiple bulk jobs are submitted, they are run in the order they are created.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Processing</strong>: The bulk job is being processed. To view the progress, click Log File Download to monitor the log file.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Succeeded</strong>: All CREATE and UPDATE operations in the bulk job were successful.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Partially Succeeded</strong>: Some CREATE or UPDATE operations were successful, and some were unsuccessful.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Failed</strong>: All CREATE and UPDATE operations were unsuccessful.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Cancelled</strong>: A bulk job is cancelled when another bulk job failed while this job was in the queued state</td>
</tr>
<tr>
<td>Host</td>
<td>The host name of the Administration and Data server where the bulk job was initiated and will be stored. When deleted, bulk job content files and log files will be deleted from this host.</td>
</tr>
<tr>
<td>Created</td>
<td>The time the bulk job was submitted.</td>
</tr>
<tr>
<td>Started</td>
<td>The time the bulk job entered the processing state.</td>
</tr>
<tr>
<td>Finished</td>
<td>The time the bulk job completed or failed (left the processing state).</td>
</tr>
<tr>
<td>Total Time</td>
<td>The time the bulk job spent in the processing state. This is calculated as Finished - Started.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Content File</td>
<td>Click Download to open the Content csv file that was submitted for this bulk job. You must authenticate to open or save this file. If your deployment includes two Administration and Data server hosts, this button is disabled if the bulk job was created using Unified CCE Web Administration on a host that is different from the host on which the job is being viewed.</td>
</tr>
<tr>
<td>Log File</td>
<td>Click Download to open the log file for this bulk job. If the job is still processing, click Download again to the review updates the job progresses. You must authenticate to open or save this file. If your deployment includes two Administration and Data server hosts, this button is disabled if the bulk job was created using Unified CCE Web Administration on a host that is different from the host on which the job is being viewed. A log file is generated for each bulk job. The log file is retained until the bulk job is deleted and contains detail of each operation that was executed, as well as a summary indicating if the bulk job completed successfully or had failures.</td>
</tr>
</tbody>
</table>
CHAPTER 7

Settings

For Packaged CCE, Manage > Settings has two tabs: General and Access.

This section explains the tools on the General tab, which you use to manage these settings: System Information, System Settings, System Deployment, and Agent Trace.

- Configure Deployment, page 75
- Capacity Info, page 76
- System Validation, page 77
- Agent Trace, page 77
- Packaged CCE Logs, page 78

Configure Deployment

Configure Deployment is an option on the Settings menu.

Administrators might access Configure Deployment for these reasons:

- To review and change deployment type
- To review Congestion Treatment Mode and call capacity

Fields on this window vary per deployment type.

Deployment Type

This field shows the deployment type that is currently identified with your system. Click the icon to select a different Deployment Type from the drop-down.

There are over a dozen possible deployment types. Each has its own features, capacity, and configuration limits. System deployment type is initially configured by the first administrator who signs in. An administrator can subsequently change the deployment type on this tool.
Changing deployment type can have significant impact on call processing capacity, configuration limits, and the features and configuration tools you can access. You should fully understand a deployment type change before you proceed.

**Important**

**Congestion Treatment Mode** For deployment type Packaged CCE: CCE-PAC-M1, this field shows Terminate with Dialog Fail/RouteEnd.

**System Default Label** For deployment type Packaged CCE: CCE-PAC-M1, this field is blank.

**Capacity** This field displays the current value for maximum calls per second for the deployment. For deployment type Packaged CCE: CCE-PAC-M1, this field is set to 8 and cannot be changed.

### Deployment Types

These are the deployment types that you can select from the Deployment Type drop-down list.

<table>
<thead>
<tr>
<th>Deployment Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Specified - This is an initial deployment type when none has been selected.</td>
</tr>
<tr>
<td>UCCE 12000 Agents Router/Logger</td>
</tr>
<tr>
<td>UCCE 8000 Agents Rogger</td>
</tr>
<tr>
<td>UCCE 4000 Agents Rogger</td>
</tr>
<tr>
<td>UCCE 450 Agents Progger</td>
</tr>
<tr>
<td>HCS-CC 4000 Agents</td>
</tr>
<tr>
<td>HCS-CC 1000 Agents</td>
</tr>
<tr>
<td>ICM Rogger</td>
</tr>
<tr>
<td>ICM Router/Logger</td>
</tr>
<tr>
<td>NAM</td>
</tr>
<tr>
<td>NAM Rogger</td>
</tr>
<tr>
<td>Packaged CCE: CCE-PAC-M1</td>
</tr>
<tr>
<td>Packaged CCE: CCE-PAC-M1 Lab Only</td>
</tr>
<tr>
<td>IVR-ICM</td>
</tr>
</tbody>
</table>

### Capacity Info

Open this tab to see a table with information showing capacity information:
Status The status column shows how your system stands with respect to the capacity limit. The status icons are:

- Green for 0-75% of capacity.
- Yellow for 76-95%.
- Orange for 96-99%.
- Red when you are at 100%.

**Number of Configured** ... Shows the name of the object.

**At Most** Shows the maximum capacity of each configurable object that Packaged CCE allows.

**Actual** Shows the number of objects currently configured on your system.

**% Used** Shows the percentage of the maximum represented by your configuration.

The arrow icon at the right of each row opens the page where you can view, add, edit, and delete the objects to maintain the capacity.

---

**System Validation**

Packaged CCE has been qualified to operate within certain software specifications.

This display-only tab shows the status of your system with regard to each specification rule, a description of the rule, and the actual value on your system. You can sort all rows in ascending/descending order.

A status icon of red indicates that your system is invalid (it either exceeds or is not set up to satisfy one of these rules). You cannot work in Packaged CCE or access its tools until you correct the underlying software specifications. If your goal is to redeploy an existing system as a Packaged CCE system, this page will assist partners and Cisco Technical Support in bringing your system to a valid state.

---

**Agent Trace**

Enabling agent trace allows you to track and report on every state an agent passes through. You might do this if you have concerns about the productivity or performance of one or more agents.

---

**Important** Enabling trace can impact system performance, as it requires additional network bandwidth and database space. Typically, you use this feature for short-term tracking of specific agents. The system imposes a configuration limit on the number of agents for whom you can enable trace.

---

Agent Trace is an option on the Settings menu.

Open this tool to view, add, and remove agents for whom agent trace is enabled.

To add trace to an agent:

1. Click the Add magnifying glass to display Add Agents with Trace Enabled. Use the sort and search features to navigate the list.

2. Click one or more agent usernames to give them the trace-enabled status.
3 Close Add Agents with Trace Enabled to return to the list.
4 Click Save on the List window to confirm the trace status for the agents you added. Click **Revert** before you save to remove an agent from the Trace Enabled list.

To remove trace from an agent:
1 On the List of Agents with Trace Enabled window, locate the agent whose trace status you want to remove.
2 Click x to clear trace status for that agent.
3 Click **Save** on the List window to confirm the removal. To cancel, click **Revert**.

## Packaged CCE Logs

You can download two types of Packaged CCE log files from the Unified CCE Administration interface.

### System Validation Logs

When you configure your deployment, if either the server or any of the VMs do not meet requirements, you see a message indicating connection problems. This message has a link to a log file. Open this file to see whether the servers are valid and whether all VMs match the deployment profiles.

#### Sample log file showing summary of invalid results:

```
VM Validation Results: Wed Aug 20 08:05:36 EDT 2012
Overall: false
Valid Systems: 0 of 1
Summary:
ESX Server: sideB
ESX Server Properties Valid: true
VM Layout Valid: false.
```

The information at the top of the log is a summary of the results. This log shows that the server is valid but the VM layout is not.

#### Sample log showing invalid server

This shows that the server does not have the required number of CPU Cores.

```
Server Result:
Required Version: 5.0.0
Required Min CPU Cores: 20
Required Min Memory (MB): 95000
Required HD(s) (GB): [1392, 1949, 273]
Required Bios <Major version>: C260
Required Vendor: Cisco Systems Inc
Found Version: 5.0.0
Found CPU Cores: 10
Found Memory (MB): 98185
Found HD(s) (GB): [1392, 273, 1949]
Found Bios: C260.1.4.2b.0.102620111637
Found Vendor: Cisco Systems Inc
```

There are three log entries for invalid VMs:

- **Required Profiles without Matching Virtual Machines**
  
  This means system does not have VMs present that match our requirements

- **Optional Profiles without Matching Virtual Machines**
  
  This means that the CVP Reporting profile, which is defined as optional, does not exist on the system. This does not block validation.
• Virtual Machines without Matching Profiles

This means the system has VMs that do not match requirements. They might be extra VMs or incorrectly-configured VMs.

Sample log showing valid VM

Virtual Machines Matching Defined Profiles:
VM: BB-CCE-DataSvr-A
Profile: Unified CCE Data Server
OS: Microsoft Windows Server 2008 R2 (64-bit)
CPU Cores: 4
Reservation: 5100
RAM (MB): 8192
HD(s) (GB): [80, 750]
VMware Tools Version: 8384

Bulk Job Logs

A log file is generated for each bulk job. The log file is retained until the bulk job is deleted and contains detail of each operation that was executed, as well as a summary indicating if the bulk job completed successfully or had failures.

To open the log:

1. Open the Bulk Jobs tool.
2. From the List of Bulk Jobs, click the ID to go to the View Bulk Job page.
3. Click Log File Download. If the job is still processing, click Download again to review updates the job progresses.

You must authenticate to open or save this file.

If your deployment includes two Administration and Data server hosts, Download is disabled if the bulk job was created using Unified CCE Web Administration on a host that is different from the host on which the job is being viewed.

Sample log file:

```
2012-08-03T14:31:05-04:00 - Job created
2012-08-03T14:31:05-04:00 - Job started
2012-08-03T14:31:05-04:00 - Processing line 1: Header
2012-08-03T14:31:05-04:00 - Processing line 2: operation=create, agentId=1000, userName=asmith, firstName=Agent, lastName=Smith, password=secret, loginEnabled=true, description=Agent Smith, agentStateTrace=false, agentDeskSettingsName=Default_Agent_Desk_Settings, attributes=, supervisorUserName=, domainName=, supervisorTeams=, skillGroups=sg1;sg2, defaultSkillGroup=sg1,
2012-08-03T14:31:06-04:00 - Created /unifiedconfig/config/agentteam/5097
2012-08-03T14:31:08-04:00 - Created /unifiedconfig/config/skillgroup/12867
2012-08-03T14:31:09-04:00 - Created /unifiedconfig/config/skillgroup/12868
2012-08-03T14:31:11-04:00 - Created /unifiedconfig/config/agent/12869
2012-08-03T14:31:11-04:00 - Processing line 3: operation=update, agentId=, userName=neo, firstName=Mister, lastName=Anderson, password=passw0rd, loginEnabled=true, description=Neo, agentStateTrace=true, agentDeskSettingsName=, agentTeamName=, skillGroups=, defaultSkillGroup=, attributes=kungFu=9;actuallyKnowsKungFu=false, supervisorUserName=neo, domainName=foo.com, supervisorTeams=team1;team2
2012-08-03T14:31:11-04:00 - Error processing line 3: agentUserName: The specified agent userName does not exist neo.
2012-08-03T14:31:11-04:00 - Processing line 4: operation=update, agentId=1001, userName=, firstName=, lastName=, password=, loginEnabled=, description=, agentStateTrace=, agentDeskSettingsName=, agentTeamName=, skillGroups=, defaultSkillGroup=, attributes=, supervisorUserName=, domainName=, supervisorTeams=
2012-08-03T14:31:11-04:00 - Error processing line 4: agentId:
```
The specified agent Id does not exist 1001.
2012-08-03T14:31:11-04:00 - Job partially completed due to errors
2012-08-03T14:31:11-04:00 - 3 lines processed, 1 succeeded, 2 failed
2012-08-03T14:31:11-04:00 - 1 agents created, 2 skill groups created,
1 agent teams created
You perform most Packaged CCE configuration with the Unified CCE Web Administration gadgets. Limited configuration is done in the legacy Configuration Manager toolset. This section explains the tools in Configuration Manager, how to access them, and why you might need them for Packaged CCE.

- Access Configuration Manager, page 81
- Configuration Manager Tools for Call Flow, page 82
- Configuration Manager Tools for Global Values and Actions, page 85
- Configuration Manager Tools for Multichannel, page 87
- Configuration Manager Tools for Scripting, page 90
- Configuration Manager Tools to Control User Access, page 91

**Access Configuration Manager**

Configuration Manager is a suite of tools installed with both CCE Data Server Virtual Machines and opened from a desktop icon on those VM consoles.

Although most configuration is accomplished with the Packaged CCE administration tools, you must use Configuration Manager for some configuration.

**How to Access Configuration Manager**

1. Open the VM Console for Packaged CCE. Select the Virtual Machine for either of the CCE Data Servers that are installed for your Packaged CCE deployment.
2. Click the Console tab.
3. Log in to the Data Server if necessary.
4. Select **All Programs -> Cisco Unified CCE Tools -> Administration Tools -> Configuration Manager**.
5. Select the type of tool indicated in the procedure you need to perform. This example shows the selection of Media Class List.
Configuration Manager Tools for Call Flow

Access these tools in Configuration Manager to facilitate or troubleshoot call flow.

Related Topics

- PG Explorer, on page 82
- Agent Targeting Rule, on page 84

PG Explorer

PG Explorer is listed in Unified CCE Configuration Manager under the Explorer Tools. Open the tool and click Retrieve to see the two peripheral gateways that are configured for Packaged CCE: the Generic_PG and the MR_PG. The Generic_PG has five Peripheral Interface Managers (PIMs)—one for Unified Communications Manager and four for the Voice Response Units (VRUs).
To retain the Packaged CCE: CCE-PAC-M1 deployment type, do not delete either PG and do not add other PGs. The MR-PG is required, even if your contact center does not used multichannel (email and web).

**Related Topics**

- Change Default Desk Settings, on page 83
- Enable Agent Reporting

**Change Default Desk Settings**

Follow these steps to locate and change the default desk settings configuration that appears in Unified CCE Web Administration tools.
Procedure

Step 1 From Configuration Manager, select **Tools > Explorer Tools > PG Explorer**.
Step 2 Select the GenericPG.
Step 3 Select CUCM_PG_1.
Step 4 On the Peripheral tab, locate the Default desk settings field. The first selection on the drop-down shows the option for the current default agent desk settings.
Step 5 Select a different agent desk settings from the drop-down. Your selection moves to the top of the drop-down and becomes the new default desk settings.
Step 6 Click **Save**.

Related Topics

- Desk Settings, on page 22

Agent Targeting Rule

To configure call routing in Packaged CCE, you must create one or more Agent Targeting Rules to specify the range of agent extension ranges. Based on agent targeting rules, if an agent attempts to log in to an extension to which the router cannot target a call, the peripheral gateway rejects the login request and returns an error that includes why the login request failed.

You can define one or more rules for a peripheral. However, each rule must cover a different agent extension range for the same routing client. In other words, if there are multiple rules that a routing client can use to target a peripheral, there must be no overlapping extension ranges in those rules.

Procedure

Step 1 From either CCE Data Server, navigate to **Unified CCE Administration Manager > Configuration Manager**.
Step 2 Select **Tools > List Tools > Agent Targeting Rule**.
Step 3 Click **Retrieve**. Then click **Add**.
Step 4 In the Attributes dialog box:
   a) Name the rule
   b) For Peripheral, select CUCM_PG_1.
   c) For Rule type, select Agent Extension.
   d) In the Routing client panel, select all four CVP clients—the two CVP_PGs on Side A and the two CVP_PGs on Side B.
   e) In the Extension Ranges panel, click Add to enter the Low and High extensions for the agents. Then click **OK**.
Step 5 Click **Save**.
What to Do Next

The CCE Call Server can only use an Agent Targeting Rule if *Agent targeting mode: Rule Preferred* is configured for the CUCM_PG_1 in PG Explorer (Advanced tab). This is set by default. If your Agent Targeting Rule does not work, verify that the configuration is still Rule Preferred.

Related Topics

PG Explorer, on page 82

Configuration Manager Tools for Global Values and Actions

This section explains Configuration Manager tools you can use to perform for system-wide actions.

System Information, on page 85
Permanent Deletion, on page 87

System Information

The System Information tool in Configuration Manager allows you to define global values.

Minimum Password Length

To configure a minimum password length for agents:
1. Launch Configuration Manager

2. Select **Miscellaneous Tools > System Information**.

3. Locate the Minimum password length field in the Person Security panel. Enter a value from 0 to 32.

4. Click **Save**. The length you configure becomes the required password length for agents.

**Service Threshold Length**

This field in the Call Types tool defaults to the value that is currently configured in System Information. It is the number of seconds that a caller spends in a queue before being connected to an agent. To change it:

1. Launch Configuration Manager.

2. Select **Miscellaneous Tools > System Information**.

3. Locate the Service level threshold field in the Call Type panel. It is a number. To change it, enter a different number.

4. Click **Save**. The value you enter becomes the system default target number of seconds that a caller spends in a queue before being connected to an agent.

**Service Level Type**

This field in the Call Types tool defaults to the setting that is currently configured in System Information. One of three options is set here to determine how the system software calculates the service level for the service for abandoned calls.

- Are they ignored, which means they have no impact on service level calculation?
- Do they have negative impact, which means only calls that are answered within the Service level threshold time are counted as treated calls?
- Do they have positive impact, which means abandoned calls will have a positive impact on the Service Level?

1. Launch Configuration Manager.

2. Select **Miscellaneous Tools > System Information**.

3. Locate the Service level type drop-down in the Call Type panel. The drop-down has three options, and the option that appears at the top is the current default. To change it, select another option.

4. Click **Save**. The option you chose becomes the new system default service level type.

**System Bucket Interval**

This field in the Call Type panel establishes which set of time ranges ("buckets") is used to capture call-handling activity.

The software is installed with one system bucket interval, but once you create custom bucket intervals, you can make one of them the default.

To do this:

1. Launch Configuration Manager.

2. Select **Miscellaneous Tools > System Information**.
3 Locate the Bucket intervals drop-down in the Call Type panel. The first option on the list is the current default. This option also appears in the Bucket Intervals gadget in Unified CCE Web Administration and serves as the system default until you change it here.

4 To change it, use the drop-down to select a different bucket interval.

5 Click Save. The bucket interval you selected becomes the first option in the drop-down and is now the new system default bucket interval.

Related Topics

Bucket Intervals, on page 49
Add and Maintain Agents, on page 16
Add and Maintain Call Types, on page 51

Permanent Deletion

Some objects are only "marked for deletion" in Unified CCE Administration. They remain in the system for reporting and record-keeping purposes. To delete them permanently:

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Launch Configuration Manager.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Select Using the Configuration Manager &gt; Deleted Objects.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Click the table name for the object you want to delete. This opens a panel showing all records for that table that have been marked for deletion.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Select one, several, or all records.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Click Delete Permanently.</td>
</tr>
</tbody>
</table>

Configuration Manager Tools for Multichannel

This section explains the Configuration Manager tools applicable to multichannel. Multichannel refers to the routing of calls to non-voice applications: email or web chat, also known as Email Interaction Manager (EIM) and Web Interaction Manager (WIM).

Media Class List, on page 88
Application Instance List, on page 89
Application Path List, on page 90
Media Routing Domain List, on page 88

Related Topics

Overview of Multichannel Services, on page 136
Media Class List

The Media Class List shows these Media Classes. The first four are not supported for Packaged CCE:

- Cisco_Blended_Collaboration
- Cisco_Email
- Cisco_Multi_Session_Chat
- Cisco_Single_Session_Chat
- Cisco_Voice
- CIM_BC
- CIM_EIM
- CIM_OUTBOUND
- CIM_WIM

You cannot delete or edit the attributes for Cisco_Voice. You can delete and edit attributes for CIM_BC, CIM_EIM, CIM_OUTBOUND, and CIM_WIM. You can add Media Classes.

Procedure

Step 1 Launch Configuration Manager and navigate to Tools > List Tools > Media Class List.
Step 2 Click Retrieve.
Step 3 Select a Media Class to review or edit its attributes.
Step 4 Add or delete Media Classes as appropriate.
Step 5 Save the configuration.

Media Routing Domain List

If you plan to use multichannel routing, you must configure one or more Media Routing Domains (MRDs) for them. The system uses MRDs to organize how requests from different media are routed to agents. After you add them in Configuration Manager, these MRDs will populate the Select Media Routing Domain popup window in the Dialed Number tool. If MRDs have already been configured, you might need to access this tool to change settings such as Max time in queue or Service Level Threshold.
Procedure

Step 1 Launch Configuration Manager and navigate to Tools > List Tools > Media Routing Domain List.
Step 2 Click Retrieve. Cisco_Voice is configured by default.
Step 3 Click Add.
Step 4 Enter the name of the MRD.
Step 5 From the Media Class drop-down menu, select any of the CIM_BC, CIM_WIM, or CIM_EIM media classes (or a custom media class if you have added one) depending on the purpose of the MRD. Typically you would have at least one MRD for each media class.
Step 6 The MRDomainID is a required read-only field. (An ID number is automatically created when you save your entry.)
Step 7 The Service Level Threshold value is the system-default that is used when in configuring a skill group. See Add and Maintain Skill Groups, on page 44.
Step 8 Enter required fields. Refer to the online help for field explanations.
Step 9 Save the configuration.

Application Instance List

If you plan to use multichannel routing, you must configure an application instance in Configuration Manager to allow the Email and Web applications to access Unified CCE. In adding an application instance, define an application ID and application key (password) that identifies the application. You need to enter the same information on the Email and Web application.

Procedure

Step 1 Launch Configuration Manager and navigate to Tools > List Tools > Application Instance List.
Step 2 Click Retrieve. Then click Add.
Step 3 Enter a name for the Application Instance.
Step 4 Enter and confirm the Application key.
Step 5 Select Other from the Application type drop-down menu.
Step 6 Select Full read/write from the Permission level drop-down menu.
Step 7 Save the configuration.

What to Do Next
Configure the Application Path for the Application Instance

Related Topics
Application Path List, on page 90
Application Path List

If you plan to use multichannel routing, you must configure an application path to associate the Application Instance with peripheral gateway (PG). For Packaged CCE, the PG for the multichannel application instance is CUCM_PG in the Generic_PG.

Procedure

Step 1 Launch Configuration Manager and navigate to Tools > List Tools > Application Path List.
Step 2 Click Retrieve. Then click Add.
Step 3 From the Application instance drop-down menu, select the Application Instance you have created for multichannel.
Step 4 From the Peripheral gateway drop-down menu, select Generic_PG. The Name field populates with Generic_PG.YourApplicationInstanceName. You can change it.
Step 5 Enter an optional Description.
Step 6 Click Add. This adds a row to the Application Path Members table that defaults to the first peripheral and the first MRD. Add a record for each multichannel MRD associated to the CUCM_PG_1 peripheral.
To change the MRD selection in the row, click on the MRD column and then click on the arrow that appears to the right of the row, which displays a drop-down list of all possible MRDs. Click on the one you want to select.
Step 7 Save the configuration.

Configuration Manager Tools for Scripting

This section explains the Configuration Manager tool applicable to scripting:

• Script Reference, on page 90
• User Variable List, on page 91

Related Topics
Scripting with Packaged CCE, on page 95

Script Reference

The Script Reference tool is an interface for identifying which scripts reference which objects. Not all target types are applicable to Packaged CCE.
To access this tool, navigate to Configuration Manager > Miscellaneous Tools > Script Reference.
Select a Target type to see the scripts where that target is referenced.
User Variable List

User Variables are used in scripting for routing calls. After you have defined a user variable, you can then use the Script Editor Formula Editor to access the variable and reference it in expressions.

User variables can be:

- Associated with object types, such as Call Types or Skill Groups. This enables the system software to maintain an instance of that variable for each object of that type in the system.
- Identified as persistent (retain value across CallRouter restarts)
- Identified as non-persistent (do not retain value across CallRouter restarts)

**Procedure**

1. Launch the Configuration Manager and select **Tools > List Tools > User Variable List**.
2. Click Retrieve. Then click **Add**.
3. Complete the Attributes property tab. Variable name, Object type, and Data type fields are required. All other fields are optional.
4. Click **Save**.

Configuration Manager Tools to Control User Access

There are two tools used to limit access Packaged CCE and to Unified CCE Configuration Manager, and you must use both of them.

**Related Topics**

- Feature Control Set List, on page 91
- User List Tool, on page 92

Feature Control Set List

Use this tool to create a Feature Control set. Once you create a feature control set, you can access the User List Tool to associate that set with users to determine which tools they can access.

**Procedure**

1. Launch **Configuration Manager > List Tools > Feature Control Set List**.
2. Click **Retrieve**; then click **Add**.
3. Enter a name and a description for the new Feature Control Set.
4. Establish access to tools by marking checkboxes for the application names on the feature set. Deny access by leaving the boxes unchecked.
Note that most application names on the Feature Control Set List do not correspond to Packaged CCE tools. This table calls out the ones that apply.

<table>
<thead>
<tr>
<th>To enable this Packaged CCE tool:</th>
<th>Check this application name in the Feature Control Set List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agents (UI and API)</td>
<td>Agent Explorer</td>
</tr>
<tr>
<td>Attributes (UI and API)</td>
<td>Attribute</td>
</tr>
<tr>
<td>Desk Settings (UI and API)</td>
<td>Agent Desk Settings List</td>
</tr>
<tr>
<td>Precision Queues (UI and API)</td>
<td>Precision Queue</td>
</tr>
<tr>
<td>Skill Groups (UI and API)</td>
<td>Skill Group Explorer</td>
</tr>
<tr>
<td>Teams (UI and API)</td>
<td>List Agent team</td>
</tr>
<tr>
<td>Reason Codes (UI and API)</td>
<td>Reason Code List</td>
</tr>
<tr>
<td>Bucket Intervals (UI and API)</td>
<td>Bucket Intervals List</td>
</tr>
<tr>
<td>Call Types (UI and API)</td>
<td>Call Type List</td>
</tr>
<tr>
<td>Dialed Numbers (UI and API)</td>
<td>Dialed Number/Script Selector List</td>
</tr>
<tr>
<td>Expanded Call Variables (UI and API)</td>
<td>Expanded Call Variable List</td>
</tr>
<tr>
<td>Network VRU Scripts (UI and API)</td>
<td>Network VRU Script List</td>
</tr>
<tr>
<td>Bulk Jobs (UI and API)</td>
<td>Dialed Number Bulk Edit AND Agent Bulk Edit</td>
</tr>
<tr>
<td>Deployment, Configure Deployment, and Agent Trace (UI)</td>
<td>System Information</td>
</tr>
</tbody>
</table>

**User List Tool**

With this tool, you can associate Feature Control Set with a user and/or limit access to "read-only." These are independent options.
Procedure

**Step 1** To assign a feature control set:

a) Launch **Configuration Manager > List Tools > User List**.

b) Click **Retrieve**.

c) Select the user you want to limit, or add the user to the list.

d) Select the feature control set from the drop-down list. The user will be able to access the applications you checked for that feature control list. The user will not be able to access the unchecked applications.

**Step 2** To limit access to "read-only":

a) Launch **Configuration Manager > List Tools > User List**.

b) Click **Retrieve**.

c) Select the user you want to limit, or add the user to the list.

d) Check Read only.

**Note**

Users who have *no* feature control set limitations but have "read-only" checked can access all tools but have no permissions to make changes in tools.

Users who have feature control set limitations and have "read only" checked cannot see the tools the feature control set excludes. They can see the other tool but cannot make changes in those tools.

Users who try to access a tool for which they do not have permissions are redirected to the first tool for which they have permissions.
Scripting with Packaged CCE

- Common Tasks, page 95
- Call Types, Contact Data, and Scripting, page 104
- Contact Categorization, page 106
- Selection of Routing Targets, page 118
- Network VRUs, page 124
- Multichannel Routing, page 136
- Use of Formulas, page 142
- Scripts in a Packaged CCE Environment, page 161
- Utility nodes, page 163
- CVP Scripting, page 165

Common Tasks

This topic contains information about common tasks you perform in Script Editor. This topic does not contain information about every possible task you can perform. For more information on Script Editor, see the Script Editor online help.

The Palette

*Figure 1: Palette Icon*

You can display the Palette by clicking the Palette icon in the Main toolbar or by selecting Palette from the View menu. The Palette contains the icons that represent the nodes used in scripts.

The available nodes are divided into the following tabs:
General Tab

The General tab contains icons for the following scripting activities:

- Comment Node, on page 164
- Categorization by Time and Date, on page 111—Uses Day of Week and Time nodes.
- Nodes Used to Stop Script Processing, on page 121—Uses End node.
- Line Connector Node, on page 164
- Formula Usage, on page 142—Uses Set Variable node.
- Start Node, on page 163

Routing Tab

The Routing tab contains icons for the following scripting activities:

- Categorization and Call Type, on page 106—Uses the Call Type and Requalify nodes.
- Media Routing Domains, on page 137—Uses the Media Domain node.

Targets Tab

The Targets tab contains icons for the following scripting activities:

- Agent Routing Nodes
- Transfer Calls from Agents to Agents
- Nodes Used to Stop Script Processing

Queue Tab

The Queue tab contains icons for the following scripting activities:

- Remove Call from a Queue
- Place a Call in Queue
- Adjust Priority of a Call in a Queue
- Queue to Agent Node
Create Routing Script

Procedure

Step 1  In Script Editor, choose File > New or click New. You are prompted to select a Routing Script or an Administrative Script:

![Figure 2: New Dialog Box](image)

Step 2  Click the following icon.

![Figure 3: Routing Script](image)

The new script opens in the Edit window, with a Start Node.

Step 3  Build the script.

Step 4  To save the script, choose File > Save or click Save. You are prompted for a script name.
Add Comments to a Node

Most nodes have a Comment tab:

**Figure 4: Comment Tab**

<table>
<thead>
<tr>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
</tr>
</tbody>
</table>

Specify a Connection Label Location for a Node

Most nodes have a **Connection Labels** tab.

**Procedure**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>You can specify the location of connection labels by moving the slider in the Label position area (when viewing a script in monitor mode). You can move the slider to the following locations:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) <strong>Origin</strong>, for the connection label to display close to the node you are editing.</td>
<td></td>
</tr>
<tr>
<td>b) <strong>Destination</strong>, for the connection label to display close to the targeted node.</td>
<td></td>
</tr>
<tr>
<td>c) <strong>Center</strong>, for the connection label to display between the nodes.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>You can also select to show the connection label by clearing the Display monitor labels check box.</td>
</tr>
</tbody>
</table>
Validate Scripts

Procedure

Step 1
To validate a single script, with the script open in the active window, choose **Script > Validate** or click the **Validate** Icon on the toolbar.

Step 2
To validate multiple scripts, choose **Script > Validate All** or on the toolbar, click the **Validate All** Icon. You are prompted to choose between validating active versions of all scripts or all the opened scripts.

![Figure 5: Validate All Query Dialog](image)

Step 3
Make the appropriate selection and click **OK**.

a) If a script is valid, a dialog box opens stating that script is valid.

b) If the script is not valid, the Validate Script dialog box opens with a list of the errors. When you select an error, the node where the error occurs is highlighted in the Edit window.

Open Script Explorer

Procedure

In Script Editor, choose **File > Script Explorer** or on the toolbar, click the **Explorer** Icon. The Script Explorer dialog box opens, listing scripts by customer and business entity:
You can then set the active version of the script, view its properties, rename it, or delete it. For more information, see the *Script Editor Online Help*.

**Schedule Routing Script**

You schedule a script by associating it with a call type as follows:

**Procedure**

**Step 1** Choose **Script > Call Type Manager**. The Call Type Manager dialog box opens.

*Figure 6: Call Type Manager Dialog Box—Schedules Tab*
Step 2 Select the call type to associate with the script.

Step 3 Click Add. The Add Call Type Schedule dialog box opens.

Step 4 In the Script tab, select the script to schedule:

Figure 7: Add Call Type Dialog Box - Script Tab
Step 5  In the Period tab, choose the information to define the period for which the schedule will be effective.

Figure 8: Add Call Type Schedule Dialog Box - Period Tab

Step 6  Optionally, in the Description tab, enter a description of the schedule.

Step 7  Click OK in the Add Call Type Schedule dialog box.

Step 8  Click OK in the Call Type Manager dialog box.

Note  The schedule is not saved until you click OK in the Call Type Manager dialog box.

Viewing Modes

You can view a script in four different modes:

- **Browse** - Allows you to view the script and edit it.
- **Edit** - Allows you to edit the script.
- **Monitor** - Allows you to monitor the script.
- **Quick Edit** - Allows you to make certain modifications to a script, with the following guidelines:
  - In Quick Edit mode, you cannot add or delete a node.
  - In Quick Edit mode, you can adjust most of the properties of the script nodes you select in the Node Control table of your assigned feature control set. However, in Quick Edit mode you cannot
edit any properties of the selected nodes that change the structure of a script or that reset previous reporting data.

• As a Quick Edit Only User:
  ◦ You can only edit scripts through Quick Edit mode.
  ◦ You cannot create or delete a script.
  ◦ You can access the Properties of any script node in any mode by either right-clicking the node and selecting Properties, or by double-clicking the node.
  ◦ You cannot edit the Call Type Manager dialog box (Script > Call Type Manager).
  ◦ You cannot edit the Administrative Manager dialog box (Script > Administrative Manager).
  ◦ You cannot edit the Custom Functions dialog box (Script > Custom Functions).
  ◦ You can choose the viewing mode from the Scripting toolbar, or from the Script menu.

Making Packaged CCE Work with Unified CVP

The following sections describe the differences between Packaged Contact Center Enterprise (Packaged CCE) and Unified Customer Voice Portal (Unified CVP) scripting and show how they work together in common tasks.

Difference between CCE and CVP Scripting

Packaged CCE scripting offers call control such as how a call should be treated based on time of day, call type, and so on. It also handles queuing for an agent based on skill group or service. It determines when to send the call to Unified CVP (for example, to play prompts, collect caller entered digits, and get or put information in a database), or for queuing the call while waiting for an agent.

Packaged CVP scripting offers IVR interaction, like playing a prompt based on an audio file or text-to-speech or collecting caller-entered digits via touch tone or speech. It also offers advanced features such as accessing an external database or web service for information used in creating a dynamic caller interaction experience. Examples include accessing current balance or storing collected customer information in a database.

Packaged CCE scripting is used for routing the call; but when the call needs to go to the CVP, a self-service component is enlisted with Unified CVP scripts that have been created in Call Studio. For example, if a customer calls a credit card company and gets a voice recorded message, the Packaged CCE component makes the decision which script to run, whether the interaction is treated as a sales call or a service call and then selects which VRU (voice response unit) scripts get run. The call is then sent to a VRU, which connects the call to the Unified CVP "self-service engine". It accomplishes these tasks without the customer talking to an agent, such as getting the account balance with touch tone activation or speech. Once the information is collected control is then returned to the Packaged CCE script. The Packaged CCE script queues the customer for an agent, and connects her to an agent.

How Packaged CCE and Unified CVP Work Together

To summarize, Packaged CCE 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.

Packaged CCE uses CVP messaging technology to direct Unified CVP and to receive the responses from Unified CVP.

For more information about Packaged CCE working with Unified CVP, proceed to Before You Begin, on page 166.

Call Types, Contact Data, and Scripting

When writing scripts to route contacts, you must understand Call Types and contact data. Call Types are typically created during use of the call type gadget of the Unified CCE Administration tool and are therefore discussed in greater detail in the Access Configuration Manager, on page 81 in Packaged CCE Administration Guide for Release 9.0(x).

Call Types

A Call Type is the first-level category of a contact and is determined by data associated with the contact. You associate a script with a Call Type. When a contact of a certain Call Type is received, the associated script runs on that contact.

Default Call Types

A default Call Type is the Call Type used when a contact does not map to a defined Call Type.

You define a default Call Type for each routing client through the Configuration Manager in the PG Explorer tool. The General Default call type is set through the Configuration Manager in the System Information tool. You also define a general default Call Type that is not specific to a routing client. For more information, see the Call Types, on page 51 chapter in Packaged CCE Administration Guide for Release 9.0(x).

Relation between Call Types and Scripts

Scripts are scheduled by Call Type. In other words, when Packaged CCE receives a request to route a contact, it determines the Call Type of that contact, then runs the associated script.

Call Types provide the first level of categorization of contacts, enabling you to write scripts to route contacts differently depending on their call type. While other types of categorization take place within a script, Call Types enable you to provide contacts with different treatment by running different scripts to begin with. Call Types enable categorization before a script begins to execute.

Call Type Qualifiers

The Call Type is determined by the following data, which are referred to as Call Type qualifiers:

• Dialed Number
The Call Type qualifiers described in this section apply to contacts from all media. The terminology used is applicable to voice contacts; where the terminology differs for other media, the differences are explained in this section, as well as the following sections in this topic:

- Contact Data for Chat
- Contact Data for E-Mail

### Dialed Number (DN)

A Dialed Number (DN) is a string that represents the telephone number dialed by the caller, preceded by the name of the routing client and a period. For example, “ucm.18005551212” might be a Dialed Number.

Typically, a Dialed Number is associated with one or more Call Types.

### Association of Contacts with Call Types in Packaged CCE

Following is the general process of how Packaged CCE attempts to associate a contact with a Call Type.

1. If the Dialed Number of the contact maps to a defined Call Type, Packaged CCE uses that Call Type.
2. If no Call Type matches the contact, Packaged CCE uses the default Call Type for the routing client.
3. If no default Call Type is defined for the routing client, Packaged CCE uses the general default Call Type.
4. If no general default Call Type is defined, Packaged CCE returns an error to the routing client.

### Determination of Call Type for Voice Contact

The following example demonstrates how Packaged CCE determines the Call Type for a voice contact and runs the appropriate script:

1. When configuring Packaged CCE, you create a Call Type called "MASSACHUSETTS_SALES". This Call Type is defined as:
   - Having a Dialed Number of "ucm.8005551234".
2. You create a script called "MASSACHUSETTS_SALES_SCRIPT," which finds the longest available agent in the "NORTHEAST_SALES" skill group.
3. You schedule the script to run for the "MASSACHUSETTS_SALES" Call Type.
4. Packaged CCE determines that the Call Type is "MASSACHUSETTS_SALES" and executes the "MASSACHUSETTS_SALES_SCRIPT" script.
5. Packaged CCE assigns the task to a particular agent.

### Determination of Call Type for E-mail or Web Request

The following basic example demonstrates how Packaged CCE determines the Call Type for e-mail or chat web request:
When configuring Packaged CCE, you create a Call Type called "SSC_CT". This Call Type is defined as having a Script Selector (Dialed Number) of "SSC_DN".

When configuring Unified EIM/WIM, set the value of the Script Selector for Media Routing Domain to "SSC_DN".

You create a script called "SSC_SCRIPT," which finds the longest available agent in the "COLLABORATION_SALES" skill group.

You schedule the script to run for the "SSC_CT" Call Type.

An e-mail is sent or a web user requests a chat session.

A route request is sent to Packaged CCE.

Packaged CCE determines that the Call Type is "SSC_CT" and executes the "SSC_SCRIPT" script.

Packaged CCE instructs the Unified EIM/WIM to assign the task to a particular agent.

Contact Categorization

When you create a routing script, you typically use the nodes available in Script Editor to define how the script is to categorize contacts. By categorizing contacts, a script can provide unique solution for different customer needs.

Categorization and Call Type

Categorization is the process of classifying a contact based on certain data associated with the contact. Through categorization, a script can determine the best way to process a contact.

Categorization through Scheduling Scripts by Call Type

Call Types provide the first level of categorization for routing scripts. You schedule scripts by Call Type; therefore, the Call Type of a contact determines which script is executed, enabling you to create different scripts for different types of contacts.

Change Call Type to Static

You can change the Call Type of a contact from within a script by using the Call Type node (in the Routing tab of the Palette).

Figure 9: The Call Type Icon
The following figure is the Call Type Properties dialog box of the Static Call Type node:

**Figure 10: Call Type Properties Dialog Box - Static Call Type**

To define a static call type node, complete the following steps.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>In the Call Type tab, click the <strong>Statically</strong> radio button.</td>
</tr>
<tr>
<td>Step 2</td>
<td>From the Call Type list, click the call type to assign to the contact.</td>
</tr>
</tbody>
</table>

**What to Do Next**

The Call Type node changes the Call Type and continues the current script execution. The Requalify Call node terminates the current script execution and executes a new script associated with that Call Type.

**Warning**

Change Call Type to Dynamic

You can change the Call Type of a contact from within a script by using the call type node (on the Routing tab of the Palette).

**Figure 11: Call Type icon**
Dynamic Call Type selection is not available when an External Authorization server is used with Internet Script Editor and will be grayed out in the interface.

The following figure is the Call Type Properties dialog box of a dynamic call type node:

**Figure 12: Call Type Properties Dialog Box - Dynamic Call Type**

To define a dynamic call type node, complete the following steps.

**Procedure**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>On the call type tab, select the <strong>Dynamically</strong> radio button.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>To dynamically change the call type of a contact by call type name, in the Find By section, select the <strong>Call Type Name</strong> radio button.</td>
</tr>
<tr>
<td>Step 3</td>
<td>To dynamically change the call type of a contact by call type ID, in the Find By section, select the <strong>Call Type ID</strong> radio button.</td>
</tr>
<tr>
<td>Step 4</td>
<td>To determine which call type name or ID to use to change the call type of a contact, click the Formula Editor button to create a formula.</td>
</tr>
</tbody>
</table>
What to Do Next

Change Call Type and Execute a New Script

You can change the Call Type of a contact from within a script and execute a new script associated with the Call Type by using the Requalify Call node (in the Routing tab of the Palette).

Figure 13: The Requalify Call Icon

Following is the Requalify Properties dialog box of the Requalify Call node:

Figure 14: The Requalify Properties - Requalify Call Tab

Define Requalify node properties as follows:

Procedure

- **Step 1** In the Requalify Call Tab, select the Call type to assign to the contact.
- **Step 2** Optionally, add comments.
What to Do Next

The Call Type node changes the Call Type and continues the current script execution. The Requalify Call node terminates the current script execution and executes a new script associated with that Call Type.

Categorization by Call Type Qualifiers

As described in the topic Call Types, Contact Data, and Scripting, a contact's Call Type is determined by the Dialed Number qualifier.

When Packaged CCE determines a contact's Call Type based on these qualifiers, it executes the associated script.

However, after the script executes, you can further categorize the contact based on the values of the Call Type qualifiers.

Categorize Contact by Dialed Number

You can categorize a contact based on its Dialed Number by using the Dialed Number (DN) node (in the Routing tab of the Palette).
Following is the DN Properties dialog box of the Dialed Number node:

**Figure 16: DN Properties - Dialed numbers rab**

You can define the Dialed Number node properties as follows:

**Procedure**

**Step 1** Select one or more dialed numbers or Script Selectors from the Dialed numbers list and click Add> to move them to the Target dialed numbers list. If the current contact matches one of the selections in the Target dialed numbers list, processing continues on the node's success branch; otherwise, processing continues on the failure branch.

**Step 2** Optionally, add comments and connection labels.

**Categorization by Time and Date**

You schedule a script by associating it with a Call Type. When a contact of a certain Call Type is received, the associated script runs for that contact.

However, after the script executes, you can further categorize the contact based on the time and day of week. This categorization refines the schedule.

**Note**

The time and day of the week are determined by the settings on the CallServer virtual machine.

For example, a Call Type named "CHAT_CT" may be defined to include all chat web requests. A script named "CHAT_SCRIPT" executes every time a contact with the Call Type "CHAT_CT" is received. Typically, this script instructs Unified WIM to assign the request to the longest available agent in the "Chat" skill group.
However, the contact center is staffed differently over the weekend and the supervisor wants reports to better reflect weekend activity. Therefore, for chat web requests received on Saturday or Sunday, the script branches differently and instructs Unified WIM to assign the request to the longest available agent in the "WKEND_SUPPORT" skill group.

As another example, for a contact center where no phone support is available during night hours or weekends, you may choose to design a script that routes a phone call to an announcement instead to an agent, during those off hours.

**Categorize Contact by Date and Time**

You use the Time node (in the General tab of the Palette) to choose from among several paths within the script based on the current time at Packaged CCE Central Controller. Following is the Time Properties dialog box of the Time node.

*Figure 17: Time Icon*

Following is the Properties dialog box of the Time node:

*Figure 18: Time Properties*
You must insert targets and connections from the Time node before you can define the node's properties. Then define Time node properties as follows:

**Procedure**

**Step 1** For each branch listed in the Connections list, define a Time Range. You can define multiple time ranges for a single branch. Click Add Time to add a new time range to the branch, or select a time range listed and click Modify Time to modify it. A dialog box opens in which you can define the time range (the Add Time dialog box is shown below; the Modify Time dialog box looks and functions similarly):

**Figure 19: Add Time Dialog**

![Add Time Dialog](image)

**Step 2** To delete a time associated with the branch, select the time and click Delete Time.

**Step 3** You can define a branch as Otherwise by selecting the branch and clicking Make Otherwise. Execution follows this branch if none of the specified time ranges apply. You can specify only one Otherwise branch for the node. If you do not want to define the branch as Otherwise, select the branch and click Delete Otherwise.

**Step 4** Optionally, add comments and connection labels.

**What to Do Next**

If you delete a connection, the time-range information you specified in the Properties dialog box is also deleted.

---

**Note**

Packaged CCE Administration Guide for Release 9.0(x)
Categorize Contact by the Day of Week

You use the Day of Week node (in the General tab of the Palette) to transfer control to one of several branches depending on the current day of week (Sunday, Monday, etc.).

Following is the Properties dialog box of the Day of Week node:

You can define multiple output connections from the Day of Week node and associate each with one or more days of the week.

You must insert targets and connections from the Day of Week node before you can define the node's properties.

Define Day of Week node properties as follows:

**Procedure**

**Step 1**
For each branch listed in the Connection list, check the days of the week in which processing should continue on that branch. To check the day for that connection, left-click in a spot in the grid corresponding to that connection and day. A check mark appears in the grid. You can associate each day of week with one connection. However, you can associate each connection with one or more days of the week.

**Step 2**
Optionally, add comments and connection labels.
Categorization by Branching

Within a script, you can create multiple branches to direct script processing based on certain conditions. Branching allows you to use a single script that processes contacts differently, depending on data associated with the contact, or on conditions at the contact center.

Execute a Different Script

You use the **Go To Script** node (in the General tab of the Palette) to direct contact processing to another script without changing the call type. When Packaged CCE encounters a Go To Script node, it stops executing the current script and starts the script indicated in the node.

*Figure 22: Go To Script Icon*

For example, you might have several scripts that check for exception conditions and, if none are found, execute a standard subroutine. Instead of including that subroutine as a branch from the failure output terminal of each of the exception conditions, you could use a Go To Script node pointing to a separate script containing the subroutine.

Following is the Properties dialog box of the Go To Script node:

*Figure 23: Go To Script Properties*

Define "Go to Script" node properties as follows:
**Procedure**

**Step 1** Select the Business entity that owns the script that the node should execute. By default, Packaged CCE consists of one business entity.

**Step 2** Select a script from the Scripts list. From within an administrative script, you can only go to another administrative script. Within a routing script, you can only go to another routing script.

**Step 3** Optionally, add comments and connection labels.

**Direct Script Execution to Different Branches by Percentage**

You can direct specific percentages of contacts to different branches in a script by using the Percent Allocation node (in the General tab of the Palette).

*Figure 24: Percent Allocation Icon*

Each branch may lead directly to a target, or may include additional processing. Because contacts are distributed by percentage and without tests of the targets’ data, distributing by percentage never fails.

For example, in a geographically diverse environment, you can create a script that sends 10% of contacts to Boston, 5% to Chicago, and distributes the remaining 85% to another set of targets.

Following is the Properties dialog box of the Percent Allocation node:

*Figure 25: % Allocation Properties*
Define Percent Allocation node properties as follows:

**Procedure**

**Step 1** In the Percent column for each connection, enter a percent number for the percentage of contacts to process on that branch. **Note**: The percent total for all rows must equal 100.

**Step 2** Optionally, modify the Connection name. Changes appear in the connector labels when you save the properties and view the script.

**Step 3** Optionally, add comments and connection labels.

---

**Categorize Contact Based on a Condition**

You use the **If** node (in the General tab in the Palette) to direct script execution to one of two branches based on the result of an evaluation. You can use formulas to define the If node.

**Figure 26: If Icon**

When Packaged CCE executes an If node, it first evaluates the condition specified in the node Properties dialog box Define condition field. If Packaged CCE determines that the condition is true, control flows through the success output terminal; if it determines the condition is false, control flows through the failure output terminal.
Following is the Properties dialog box of the If node:

**Figure 27: If Properties**

![If Properties dialog box]

Define If node properties as follows:

**Procedure**

- **Step 1** In the Define condition field, enter a condition or use the Formula Editor to create a formula.
- **Step 2** Optionally, add comments and connection labels.

**Categorize a Contact Based on its Media Routing Domain**

You use the Media Routing Domain node to categorize contacts based on their media routing domains. This node is described in the Universal Queues section of this document.

**Selection of Routing Targets**

After defining how a script is used to categorize contacts, you typically use the nodes available in Script Editor to specify how the contact is to be routed to a target. By selecting routing targets, you determine the destination for contacts.

**Routing Targets**

A routing target is an entity to which Packaged CCE can route a contact. The routing target receives the contact and processes it accordingly.

The routing targets available for Packaged CCE are peripheral-level skill targets.
Agent Routing Nodes

The following nodes available for agent routing:

1. Queue to Agent Node. For more information, see Specify an Agent Directly
2. Agent to Agent Node. For more information, see Transfer Calls from Agents to Agents

Transfer Calls from Agents to Agents

You can transfer a call from an agent to an agent by using the Agent to Agent node in the Targets tab of the Palette.

The Agent to Agent node routes the call to the specified agent. You define the agent either by directly selecting the agent from the database or by providing an expression using a formula. The expression must translate to agent peripheral number or SkillTargetID. The router then finds a valid label for the agent. If there are no labels configured for the specified agent, the failure node of the Agent to Agent node is executed.
Following is the Properties dialog box for the Agent to Agent node:

**Figure 29: Agent to Agent Properties**

Define Agent to Agent node properties as follows:

**Procedure**

**Step 1** Choose an option from the Select agent by drop-down list:

a) Peripheral number - To select a peripheral and a provide formula that translates to the agent's peripheral number.

b) Enterprise Name - To select the agent from the list of configured agents.

c) Skill target ID - To select the agent by providing an expression that translates into the agent's SkillTargetID. In the Packaged CCE/Unified CCH supervisory case, the expression should use the call's PreferredAgentID.

**Step 2** Based on your selection in Step 1, select the peripheral or agent, or enter an expression, as necessary.

**Step 3** Optionally, check or uncheck Fail node if agent is unavailable:

a) When checked, the success branch of the Agent to Agent node is executed and the router sends the call if the router finds a valid label for the agent, the agent is available, and the agent state is Ready.

b) The failure branch of the Agent to Agent node is executed if the router does not find a valid label for the agent, or the agent is not available or the agent is in TempUnavailable mode (the router has just send a call to the agent).
c) When not checked, the success branch of the Agent to Agent node is executed and the router sends the call if the router finds a valid label for the agent. The failure branch of the Agent to Agent node is executed if the Router does not find a valid label for the agent.

Step 4  Optionally, add comments and connection labels.

Nodes Used to Stop Script Processing

You can use the following nodes to stop script processing:

- End Node
- Termination Node
- Release Call Node

End Node

You can terminate the script by using the End node in the General tab of the Palette.

Figure 30: End icon

If the script reaches the End node, it has failed to find a target for the contact. Packaged CCE then uses the default route for the Dialed Number.

Several End nodes can appear in the same script. The End node is never required; a script can terminate with any node.

You do not define any properties for the End node. You can optionally add comments.

Release Call Node

You can terminate the script and disconnect the caller by using the Release Call node in the Targets tab of the Palette.

Figure 31: Release Call Icon

You can use a Release Call node in situations where the caller needs no further service after executing several CVP scripts.

You do not define any properties for the Release Call node. You can optionally add comments.
Target Requery

Target Requery is a script node feature that addresses routing failures due to transient failures in the network (such as network congestion). If the determined destination for a contact is available but not reachable, Target Requery attempts to find a different valid destination.

You need Target Requery to address the following failures:

- Failure to deliver a call to an agent.
- Failure of the outbound leg of a blind-mode Network Transfer.
- Target Requery works on a per call basis; that is, the routing information for one call does not affect the state for other calls. If the first target selected for the contact was not reachable, the target is not eliminated from the potential routing destinations for other contacts.

Target Requery Functionality

In the Packaged CCE system, when queried, the CallRouter returns a label to the routing client. The routing client then routes the call to the destination specified by the label. If the destination is not reachable (for example, because of a busy signal or no answer), the call is routed to the default destination.

With Target Requery, if the router fails to route to the target node, a second attempt is made. If the failure occurs a second time, then the router continues from the failure path in the node.

Note

In a Queue node, just one target is used and the failure path of the node is immediately taken. You can create a script that increases the priority and requeries the call from the failure path to the same queue.

In the event of a failure, you can handle requerying in the scripting environment, as you deem appropriate.

Target Requery does not require different definitions for different failure cases. However, you can choose to handle different failures differently.

Test of the RequeryStatus Variable

You can test the error path of these script nodes using Target Requery to determine the specific network cause of failure and conditionally retry the attempt as necessary. You can accomplish this using an If node to check the value of the Call variable RequeryStatus. The decision path for the script is then determined by the value of the RequeryStatus variable.

The following are possible values for the RequeryStatus variable:

<table>
<thead>
<tr>
<th>Requery Status Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUERY_ANSWER (0)</td>
<td>CallRouter internal use. Script ends and the call was successfully sent to the chosen target. Not visible to users.</td>
</tr>
<tr>
<td>Requery Status Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REQUERY_ROUTE_SELECT_FAILURE (1)</td>
<td>Routing client generated an error code from ReRouteReq msg indicating a Route Select failure.</td>
</tr>
<tr>
<td>REQUERY_CALLED_PARTY_BUSY (2)</td>
<td>Routing client generated error code from ReRouteReq msg indicating the called party is busy.</td>
</tr>
<tr>
<td>REQUERY_NO_ANSWER (3)</td>
<td>Routing client generated an error code from ReRouteReq msg indicating no answer.</td>
</tr>
<tr>
<td>REQUERY_ERROR (4)</td>
<td>CallRouter generated an error code. The attempt to send the call to target failed because the target was not reachable (i.e., busy, ring no answer).</td>
</tr>
<tr>
<td>REQUERY_TIMED_OUT (5)</td>
<td>CallRouter internal use. Script ends. The attempt to send the call to target failed because the Routing Client did not respond within the DivertOnBusyCallTimeout period. Not visible to users.</td>
</tr>
<tr>
<td>REQUERY_ABORTED (6)</td>
<td>CallRouter internal use. Script ends. The attempt to send the call to target failed because the caller hung up or the call was lost. Not visible to users. In the case of ABANDON and DISCONNECT, the CallRouter assumes the call has ended and ends the script. The RequeryStatus value is set to 6, indicating REQUERY_ABORTED. This is used for the internal workings of the CallRouter but is not visible to users. Which Nodes Support Target.</td>
</tr>
</tbody>
</table>

**Nodes that Support Target Requery**

The following nodes support Target Requery:

- Queue
- Queue to Agent
- Route Select
- Precision Queue

**Use Target Requery**

You define nodes to enable Target Requery. For the Queue, Queue to Agent, and Route Select nodes:
### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Open the node properties.</td>
</tr>
<tr>
<td>2</td>
<td>Click <strong>Change</strong>. A dialog box opens.</td>
</tr>
<tr>
<td>3</td>
<td>Check <strong>Enable target requery</strong>.</td>
</tr>
<tr>
<td>4</td>
<td>Click <strong>OK</strong> to close the dialog box.</td>
</tr>
<tr>
<td>5</td>
<td>Click <strong>OK</strong> to close the properties dialog box.</td>
</tr>
</tbody>
</table>

### What to Do Next

For Select nodes:

1. Open the node properties.
2. Check **Enable target requery**.
3. Click **OK** to close the properties dialog box.

### Network VRUs

Through routing scripts, you can divert a call to a Network VRU for additional call processing.

### VRU Functionality

A VRU, or Voice Response Unit, is a telecommunications device, also called an Interactive Voice Response Unit (IVR), that plays recorded announcements and responds to caller-entered touch-tone digits. Cisco Voice Portal (CVP) is the supported VRU for Packaged CCE. A VRU can also be equipped with Automatic Speech Recognition (ASR) or Text-to-Speech (TTS) capabilities.

### Access to VRU Scripts in Unified ICM Scripts

After you set up VRUs through the Unified ICM Configuration Manager, you can write routing scripts that send calls to the VRU, queue the call, and invoke specific VRU scripts.

### Send Call to a VRU with Send to VRU Node

You can send a call to CVP for further processing by using the Send to VRU node (in the Queue tab of the Palette).

---

**Figure 32: The Send to VRU icon**

---

---

---
When Packaged CCE executes a Send to VRU Node, it looks up the call's Dialed Number, the Dialed Number's Customer, and the Customer's Network VRU. If that fails to retrieve a Network VRU, the router uses the system default Network VRU.

There are two failure cases:

- If the label does not exist, script execution continues with control flowing through the nodes failure output terminal.
- If Packaged CCE does not receive confirmation, execution continues with control flowing through the nodes failure output terminal.

In all other cases script execution continues with control flowing through the nodes success output terminal.

Notes:

- If the Run External Script, Play, Menu, Collect Data, or Queue node is used in a script before a Send To VRU node, an implicit Send To VRU node is assumed. You do not have to use the Send To VRU node. However, Cisco recommends that you make a practice of including the node in routing scripts, as it can act as a visual aid if you ever need to troubleshoot the script.
- If the call is delivered to the CVP but then abandoned, script execution ends. In monitor mode, a special label on the Send To VRU node accounts for these cases.

You do not need to set properties for the Send to VRU node. However, you can optionally add comments or connection labels.

Run External Scripts

You can instruct a Network CVP to run a specific script by using the Run External Script node (in the Queue tab of the Palette).

You can use multiple Run External Script nodes to execute a series of scripts on the CVP. The Run External Script node is valid for use with all the VRU types.

The execution of Packaged CCE routing script waits for the external script to finish:

- If the external script runs successfully, control then passes through the success branch of the Run External Script node.
- If the external script does not run successfully for any reason, then control passes through the failure branch of the Run External Script node.

Notes:

- If the current call is not at a CVP when the Run External Script node executes, the Packaged CCE sends the call to the associated Network VRU, as executing a Send to VRU node.
Design scripts so that the Failure branch of a Run External Script Node contains a test for the Call.VRUStatus variable. If the value is 2, the VRU is likely to be not functioning properly. Therefore, the script avoids executing any subsequent Run External Script nodes on this Failure branch.

When an uninterruptible script is used in a Run External Script node, the CallRouter waits for the script result from the VRU. It then executes the next node. Calls can only be routed when they reach an interruptible node. The Wait node and interruptible Run External Script node (micro apps) are interruptible. Every other node is uninterruptible.

Following is the Properties dialog box for the Run External Script node:

**Figure 34: Run External Script Properties**

Define Run External Script node properties as follows:

**Procedure**

- **Step 1** Select the Packaged CCE Script/External Script Name you want to execute.
- **Step 2** Optionally, add comments and connection labels.

### VRU Errors

The following table lists the possible values for the VruStatus variable:
### Table 2: VruStatus Variable Codes

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>VRU_SUCCESS</td>
<td>The last VRU node was successful.</td>
</tr>
<tr>
<td>1</td>
<td>VRU_ERROR</td>
<td>The last VRU node failed because of a routing or configuration error.</td>
</tr>
<tr>
<td>2</td>
<td>VRU_TIMEOUT</td>
<td>The last Send To VRU failed because the routing client did not respond within 20 seconds or the last Run External Script node failed because the timeout limit defined for the script expired.</td>
</tr>
<tr>
<td>3</td>
<td>VRU_ABORTED</td>
<td>The last VRU node did not complete because the caller hung up or was otherwise lost. (Because this causes the routing script to terminate immediately, this value is never seen.)</td>
</tr>
<tr>
<td>4</td>
<td>VRU_DIALOG_FAILED</td>
<td>The last VRU node failed because communication with the VRU ended unexpectedly.</td>
</tr>
<tr>
<td>5</td>
<td>VRU_SCRIPT_NOT_FOUND</td>
<td>The VRU failed because the referenced VRU script was not found in the Packaged CCE configuration.</td>
</tr>
</tbody>
</table>

---

### Call Queuing at VRUs

You can queue a call at a Network VRU until a specific resource becomes available. A call can be queued for one or more skill groups, or a Precision Queue. As soon as an agent becomes available at one of the specified targets, the call is removed from the queue and sent to the target.

Specifically, you can:

- Place a call in a Precision Queue.
- Place the call in a Precision Queue for one or more skill groups.
- Adjust the priority of call in a queue for one or more skill groups.
- Remove the current call from any queues to which it is assigned.

For example:

1. The call is first sent to the Network VRU. This step is required before you queue the call.
The call is queued for three skill groups.

If the call is successfully queued, the script cycles between a Wait node and a Run External Script node so that the caller hears an announcement every 30 seconds. If an agent in one of the skill groups becomes available, the call is removed from queue and taken back from the CVP. Routing script execution ends and the call is delivered to the target.

Note

In this scenario, you would likely make the VRU script interruptible so that the routing script can retrieve the call immediately when the resource becomes available.

Place a Call in Queue

You can place a call in queue at a CVP for one or more skill groups using the Queue node (in the Queue tab of the Palette).

Figure 35: The Queue icon

If an agent becomes available in one of the skill groups, the call is routed to that resource.

Note

If the current call is not at a CVP when the Queue node executes, Packaged CCE sends the call to the associated Network VRU. (This does not apply to Type 2 VRUs, which are VRUs at customer premises.)

The Queue node includes a Priority field, which sets the initial queuing priority for the calls processed through this node versus other calls queued for the same target. The Priority is expressed as an integer from 1 (top priority) to 10 (least priority). The default value is 5.

If more than one call is queued to a group when an agent becomes available, the queued call with the lowest priority number is routed to the target first. For example, assume an agent in a skill group becomes available and two calls are queued to that skill group. If one call has priority 3 and the other has priority 5, the call with priority 3, the lower value, is routed to the skill group while the other call continues to wait.

Note

The Queue node does not actually result in instructions being sent to the VRU. When queuing occurs the Queue node exits immediately through the success branch and the call is assumed to be at the VRU; the script should then continue with a Run External Script node to instruct the VRU what to do while holding the call until an agent becomes available. Typically this would invoke a Network VRU Script that plays music-on-hold, possibly interrupted on a regular basis with an announcement.
Following is the Properties dialog box for the Queue node:

**Figure 36: Queue to Skill Group Properties**

Define Queue node properties as follows:

**Procedure**

**Step 1** To change the queue type:

a) Click **Change**. The Queue Type dialog box opens.

b) Select a Target Type (Skill Group). You cannot reference more than one type of target within a single Queue node. To queue a call to more than one target type, execute multiple Queue nodes sequentially.

c) Optionally, select a Priority to set the initial queuing priority for calls processed through this node versus other calls queued for the same target: 1 for top priority to 10 for least priority. (The default is 5.)

d) Optionally, check **Enable Target Requery**.

Note When Target Requery is enabled in a Queue node and a Requery happens, for example because the call is presented to an available agent, but the agent does not answer, the script continues through the failure terminal. The script can then inspect the call variable RequeryStatus to determine what to do next. The typical action in case of a No Answer would be to Queue the call again, possibly to other skill groups, and possibly increase the priority so that it is taken out of the queue before regular queued calls.

e) Click **OK** to close the Queue Type dialog box.

**Step 2** To add targets:

a) Click **Add Targets**. The Add Targets dialog box opens, listing available targets of the type you specified.

b) Use the Available Targets list and the Add button to select targets.
c) Click **OK** to close the Add Targets dialog box. The target members you selected appear in the Properties dialog box.

**Step 3** Optionally, continue defining Target Type information for the Route (Drop-down list) member. This is the route to send the call to when an agent in the target type becomes available. (The drop-down list includes all routes associated with the target.)

**Step 4** Optionally, add connection labels.

---

**What to Do Next**

When processing a Queue node, the router first checks for an available target, if there is none available then the router attempts to queue the call. The call does not move to the VRU if there is an available agent.

---

**Precision Queue Script Node**

You can use the Precision Queue script node to queue a call in the Unified CCE based on caller requirements until agents with desired proficiency become available. This node contains multiple agent selection criterion which are separated into steps.

**Figure 37: Precision Queue Script Node**

You must use the Unified CCE Web Administration or the Web Services API (see the Unified Contact Center Enterprise Developer Guide) to configure a Precision Queue.

A single call can be queued on multiple Precision Queues. If an agent becomes available in one of the Precision Queues, the call is routed to that resource. You cannot reference multiple Precision Queues with a single Precision Queue node. However, you can execute multiple Precision Queue nodes sequentially to achieve this.

The Precision Queue node includes a Priority field, which sets the initial queuing priority for the calls processed through this node versus other calls queued to the other targets using different nodes. The priority is expressed as an integer from 1 (top priority) to 10 (least priority). The default value is 5.

If more than one call is queued to a Precision Queue when an agent becomes available, the queued call with the lowest priority number is routed to the target first. For example, assume an agent in a Precision Queue becomes available and two calls are queued to that Precision Queue. If one call has priority 3 and the other has priority 5, the call with priority 3, the lower value, is routed to the Precision Queue while the other call continues to wait. If the priorities of the two calls are same, then the call queued first is routed first.

VRU script instructions are not sent to the VRU. If a call enters the Precision Queue node and no resource is available, the call is queued to the Precision Queue and the node transfers the call to the default VRU, if the call is not already on a VRU. The script flow then exits immediately through the success branch and continues to a Run External Script node to instruct the VRU what to do while holding the call until an agent becomes available. Typically, this invokes a Network VRU Script that plays music-on-hold, possibly interrupted on a
regular basis with an announcement. The script flow can also use other queuing nodes to queue the same call to other targets, for example, Queue to Skill Group and Queue to Agent.

**Precision Queue Properties Dialog Box - Static Precision Queue**

The following list describes the Precision Queue Properties dialog box for a static Precision Queue script node.

![Figure 38: Precision Queue Properties dialog box — Static Precision Queue](image)

The following property is unique to static precision queues:

- **Drop-down list**—To route calls that enter this node to a static Precision Queue, you must select a Precision Queue from the list.

The following properties are common to static and dynamic precision queues:

- **Select Precision Queue radio buttons**—You can select one of the following options for each a Precision Queue:
  - **Statically**—Select this option to choose a single Precision Queue to be selected for all the calls that enter this node.
  - **Dynamically**—Select this option to select a Precision Queue on a call-by-call basis based on a formula.

**Note**

The Dynamic Precision Queue selection is not available when an External Authorization server is used with Internet Script Editor and will be grayed out in the interface.

- **Priority selection**—To select the initial queuing priority for calls processed through this node, you can select from 1 to 10. The default is 5.
- **Enable target requery check box**—To enable the requery feature for calls processed through this node, select this check box. When a requery occurs, for example if a call is presented to an available agent and the agent does not answer, the script continues through the failure terminal. The script can then inspect the call variable RequeryStatus to determine what to do next. The typical action in case of a No Answer is to queue the call again to other precision queues, and increase the priority so that it is taken out of the queue before regular queued calls.
Precision Queue Properties Dialog Box - Dynamic Precision Queue

The following list describes the Precision Queue Properties dialog box for a dynamic precision queue script node.

Use dynamic precision queues when you want a single routing script for multiple precision queues (for example, when the overall call treatment does not vary from one precision queue to another). Dynamic precision queues can simplify and reduce the overall number of routing scripts in the system.

**Figure 39: Precision Queue Properties dialog box—Dynamic Precision Queue**

Dynamic Precision Queue selection is not available when an External Authorization server is used with Internet Script Editor and will be grayed out in the interface.

The following properties are unique to dynamic precision queues:

- **Find By radio buttons**—To dynamically route calls that enter this node to a Precision Queue name or ID, use the Find By radio buttons.
  - **Precision Queue Name radio**—Select this option to dynamically route calls that enter this node to a Precision Queue name.
  - **Precision Queue ID**—Select this option to dynamically route calls that enter this node to a Precision Queue ID.

- **Formula Editor button**—To determine to which Precision Queue name or ID to route calls that enter this node, click the Formula Editor button to create a formula. The formula is then evaluated at run time to select a precision queue by either name or by database ID. For example, you can use the formula "Call.PeripheralVariable4" to look up the Precision Queue if call variable 4 contained the Precision Queue name, as a result of a database lookup or from CVP call processing.

For properties which are common to static and dynamic precision queues see, *Precision Queue Properties Dialog Box - Static Precision Queue*, on page 131
Queuing Behavior of Precision Queue Node

Precision Queues internally are configured with one or more time-based steps, each with a configured wait time. After a call is queued, the first step begins and the timer starts. This occurs although the execution path of the script exited the success node and a new node may be targeted (for example, Run Ext. Script).

If the timer for the first step expires, control moves to the second step (assuming one exists), and so on. As long as the call remains in queue and there are steps left to execute, the call internally continues to move between steps regardless of the path the call takes after it leaves the Precision Queue node. If a call is queued to two or more Precision Queues, the call internally walks through the steps for each Precision Queue in parallel. After the call reaches the last step on a Precision Queue, it remains queued on that step until the call is routed, abandoned, or ended.

Adjust Priority of a Call in a Queue

You can override the priority of a call in queue set by the Queue node by using the Queue Priority node (in the Queue tab of the Palette).

*Figure 40: The Queue Priority Icon*

For example:

1. The original priority of the call in queue is set by the Queue to Skill Group node or the Precision Queue node.
2. The call waits in queue for 20 seconds while the caller listens to an announcement.
3. Call control passes to a second Wait node.
4. If 20 more seconds pass without an agent becoming available, the Queue Priority node executes and raises the call's priority in queue.

Notes:

- Only use the Queue Priority node after a Queue to Skill Group node or a Precision Queue node. Any subsequent use of the Queue to Skill Group node or the Precision Queue node results in setting the queue priority back to the original setting for that node.
- The Queue Priority node sets the priority for a call within all queues that the call is placed in. If a call requires the priority to be raised in one queue only, you should use a subsequent Queue to Skill Group or Precision Queue node for only that skill group/queue (with the new priority).
- Queuing priorities should be handled very carefully. Just increasing Queue priority does not get a call handled sooner. The effect depends on the other call in the queue. For example, if all calls are treated using the example above, the priority increase has no net effect. If the script above is only used for the Platinum customers while the Standard customers script leaves them at the default priority level, the effect is that all Platinum customers that have been in queue for more than 20 seconds are handled first regardless of other customers in queue. This can be a dangerous practice, because while the delay for Platinum customers is greater than 20 seconds, no Standard customer will ever be handled. The solution...
is to increase the priority level for Standard customers as well, but only after they have been in queue for a longer period, for example 3 minutes.

Following is the Properties dialog box for the Queue Priority node:

**Figure 41: Queue Priority Properties**

Define Queue Priority node properties as follows:

1. Specify the queuing priority for a call processed through this node: 1 for top priority to 10 for least priority. This determines the priority the call has versus other calls queued for the same target.

2. Optionally, add comments or connection labels.

**Remove Call from a Queue**

You can remove a call from any queues by using the Cancel Queuing node (in the Queue tab of the Palette).

**Figure 42: The Cancel Queuing Icon**

You do not have to define properties for the Cancel Queuing node. You can optionally add comments or connection labels.
Temporarily Halt Script Execution

You can halt script execution for a specified number of seconds by using the Wait node (in the Queue tab of the Palette).

Figure 43: The Wait icon

The Wait node simply stops script executing for the specified number of seconds. In the meantime, the Network VRU is waiting for instructions.

Warning

You must set protocol time-out variables in the VRU system to a value greater than the longest wait node used in the script.

Following is the Properties dialog box for the Wait node:

Figure 44: Wait Properties

Define Wait node properties as follows:

Procedure

Step 1

In the Timeout in field, specify an interval to wait, in seconds.

Step 2

Optionally, add comments or connection labels.
Multichannel Routing

Overview of Multichannel Services

When Unified ICM is integrated with Unified WIM or Unified EIM, you write routing scripts to route contacts that are handled by these applications.

Supported Route Requests

Packaged CCE supports the following types of multichannel route requests:

• **Web callback** - A web callback request is one that does not involve Unified WIM. A customer clicks a button on a website that says, "Call me back." Then the caller and agent simply talk on the phone.

• **Blended collaboration** - With blended collaboration, the caller and agent talk on the phone and are linked in a collaborative web session. They can share web pages, forms, and applications, while at the same time conducting a voice conversation.

• **Text chat** - The caller and agent can conduct a text chat session when a telephone call is not desired or not possible. They can both chat and collaborate on the web.

• **E-mail message** - The customer and agent communicate using electronic mail.

Application Request Routing

The Unified WIM or Unified EIM applications route requests to Packaged CCE Media Routing Peripheral gateway (MR-PG). The Media Routing Peripheral Interface Manager (MR-PIM) on the MR-PG provides a generic interface to queue and route requests. The MR-PIM communicates with the CallRouter, which runs a routing script to determine how best to handle the request.

Packaged CCE uses a media class ID to identify the type of media or channel. A media class is a communication channel that is correlated to an application. Cisco_Voice is a predefined media class that is used for web and delayed callbacks requests and Packaged CCE inbound and outbound voice calls.

Each media class has at least one Media Routing Domain (MRD), which is a collection of skill groups associated with a medium. The Packaged CCE uses the MRD to route a task to an agent who is associated with a skill group and a particular medium. Each MRD requires a Packaged CCE script, but it is possible to route requests from different MRDs using one script.

Synchronized Agents and Skill Groups

Agents are common across the multichannel software, but skill groups are application-specific. Agents can be created in the Unified WIM or Unified EIM applications or in Packaged CCE and agents can be shared across applications. When agents or skill groups are created in either Unified WIM or Unified EIM, they are simultaneously created in Packaged CCE. If an agent is created in Packaged CCE, the agent must be enabled in the Unified WIM and Unified EIM applications if the agent wants to log in to and work on those applications.
Skill groups are application-specific, and even though they are simultaneously created in Packaged CCE when they are created in Unified WIM or Unified EIM, do not create, modify, or delete them in Packaged CCE. You cannot enable skill groups in the applications.

**Independent Media Queues**

You can configure the multichannel software to route all media through independent queues defined by media class. You can configure agents to log in to only one media type to take either e-mail, text chat, or voice. In this configuration, requests are queued only to agents who have logged in to the corresponding media application.

**Universal Queue**

**Universal Queue Functionality**

Universal Queue is the term used to describe the system's ability to route requests from different channels to agents who work with customer contacts in multiple media. With Universal Queue, the Packaged CCE treats requests from different channels as part of a single queue. Routing scripts can send requests to agents based on business rules regardless of the channel from which the request came. For example, the Packaged CCE can route phone, chat, and e-mail message requests to an agent who works with all these channels, based on the agent's skills and current tasks.

**Universal Queue Configuration Overview**

You can configure the multichannel software to manage a single universal work queue for all requests. You can configure agents to handle all media types, switching media on a task-by-task basis. For example, you can configure an agent as a member of three skill groups if the agent handles voice, e-mail, and chat, and the agent logs in to the softphone, the Unified EIM, and the Unified WIM. The agent is assigned the longest waiting request from any of the three skill groups, or you can choose to prioritize the requests using the multichannel software scripting environment.

**Multichannel scripts and media routing domains**

**Media Routing Domains**

A Media Routing Domain (MRD) is a collection of skill groups and services associated with a specific communication channel. For example, the Unified EIM uses Packaged CCE MRD to route a task to an agent who is associated with a skill group and a particular channel. MRDs are defined in the Packaged CCE configuration and have unique IDs across the enterprise.

**Media Routing Domains and Interruptibility**

When you configure MRDs, you indicate whether tasks for the MRD are interruptible. If the MRD is not interruptible, and agent working on tasks for that MRD is not assigned tasks from other MRDs. If the MRD is interruptible, the agent may be assigned tasks from another MRD.
Typically, voice calls and Unified WIM tasks are not interruptible, as agents are interacting with customers synchronously. E-mail messages, on the other hand, are typically interruptible because contact with the customer is asynchronous. Therefore, an agent responding to an e-mail message may be interrupted by a phone call or chat session.

Use Media Routing Domains to Categorize Contacts

You can categorize contacts based on the MRD based on the route request made by the MRD using the Media Routing Domain node (in the Routing tab of the Palette).

For example, you can have different MRDs for an Unified EIM instance and Unified WIM chat. You can have a single script for both types of requests that branches so that it routes e-mail messages and chats to different targets.

You must insert targets and connections from the MRD node before you define the node's properties.

A branch can include multiple MRDs, but a single MRD can only be associated with one branch.

Following is the Properties dialog box for the Media Routing Domain node:

Define Media Routing Domain node properties as follows:
Procedure

**Step 1** To associate a MRD with a branch: Select the branch:
   a) Click **Add**.
   b) Choose a MRD from the drop-down list.

**Step 2** To delete a branch, select it and click **Delete**.

**Step 3** To rename a branch, select it, click **Rename**, and type the new name.

**Step 4** You can define a branch as Otherwise by selecting the branch and clicking **Make Otherwise**. Execution follows this branch if none of the specified time ranges apply. You can specify only one Otherwise branch for the node.

What to Do Next

For more information, see *Media Routing Domain List*, on page 88.

Queue to Agent Node

You can queue a contact directly to an agent by using the Queue to Agent node (in the Queue tab of the Palette).

You can change the Queue to Agent type to:

- Specify an agent directly
- Select an agent by expression
Procedure

Step 1  In the Queue to Agent properties dialog box, click **Change**. The Queue Agent Type dialog box opens:

*Figure 47: Queue Agent Type*

Step 2  To select a specific agent, select **Explicit agent references**.

Step 3  To select and agent by an expression, select **Lookup agent references by expression**.

Step 4  Select a **Priority** between 1 (the highest) and 10 (the lowest).

Step 5  Optionally, select **Enable target requery**.
Specify an Agent Directly

Following is the properties dialog box of the Queue to Agent node when you select to specify agents directly:

**Figure 48: Agent Direct Properties**

To specify agents directly:

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>If necessary, change the Queue to Agent type to <strong>Explicit agent references</strong>.</td>
</tr>
<tr>
<td>Step 2</td>
<td>In the <strong>Agent</strong> column, select an agent.</td>
</tr>
<tr>
<td>Step 3</td>
<td>In the <strong>Media Routing Domain</strong> column, select the media routing domain for the selected agent.</td>
</tr>
<tr>
<td>Step 4</td>
<td>In the <strong>Skill Group</strong> column, select the skill group for the selected agent and media routing domain.</td>
</tr>
<tr>
<td>Step 5</td>
<td>In the <strong>Route</strong> column, select the route for the selected agent and media routing domain.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Optionally, select <strong>Queue if agent not logged in</strong>, to have the contact queued to the agent even if the agent is not currently logged in.</td>
</tr>
<tr>
<td>Step 7</td>
<td>To test the data you entered, click <strong>Validate</strong>.</td>
</tr>
<tr>
<td>Step 8</td>
<td>Optionally, modify <strong>Connection Labels</strong>.</td>
</tr>
</tbody>
</table>
Use of Formulas

You can use formulas in many routing nodes to both Contact Categorization and Selection of Routing Targets.

Formula Usage

A formula consists of one or more expressions that the Packaged CCE evaluates to produce a value that it can use for subsequent script processing. You define expressions—made up of Variable Usage, constants, operators, and functions—as part of custom selection rules or distribution criteria in scripts.

Formula Example

Following is a simple example of a formula:

\[ \text{CallerEnteredDigits} == 1 \]

In this example:

- The Left-value, CallerEnteredDigits, is a variable. More specifically, it is a call control variable.
- The operator is the "EqualTo" equality operator.
- The Right-value is the number 1.

If the value of CallerEnteredDigits is 1, the formula returns true; otherwise, the formula returns false.

Variables

Variable Usage

A variable is a named object that holds a value. You use variables in formulas to select targets and help in call tracking.

Variable Syntax

Following is the syntax for using a variable in a formula:

\[ \text{object-type.object-name.variable-name} \]

Where:

- The object-type is an object category, such as SkillGroup.
- The object-name is the name of an object contained in Packaged CCE database, such as the name of a SkillGroup (for example, BosSales).
- The variable-name is the name of an object that can hold a value, such as call information for the Skill Group; for example, (CallsInProgress).
- Each component in the variable is separated by a period (.)
Single-Target Variables

A single-target variable examines data for one specified routing target. For example, the variable:

`SkillGroup.BosSales.CallsInProgress`: Examines the number of calls in progress for the BosSales skill group.

Multiple-Target Variables

A multiple-target variable examines data across multiple routing targets. For example, the function:

`Max(SkillGroup.*.LongestAvailable)`: Finds the skill group, from all skill groups defined in the target set for the script node that calls the function, with the longest available agent.

You use an asterisk (*) as the object-name value to indicate that the variable is to examine data across multiple targets.

Call Control Variables

Call control variables provide information about the current contact that is being routed by the script. Call control variables include information about where the route request came from, contact classification data, and data to be passed to the peripheral that receives the contact.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
<th>Can be Set by the User</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallerEnteredDigits</td>
<td>String</td>
<td>Digits caller entered in response to prompts.</td>
<td>Yes</td>
</tr>
<tr>
<td>CallingLineID</td>
<td>String</td>
<td>Billing telephone number of the caller.</td>
<td>No</td>
</tr>
<tr>
<td>DialedNumberString</td>
<td>String</td>
<td>Telephone number dialed by the caller.</td>
<td>No</td>
</tr>
<tr>
<td>ExpCallVarName</td>
<td>String</td>
<td>Expanded Call Context (ECC) variable value assigned in scripts and passed with contact.</td>
<td>Yes</td>
</tr>
<tr>
<td>PeripheralVariable1-PeripheralVariable10</td>
<td>String</td>
<td>Values passed to and from the peripheral.</td>
<td>Yes</td>
</tr>
<tr>
<td>RequeryStatus</td>
<td>Integer</td>
<td>Provides the ability to test the error path of the Label, Queue, RouteSelect, and Select nodes to determine the specific network cause of failure and conditionally retry the attempt as necessary.</td>
<td>No</td>
</tr>
<tr>
<td>RouterCallDay</td>
<td>Integer</td>
<td>An encoded value that indicates the date on which Packaged CCE processes the call.</td>
<td>No</td>
</tr>
</tbody>
</table>
### Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
<th>Can be Set by the User</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterCallKey</td>
<td>Integer</td>
<td>A value that is unique among all calls Packaged CCE has processed since midnight. RouterCallDay and RouterCallKey combine to form a unique call identifier.</td>
<td>No</td>
</tr>
<tr>
<td>RoutingClient</td>
<td>String</td>
<td>The name of the routing client that made the route request.</td>
<td>No</td>
</tr>
<tr>
<td>TimeInQueue</td>
<td>Integer</td>
<td>Number of seconds a call has been queued.</td>
<td>No</td>
</tr>
<tr>
<td>UserToUserInfo</td>
<td>String</td>
<td>ISDN private network User to User information</td>
<td>Yes</td>
</tr>
<tr>
<td>VruStatus</td>
<td>Integer</td>
<td>Indicates the result of a previous VRU node.</td>
<td>No</td>
</tr>
<tr>
<td>CallGUID</td>
<td>varchar(32)</td>
<td>Globally unique call identifier.</td>
<td>No</td>
</tr>
<tr>
<td>LocationParamName</td>
<td>varchar(50)</td>
<td>Location name.</td>
<td>No</td>
</tr>
<tr>
<td>PstnTrunkGroupID</td>
<td>varchar(32)</td>
<td>The Trunk Group ID on which the call arrived on IOS Gateway.</td>
<td>No</td>
</tr>
<tr>
<td>PstnTrunkGroupChannelNumber</td>
<td>Integer</td>
<td>The Trunk Group Channel Number on which the call arrived on IOS Gateway.</td>
<td>No</td>
</tr>
<tr>
<td>SIPHeader</td>
<td>varchar(255)</td>
<td>Specific header information extracted from a SIP call that arrives at CVP (or VRU).</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Note**

The Call Variables are used in "SET" node by several customers in an Admin Script as temporary placeholders for complex calculation. However, because any call context is only existent as long as the call itself, the Variables cease to exist after the Route Request (a.k.a Call) is complete (be it by virtue of a successful Routing Script Execute Completion or an Administrative Script Execute Completion). They cannot be used to store values, so as to be re-used in Routing Scripts, as the Routing Scripts itself will have a new set of CallVariables created for the Route Request.
When comparing two Call Variables of Numeric string, you must use the Built-In Function "value()" in the IF Node to perform Numeric comparison, otherwise there is a String comparison. Ex: value(Call.PeripheralVariable1) >= value(Call.PeripheralVariable2) where Call.PeripheralVariable1 and Call.PeripheralVariable2 are given as Numeric string.

**Expanded Call Variables**

Expanded Call variables store values associated with the contact. Expanded call values are written to Termination Call Detail records only if, and when, an ECC value is explicitly set, which can be done any numbers of ways, such as using a script, a CVP, CTI, and so on. This applies to null values as well as non-null values.

If an expanded call variable is defined, but never assigned a value, it does not have a row in the Termination Call Variable table when a Termination Call Detail record is written.

Generally speaking, expanded call values are passed from leg to leg on the call. After a value is assigned, the value is recorded in the Termination Call Variable for every Termination Call Detail Segment. However, this depends on how each new call segment is created.

The Packaged CCE is delivered with some ECC variables, and you can create others through the Unified CCE Administration tool.

**Persistent vs. Non-Persistent Call Variables**

When Packaged CCE writes call data records to its historical database, it can store the values of all call variables. Storing excessive call variable data can degrade historical database performance. When you define a call variable (using the Expanded Call Variables gadget in the Unified CCE Administration web tool), you can tag it as either persistent or non-persistent. Only persistent call variables are written to the historical database. You can use non-persistent variables in routing scripts, but they are not written to the database.

**User Variables**

User variables are variables you create to serve as temporary storage for values you can test with an If Node. For example, you could create a user variable called usertemp to serve as a temporary storage area for a string value used by an If node.

You create user variables through the Unified CCE Administration tool. For more information, see the User Variable List, on page 91 in Packaged CCE Administration Guide for Release 9.0(x).

Each user variable must:

- Have a name that begins with user.
- Be associated with an object type, for example, SkillGroup. (This enables the Packaged CCE to maintain an instance of that variable for each object of that type in the system.)
- Be identified as a persistent (retains value across CallRouter restarts) or non-persistent (does not retain value across CallRouter restarts) variable.
- Store a value up to 40 characters long.
After you have define a variable, you can use the Formula Editor to access the variable and reference it in expressions, just as you would with a built-in variable.

### Set Variable Node Usage

**Figure 49: Set Properties window**

You can set the value of a variable with the Set Variable node:

- **Object type** - Choose the type of object the variable is associated with.

- **Object** - Choose the specific object the variable is associated with.

  - **Note** If you choose Call as the Object Type, this field does not apply.

- **Variable** - The specific variable you want to set.

  - **Note** The variables that are available are determined by the value you choose in the Object Type field.

  - **Note** Define all integer fields in tables accessed by a Set Variables node as NOT NULL.

- **Array index** - Enter an integer or an expression that evaluates to an integer. For example, if the Array Index expression evaluates to 2, then the Set Variable node sets the second element of the variable array.

  - **Note** This field is only available if you select an array variable in the Variable field.
Value - Enter the value to assign to the variable. The value can be:

- A constant
- A reference to another variable
- An expression

SkillGroup.Avail and SkillGroup.ICMAvailable Variables

When Packaged CCE system includes only the voice channel, the value of the SkillGroup.Avail variable is the number of agents in the available state, meaning that the agents are able to accept new calls.

However, when the web or E-mail channel is used with non-voice Media Routing Domains and agents log in to multiple domains, the value of the SkillGroup.Avail variable is calculated differently. There is also a SkillGroup.ICMAvail variable.

The following table describes the difference between the SkillGroup.Avail and the SkillGroup.ICMAvail variables:

<table>
<thead>
<tr>
<th>Case</th>
<th>SkillGroup.Avail</th>
<th>SkillGroup.ICMAvailable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only voice domain is used</td>
<td>Number of agents in the Available state.</td>
<td>Same</td>
</tr>
<tr>
<td>Multiple Domains are used</td>
<td>Number of agents in the Available state, regardless of what they may be doing in this or other domains.</td>
<td>Number of agents who can actually handle an additional task or call in the domain.</td>
</tr>
</tbody>
</table>

SkillGroup.ICMAvailable Variable

The value of the SkillGroup.ICMAvailable variable is the actual number of agents logged in to the skill group who can take new calls or tasks. Such agents must meet all the following criteria:

1. They are routable in the domain.
2. The agent's state in the domain is something other than "Not-Ready".
3. The agent is below the maximum task limit.

Note: For most domains (that is, if the agent is not a Unified WIM Multi-session agent), the maximum task limit is 1, and an agent is below the maximum only when the agent is not working on any call or task.

4. The agent is not working on another task in a non-interruptible domain.

SkillGroup.Avail Variable

SkillGroup.Avail is the number of agents in the skill group who are not doing anything in the domain. An agent who is logged in to two domains can be counted as Avail in one domain even though that agent is handling a task in another non-interruptible domain. An agent in a domain that handles multiple tasks (such
as chat) is not counted as Avail if that agent is handling a task, even though the agent has additional capacity for more tasks.

The following table shows some possible values for these variables. Assume three agents are logged in to a voice skill group, and the same three agents are also logged in to another non-interruptible domain, such as Unified WIM Chat. This table shows the voice skill group states.

<table>
<thead>
<tr>
<th>Case</th>
<th>SkillGroup.Avail</th>
<th>SkillGroup.Avail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial state</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>First agent handles a call</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Second agent handles a Collaboration chat session</td>
<td>2 (because there are two agents doing nothing in the domain)</td>
<td>1 (because there is only one agent left to handle voice calls)</td>
</tr>
<tr>
<td>Voice call ends</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Collaboration chat ends</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

If a routing script needs to check the number of available agents, using SkillGroup.Avail produces better results as it uses an extrapolation mechanism in determining the available agent.

Following is another example showing agents handling chat. Assume three agents logged in to a chat skill group, each capable of handling two chats. This table shows states for the chat group.

<table>
<thead>
<tr>
<th>Case</th>
<th>SkillGroup.Avail</th>
<th>SkillGroup.TalkingIn</th>
<th>SkillGroup.ICMAvailable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial state</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>First agent handles a Collaboration chat session</td>
<td>2 (because the agent is now in the talking state)</td>
<td>1</td>
<td>3 (because all three agents can still handle additional chats)</td>
</tr>
<tr>
<td>Second agent handles a Collaboration chat session</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>First agent handles second Collaboration chat session</td>
<td>0</td>
<td>3 (even though a total of 4 chats are in progress, only 3 agents are doing the work)</td>
<td>2 (because only the second and third agents can handle an additional chat)</td>
</tr>
</tbody>
</table>

By default, Script Editor shows the ICMAvailable value instead of Avail value when displaying skill group real-time data.
Closed Variables

Closed variables are available for use for Skill Groups, Peripherals, and Media Routing Domains. Closed variables allow administration scripts to turn dequeuing to these objects on and off. The Closed variables default to 0, meaning that the object is open. A script (usually an administration script) can change the state of the Closed variables.

If a Closed flag is set to a non-zero integer, then calls are not dequeued to affected agents, regardless of their state.

When closed variables are set to zero, the queued calls do not go to the available agents immediately, and continue to be in the queue. When the agent state changes from "Not Ready" to "Ready" state, the new calls are sent to the available agents (agents in the "Ready" state) only, and not the queued calls.

Operator Precedence

The following table shows the order in which operators are evaluated.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Operator type</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Prefix (unary)</td>
<td>+ - ! ~</td>
</tr>
<tr>
<td>2</td>
<td>Multiplication and division</td>
<td>* /</td>
</tr>
<tr>
<td>3</td>
<td>Addition and subtraction</td>
<td>+ -</td>
</tr>
<tr>
<td>4</td>
<td>Shift right and shift left</td>
<td>&gt;&gt; &lt;&lt;</td>
</tr>
<tr>
<td>5</td>
<td>Relational</td>
<td>&lt; &gt; &lt;= &gt;=</td>
</tr>
<tr>
<td>6</td>
<td>Equality</td>
<td>== !=</td>
</tr>
<tr>
<td>7</td>
<td>Bitwise And</td>
<td>&amp;</td>
</tr>
<tr>
<td>8</td>
<td>Bitwise exclusive Or</td>
<td>^</td>
</tr>
<tr>
<td>9</td>
<td>Bitwise inclusive Or</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>And</td>
<td>&amp;&amp;</td>
</tr>
<tr>
<td>11</td>
<td>Or</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Conditional</td>
<td>?</td>
</tr>
<tr>
<td>13</td>
<td>Sequential</td>
<td>,</td>
</tr>
</tbody>
</table>

Note: The operators with priority 1 are evaluated first, then those with priority 2, and so on. The order of evaluation within each priority level can also be important. Prefix operators are evaluated from right-to-left in an expression. Assignment operators are also evaluated from right-to-left. In all other cases where operators have equal priority, they are evaluated left-to-right.
Operators

Prefix Operators

The Prefix Operators in the following table take a single operand:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Comments/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Positive</td>
<td>Numeric values are positive by default, so the positive operator (+) is optional. Example: 2 and +2 represent the same value.</td>
</tr>
<tr>
<td>-</td>
<td>Negative</td>
<td>The negative operator (-) changes the sign of a value. Example: 2 represents a positive value; -2 represents a negative value.</td>
</tr>
<tr>
<td>!</td>
<td>Logical negation</td>
<td>A logical expression is any expression that evaluates to true or false. The logical negation operator (!) changes the value of a logical expression. Note: Numerically, a false value equates to 0 and a true value equates to a non-zero value. Example: If the current value of SkillGroup.Sales.Avail is 3, then SkillGroup.Sales.Avail &gt; 0 is true and (SkillGroup.Sales.Avail &gt; 0) is false.</td>
</tr>
<tr>
<td>~</td>
<td>One's complement</td>
<td>Operates on a bit value, changing each 1 bit to 0 and each 0 bit to 1. Note: This operator is rarely used.</td>
</tr>
</tbody>
</table>

Arithmetic Operators

The Arithmetic Operators in the following table take two operands:
Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Comments/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Multiplication</td>
<td>Arithmetic operators perform the basic operations of addition, subtraction, multiplication and division. You can use them in making calculations for a skill group, service, or route. Note: Multiplication (*) and division (/) operators are evaluated before addition (+) and subtraction (-) operators. Examples: Returns the number of agents who are logged in to the service but not currently available. Evaluates to because the multiplication is performed first.</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
<td></td>
</tr>
<tr>
<td>+</td>
<td>Addition</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
<td></td>
</tr>
</tbody>
</table>

Equality Operators

The Equality Operators in the following table take two operands:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Comments/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>Equal to</td>
<td>Equality operators allow you to determine whether two values are equivalent or not. Examples: Is true if any calls are currently queued for the service. Is true if all agents logged in to the service are currently available.</td>
</tr>
<tr>
<td>!=</td>
<td>Not Equal To</td>
<td></td>
</tr>
</tbody>
</table>

Relational Operators

The Relational Operators in the following table take two operands:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Comments/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>Greater than</td>
<td>Relational operators allow you to perform a more sophisticated comparison than the equality operators. Examples: Is true if more members of the skill group are in the Not Ready state. Is true if at least as many agents are Ready as Not Ready.</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
<td></td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater Than or Equal To</td>
<td></td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less Than or Equal To</td>
<td></td>
</tr>
</tbody>
</table>
**Logical Operators**

The Logical Operators in the following table take two operands. Logical operators examine the values of different logical expressions:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Comments/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;&amp;</td>
<td>And</td>
<td>The expression is true if both of the operands are true. If either is false, the overall expression is false. The following is true if the skill group has at least one agent logged in and no agents are currently available.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note**

The equality (==) and relational (>) operators are evaluated before the logical operators (&& and ||).

**Bitwise Operators**

The Bitwise Operators in the following table take two operands.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Comments/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>And</td>
<td>The &amp; Bitwise Operator turns specific bits in a value on or off. The following expression turns off the six low-order bits of AvgTalkTimeTo5. Note the use of the complement operator (~) with the constant. This is equivalent to rounding the value down to the next multiple of 64.</td>
</tr>
<tr>
<td>Operator</td>
<td>Meaning</td>
<td>Comments/Examples</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>Inclusive Or</td>
<td>Inclusive Or and Exclusive Or differ in the way they handle the case where bits in both values are 1: Inclusive Or evaluates the result as true and sets a 1 bit in the result. Exclusive Or evaluates the result as false and sets a 0 bit in the result. (An Exclusive Or applies the rule &quot;one or the other, but not both&quot;).</td>
</tr>
<tr>
<td>^</td>
<td>Exclusive Or</td>
<td></td>
</tr>
</tbody>
</table>

**Miscellaneous Operators**

The following table lists miscellaneous operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Comments/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Conditional</td>
<td>The conditional operator (?) takes three operands and its syntax is as follows: The Packaged CCE evaluates the expression by first examining the logical expression condition and then tests the following condition: If the result is true, then the overall expression evaluates to the value of the expression true-result. If the result is false, then the overall expression evaluates to the expression false-result. The following expression determines whether the number of agents available for skill group S1 is even or odd.</td>
</tr>
<tr>
<td>&amp;</td>
<td>Concatenation</td>
<td>The concatenation operator (&amp;) joins two strings end-to-end. returns the value.</td>
</tr>
</tbody>
</table>
The sequential or comma operator (,) takes two operands, each of which is an expression. Packaged CCE evaluates the left expression first and then the right expression. The value of the overall expression is the value of the right expression. The first expression typically affects the valuation of the second.

The shift left (<<) and shift right (>>) operators shift the bits within a value. The following example shifts the bits in Avail two places to the left. The two right-most positions are filled with zeroes: The following example shifts the bits in Avail two places to the right. In this case, the positions are filled with sign bits (0 if the original value is positive or zero; 1 if the original value is negative).

### Built-in functions

#### Date and Time Functions

The following table lists the built-in date and time functions:
<table>
<thead>
<tr>
<th>Function</th>
<th>Data Type</th>
<th>Return Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>date [ (date) ]</td>
<td>Integer</td>
<td>Returns the current system date or the date portion of a given date-time value. The given date can be a floating point value (as returned by the now function), a string of the form mm/dd/yy, or three integers: yyyy, mm, dd. date (with no arguments) returns the current date. = date (2001, 7, 15) tests whether the current date is July 15, 2001.</td>
</tr>
<tr>
<td>day [ (date) ]</td>
<td>Integer</td>
<td>Returns the day of month (1-31) for the current date or a given date. The given date must be an integer or a floating-point value, as returned by the date or now function. Tests whether tomorrow is the first of the month.</td>
</tr>
<tr>
<td>hour [ (time) ]</td>
<td>Integer</td>
<td>Returns the hour (0-23) of the current time or a given time. The given time must be a floating-point value, as returned by the now function. Tests whether the current time is before noon.</td>
</tr>
<tr>
<td>minute [ (time) ]</td>
<td>Integer</td>
<td>Returns the minutes (0-59) of the current time or a given time. The given time must be a floating-point value as returned by the time function. Tests whether the current time is in the second fifteen-minute interval after an hour.</td>
</tr>
<tr>
<td>month [ (date) ]</td>
<td>Integer</td>
<td>Returns the month (1-12) of the current month or a given date. The given date must be a floating-point value, as returned by the date or now function. Tests whether the current month is June.</td>
</tr>
</tbody>
</table>

**Note**: Do not use the slash (/) character in defining a date function. Because it is the division operator, the function would not return the results you are looking for. You can enclose the argument within a string.
### Built-in functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Data Type</th>
<th>Return Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>now</td>
<td>Float</td>
<td>Returns the current date and time, with the date represented as an integer and the time represented as a fraction. Note: You can use the date or time functions without any arguments to return just the current date or time. This function is useful for comparing the current date and time to a specific point in time. To test whether the current date and time is later than 10 P.M., December 24, 2001, use the expression.</td>
</tr>
<tr>
<td>second [ (time) ]</td>
<td>Integer</td>
<td>Returns the seconds (0-59) of the current time or a given time. The given time must be a floating-point value, as returned by the time function. Tests whether the current time is within the last ten seconds of a minute.</td>
</tr>
<tr>
<td>time [ (time) ]</td>
<td>Float</td>
<td>Returns the current system time or the time portion of a date-time value. The given time can be a floating point value, a string of the form hh:mm:ss, or two or three numeric values: hh, mm [, ss ]. (with no arguments) returns the current time. Tests whether the current time is after 2:00 PM.</td>
</tr>
<tr>
<td>weekday [ (date) ]</td>
<td>Integer</td>
<td>Returns the current day of week (Sunday=1, Monday=2, etc.) of the current date or given date. The given date must be an integer or floating-point value, as returned by the date or now function. Tests whether today is Tuesday.</td>
</tr>
<tr>
<td>year [ (date) ]</td>
<td>And</td>
<td>Returns the year of the current year or given date. The given date must be a floating-point value, as returned by the date or now function. Tests whether the millennium has passed.</td>
</tr>
</tbody>
</table>
**Mathematical Functions**

The following table lists the built-in mathematical functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Data Type</th>
<th>Return Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs(n)</td>
<td>Floating Point or Integer</td>
<td>Returns the absolute value of (the number with no sign). Returns the value 15.</td>
</tr>
<tr>
<td>max(n1, n2 [,n3] . . )</td>
<td>Floating Point or Integer</td>
<td>Returns the largest of the operands. Each operand must be numeric. Returns the value 3.</td>
</tr>
<tr>
<td>min(n1, n2 [,n3] . . )</td>
<td>Integer</td>
<td>Returns the smallest of the operands. Each operand must be numeric. Returns the value -2.</td>
</tr>
<tr>
<td>mod(n1,n2)</td>
<td>Floating Point or Integer</td>
<td>Returns the integer remainder of n1 divided by n2. Returns the value 99.</td>
</tr>
<tr>
<td>random()</td>
<td>Floating Point or Integer</td>
<td>Returns a random value between 0 and 1.</td>
</tr>
<tr>
<td>sqrt(n)</td>
<td>Floating Point or Integer</td>
<td>Returns the square root. (The operand n must be numeric and non-negative). Returns the value 7.</td>
</tr>
<tr>
<td>trunc(n)</td>
<td>Floating Point or Integer</td>
<td>Returns the value of truncated to an integer. Returns the value 28.</td>
</tr>
</tbody>
</table>

**Miscellaneous Functions**

The following table lists the built-in miscellaneous functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Data Type</th>
<th>Return Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>after(string1,string2)</td>
<td>String</td>
<td>That portion of string2 following the first occurrence of string1. If string1 does not occur in string2, the null string is returned. If string1 is the null string, string2 is returned. Returns the value defg.</td>
</tr>
</tbody>
</table>
### Function | Data Type | Return Value/Example
--- | --- | ---
before(string1,string2) | String | That portion of string2 that precedes the first occurrence of string1. If string1 does not occur in string2, string2 is returned. If string1 is the null string, the null string is returned. Returns the value abc.

ClidInRegion | Logical | Indicates whether the CLID for the current contact is in the geographical region specified by string. The value string must be the name of a defined region. You can use the Name variable of a region to avoid entering a literal value. Tests whether the CLID is from the Maryland region.

concatenate(string1,string2, ...) | String | Returns the concatenation of the arguments. The function takes up to eight arguments. Returns the value abcde.

find(string1, string2 [,index ]) | Integer | Returns the starting location of string1 within string2. If you specify an index value, searching starts with the specified character of string2. Returns the value 6.

if(condition,true-value,false-value) | Logical | Returns a value of true-value if the condition is true; false-value if the condition is false. Returns the current hour in 12-hour format rather than 24-hour format.

left(string,n) | String | Returns left-most n characters of string. Returns the value abc.

len(string) | Integer | Returns the number of characters in string. Returns the value 3.

mid(string,start,length) | String | Returns a substring of string beginning with the start character and continuing for length characters. Returns the value bed.
<table>
<thead>
<tr>
<th>Function</th>
<th>Data Type</th>
<th>Return Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>result</td>
<td>Floating Point or Integer</td>
<td>Returns the result of the current Select node. (This function is valid only in a Select node.) If you are using the LAA rule in the Select node, result Returns the number of seconds the selected agent has been available.</td>
</tr>
<tr>
<td>right(string,n)</td>
<td>String</td>
<td>Returns right-most n characters of string. Returns the value cde.</td>
</tr>
<tr>
<td>substr(string,start [, length ])</td>
<td>String</td>
<td>Returns a substring of string beginning with start character and continuing for length characters. Returns the value 01851.</td>
</tr>
<tr>
<td>text(n)</td>
<td>String</td>
<td>Converts a numeric value to a string. Returns the value &quot;5&quot;.</td>
</tr>
<tr>
<td>valid(variable)</td>
<td>Logical</td>
<td>Returns whether variable has a valid value. Tests whether the database C name is a valid value.</td>
</tr>
<tr>
<td>ValidValue(variable,value)</td>
<td>String</td>
<td>If variable has a valid value, returns that value; otherwise, returns &quot;value&quot;. Returns either a name from the database or the string value None.</td>
</tr>
<tr>
<td>value(string)</td>
<td>Floating Point or Integer</td>
<td>Converts a string to a numeric value. Returns the value 5.</td>
</tr>
</tbody>
</table>

### Custom Functions

Custom functions are those functions you create for use within scripts, as opposed to built-in functions.

### Add Custom Functions

**Procedure**

1. **Step 1** In Script Editor, from the Script menu, choose **Custom Functions**. The Custom Functions dialog box opens, listing all the custom functions currently defined.
2. **Step 2** Click **Add** to open the Add Custom Function dialog box.
3. **Step 3** Specify the following:
a) Function name. All custom function names must begin with user.
b) Number of Parameters. The number of parameters to be passed to the function. A function may take 0, 1, or more parameters.
c) Function definition. The expression to be evaluated when the function is called. When entering the function definition, keep the following in mind:
The parameters to a function are numbered beginning with 1. To reference a parameter within the expression, surround it with percent signs (%). For example, %3% is a reference to the third parameter.

The lower portion of the dialog box is just like the Formula Editor. You can use it to help build the expression.

Step 4 When finished, click Test. The Test Function dialog box opens.
Step 5 Test the function by entering an example of how you might reference the function. Include a specific value for each parameter.
Step 6 Click Evaluate to see how the Script Editor interprets the function call and click Close to return to the Add Custom Function dialog box.
Step 7 Use one of the Validate buttons to validate the scripts that reference a selection function. (The Validate All button lets you validate all the scripts that reference any custom function.)
Step 8 When finished, click OK to apply changes and to close the dialog box.

Import Custom Functions

Procedure

Step 1 In Script Editor, from the Script menu, choose Custom Functions. The Custom Functions dialog box opens, listing all the custom functions currently defined.
Step 2 Click Import. The Import Custom Function dialog box opens.
Step 3 Choose a file name with an ICMF extension (.ICMF) and click Open. The Script Editor examines the file for naming conflicts. If a conflict is found, a dialog box appears listing options for resolving the conflict.
Step 4 Choose one of the options and click OK.
Note If you choose to rename the function, the new name must begin with user.

The Script Editor performs automapping and the following happens:

• If all imported objects were successfully auto-mapped, a message window appears prompting you to review the mappings. Click OK to access the Object Mapping dialog box.

• If some imported objects were not successfully auto-mapped, the Object Mapping dialog box appears, with all unmapped objects labeled Unmapped.

The Object Mapping dialog box contains three columns:

• Object Types. The type of imported objects.
  • Imported Object. Name of imported object.
Mapped To. What this imported object will be mapped to.

- (Optional.) Click an Imported Object value. The Mapped To column displays all the valid objects on the target system.
- (Optional.) Choose an object from the Mapped To columns drop-down list on the target system that you want to map the imported object to.

Note

Multiple objects may be mapped to the same target. Objects may be left unmapped; however, the resulting custom function are not valid until all objects are mapped.

When the mapping is complete, click Apply and Finish.

Export Custom Functions

Procedure

Step 1
In Script Editor, from the Script menu, choose Custom Functions. The Custom Functions dialog box opens, listing all the custom functions currently defined.

Step 2
Choose the custom function(s) from the list and click Export. The Export Custom Function dialog box opens.

Note
If you selected a single function, that functions name appears in the File Name field. If you selected more than one function, the File Name field is blank.

Step 3
(Optional.) Change the File Name.

Step 4
Click Save. If the file name already exists, the system prompts you to confirm the save.

Step 5
If prompted, click OK. The custom function(s) are saved to the specified file in text format.

Scripts in a Packaged CCE Environment

Call Priority

When a call is queued to a skill group because there are no agents available, the Queue to Skill Group node sets the call's priority. The Queue Priority node can then promote the call's priority based on time the caller has waited. The call can be queued to multiple skill groups with the same or different priorities.

If there are calls in the agent's skill group queues when an agent becomes available, the agent is presented with the highest priority (1-10 with 1 being the highest priority) call that has waited the longest within the skill group(s) that the agent is assigned to.
Check for Available Agents

A script that routes to Packaged CCE agents needs to check for an available agent within a skill group. If an agent is not available, then the script should use a Queue to Skill Group node. The script execution ends when an agent becomes available or when the caller disconnects.

Queue to Skill Group Node

If you use the Queue to Skill Group node, then do not use the base priority within the Queue node unless the option is enabled in the router through the registry.

Cancel Queuing Node

If the call needs to be taken out of a skill group, then use the Cancel Queuing node. The Cancel Queuing node takes the call out of all the skill groups it is queued to.

End Node

The End node either tries default routing, or if there is no default label, it sends an error (dialog fail) to the routing client.

Agent to Agent Node

You can use the Agent to Agent node for agent to agent transfers; the router checks agent availability before sending the call to the agent. If the agent is not available, the script queues the call to a skill group. You can also use the Agent to Agent node to send a call to the agent: the "caller" is not required to be an agent.

CVP as a Queue Point

Packaged CCE relies on the CVP to queue the call while it is waiting for an available agent.

1. To provide the call with a termination point that allows the VoIP Gateway to return the correct signals or messages back to the PSTN.

2. Provide announcements or music or expected wait time or initial position in queue to the caller while they are waiting for an agent. Allow the caller an option to leave a message if the caller does not want to wait for an agent.

3. To obtain further information from the caller that is not sent from the network

The CVP lets Packaged CCE know when the caller disconnects through the Event Report Message with an Event Type of either DISCONNECT or ABANDON. When an agent becomes available, Packaged CCE automatically instructs CVP to route the call to the agent through the Connect message.
Interruptible vs. Non-Interruptible

If the VRU script is collecting digits from the customer to ascertain information regarding the caller that is crucial for a screen pop or call routing, put the VRU script in a non-interruptible mode.

If a call was queued to a skill group through a Queue to Skill Group node and then sent to VRU to hear a non-interruptible VRU script, if during the time that the caller is interacting and listening to the non-interruptible VRU script, an agent becomes available, the call will not be connected to the agent. The Packaged CCE only looks for available agents for that call when the VRU script is finished and the call executes an interruptible node such as a Wait node or a Run External Script node for a VRU script that is interruptible. The call does, however, maintain its place in the queue so when the call does become available for an agent, it is answered before calls that came in afterward it (assuming the same priority).

For announcement and music type of treatments, put the VRU Scripts in interruptible mode. This allows the call to be connected to the first available agent even while the caller is listening to a VRU script.

You set the interruptibility of a VRU script through the Network VRU Scripts gadget in the Unified CCE Administration Web tool. Neither the VRU or the Packaged CCE script can overwrite this setting.

Utility nodes

Start Node

The Start node marks the beginning of a script. The Script Editor automatically inserts the Start node when you create a new script; a script must have one and only one Start node.

You do not define any properties for the Start node. However, you can add comments and connection labels:

Figure 50: Start Properties
Comment Node

Use the Comment node (in the General tab of the Palette) to include a block comment in a script. A block comment provides general documentation for a script or section of a script:

*Figure 51: The Comment Icon*

For example, you might add a comment describing the purpose of the script.
You can move and resize the comment box within the script.

*Note*
If you choose the Auto-Size Height option, you cannot adjust the height of the comment.

Line Connector Node

Use the Line Connector node (in the General tab of the Palette) to make routing and administrative scripts clear and understandable.

*Figure 53: The Line Connector Icon*

A script can be difficult to understand and the call flow hard to follow if:

- The connecting lines between nodes are too long.
The connecting lines go in different directions.

The connecting lines run over other nodes and other connection lines.

The Line Connector node allows you to break and reconnect lines using one or more of its multiple input connections and single output connection. Any request coming into this node (on any one of the multiple inputs) goes to the single output connection of the line connector node.

For the Line Connector node, you define the connection labels:

![Line Connector Properties](image)

**CVP Scripting**

**Writing Scripts for Unified CVP**

This section discusses using Packaged CCE configuration and script editing to access the Unified CVP solution. It includes information about how to:

- Set up Packaged CCE to interact with Unified CVP
- Write applications for Unified CVP

---

**Note**

This section contains important information for CVP application developers. It also may be of interest to Call Center Managers, Unified CVP System Managers, and Packaged CCE system managers.
Before You Begin

This chapter makes the following assumptions:

- The information in this chapter assumes that you are already familiar with using the Unified CCE Administration and Script Editor tools for call center operations and management.

- 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.

Scripts to Access Unified CVP from Packaged CCE

Both Packaged CCE and Unified CVP use scripts to invoke their features. In fact, Packaged CCE references Unified CVP scripts from within its own scripts. This method of invoking Unified CVP from within Packaged CCE enables Packaged CCE to take advantage of the features of Unified CVP.

The two products (Packaged CCE 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 CVP applications.

| Note | For more information, refer to "Scripting for Unified CVP with Call Studio." |

Packaged CCE Scripting

The Packaged CCE 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 Packaged CCE routing script to produce a viable voice interaction with the caller.

While it is possible to develop full scale CVP applications using micro-applications, it is not recommended. 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 CVP scripts developed using Call Studio to create the CVP applications.

In an environment where Packaged CCE script works with Call Studio script (the 2-script implementation for Unified CCE-integrated models described here), the Packaged CCE script remains in control (and receives control back), even while it **delegates** the more complex self-service activity to the Call Studio script. Data can be passed from one script to the other and back through ECC variables.
The capability of using Packaged CCE 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 CVP applications.

**Unified CVP Call Studio Scripting**

Sophisticated CVP 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 for execution. To invoke a VXML Server application, the script writer includes a Get Speech (GS) micro-application via the Run External Script node in his Packaged CCE 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 Packaged CCE.

Some of the Call Studio scripting environment features include:

- A drag-and-drop interface with a palette of CVP functions
- The ability to do database queries
- Extensibility with Java code written to perform any task a Java application can perform

**Note**

Packaged CCE 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 Writing Packaged CCE 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 "Writing Packaged CCE Applications for Unified CVP."

**Scripting for Unified CVP with Packaged CCE**

The sections that follow include:

- A discussion of micro-applications.
- A sample Packaged CCE script.
- A discussion of how Packaged CCE and Unified CVP exchange information.

**Micro-Applications**

*Micro-applications* are a set of specific CVP functions that can be invoked by Packaged CCE, 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 playback type.

• **Get Digits.** Plays a media file and retrieves digits from the caller.

• **Menu.** Plays a media menu file and retrieves a single telephone keypad entry from the caller.

• **Get Speech.** Executes a Call Studio script on VXML Server.

• **Capture.** The Capture (CAP) micro-application enables you to trigger the storage of current call data at multiple points in the Packaged CCE routing script.

Micro-applications are interpreted by the CVP Service, which resides on the Call Server. The CVP Service sends VoiceXML code to the VoiceXML Gateway Voice Browser.

---

**Simple Example Script: Welcome to XYZ Corporation**

Suppose you want to create a Packaged CCE script that has an example call flow as follows. This simple script performs the following functions:

- Executes the GetInfo Call Studio script on the VXML Server to collect some caller input based on the example callflow.

- Based on caller input, queues for a sales or support agent .

- If an agent is not available, executes the MOH Call Studio script which will play music-on-hold to the caller until an agent becomes available.
### Procedure

**Step 1**  
A call arrives at CCE and executes a CCE script.

**Step 2**  
The caller hears a welcome prompt.

**Step 3**  
The CCE script sends the call to CVP for collecting some information from the caller before queuing the call for an agent. For example, a menu is offered such as "press 1 for sales and 2 for support," as well as entering an account number.

**Step 4**  
If the caller is an existing customer, the caller-entered account number is used to retrieve additional information about the caller from an external database.

**Step 5**  
Caller-entered digits and the additional information about the caller are returned back to the CCE script to be shown a screen pop to the agent, when an agent becomes available.

**Step 6**  
The call is then queued waiting for an agent in a particular skill group, based on the caller selection of the type of service.

**Step 7**  
If an agent is available, the caller is connected to that agent. The agent desktop displays the caller information collected via caller input as well as database lookup.

**Step 8**  
If an agent is not available, the call is sent back to CVP for playing music-on-hold while the caller waits for an agent to become available.

**Step 9**  
The information collected from the caller is preserved as call context on the call until the agent becomes available.

You can create a script such as the one shown in the following figure.

**Figure 55: Packaged CCE Script with Call Flow**

This simple script performs the following functions:

- Executes the GetInfo Call Studio script on the VXML Server to collect some caller input based on the example callflow.
- Based on caller input, queues for a sales or support agent.

---

**Packaged CCE Administration Guide for Release 9.0(x)**

169
• If an agent is not available, executes the MOH Call Studio script which will play music-on-hold to the caller until an agent becomes available.

Note  In a “real life” application, any Packaged CCE script you create would include error checking to ensure that micro-applications instructions are properly executed.

Packaged CCE Unified CVP Micro-App Connection

Before the Unified CVP can be accessible through the Script Editor’s Run External Script node, you must first set up Packaged CCE with special Unified CVP parameters using the Unified CCE Administration tool. Begin by using the Packaged CCE Administration's New Network VRU Script window to define Unified CVP parameters.

**Figure 56: New Network VRU Script Example**

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.
### Table 1: Attribute Values for Scripting with Packaged CCE

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Allowed Values</th>
<th>Applies to</th>
<th>Case-Sensitive?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attribute:</strong> VRU Script Name (for example, PM, GD).</td>
<td>PM, GD</td>
<td>All micro-applications</td>
<td>N</td>
</tr>
<tr>
<td><strong>Attribute:</strong> Media Library Type (A, S, V)</td>
<td>A, S, V</td>
<td>All micro-applications</td>
<td>N</td>
</tr>
<tr>
<td>Barge-in Allowed</td>
<td>Y/N</td>
<td>All micro-applications</td>
<td>N</td>
</tr>
<tr>
<td>Data playback type</td>
<td>Number, Char</td>
<td>PlayData (PD)</td>
<td>N</td>
</tr>
<tr>
<td>Time Format</td>
<td>HHMM, HHMMSS, HHMMAP</td>
<td>PlayData (PD)</td>
<td>N</td>
</tr>
<tr>
<td>Timeout Message Override</td>
<td>Y/N</td>
<td>Get Digits (GD), Get Speech (GS), Menu (M)</td>
<td>N</td>
</tr>
<tr>
<td>Invalid Entry Message Override</td>
<td>Y/N</td>
<td>Get Digits (GD), Get Speech (GS), Menu (M)</td>
<td>N</td>
</tr>
<tr>
<td>DTMF Termination Key</td>
<td>N</td>
<td>All micro-applications</td>
<td>N</td>
</tr>
<tr>
<td>Media File Name</td>
<td></td>
<td>All micro-applications</td>
<td>Y</td>
</tr>
</tbody>
</table>

Once the network VRU script configuration 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.

**Figure 57: Run External Script Node**

<table>
<thead>
<tr>
<th>ICM Script Name</th>
<th>External Script Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play_Welcome</td>
<td>PM_Welcome</td>
</tr>
<tr>
<td>Play_July</td>
<td>PM_July_S</td>
</tr>
<tr>
<td>Play_Number</td>
<td>TD_Number</td>
</tr>
<tr>
<td>Play_Cover</td>
<td>TD_Cover</td>
</tr>
<tr>
<td>Play_Time_HHMM</td>
<td>PD_Time_0</td>
</tr>
<tr>
<td>Play_TDD_0</td>
<td>PD_TDD_0</td>
</tr>
<tr>
<td>Get_Password_vwDef</td>
<td>GD_Password_AH_D0</td>
</tr>
<tr>
<td>Menu_Banking</td>
<td>MBanking</td>
</tr>
<tr>
<td>MAP_Menu</td>
<td>M_Main_Menu</td>
</tr>
</tbody>
</table>

Information Exchange Between Packaged CCE and Unified CVP

When Packaged CCE 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 CVP 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 (Packaged CCE /CVP Service Control Interface).

Packaged CCE 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 Packaged CCE scripts:

**ASR Error**

Failure of an Advanced Speech Recognition component.

**General error**

General error occurred.

**Invalid Configuration Param**

Data passed from Packaged CCE to the CVP 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 Packaged CCE to the CVP 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 Media, Get Digits, Menu, Get Speech, or Capture micro-applications.)

**Misconfigured ECC variable**

An ECC variable was set to a value the CVP Service did not recognize. ECC variable definitions must be the same in Packaged CCE 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 Packaged CCE to the CVP Service did not exist on the Media Server.

Unknown micro-application

Micro-application name passed from Packaged CCE to the CVP Service did not exist on the CVP Service.

Unsupported locale

The VoiceXML Interpreter (that is, gateway) did not recognize the locale passed from the CVP Service.

Unsupported VoiceXML element

The VoiceXML Interpreter (that is, gateway) did not recognize a VoiceXML element passed from the CVP Service, VXML Server, or media server.

Unsupported VoiceXML format

The VoiceXML Interpreter (that is, gateway) did not recognize a VoiceXML format passed from the CVP Service, VXML Server, or media server.

Each Unified CVP micro-application has individualized settings for `user микроаппп.ерроr_ код` as shown in the following table.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Play Media</th>
<th>Play Data</th>
<th>Get Digits</th>
<th>Menu</th>
<th>Get Speech</th>
<th>Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No error</td>
<td>No error</td>
<td>No error</td>
<td>No error</td>
<td>No error</td>
<td>No error</td>
</tr>
<tr>
<td>1</td>
<td>Caller Hangup</td>
<td>Caller Hangup</td>
<td>Caller Hangup</td>
<td>Caller Hangup</td>
<td>Caller Hangup</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>Network Error</td>
<td>Network Error</td>
<td>Network Error</td>
<td>Network Error</td>
<td>Network Error</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>System Error</td>
<td>System Error</td>
<td>System Error</td>
<td>System Error</td>
<td>System Error</td>
<td>System Error</td>
</tr>
<tr>
<td>4</td>
<td>Unknown micro-application</td>
<td>Unknown micro-application</td>
<td>Unknown micro-application</td>
<td>Unknown micro-application</td>
<td>Unknown micro-application</td>
<td>Unknown micro-application</td>
</tr>
<tr>
<td>5</td>
<td>Invalid VRU Script Name format</td>
<td>Invalid VRU Script Name format</td>
<td>Invalid VRU Script Name format</td>
<td>Invalid VRU Script Name format</td>
<td>Invalid VRU Script Name format</td>
<td>N/A</td>
</tr>
<tr>
<td>6</td>
<td>Invalid VRU Script Name format</td>
<td>Invalid VRU Script Name format</td>
<td>Invalid VRU Script Name format</td>
<td>Invalid VRU Script Name format</td>
<td>Invalid VRU Script Name format</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 3: Possible user микроаппп.ерроr_ код ECC Variable Settings for Non-Video
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Play Media</th>
<th>Play Data</th>
<th>Get Digits</th>
<th>Menu</th>
<th>Get Speech</th>
<th>Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Invalid Configuration Param</td>
<td>Invalid Configuration Param</td>
<td>Invalid Configuration Param</td>
<td>Invalid Configuration Param</td>
<td>Invalid Configuration Param</td>
<td>N/A</td>
</tr>
<tr>
<td>8</td>
<td>Misconfigured ECC variable</td>
<td>Misconfigured ECC variable</td>
<td>Misconfigured ECC variable</td>
<td>Misconfigured ECC variable</td>
<td>Misconfigured ECC variable</td>
<td>N/A</td>
</tr>
</tbody>
</table>
| 9          | One of the following:  
• Media file does not exist.  
• Invalid URL for Media file. | One of the following:  
• Media file does not exist  
• Invalid URL for Media file | One of the following:  
• Media file does not exist  
• Invalid URL for Media file | One of the following:  
• Media file does not exist  
• Invalid URL for Media file | One of the following:  
• Media file does not exist  
• Invalid URL for Media file | N/A     |
<p>| 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     |
| 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                | Reached Maximum Invalid Tries | Reached Maximum Invalid Tries | Reached Maximum Invalid Tries | Reached Maximum Invalid Tries | N/A     |
| 17         | N/A                | Reached Maximum No Entry Tries | 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     |</p>
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Play Media</th>
<th>Play Data</th>
<th>Get Digits</th>
<th>Menu</th>
<th>Get Speech</th>
<th>Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>No answer</td>
<td>No answer</td>
<td>No answer</td>
<td>No answer</td>
<td>No answer</td>
<td>N/A</td>
</tr>
<tr>
<td>24</td>
<td>Busy</td>
<td>Busy</td>
<td>Busy</td>
<td>Busy</td>
<td>Busy</td>
<td>N/A</td>
</tr>
<tr>
<td>25</td>
<td>General transfer error</td>
<td>General transfer error</td>
<td>General transfer error</td>
<td>General transfer error</td>
<td>General transfer error</td>
<td>N/A</td>
</tr>
<tr>
<td>26</td>
<td>Invalid extension</td>
<td>Invalid extension</td>
<td>Invalid extension</td>
<td>Invalid extension</td>
<td>Invalid extension</td>
<td>N/A</td>
</tr>
<tr>
<td>27</td>
<td>Called party hung up</td>
<td>Called party hung up</td>
<td>Called party hung up</td>
<td>Called party hung up</td>
<td>Called party hung up</td>
<td>N/A</td>
</tr>
<tr>
<td>28</td>
<td>Error after transfer established</td>
<td>Error after transfer established</td>
<td>Error after transfer established</td>
<td>Error after transfer established</td>
<td>Error after transfer established</td>
<td>N/A</td>
</tr>
<tr>
<td>30</td>
<td>Unsupported locale</td>
<td>Unsupported locale</td>
<td>Unsupported locale</td>
<td>Unsupported locale</td>
<td>Unsupported locale</td>
<td>N/A</td>
</tr>
<tr>
<td>31</td>
<td>ASR error</td>
<td>ASR error</td>
<td>ASR error</td>
<td>ASR error</td>
<td>ASR error</td>
<td>N/A</td>
</tr>
<tr>
<td>32</td>
<td>TTS error</td>
<td>TTS error</td>
<td>TTS error</td>
<td>TTS error</td>
<td>TTS error</td>
<td>N/A</td>
</tr>
<tr>
<td>33</td>
<td>General ASR/TTS error</td>
<td>General ASR/TTS error</td>
<td>General ASR/TTS error</td>
<td>General ASR/TTS error</td>
<td>General ASR/TTS error</td>
<td>N/A</td>
</tr>
<tr>
<td>34</td>
<td>Unknown error</td>
<td>Unknown error</td>
<td>Unknown error</td>
<td>Unknown error</td>
<td>Unknown error</td>
<td>N/A</td>
</tr>
<tr>
<td>40</td>
<td>VXML Server system unavailable</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>VXML Server system unavailable</td>
<td>N/A</td>
</tr>
<tr>
<td>41</td>
<td>VXML Server application error</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>VXML Server application error</td>
<td>N/A</td>
</tr>
<tr>
<td>42</td>
<td>VXML Server application used hangup element instead of subdialog return element</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>VXML Server application used hangup element instead of subdialog return element</td>
<td>N/A</td>
</tr>
<tr>
<td>43</td>
<td>VXML Server application is suspended</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>VXML Server application is suspended</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### Writing Packaged CCE Applications for Unified CVP

Once Packaged CCE-to-Unified CVP initial setup is complete, you can create Packaged CCE applications to access Unified CVP micro-applications.

You do this using two Packaged CCE software tools:

- Unified CCE Administration
- Packaged CCE Script Editor

The sections that follow give a brief overview of how to use these tools to access Unified CVP functionality.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Play Media</th>
<th>Play Data</th>
<th>Get Digits</th>
<th>Menu</th>
<th>Get Speech</th>
<th>Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>VXML Server session error (for example, application has not yet been loaded)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>VXML Server session error (for example, application has not yet been loaded)</td>
<td>N/A</td>
</tr>
<tr>
<td>45</td>
<td>VXML Server encounters a bad fetch error (for example, media or grammar file not found)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>VXML Server encounters a bad fetch error (for example, media or grammar file not found)</td>
<td>N/A</td>
</tr>
<tr>
<td>46</td>
<td>Audio streaming error</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**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.)

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.
Configure a Unified CVP Network VRU Script

Procedure

Step 1  Within Packaged CCE Administration, select Tools > List Tools> Network VRU Scripts.

Step 2  In the Network VRU Scripts 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.

   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).

   • Name. A unique name for the VRU script.

   • VRU Script Name. A 39-character, comma-delimited string used by Unified CVP to pass parameters to the CVP 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

   • Timeout. The number of seconds Packaged CCE waits for a response from the VRU after invoking the script before assuming that the Unified CVP script has failed.

      Note 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.

   • Interruptible. (Checkbox.) Whether Packaged CCE can interrupt the script (for example, if a routing target becomes available).

   • Description. Any additional information about the script.

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 Scripts tool.

Step 2
Select the [ICM Script/VRU Script Name] you want to execute.

Step 3
Modify the Comments tab, if desired.

Step 4
Modify the Labels tab, if desired.

Step 5
When finished, click OK to submit the changes and close the dialog box.

Unified CVP Micro-Applications

The sections that follow describe the parameters that can be defined through Unified CCE Administration 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’s Attributes tab must be separated by a comma.
- If a parameter value is not specified, the micro-application uses its default.

Dynamic Audio File Support for Micro-Applications

Unified CVP lets you use a single micro-application and specify the prompt using call variables and the Packaged CCE 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, and Get Digits micro-applications.

Note: When A is specified as the Media Library, it means CVP looks for the media file under the C:\inetpub\wwwroot\en-us\app folder by default and when S is specified, it looks under the C:\inetpub\wwwroot\en-us\sys folder by default.
For an example of how to use a dynamic audio file, see the following table.

<table>
<thead>
<tr>
<th>VRU Script Parameter Example</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM, -3,A</td>
<td>PM - Uses the Play Media micro-application. -3 - Plays the file specified in Call.PeripheralVariable3. A - Acquires the file from the application media files folder (for example, C:\inetpub\wwwroot\en-us\app).</td>
</tr>
</tbody>
</table>

**Notes**

- If you do not specify a file extension for the file name in the Call.PeripheralVariable, the default media file extension is applied (for example, .wav for audio files).
- 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 CVP Service creates a VoiceXML that does not contain a media prompt.
- 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 (before Unified CVP 8.5), 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 Packaged CCE 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)

The following screen shot shows the Packaged CCE script where Play Media micro-application plays a media file using the ECC variable `user.microapp.media_server`.

**Figure 58: Packaged CCE Script with Play Media using ECC Variable**

The following screen shot shows the Packaged CCE script where Play Media micro-application plays a media file using a default media server configured in the Operations Console.

**Figure 59: Packaged CCE Script with Play Media Using Default Media Server**

**Capture Micro-Application**

The Capture (CAP) micro-application allows you to trigger the storage of current call data at multiple points in the Packaged CCE 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 Packaged CCE 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.

Packaged CCE 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 Packaged CCE. 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 Packaged CCE 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 CVP VRU leg.

The Capture micro-application places a heavy demand on Packaged CCE resources. Each time you use it, Packaged CCE 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 Packaged CCE in terms of processing time and disk space, which despite the minimal indexing, may impact Packaged CCE’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 Packaged CCE Administration’s Network VRU Scripts tool to specify parameters.

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).
  * **null** - (default) If this field 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.
  * **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 Packaged CCE.
• **Media Library Type.** Flag indicating the location of the media files to be played.

  The valid options are:
  
  - **A** - (default) Application
  - **S** - System

• **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.

  For more information about barge-in, see *How Unified CVP Handles Barge-In, on page 183.*

• **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 normally used when two or more PM and/or PD microapps are used in a loop in the CCE 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 CCE script when the user barges in.

---

*How Unified CVP Handles Barge-In*

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 Voice XML

Configure Play Media Micro-Application to Use Streaming Audio

Use the CCE 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.

---

Note

The IOS gateway only supports μ-law wav files in 8-bit format.

You must enclose the stream URL and stream name values in quotation marks.

---

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 CCE 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.
In the example above, the CVP_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 CCE Script Editor to configure CCE 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 CCE Script Name desired and click **OK**.

**Step 5** Optionally, you can use the Packaged CCE Administration's Network VRU Scripts tool 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 CVP 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 CVP Service, and as a result a code 38 rejection is generated in the Gateway logs.

**Related Topics**

Configure Custom Streaming Ringtones, on page 186

**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.
**Play Media Examples: Play Welcome Message**

The following table shows some Network VRU Script configuration examples for Play Media.

**Table 4: Network VRU Script configuration examples**

<table>
<thead>
<tr>
<th>Example</th>
<th>Field Name</th>
<th>Field Contents</th>
<th>Tells Unified CVP:...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VRU Script Name</td>
<td>PM,Welcome</td>
<td>To use the Play Media (PM) micro-application to play the &quot;Welcome.wav&quot; Media file and accept the defaults for remaining settings.</td>
</tr>
<tr>
<td></td>
<td>Configuration Param</td>
<td>N</td>
<td>That Barge-in <em>is not</em> allowed.</td>
</tr>
<tr>
<td>2</td>
<td>VRU Script Name</td>
<td>pm,July,S</td>
<td>To use the Play Media (PM) micro-application to play the &quot;July.wav&quot; Media file, using the System (S) Media library.</td>
</tr>
<tr>
<td></td>
<td>Configuration Param</td>
<td>Null (Accept default.)</td>
<td>That Barge-in <em>is</em> allowed.</td>
</tr>
<tr>
<td>3</td>
<td>VRU Script Name</td>
<td>PM,Website,,0</td>
<td>To use the Play Media (PM) micro-application to play the &quot;Website.wav&quot; Media file, using the default Media Type (Application library), and setting 0 as the Uniqueness value.</td>
</tr>
<tr>
<td></td>
<td>Configuration Param</td>
<td>Null (Accept default.)</td>
<td>That Barge-in <em>is</em> allowed.</td>
</tr>
<tr>
<td>4</td>
<td>VRU Script Name</td>
<td>PM, Website,,1</td>
<td>To use the Play Media (PM) micro-application to play the &quot;Website.wav&quot; Media file, using the default Media Type (Application library), and setting 1 as the Uniqueness value.</td>
</tr>
<tr>
<td></td>
<td>Configuration Param</td>
<td>N</td>
<td>That Barge-in <em>is not</em> allowed.</td>
</tr>
<tr>
<td>5</td>
<td>VRU Script Name</td>
<td>PM, -3, A</td>
<td>To use the Play Media (PM) micro-application, using the file listed in Call.PeripheralVariable3, acquiring the file from the Application (A) media library.</td>
</tr>
<tr>
<td></td>
<td>Configuration Param</td>
<td>N</td>
<td>That Barge-in <em>is not</em> allowed.</td>
</tr>
</tbody>
</table>
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 playback 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 Packaged CCE Peripheral Variables (PeripheralVariable1 through PeripheralVariables10).
- The `user.microapp.play_data` elements.

Configure Network VRU Script Settings for Play Data Micro-Application

Use the CCE Configuration Manager’s Network VRU Script tool’s Attributes tab to specify parameters.

\[ \text{Note} \]

Voice barge-in is not supported by Play Media and Play Data micro-applications. However, DTMF barge-in is supported for these micro-applications.

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. For more information about barge-in, see How Unified CVP Handles Barge-In, on page 183.

- **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
- **HHMSS** - 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 normally used when two or more PM and/or PD microapps are used in a loop in the CCE 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 CCE 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. 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.

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 CVP 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."

Table 5: Data Play Back Types

<table>
<thead>
<tr>
<th>Data Play Back Type</th>
<th>Description</th>
<th>Input Format</th>
<th>Output Examples (When Not Using TTS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Play the stored data as a number.</td>
<td>-#----------.#-</td>
<td>en-us and en-gb typical spoken form:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• -123 = “minus one hundred twenty three”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 35.67 = “thirty five point six seven”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 1234.0 = “one thousand, two hundred, thirty four point zero”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>es-mx and es-es typical spoken form:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• -120 = “menos ciento veinte”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 10.60 = “diez coma seis cero”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 1,100 = “mil cien”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Play the stored data as an number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All printable American National Standards Institute (ANSI) characters are supported.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code Page 1252 is ANSI standard. It contains ASCII (characters 0-127) and extended characters from 128 to 255</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>en-us and en-gb typical spoken form:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• abc123 = “A, B, C, one, two, three”</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>es-mx and es-es typical spoken form:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• abc123 = “A, B, C, uno, dos, tres”</td>
<td></td>
</tr>
</tbody>
</table>

Char                Play the stored data as individual characters.  

Note: Code Page 1252 is ANSI standard. It contains ASCII (characters 0-127) and extended characters from 128 to 255.
<table>
<thead>
<tr>
<th>Data Play Back Type</th>
<th>Description</th>
<th>Input Format</th>
<th>Output Examples (When Not Using TTS)</th>
</tr>
</thead>
</table>
| Date                | Play the stored data as a date. | YYYYMMDD, regardless of locale.  
YYY 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"  
- HHMMSS format: 020101 = "dos horas un minuto un segundo" |
### Data Play Back Type

<table>
<thead>
<tr>
<th>Data Play Back Type</th>
<th>Description</th>
<th>Input Format</th>
<th>Output Examples (When Not Using TTS)</th>
</tr>
</thead>
</table>
| 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.” |
| 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” |

---

**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>
<thead>
<tr>
<th>Symbol (where applicable)</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>point</td>
<td>point</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>minus</td>
<td>minus</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>48</td>
<td>0</td>
<td>zero</td>
<td>All except DOW</td>
</tr>
<tr>
<td></td>
<td>49</td>
<td>1</td>
<td>one (masculine version), uno (es-mx and es-es)</td>
<td>All except DOW</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>2</td>
<td>two</td>
<td>All except DOW</td>
</tr>
<tr>
<td>3</td>
<td>51</td>
<td>3</td>
<td>three</td>
<td>All except DOW</td>
</tr>
<tr>
<td>4</td>
<td>52</td>
<td>4</td>
<td>four</td>
<td>All except DOW</td>
</tr>
<tr>
<td>5</td>
<td>53</td>
<td>5</td>
<td>five</td>
<td>All except DOW</td>
</tr>
<tr>
<td>6</td>
<td>54</td>
<td>6</td>
<td>six</td>
<td>All except DOW</td>
</tr>
<tr>
<td>7</td>
<td>55</td>
<td>7</td>
<td>seven</td>
<td>All except DOW</td>
</tr>
<tr>
<td>8</td>
<td>56</td>
<td>8</td>
<td>eight</td>
<td>All except DOW</td>
</tr>
<tr>
<td>9</td>
<td>57</td>
<td>9</td>
<td>nine</td>
<td>All except DOW</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>ten</td>
<td>Same for the rest of all the numbers</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td></td>
<td>eleven</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td></td>
<td>twelve</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td></td>
<td>thirteen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td></td>
<td>fourteen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td></td>
<td>fifteen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td></td>
<td>sixteen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td></td>
<td>seventeen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18</td>
<td></td>
<td>eighteen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td></td>
<td>nineteen</td>
<td></td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>twenty</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>twenty-one</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>twenty-two</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>twenty-three</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>twenty-four</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>twenty-five</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>twenty-six</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>twenty-seven</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>twenty-eight</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>twenty-nine</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>thirty</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>thirty-one</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>thirty-two</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>thirty-three</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>thirty-four</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>thirty-five</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>thirty-six</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>thirty-seven</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>38</td>
<td>thirty-eight</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>thirty-nine</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>forty</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>forty-one</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>forty-two</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>forty-three</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td></td>
<td>forty-four</td>
<td></td>
</tr>
<tr>
<td></td>
<td>45</td>
<td></td>
<td>forty-five</td>
<td></td>
</tr>
<tr>
<td></td>
<td>46</td>
<td></td>
<td>forty-six</td>
<td></td>
</tr>
<tr>
<td></td>
<td>47</td>
<td></td>
<td>forty-seven</td>
<td></td>
</tr>
<tr>
<td></td>
<td>48</td>
<td></td>
<td>forty-eight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>49</td>
<td></td>
<td>forty-nine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td></td>
<td>fifty</td>
<td></td>
</tr>
<tr>
<td></td>
<td>51</td>
<td></td>
<td>fifty-one</td>
<td></td>
</tr>
<tr>
<td></td>
<td>52</td>
<td></td>
<td>fifty-two</td>
<td></td>
</tr>
<tr>
<td></td>
<td>53</td>
<td></td>
<td>fifty-three</td>
<td></td>
</tr>
<tr>
<td></td>
<td>54</td>
<td></td>
<td>fifty-four</td>
<td></td>
</tr>
<tr>
<td></td>
<td>55</td>
<td></td>
<td>fifty-five</td>
<td></td>
</tr>
<tr>
<td></td>
<td>56</td>
<td></td>
<td>fifty-six</td>
<td></td>
</tr>
<tr>
<td></td>
<td>57</td>
<td></td>
<td>fifty-seven</td>
<td></td>
</tr>
<tr>
<td></td>
<td>58</td>
<td></td>
<td>fifty-eight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>59</td>
<td></td>
<td>fifty-nine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60</td>
<td></td>
<td>sixty</td>
<td></td>
</tr>
<tr>
<td></td>
<td>61</td>
<td></td>
<td>sixty-one</td>
<td></td>
</tr>
<tr>
<td></td>
<td>62</td>
<td></td>
<td>sixty-two</td>
<td></td>
</tr>
<tr>
<td></td>
<td>63</td>
<td></td>
<td>sixty-three</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td></td>
<td>sixty-four</td>
<td></td>
</tr>
<tr>
<td></td>
<td>65</td>
<td></td>
<td>sixty-five</td>
<td></td>
</tr>
<tr>
<td></td>
<td>66</td>
<td></td>
<td>sixty-six</td>
<td></td>
</tr>
<tr>
<td></td>
<td>67</td>
<td></td>
<td>sixty-seven</td>
<td></td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>68</td>
<td></td>
<td>sixty-eight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>69</td>
<td></td>
<td>sixty-nine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>70</td>
<td></td>
<td>seventy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>71</td>
<td></td>
<td>seventy-one</td>
<td></td>
</tr>
<tr>
<td></td>
<td>72</td>
<td></td>
<td>seventy-two</td>
<td></td>
</tr>
<tr>
<td></td>
<td>73</td>
<td></td>
<td>seventy-three</td>
<td></td>
</tr>
<tr>
<td></td>
<td>74</td>
<td></td>
<td>seventy-four</td>
<td></td>
</tr>
<tr>
<td></td>
<td>75</td>
<td></td>
<td>seventy-five</td>
<td></td>
</tr>
<tr>
<td></td>
<td>76</td>
<td></td>
<td>seventy-six</td>
<td></td>
</tr>
<tr>
<td></td>
<td>77</td>
<td></td>
<td>seventy-seven</td>
<td></td>
</tr>
<tr>
<td></td>
<td>78</td>
<td></td>
<td>seventy-eight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>79</td>
<td></td>
<td>seventy-nine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80</td>
<td></td>
<td>eighty</td>
<td></td>
</tr>
<tr>
<td></td>
<td>81</td>
<td></td>
<td>eighty-one</td>
<td></td>
</tr>
<tr>
<td></td>
<td>82</td>
<td></td>
<td>eighty-two</td>
<td></td>
</tr>
<tr>
<td></td>
<td>83</td>
<td></td>
<td>eighty-three</td>
<td></td>
</tr>
<tr>
<td></td>
<td>84</td>
<td></td>
<td>eighty-four</td>
<td></td>
</tr>
<tr>
<td></td>
<td>85</td>
<td></td>
<td>eighty-five</td>
<td></td>
</tr>
<tr>
<td></td>
<td>86</td>
<td></td>
<td>eighty-six</td>
<td></td>
</tr>
<tr>
<td></td>
<td>87</td>
<td></td>
<td>eighty-seven</td>
<td></td>
</tr>
<tr>
<td></td>
<td>88</td>
<td></td>
<td>eighty-eight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>89</td>
<td></td>
<td>eighty-nine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>90</td>
<td></td>
<td>ninety</td>
<td></td>
</tr>
<tr>
<td></td>
<td>91</td>
<td></td>
<td>ninety-one</td>
<td></td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>92</td>
<td></td>
<td>ninety-two</td>
<td>24TOD, Date</td>
</tr>
<tr>
<td></td>
<td>93</td>
<td></td>
<td>ninety-three</td>
<td>Number, 24TOD, Date, Currency</td>
</tr>
<tr>
<td></td>
<td>94</td>
<td></td>
<td>ninety-four</td>
<td></td>
</tr>
<tr>
<td></td>
<td>95</td>
<td></td>
<td>ninety-five</td>
<td></td>
</tr>
<tr>
<td></td>
<td>96</td>
<td></td>
<td>ninety-six</td>
<td></td>
</tr>
<tr>
<td></td>
<td>97</td>
<td></td>
<td>ninety-seven</td>
<td></td>
</tr>
<tr>
<td></td>
<td>98</td>
<td></td>
<td>ninety-eight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>99</td>
<td></td>
<td>ninety-nine</td>
<td></td>
</tr>
<tr>
<td>oh</td>
<td></td>
<td>oh</td>
<td></td>
<td>24TOD, Date</td>
</tr>
<tr>
<td>hundred</td>
<td></td>
<td>hundred</td>
<td></td>
<td>Number, 24TOD, Date, Currency</td>
</tr>
<tr>
<td>thousand</td>
<td></td>
<td>thousand</td>
<td></td>
<td>Number, Date, Currency</td>
</tr>
<tr>
<td>million</td>
<td></td>
<td>million</td>
<td></td>
<td>Number, Currency</td>
</tr>
<tr>
<td>billion</td>
<td></td>
<td>billion</td>
<td></td>
<td>Number, Date, Currency</td>
</tr>
<tr>
<td>trillion</td>
<td></td>
<td>trillion</td>
<td></td>
<td>Number, Currency</td>
</tr>
</tbody>
</table>

The table that follows lists the System Media File information for ordinal numbers.

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 7: System Media Files, Ordinal Numbers

<table>
<thead>
<tr>
<th>Symbol (where applicable)</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ord</td>
<td></td>
<td>first</td>
<td></td>
<td>Date</td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>2ord</td>
<td>second</td>
<td></td>
<td>Date for all ordinal numbers</td>
</tr>
<tr>
<td></td>
<td>3ord</td>
<td>third</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4ord</td>
<td>fourth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5ord</td>
<td>fifth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6ord</td>
<td>sixth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7ord</td>
<td>seventh</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8ord</td>
<td>eighth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9ord</td>
<td>nineth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10ord</td>
<td>tenth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11ord</td>
<td>eleventh</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12ord</td>
<td>twelveth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13ord</td>
<td>thirteenth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14ord</td>
<td>fourteenth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15ord</td>
<td>fifteenth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16ord</td>
<td>sixteenth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>17ord</td>
<td>seventeenth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>18ord</td>
<td>eighteenth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>19ord</td>
<td>nineteenth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20ord</td>
<td>twentieth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>21ord</td>
<td>twenty-first</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>22ord</td>
<td>twenty-second</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23ord</td>
<td>twenty-third</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>24ord</td>
<td>twenty-fourth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>25ord</td>
<td>twenty-fifth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26ord</td>
<td>twenty-sixth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27ord</td>
<td>twenty-seventh</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28ord</td>
<td>twenty-eighth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29ord</td>
<td>twenty-nineth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30ord</td>
<td>thirtieth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31ord</td>
<td>thirty-first</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The table that follows lists the System Media File information for measurements.

*Table 8: System Media Files, Measurements*

<table>
<thead>
<tr>
<th>Symbol (where applicable)</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>½</td>
<td>189</td>
<td>one_half</td>
<td>one half</td>
<td>Char</td>
</tr>
<tr>
<td>¼</td>
<td>188</td>
<td>one_quarter</td>
<td>one quarter</td>
<td>Char</td>
</tr>
<tr>
<td>¾</td>
<td>190</td>
<td>three_quarters</td>
<td>three quarters</td>
<td>Char</td>
</tr>
<tr>
<td>A , a</td>
<td>65,97</td>
<td>a</td>
<td>A</td>
<td>Char</td>
</tr>
<tr>
<td>B , b</td>
<td>66,98</td>
<td>b</td>
<td>B</td>
<td>Char</td>
</tr>
<tr>
<td>C , c</td>
<td>67,99</td>
<td>c</td>
<td>C</td>
<td>Char</td>
</tr>
<tr>
<td>D , d</td>
<td>68,100</td>
<td>d</td>
<td>D</td>
<td>Char</td>
</tr>
<tr>
<td>E , e</td>
<td>69,101</td>
<td>e</td>
<td>E</td>
<td>Char</td>
</tr>
<tr>
<td>F , f</td>
<td>70,102</td>
<td>f</td>
<td>F</td>
<td>Char</td>
</tr>
<tr>
<td>G , g</td>
<td>71,103</td>
<td>g</td>
<td>G</td>
<td>Char</td>
</tr>
<tr>
<td>H , h</td>
<td>72,104</td>
<td>h</td>
<td>H</td>
<td>Char</td>
</tr>
<tr>
<td>I , I</td>
<td>73,105</td>
<td>I</td>
<td>I</td>
<td>Char</td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>J, j</td>
<td>74,106</td>
<td>j</td>
<td>J</td>
<td>Char</td>
</tr>
<tr>
<td>K, k</td>
<td>75,107</td>
<td>k</td>
<td>K</td>
<td>Char</td>
</tr>
<tr>
<td>L, l</td>
<td>76,108</td>
<td>l</td>
<td>L</td>
<td>Char</td>
</tr>
<tr>
<td>M, n</td>
<td>77,109</td>
<td>m</td>
<td>M</td>
<td>Char</td>
</tr>
<tr>
<td>N, o</td>
<td>78,110</td>
<td>n</td>
<td>N</td>
<td>Char</td>
</tr>
<tr>
<td>P, p</td>
<td>80,112</td>
<td>p</td>
<td>P</td>
<td>Char</td>
</tr>
<tr>
<td>Q, q</td>
<td>81,113</td>
<td>q</td>
<td>Q</td>
<td>Char</td>
</tr>
<tr>
<td>R, r</td>
<td>82,114</td>
<td>r</td>
<td>R</td>
<td>Char</td>
</tr>
<tr>
<td>S, s</td>
<td>83,115</td>
<td>s</td>
<td>S</td>
<td>Char</td>
</tr>
<tr>
<td>T, t</td>
<td>84,116</td>
<td>t</td>
<td>T</td>
<td>Char</td>
</tr>
<tr>
<td>U, u</td>
<td>85,117</td>
<td>u</td>
<td>U</td>
<td>Char</td>
</tr>
<tr>
<td>V, v</td>
<td>86,118</td>
<td>v</td>
<td>V</td>
<td>Char</td>
</tr>
<tr>
<td>W, w</td>
<td>87,119</td>
<td>w</td>
<td>W</td>
<td>Char</td>
</tr>
<tr>
<td>X, x</td>
<td>88,120</td>
<td>x</td>
<td>X</td>
<td>Char</td>
</tr>
<tr>
<td>Y, y</td>
<td>89,121</td>
<td>y</td>
<td>Y</td>
<td>Char</td>
</tr>
<tr>
<td>Z, z</td>
<td>90,122</td>
<td>z</td>
<td>Z</td>
<td>Char</td>
</tr>
<tr>
<td>OE, œ</td>
<td>140,156</td>
<td>oe_140_156</td>
<td>Ligature OE</td>
<td>Char</td>
</tr>
<tr>
<td>À, à</td>
<td>192,224</td>
<td>a_192_224</td>
<td>A grave</td>
<td>Char</td>
</tr>
<tr>
<td>Á, â</td>
<td>193,225</td>
<td>a_193_225</td>
<td>A acute</td>
<td>Char</td>
</tr>
<tr>
<td>Å, å</td>
<td>194,226</td>
<td>a_194_226</td>
<td>A circumflex</td>
<td>Char</td>
</tr>
<tr>
<td>À, à</td>
<td>195,227</td>
<td>a_195_227</td>
<td>A tilde</td>
<td>Char</td>
</tr>
<tr>
<td>Ä, ä</td>
<td>196,228</td>
<td>a_196_228</td>
<td>A umlaut</td>
<td>Char</td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Å, å</td>
<td>197,229</td>
<td>a_197_229</td>
<td>A with ring above</td>
<td>Char</td>
</tr>
<tr>
<td>Æ, æ</td>
<td>198,230</td>
<td>ae_198_230</td>
<td>Ligature AE</td>
<td>Char</td>
</tr>
<tr>
<td>È, è</td>
<td>200,232</td>
<td>e_200_232</td>
<td>E grave</td>
<td>Char</td>
</tr>
<tr>
<td>É, é</td>
<td>201,233</td>
<td>e_201_233</td>
<td>E acute</td>
<td>Char</td>
</tr>
<tr>
<td>Ë, ë</td>
<td>202,234</td>
<td>e_202_234</td>
<td>E circumflex</td>
<td>Char</td>
</tr>
<tr>
<td>Ì, ì</td>
<td>204,236</td>
<td>i_204_236</td>
<td>I grave</td>
<td>Char</td>
</tr>
<tr>
<td>Ì, ì</td>
<td>205,237</td>
<td>i_205</td>
<td>I acute</td>
<td>Char</td>
</tr>
<tr>
<td>Ì, ì</td>
<td>206,238</td>
<td>i_206</td>
<td>I circumflex</td>
<td>Char</td>
</tr>
<tr>
<td>Ì, ì</td>
<td>207,239</td>
<td>i_207</td>
<td>I umlaut</td>
<td>Char</td>
</tr>
<tr>
<td>Ð</td>
<td>208</td>
<td>char_208</td>
<td>character 208</td>
<td>Char</td>
</tr>
<tr>
<td>ð</td>
<td>240</td>
<td>char_240</td>
<td>character 240</td>
<td></td>
</tr>
<tr>
<td>Ô, ò</td>
<td>210,242</td>
<td>o_210_242</td>
<td>O grave</td>
<td>Char</td>
</tr>
<tr>
<td>Ô, ó</td>
<td>211,243</td>
<td>o_211_243</td>
<td>O acute</td>
<td>Char</td>
</tr>
<tr>
<td>Ô, ò</td>
<td>212,244</td>
<td>o_212_244</td>
<td>O circumflex</td>
<td>Char</td>
</tr>
<tr>
<td>Ô, ô</td>
<td>213,245</td>
<td>o_213_245</td>
<td>O tilde</td>
<td>Char</td>
</tr>
<tr>
<td>Ö, ö</td>
<td>214,246</td>
<td>o_214_246</td>
<td>O umlaut</td>
<td>Char</td>
</tr>
<tr>
<td>x</td>
<td>215</td>
<td>multiply</td>
<td>multiplication sign</td>
<td>Char</td>
</tr>
<tr>
<td>Ø, ø</td>
<td>216,248</td>
<td>o_216_248</td>
<td>oh stroke</td>
<td>Char</td>
</tr>
<tr>
<td>Ù, ü</td>
<td>217,249</td>
<td>u_217_249</td>
<td>U grave</td>
<td>Char</td>
</tr>
<tr>
<td>Ù, ü</td>
<td>218,250</td>
<td>u_218_250</td>
<td>U acute</td>
<td>Char</td>
</tr>
<tr>
<td>Ù, ù</td>
<td>219,251</td>
<td>u_219_251</td>
<td>U circumflex</td>
<td>Char</td>
</tr>
<tr>
<td>Ù, ü</td>
<td>220,252</td>
<td>u_220_252</td>
<td>U umlaut</td>
<td>Char</td>
</tr>
<tr>
<td>Ý, ý</td>
<td>221,253</td>
<td>y_221_253</td>
<td>Y acute</td>
<td>Char</td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>--------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>ß</td>
<td>223</td>
<td>ss</td>
<td>double s</td>
<td>Char</td>
</tr>
<tr>
<td>÷</td>
<td>247</td>
<td>÷</td>
<td>division sign</td>
<td>Char</td>
</tr>
<tr>
<td>ß, ý</td>
<td>254</td>
<td>y_159_255</td>
<td>character 159 or 255</td>
<td>Char</td>
</tr>
</tbody>
</table>

The table that follows lists the System Media File information for month values.

**Table 9: System Media Files, Months**

<table>
<thead>
<tr>
<th>Symbol (where applicable)</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td></td>
<td>January</td>
<td>January</td>
<td>Date</td>
</tr>
<tr>
<td>February</td>
<td></td>
<td>February</td>
<td>February</td>
<td>Date</td>
</tr>
<tr>
<td>March</td>
<td></td>
<td>March</td>
<td>March</td>
<td>Date</td>
</tr>
<tr>
<td>April</td>
<td></td>
<td>April</td>
<td>April</td>
<td>Date</td>
</tr>
<tr>
<td>May</td>
<td></td>
<td>May</td>
<td>May</td>
<td>Date</td>
</tr>
<tr>
<td>June</td>
<td></td>
<td>June</td>
<td>June</td>
<td>Date</td>
</tr>
<tr>
<td>July</td>
<td></td>
<td>July</td>
<td>July</td>
<td>Date</td>
</tr>
<tr>
<td>August</td>
<td></td>
<td>August</td>
<td>August</td>
<td>Date</td>
</tr>
<tr>
<td>September</td>
<td></td>
<td>September</td>
<td>September</td>
<td>Date</td>
</tr>
<tr>
<td>October</td>
<td></td>
<td>October</td>
<td>October</td>
<td>Date</td>
</tr>
<tr>
<td>November</td>
<td></td>
<td>November</td>
<td>November</td>
<td>Date</td>
</tr>
<tr>
<td>December</td>
<td></td>
<td>December</td>
<td>December</td>
<td>Date</td>
</tr>
</tbody>
</table>

The table that follows lists the System Media File information for month values.
Table 10: System Media Files, Days

<table>
<thead>
<tr>
<th>Symbol (where applicable)</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sunday</td>
<td>Sunday</td>
<td></td>
<td></td>
<td>DOW</td>
</tr>
<tr>
<td>Monday</td>
<td>Monday</td>
<td></td>
<td></td>
<td>DOW</td>
</tr>
<tr>
<td>Tuesday</td>
<td>Tuesday</td>
<td></td>
<td></td>
<td>DOW</td>
</tr>
<tr>
<td>Wednesday</td>
<td>Wednesday</td>
<td></td>
<td></td>
<td>DOW</td>
</tr>
<tr>
<td>Thursday</td>
<td>Thursday</td>
<td></td>
<td></td>
<td>DOW</td>
</tr>
<tr>
<td>Friday</td>
<td>Friday</td>
<td></td>
<td></td>
<td>DOW</td>
</tr>
<tr>
<td>Saturday</td>
<td>Saturday</td>
<td></td>
<td></td>
<td>DOW</td>
</tr>
</tbody>
</table>

The table that follows lists the System Media File information for month values.

Table 11: System Media Files, Time

<table>
<thead>
<tr>
<th>Symbol (where applicable)</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hour</td>
<td>hour</td>
<td></td>
<td></td>
<td>Etime, 24TOD per locale, TOD per locale</td>
</tr>
<tr>
<td>hours</td>
<td>hours</td>
<td></td>
<td></td>
<td>Etime, 24TOD per locale, TOD per locale</td>
</tr>
<tr>
<td>minute</td>
<td>minute</td>
<td></td>
<td></td>
<td>Etimate</td>
</tr>
<tr>
<td>minutes</td>
<td>minutes</td>
<td></td>
<td></td>
<td>Etimate</td>
</tr>
<tr>
<td>second</td>
<td>second</td>
<td></td>
<td></td>
<td>Etimate, 24TOD</td>
</tr>
<tr>
<td>seconds</td>
<td>seconds</td>
<td></td>
<td></td>
<td>Etimate, 24TOD</td>
</tr>
<tr>
<td>on</td>
<td>on</td>
<td></td>
<td></td>
<td>per locale(unused for en-us)</td>
</tr>
<tr>
<td>at</td>
<td>at</td>
<td></td>
<td></td>
<td>per locale(unused for en-us)</td>
</tr>
</tbody>
</table>
The table that follows lists the System Media File information for currency values.

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 12: System Media Files, Currency

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>currency_minus</td>
<td></td>
<td>minus</td>
<td></td>
<td>Currency</td>
</tr>
<tr>
<td>currency_and</td>
<td></td>
<td>and</td>
<td></td>
<td>Currency</td>
</tr>
<tr>
<td>$</td>
<td>36</td>
<td>USD_dollar</td>
<td>dollar</td>
<td>Currency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USD_dollars</td>
<td>dollars</td>
<td>Currency</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>£</td>
<td>163</td>
<td>GBP_pound</td>
<td>pound</td>
<td>Currency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>euro_dollar</td>
<td>euro</td>
<td>Currency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cent</td>
<td>cents</td>
<td>Currency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pound</td>
<td>pounds</td>
<td>Currency</td>
</tr>
</tbody>
</table>
The table that follows lists the System Media File information for gaps of silence and miscellaneous phrases.

**Table 13: System Media Files, Silence and Miscellaneous Phrases**

<table>
<thead>
<tr>
<th>Symbol (where applicable)</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>GBP_pounds</td>
<td>pounds</td>
<td>Currency</td>
</tr>
<tr>
<td>penny</td>
<td></td>
<td>penny</td>
<td>pounds</td>
<td>Currency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pence</td>
<td>pence</td>
<td>Currency</td>
</tr>
<tr>
<td>MXN_peso</td>
<td></td>
<td>peso</td>
<td>pesos</td>
<td>Currency</td>
</tr>
<tr>
<td>MXN_pesos</td>
<td></td>
<td>pesos</td>
<td>pesos</td>
<td>Currency</td>
</tr>
<tr>
<td>centavo</td>
<td></td>
<td>centavo</td>
<td>centavos</td>
<td>Currency</td>
</tr>
<tr>
<td>centavos</td>
<td></td>
<td>centavos</td>
<td>centavos</td>
<td>Currency</td>
</tr>
</tbody>
</table>

The table that follows lists the System Media File information for ANSI characters.

**Table 14: System Media Files, ANSI Characters**

<table>
<thead>
<tr>
<th>Symbol (where applicable)</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>32</td>
<td>space</td>
<td>space</td>
<td>Etime,TOD,25TOD</td>
</tr>
</tbody>
</table>

Scripting with Packaged CCE
<table>
<thead>
<tr>
<th>Symbol (where applicable)</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>33</td>
<td>exclamation_mark</td>
<td>exclamation mark</td>
<td>Char</td>
</tr>
<tr>
<td>&quot;</td>
<td>34</td>
<td>double_quote</td>
<td>double quote</td>
<td>Char</td>
</tr>
<tr>
<td>#</td>
<td>35</td>
<td>pound</td>
<td>pound</td>
<td>Char</td>
</tr>
<tr>
<td>%</td>
<td>37</td>
<td>percent</td>
<td>percent</td>
<td>Char</td>
</tr>
<tr>
<td>&amp;</td>
<td>38</td>
<td>ampersand</td>
<td>ampersand</td>
<td>Char</td>
</tr>
<tr>
<td>’</td>
<td>39</td>
<td>apostrophe</td>
<td>apostrophe</td>
<td>Char</td>
</tr>
<tr>
<td>(</td>
<td>40</td>
<td>open_parenthesis</td>
<td>open parenthesis</td>
<td>Char</td>
</tr>
<tr>
<td>)</td>
<td>41</td>
<td>close_parenthesis</td>
<td>close parenthesis</td>
<td>Char</td>
</tr>
<tr>
<td>*</td>
<td>42</td>
<td>asterisk</td>
<td>asterisk</td>
<td>Char</td>
</tr>
<tr>
<td>+</td>
<td>43</td>
<td>plus</td>
<td>plus</td>
<td>Char</td>
</tr>
<tr>
<td>,</td>
<td>44</td>
<td>comma</td>
<td>comma</td>
<td>Char</td>
</tr>
<tr>
<td>-</td>
<td>45</td>
<td>hyphen</td>
<td>hyphen</td>
<td>Char</td>
</tr>
<tr>
<td>:</td>
<td>46</td>
<td>period</td>
<td>period</td>
<td>Char</td>
</tr>
<tr>
<td>/</td>
<td>47</td>
<td>slash</td>
<td>slash</td>
<td>Char</td>
</tr>
<tr>
<td>;</td>
<td>58</td>
<td>colon</td>
<td>colon</td>
<td>Char</td>
</tr>
<tr>
<td>&lt;</td>
<td>60</td>
<td>less_than</td>
<td>less than</td>
<td>Char</td>
</tr>
<tr>
<td>=</td>
<td>61</td>
<td>equal</td>
<td>equal</td>
<td>Char</td>
</tr>
<tr>
<td>@</td>
<td>62</td>
<td>greater_than</td>
<td>greater than</td>
<td>Char</td>
</tr>
<tr>
<td>?</td>
<td>63</td>
<td>question_mark</td>
<td>question mark</td>
<td>Char</td>
</tr>
<tr>
<td>[</td>
<td>91</td>
<td>left_square_bracket</td>
<td>left square bracket</td>
<td>Char</td>
</tr>
<tr>
<td>\</td>
<td>92</td>
<td>backslash</td>
<td>backslash</td>
<td>Char</td>
</tr>
<tr>
<td>]</td>
<td>93</td>
<td>right_square_bracket</td>
<td>right square bracket</td>
<td>Char</td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>--------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>^</td>
<td>94</td>
<td>caret</td>
<td>caret</td>
<td>Char</td>
</tr>
<tr>
<td>_</td>
<td>95</td>
<td>underscore</td>
<td>underscore</td>
<td>Char</td>
</tr>
<tr>
<td>`</td>
<td>96</td>
<td>single_quote</td>
<td>single quote</td>
<td>Char</td>
</tr>
<tr>
<td>{</td>
<td>123</td>
<td>open_brace</td>
<td>open brace</td>
<td>Char</td>
</tr>
<tr>
<td></td>
<td></td>
<td>124</td>
<td>pipe</td>
<td>pipe</td>
</tr>
<tr>
<td>}</td>
<td>125</td>
<td>close_brace</td>
<td>close brace</td>
<td>Char</td>
</tr>
<tr>
<td>~</td>
<td>126</td>
<td>tilde</td>
<td>tilde</td>
<td>Char</td>
</tr>
<tr>
<td>′</td>
<td>130</td>
<td>char_130</td>
<td>low single quote</td>
<td>Char</td>
</tr>
<tr>
<td>‼</td>
<td>131</td>
<td>char_131</td>
<td>F with hook</td>
<td>Char</td>
</tr>
<tr>
<td>′</td>
<td>132</td>
<td>low double quote</td>
<td>low double quote</td>
<td>Char</td>
</tr>
<tr>
<td>′</td>
<td>133</td>
<td>ellipsis</td>
<td>ellipsis</td>
<td>Char</td>
</tr>
<tr>
<td>†</td>
<td>134</td>
<td>char_134</td>
<td>character 134</td>
<td>Char</td>
</tr>
<tr>
<td>‡</td>
<td>135</td>
<td>char_135</td>
<td>character 135</td>
<td>Char</td>
</tr>
<tr>
<td>‡</td>
<td>136</td>
<td>char_136</td>
<td>character 136</td>
<td>Char</td>
</tr>
<tr>
<td>‰</td>
<td>137</td>
<td>per_mille</td>
<td>per mile</td>
<td>Char</td>
</tr>
<tr>
<td>Š</td>
<td>138</td>
<td>char_138</td>
<td>character 138</td>
<td></td>
</tr>
<tr>
<td>&lt;</td>
<td>139</td>
<td>left_pointing_angle</td>
<td>left pointing angle</td>
<td>Char</td>
</tr>
<tr>
<td>′</td>
<td>145</td>
<td>left_single_quote</td>
<td>left single quote</td>
<td>Char</td>
</tr>
<tr>
<td>′</td>
<td>146</td>
<td>right_single_quote</td>
<td>right single quote</td>
<td>Char</td>
</tr>
<tr>
<td>′</td>
<td>147</td>
<td>left_double_quote</td>
<td>left double quote</td>
<td>Char</td>
</tr>
<tr>
<td>″</td>
<td>148</td>
<td>right_double_quote</td>
<td>right double quote</td>
<td>Char</td>
</tr>
<tr>
<td>–</td>
<td>149</td>
<td>bullet</td>
<td>bullet</td>
<td>Char</td>
</tr>
<tr>
<td>–</td>
<td>150</td>
<td>en_dash</td>
<td>en dash</td>
<td>Char</td>
</tr>
<tr>
<td>Symbol (where applicable)</td>
<td>Decimal Value</td>
<td>Media File Name</td>
<td>Media File Content</td>
<td>Data Play Back Types / When Media File Is Used</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>—</td>
<td>151</td>
<td>em_dash</td>
<td>em dash</td>
<td></td>
</tr>
<tr>
<td>‾</td>
<td>152</td>
<td>small_tilde</td>
<td>small tilde</td>
<td>Char</td>
</tr>
<tr>
<td>™</td>
<td>153</td>
<td>trade_mark</td>
<td>trade mark</td>
<td>Char</td>
</tr>
<tr>
<td>§</td>
<td>154</td>
<td>char_154</td>
<td>character 154</td>
<td>Char</td>
</tr>
<tr>
<td>›</td>
<td>155</td>
<td>char_155</td>
<td>character 155</td>
<td>Char</td>
</tr>
<tr>
<td>i</td>
<td>161</td>
<td>exclamation_mark_inverted</td>
<td>inverted exclamation mark</td>
<td>Char</td>
</tr>
<tr>
<td>□</td>
<td>164</td>
<td>char_164</td>
<td>character 164</td>
<td>Char</td>
</tr>
<tr>
<td>ŏ</td>
<td>166</td>
<td>broken_pipe</td>
<td>broken pipe</td>
<td>Char</td>
</tr>
<tr>
<td>$</td>
<td>167</td>
<td>section</td>
<td>section</td>
<td>Char</td>
</tr>
<tr>
<td>±</td>
<td>168</td>
<td>char_168</td>
<td>character 168</td>
<td>Char</td>
</tr>
<tr>
<td>‹</td>
<td>169</td>
<td>copyright</td>
<td>copyright</td>
<td>Char</td>
</tr>
<tr>
<td>a</td>
<td>170</td>
<td>char_170</td>
<td>character 170</td>
<td>Char</td>
</tr>
<tr>
<td>«</td>
<td>171</td>
<td>left_double_angle_quote</td>
<td>left double angle quote</td>
<td>Char</td>
</tr>
<tr>
<td>¬</td>
<td>172</td>
<td>not</td>
<td>not</td>
<td>Char</td>
</tr>
<tr>
<td>′</td>
<td>173</td>
<td>char_173</td>
<td>character 173</td>
<td>Char</td>
</tr>
<tr>
<td>®</td>
<td>174</td>
<td>registered</td>
<td>registered</td>
<td>Char</td>
</tr>
<tr>
<td>‰</td>
<td>175</td>
<td>char_175</td>
<td>character 175</td>
<td>Char</td>
</tr>
<tr>
<td>°</td>
<td>176</td>
<td>degree</td>
<td>degree</td>
<td>Char</td>
</tr>
<tr>
<td>±</td>
<td>177</td>
<td>plus_minus</td>
<td>plus or minus</td>
<td>Char</td>
</tr>
<tr>
<td>²</td>
<td>178</td>
<td>superscript_2</td>
<td>superscript two</td>
<td>Char</td>
</tr>
<tr>
<td>³</td>
<td>179</td>
<td>superscript_3</td>
<td>superscript three</td>
<td>Char</td>
</tr>
<tr>
<td>′</td>
<td>180</td>
<td>acute_accent</td>
<td>acute accent</td>
<td>Char</td>
</tr>
<tr>
<td>µ</td>
<td>181</td>
<td>micro</td>
<td>micro</td>
<td>Char</td>
</tr>
</tbody>
</table>
### Play Data Configuration Examples

The following table shows several configuration examples for Play Data.

**Table 15: Play Data configuration examples**

<table>
<thead>
<tr>
<th>Symbol (where applicable)</th>
<th>Decimal Value</th>
<th>Media File Name</th>
<th>Media File Content</th>
<th>Data Play Back Types / When Media File Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>¶</td>
<td>182</td>
<td>paragraph</td>
<td>paragraph</td>
<td>Char</td>
</tr>
<tr>
<td>·</td>
<td>183</td>
<td>middle_dot</td>
<td>middle dot</td>
<td>Char</td>
</tr>
<tr>
<td>,</td>
<td>184</td>
<td>cedilla</td>
<td>cedilla</td>
<td>Char</td>
</tr>
<tr>
<td>′</td>
<td>185</td>
<td>superscript_1</td>
<td>superscript one</td>
<td>Char</td>
</tr>
<tr>
<td>º</td>
<td>186</td>
<td>char_186</td>
<td>character 186</td>
<td>Char</td>
</tr>
<tr>
<td>»</td>
<td>187</td>
<td>right_double_angle_quote</td>
<td>right double angle quote</td>
<td>Char</td>
</tr>
<tr>
<td>¿</td>
<td>191</td>
<td>question_mark_inverted</td>
<td>inverted question mark</td>
<td>Char</td>
</tr>
</tbody>
</table>

**Play Data Configuration Examples**

The following table shows several configuration examples for Play Data.

<table>
<thead>
<tr>
<th>If the VRU Script Name field setting is…</th>
<th>It means…</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD,Number</td>
<td><strong>PD</strong> - Use the Play Data micro-app.  <strong>Number</strong> - Play back the data as a number.</td>
</tr>
<tr>
<td><strong>Note</strong> 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.</td>
<td>It means…</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>If the Configuration Param field is…</th>
<th>It means…</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty</td>
<td>Play the data in the default ECC, <code>user.microapp.play_data</code>, as a number.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PD, Char</th>
<th>It means…</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pd</strong> - Use the Play Data micro-app.</td>
<td><strong>Char</strong> - Play back the data as individual characters.</td>
</tr>
<tr>
<td>1</td>
<td>1 - Play the data in Call Peripheral Variable 1 as a character.</td>
</tr>
</tbody>
</table>
### If the VRU Script Name field setting is...

<table>
<thead>
<tr>
<th>Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD, Etime, 0</td>
<td><strong>Note</strong>: 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.</td>
</tr>
<tr>
<td><strong>PD</strong></td>
<td>Use the Play Data micro-app.</td>
</tr>
<tr>
<td><strong>Etime</strong></td>
<td>Play back the data as a Time.</td>
</tr>
<tr>
<td><strong>HHMM</strong></td>
<td></td>
</tr>
</tbody>
</table>

### If the Configuration Param field is...

<table>
<thead>
<tr>
<th>Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD, Date</td>
<td><strong>PD</strong> - Use the Play Data micro-app.</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>Play back the data as a Date.</td>
</tr>
<tr>
<td><strong>N</strong></td>
<td>No barge-in allowed.</td>
</tr>
</tbody>
</table>

### PD, Currency

<table>
<thead>
<tr>
<th>Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD</td>
<td><strong>PD</strong> - Use the Play Data micro-app.</td>
</tr>
<tr>
<td><strong>Currency</strong></td>
<td>Play back the data as a Currency.</td>
</tr>
<tr>
<td><strong>N</strong></td>
<td>No barge-in allowed.</td>
</tr>
</tbody>
</table>

**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.

## 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 Packaged CCE for further processing using the Caller-Entered Digits (CED) field in the CCE/CVP Messaging interface. (This is available in the Packaged CCE script through the variable Call.CallerEnteredDigits).

### Configure Network VRU Script Settings for Get Digits Micro-Application

Use the Packaged CCE Administration’s Network VRU Script tool 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 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, 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

• **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

  For more information about barge-in, see [How Unified CVP Handles Barge-In](#), on page 183.

  **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 Voice XML](#).

  • **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**).

  • **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

• **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, 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**.

• **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** If the value is set to 0, the CVP 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 16: Get Digits configuration examples for .wav files

<table>
<thead>
<tr>
<th>If the VRU Script Name field setting is…</th>
<th>It means…</th>
<th>If the Configuration Param field setting is…</th>
<th>It means…</th>
</tr>
</thead>
<tbody>
<tr>
<td>GD,Password,A,0</td>
<td>GD - Use the Get Digits micro-app. Password - Play the Media file named “Password.wav.” A - Application Media Library. 0 - Uniqueness value.</td>
<td>6,12</td>
<td>6 - Minimum field length 12 - Maximum field length Accept defaults for all other settings.</td>
</tr>
<tr>
<td>GD,Password,A,1</td>
<td>gd - Use the Get Digits micro-app. Password - Play the Media file named “Password.wav.” A - Application Media Library. 1 - Uniqueness value.</td>
<td>6,12,N,3,5,2,2,N,Y,#</td>
<td>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</td>
</tr>
<tr>
<td>GD,ssn</td>
<td>GD - Use the Get Digits micro-app. ssn - Play the Media file named “ssn.wav.”</td>
<td>9,9,</td>
<td>6 - Minimum field length 12 - Maximum field length Accept defaults for all other settings.</td>
</tr>
<tr>
<td>GD, -4, S</td>
<td>gd - Use the Get Digits micro-app -4 - Calls the file specified in Call.PeripheralVariable4 S - Acquires the file from the System media library</td>
<td>6,12,</td>
<td>6 - Minimum field length 12 - Maximum field length Accept defaults for all other settings.</td>
</tr>
</tbody>
</table>

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.

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 Voice XML.

Get Speech and External Voice XML

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 CVP 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 CVP Service "NoEntry" and "InvalidEntry" retry logic are used if the external VoiceXML returns a <noinput> or <nomatch> event.

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 Packaged CCE.

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 Packaged CCE 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 \( \geq \) (greater than or equal to) the minimum length and \( \leq \) (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 CVP Service, SIP Service discards the termination key (only the password digits are included in the CED returned to Packaged CCE).
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 Packaged CCE 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 Packaged CCE.

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 Packaged CCE.

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 Packaged CCE for further processing using the Caller-Entered Digits (CED) field in the Packaged CCE / CVP Messaging interface.

Configure Network VRU Script Settings for the Menu Micro-Application

Use the Packaged CCE Administration Network VRU Script tool 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 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

**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 delimit ed by a `/` (forward slash)
  - Ranges delimit ed 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

  For more information about barge-in, see How Unified CVP Handles Barge-In, on page 183.

- **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 application where input type is DTMF.

Table 17: Menu Configuration Example - DTMF Application

<table>
<thead>
<tr>
<th>If the VRU Script Name field setting is...</th>
<th>It means...</th>
<th>If the Config Param setting is...</th>
<th>It means...</th>
</tr>
</thead>
<tbody>
<tr>
<td>M,Banking</td>
<td>M - Use the Menu micro-app.</td>
<td>1-3</td>
<td>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).</td>
</tr>
<tr>
<td><strong>Note</strong> This file may contain a message such as: &quot;For Checking, press 1. For Savings, press 2. For Money Market, press 3.&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Banking</td>
<td>Banking - Play the Media file named &quot;Banking.wav.&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If the VRU Script Name field setting is…</td>
<td>It means…</td>
<td>If the Config Param setting is…</td>
<td>It means…</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>------------</td>
<td>--------------------------------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| 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.
It is important that you set up your Packaged CCE 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 CCE 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 Packaged CCE.
- 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 Packaged CCE.

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 Packaged CCE.

Get Speech Micro-Application

The Get Speech (GS) micro-application is used to execute a Call Studio script on VXML Server.

Configure Network VRU Script Settings for the Get Speech Micro-Application

Use the Packaged CCE Administration’s Network VRU Script tool to specify parameters.

Procedure

Step 1 Configure VRU Script field parameters:

Note: By default a pre-configured network VRU script called VXML_Server has already been configured in Packaged CCE. This should be used in all Run External Script nodes that intend to execute a Call Studio script. When using an optional feature like Courtesy Callback, you must configure additional GS network VRU scripts.
• **Micro-application type.** For Get Speech, valid options are: **GS** or **gs**.

• **Media File Name.** Only the value **Server** is supported for this field for GS.

• **Media Library Type.** Only the value **V** is supported for this field for GS.

• **Uniqueness value.** Optional. A string identifying a VRU script name as unique.

**Step 2** Configure the Configuration Param field parameters:

**Note** Configuration parameters 1-10 are only for non-Packaged CCE deployments with CVP where GS is supported with external VXML. Only the Pass FTP Information parameter (parameter 11) is configurable when using the Agent Greeting recording feature.

• **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 *Elements Specifications for Cisco Unified CVP VXML Server and Cisco Unified Call Studio* guide for more information.

• **ftpPath** - A 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`.

---

**Passing Information to the Call Studio Scripts Executing on VXML Server**

You can pass up to 1050 characters to the Call Studio scripts executing on VXML server by using an ECC Variable array.

**Table 18: To External VoiceXML ECC Variable Array**

<table>
<thead>
<tr>
<th>ECC Variable Name</th>
<th>Type</th>
<th>Max. Number of Elements</th>
<th>Max. Size of Each Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>user.microapp.ToExtVXML</td>
<td>Array</td>
<td>5</td>
<td>210</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>user.microapp.ToExtVXML[0]</td>
<td>&quot;Company=Cisco;Job=technical writer&quot;</td>
</tr>
<tr>
<td>user.microapp.ToExtVXML[1]</td>
<td>&quot;Location=Boxborough;Street=Main&quot;</td>
</tr>
<tr>
<td>user.microapp.ToExtVXML[2]</td>
<td>&quot;FirstName=Gerrard;LastName=Thock&quot;</td>
</tr>
<tr>
<td>user.microapp.ToExtVXML[3]</td>
<td>&quot;Commute=1hour;Car=Isuzu&quot;</td>
</tr>
</tbody>
</table>

Unified CVP sends each name/value pair as a session variable on the call to VXML server (for example, a session variable named **Company** with a value of **Cisco**). The session variables are accessible in the Call Studio scripts.

**Passing Data Back to Packaged CCE from the VXML Server**

Unified CVP can return 840 characters from the VXML server.

The following ECC Variable array is added:

Table 20: From External VoiceXML ECC Variable Array

<table>
<thead>
<tr>
<th>ECC Variable Name</th>
<th>Type</th>
<th>Max. Number of Elements</th>
<th>Max. Size of Each Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>user.microapp.FromExtVXML</td>
<td>Array</td>
<td>4</td>
<td>210</td>
</tr>
</tbody>
</table>

The Get Speech micro-app returns up to 840 characters by populating the **user.microapp.caller_input** variable and each element of the **user.microapp.FromExtVXML** array.

**Note**

By default **user.microapp.FromExtVXML** ECC variable is pre-defined for Packaged CCE but not enabled. You can use the pre-defined ECC variable or update the length based on your needs.

**Scripting for Unified CVP with Call Studio**

You can use Call Studio to build sophisticated CVP applications which can then be loaded onto a VXML Server machine for execution.

To invoke a VXML Server application, create a Packaged CCE 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 Packaged CCE

This section describes...
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 CVP 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 Packaged CCE Lookup call flow model. See High-Level Configuration Instructions.

**Figure 60: Call Studio Application Used with Unified CVP Standalone**

---

**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, Packaged CCE VRU handling and Unified CVP Call Server (including the SIP Service, Packaged CCE service, and CVP Service).

### Call Studio ReqICMLabel Element to Pass Data

The ReqICMLabel element allows a Call Studio script to pass caller input, Call Peripheral variables, and Expanded Call Context (ECC) variables to a Packaged CCE script. The ReqICMLabel must be inserted into a Call Studio script as a decision element. In Call Studio, the returned Packaged CCE 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 Packaged CCE 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 Packaged CCE is \{Data.Element.GetICMLabel.result\}, where result is the variable of the ReqICMLabel element that contains the Packaged CCE label.

**Table 21: Settings**

<table>
<thead>
<tr>
<th>Name (Label)</th>
<th>Type</th>
<th>Required</th>
<th>Single Setting Value</th>
<th>Substitution Allowed</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call Peripheral Variables 1 - 10 (callvar1 - callvar10)</td>
<td>String</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Call Peripheral variables passed by the Call Studio script to the Packaged CCE server. This setting can be a maximum of 40 characters. The Packaged CCE server returns a name-value pair for up to 10 Call Peripheral Variables in a result. Any value that is placed in callvar&lt;n&gt; from a Call Studio script is returned unchanged, if the Packaged CCE script does not change it.</td>
</tr>
<tr>
<td>Call Peripheral Variables Return 1 - 10 (callvarReturn1 - callvarReturn10)</td>
<td>String</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Call Peripheral variables created upon the return of the Packaged CCE Label request, regardless of whether or not these variables are filled by the Packaged CCE script. You need two sets of these variables to keep reporting to the Packaged CCE Call Peripheral Variables separate from what is returned from Packaged CCE.</td>
</tr>
</tbody>
</table>
Table 22: Element Data

<table>
<thead>
<tr>
<th>Name (Label)</th>
<th>Type</th>
<th>Required</th>
<th>Single Setting Value</th>
<th>Substitution Allowed</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FromExtVXML0 - 3 (External VXML 0 - External VXML 3)</td>
<td>String Array</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>External Call Context (ECC) variables passed by the Call Studio script to the Packaged CCE Packaged CCE 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.</td>
</tr>
<tr>
<td>ToExtVXML0 - 4 (External VXML 0 - External VXML 4)</td>
<td>String Array</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>External Call Context (ECC) variables received from the Packaged CCE server. The Packaged CCE server returns a string of name-value pairs, separated by semicolons, for up to five external VoiceXML variables.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Integer</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>3000 (ms)</td>
<td>The number of milliseconds that the transfer request waits for a response from the Packaged CCE server before timing out. <strong>Note</strong>: This value is increased or decreased by increments of 500 ms.</td>
</tr>
<tr>
<td>caller_input (Caller Input)</td>
<td>String</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>This setting can be a maximum of 210 characters. The caller_input is only passed to Packaged CCE from Call Studio.</td>
</tr>
</tbody>
</table>

Table 22: Element Data

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>result</td>
<td>String</td>
<td>Packaged CCE label returned from a Packaged CCE 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}.</td>
</tr>
</tbody>
</table>
CallPeripheralvariablesarethattheCallStudio scripts pass to the Packaged CCE server. Valid Call Peripheral Variables are callvar1-callvar10.

CallPeripheralvariables that the Packaged CCE script returns to the VXML Server. Valid Call Peripheral Variables are callvarReturn1-callvarReturn10.

For example, if a Packaged CCE 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 Packaged CCE script by using

Data.Element.ReqICMLabelElement.callvarReturn3

The returned value is "Cisco Systems, Inc."

Table 23: Session Data

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Value for a name-value pair contained in a ToExtVXML variable returned in the Packaged CCE label. You must know which name-value pairs are set in the Packaged CCE script to retrieve the correct value from the Call Studio script. For example, if a Packaged CCE script contains a user.microapp.ToExtVXML0 variable with the string value &quot;CustomerName=Mantle&quot;, specify Data.Session.CustomerName. If the same Packaged CCE script contains a user.microapp.ToExtVXML0 variable with the string value &quot;BusinessType=Manufacturing&quot;, you can access the customer business type returned by the Packaged CCE script by using Data.Session.BusinessType.</td>
</tr>
</tbody>
</table>

Table 24: Exit States

<table>
<thead>
<tr>
<th>Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>done</td>
<td>The element execution is complete and the value is successfully retrieved.</td>
</tr>
<tr>
<td>error</td>
<td>The element failed to retrieve the value.</td>
</tr>
</tbody>
</table>

Studio Element Folder is "Cisco."
Integrate Call Studio Scripts with Unified CCE 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 CCE script with ECC variables configured for VXML Server
• Creating a VRU Script to run in the Packaged CCE script

Integrate Call Studio Scripts with Packaged CCE Scripts

The following steps describe how to integrate Call Studio scripts with Packaged CCE:

Procedure

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Set the user.microapp.ToExtVXML[0] ECC variable to application=HelloWorld.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note</td>
<td>This example indicates that the VXML Server executes the &quot;HelloWorld&quot; application. To execute a different application, change the value of user.microapp.ToExtVXML[0] accordingly.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Create a Run External Script node within the Packaged CCE script with a VRU Script Name value of GS,Server,V.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• 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.)</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>After you configure the Packaged CCE script, configure a corresponding VXML Server script with Call Studio. The VXML Server script must</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Begin with a Unified CVP Subdialog_Start element (immediately after the Call Start element)</td>
</tr>
<tr>
<td></td>
<td>• Contain a Unified CVP Subdialog_Return element on all return points (script must end with a Subdialog_Return element)</td>
</tr>
<tr>
<td></td>
<td>• Must include a value for the call input for the Unified CVP Subdialog_Return element</td>
</tr>
<tr>
<td></td>
<td>• Must add Data Feed/SNMP loggers to enable reporting</td>
</tr>
</tbody>
</table>

Outbound Option Scripting

Outbound Option uses Packaged CCE Enterprise scripting configured on the Administrative Workstation to manage campaigns.

There are two types of scripts:

• Outbound Option Administrative Scripts, on page 229
• Outbound Option Agent Routing Scripts, on page 229
Outbound Option Administrative Scripts

Outbound Option administrative scripts enable, disable, or throttle campaign skill groups for outbound campaigns. The scripts can also automatically close out a skill group for a specific campaign based on time or any other conditional factor that the admin script can access. This scripting is performed at the skill group level to provide more flexibility when managing larger campaigns that are distributed across multiple skill groups.

Enable a campaign skill group by setting the campaign mode to one of the available modes: Preview, Direct Preview, Progressive or Predictive. (For more information about these modes, refer to this section: Dialing Modes.) Schedule an administrative script to run at regular intervals. Disable the campaign skill group by the administrative script, using a script node to change the campaign mode to inbound for that skill group.

This script also provides the ability to control the percentage of agents to be used in a campaign skill group and whether this skill group is used for other campaigns or inbound calls.

Note

An administrative script controls a campaign skill group. A campaign skill group can be mapped only to one campaign at a time. A skill group can be reused if new campaigns are added. If a campaign skill group is recycled, its administrative scripting should be reused as well. However, note that although it is possible to have two administrative scripts controlling the same skill group, conflicting campaign mode requests for Outbound Option can result.

Outbound Option Agent Routing Scripts

Two types of routing scripts are described later in this document. One is for Agent Campaign and one is for IVR Campaign. For instructions on creating them, refer to Set Up Routing Scripts.

Call Type Reporting and Outbound Option Campaigns

The call type is an Unified Contact Center concept that maps a route point dialed number to a routing script. It is also a very useful reporting object for the enterprise to describe all calls that traversed a specific routing script. The call type mostly applies to inbound traffic in the call center because the dialer does not use the routing script when placing outbound calls. However, Outbound Option does use routing scripts to reserve agents and to transfer calls to the Unified CVP, so that it can provide some insight into how calls are being routed.

Because Outbound Option uses a routing script to reserve agents, the Call Type real-time and half-hour reports contain data for Outbound Option reservation calls. It is important to note that this call type data pertains only to reservation calls and does not include reporting information for any outbound calls. To eliminate any confusion regarding Call Type reporting, create a separate call type for Outbound Option routing scripts.

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 Call Studio, use the Deploy Remotely option to deploy the scripts to an FTP Server.
- 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** 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 level, et cetera. See the Call Studio documentation for more information.

**Step 3** 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.

Deploy Call Studio Scripts Using the Operations Console

**Procedure**

**Step 1** From the web browser, enter the following URL:

**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 into 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.
This chapter describes how reporting data is collected, the system entities that appear in reports, lists of reports for each entity, and configuration settings that affect data collection and report content.

- Reporting Data, page 231
- Reporting entities and concepts, page 236
- Data Loss and Component Failover, page 278
- Call type and Skill Group Metrics, page 282
- Reporting on Skill Groups, page 285
- Reporting on Precision Queues, page 285

### Reporting Data

Packaged CCE manages high volumes of call data, which are processed on the Packaged CCE Data Server. This section explains how reporting data are replicated in the system and presented in reports.

### Real Time Data Collection

Both the Unified CCE Peripheral Gateway and the Call Router produce real time data that is forwarded and stored in both the Data Server and the optional External AW/HDS/DDS server. Old real time data is constantly overwritten by new real time data. No history is kept. Real time data is stored in data fields that reflect four time increments, as described in the following table:
### Table 25: Real time data time increments

<table>
<thead>
<tr>
<th>Real time data time increments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half</td>
<td>&quot;Half&quot; values contain a value for the current half-hour. Real time half-hour values are not affected by Interval configuration. That is, if you set the historical reporting interval to 15 minutes, the Half values in real time tables represent the current half-hour time period falling between xx:00:00 and xx:29:59, or xx:30:00 and xx:59:59. For example, if it is currently 09:18:33, the CallsOfferedHalf column in the Call_Type_Real_Time table contains a value that reflects the first 18 minutes and 33 seconds of the specific half-hour. When a new half-hour begins, at time (09:00:00 or 09:30:00), the database element is reset to zero.</td>
</tr>
<tr>
<td>Now</td>
<td>&quot;Now&quot; contains a snapshot of the activity at a particular instant (the last check). For example, Packaged CCE software tracks CallsQNow, which is the number of calls currently in queue for a service or route. When a call is answered, the CallsQNow count is reduced immediately by one (-1) because the call has left the queue. This change is seen at the next real time update for reports that query for that value.</td>
</tr>
<tr>
<td>To5</td>
<td>The &quot;To5&quot; values track data on a rolling five-minute basis. The rolling five-minute data employs a &quot;sliding&quot; five-minute window.</td>
</tr>
<tr>
<td>Today</td>
<td>Contains the counts since midnight for each value</td>
</tr>
</tbody>
</table>

---

### Historical and Interval Data Collection

Packaged CCE stores some historical data in *Half_Hour* tables and other historical data in *Interval* tables. Interval tables contain half-hour summaries. Interval data is kept for 13 months. If you need additional detail reporting detention, add the optional database server (AW/HDS/DDS).
Table 26: Historical and interval data

<table>
<thead>
<tr>
<th>Historical data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval (30-minute)</td>
<td>Note: Two Interval tables—Dialer_Interval and Campaign_Query_Rule_Interval—always contain 30-minute data. Interval tables are: • Agent_Interval (30) • Agent_Skill_Group_Interval (30) • Skill_Group_Interval (30) • Call_Type_Interval (30) • Call_Type_Skill_Group_Interval (30) • Campaign_Query_Rule_Interval(30) • Dialer_Interval (30) • Router_Queue_Interval(30)</td>
</tr>
<tr>
<td>Half Hour</td>
<td>The Half_Hour tables are populated for completed half-hour intervals, and the data fields are stored in the database with the extension “ToHalf” (for example, Application_Gateway_Half_Hour.ErrorsToHalf). These elements contain a value for a completed half-hour interval. The completed interval is the time period falling between xx:00:00 and xx:29:59, or xx:30:00 and xx:59:59. For example, it is now 15:50:00. An error occurred at 15:47:00. The half-hour interval reported on right now is for the 15:00:00 to 15:29:59 interval. The error that occurred at 15:47:00 will be written to the database at 16:00:00, when the 15:30:00 to 15:59:59 half-hour interval is complete. Examples of Half_Hour tables are: • Application_Gateway_Half_Hour • Campaign_Half_Hour • Trunk_Group_Half_Hour • Route_Half_Hour</td>
</tr>
</tbody>
</table>

Configuration Data

Configuration tables define the entities and entity names that are defined in Configuration Manager and Web Config. They contain EnterpriseName fields that associate the key values in the historical tables with the text labels that are used in reports.
Examples of configuration tables are the Agent, Agent Team, Skill Group, and Call Type tables. For example, adding a new Agent Team in Web Config adds an EnterpriseName for that team in the Agent Team database table.

Configuration Data and Routing Scripts are created and edited on the Administration & Data Server, stored on the data server, stored on the data server database and replicated to optional external AW/HDS/DDS servers.

Note

The name field in Web Config maps to enterprise names in the database.

---

**Call Detail Data**

Packaged CCE retains five weeks of records in each of these two detail tables on the data server. If additional detail reporting retention is desired, the optional database server (AW/HDS/DDS) must be added to the configuration. Two database tables store call detail, as described below:

- **Route Call Details**
  
  For every call routing request it handles, the Router records detailed data about the call and how it was routed to a peripheral by Unified ICM. This *route call detail data* (RCD record) is stored in the Route_Call_Detail table.

  RCD data is written to the database when the script ends. Non-routed calls, such as direct dials, transfers, and conferences, have no RCD records.

  You can use the data in the Route_Call_Detail table to see the starting point of the call. For example, you can see the Automatic Number Identification (ANI) and the type of request made. A typical Route_Call_Detail record might describe a call that originated with a CVP request, and an ANI of 9785551000. In addition, route call detail tells you how long the call was held in a queue.

- **Termination Call Detail**
  
  Detailed *termination call detail data* (a TCD record) is written for each call that arrives at a peripheral. The TCD record is written after the call segment terminates and the after-call work is complete.

  For example, typical Termination_Call_Detail data might show that the call was an Inbound ACD Call, that it was handled by a particular Skill Group; and that a particular agent handled the call. The Termination_Call_Detail record also describes the final disposition of the call (for example, how the call terminated; abandoned in network, disconnect/drop, and abandoned delay).

  In Packaged CCE most inbound calls have at least two TCDs: one for the segment of the call that was at CVP and one for the segment of the call handled by the agent.

  The Termination_Call_Detail table includes records that indicate which TCDs were used to create the Call Type, Call Type Precision Queue, and the Call Type Skill Group reports.

Although Call Detail records are stored in the Route_Call_Detail and Termination_Call_Detail tables, none of the standard (stock) reports retrieve data from these two tables for performance reasons.

To use Call Detail data in reports, you must create custom reports that populate from your custom database. These two detail tables are constrained to five weeks. If additional detail record reporting is required, an external database (AW/HDS/DDS) must be added to the configuration.
Event Data

Two database tables store system and application event data that is generated by the various components of Unified CCE software:

- **Event**
  
The Event table stores all system events that are generated by the various Unified CCE components.

- **Application_Event**
  
The Application_Event table stores all application events that are generated by the various components.
  This is a subset of the events reported in the Event table.

---

**Note**

By default, the Event and Application_Event data is not replicated to the HDS database.

Database Tables that Hold Reporting Data

All report data is pulled from tables and rows in the Packaged CCE database. Many fields are direct database values, as reflected in their displayed column names in the reports.

For example:

- The direction of the active task on which the agent is currently working is derived from `Agent_Real_Time.Direction`.

Other report data fields are less obvious—because they represent calculated values, because the same data-entity name is used in multiple contexts, or because they are calling database values whose names are not clearly indicative.

**Calculated Fields.** Many report values are the result of calculated fields. For example, in reports that present Skill Group Real Time activity, the Average Active Time (AAT) is calculated as follows: `Skill_Group_Real_Time.HandledCallsTalkTimeTo5 / Skill_Group_Real_Time.CallsHandledTo5`. Refer to the *Unified Intelligence Center Report Template Guide* for details on the calculated fields.

**Fields Used in Many Tables and Context.** Examples are the fields `Deleted`, `Description`, and `EnterpriseName`, which appear in many tables.

Why Report Data Can Differ

This chapter explains how and why data can differ in reports.

Real Time and Historical Reports

Counts in real time data (for example CallsHandledTo5) do not match up with counts in the historical interval records (for example, CallsHandled) because the real time data is moved to the historical database at the end of each half-hour interval.

Consider this example: at 8:55 a call comes into the contact center and is answered by an agent.
The real time count for CallsAnswered increases by one (+1).

- Between 8:55 and 9:00, the real time data shows the answered call.
- The answered call does not populate the half-hour data until 9:00, when the 8:00 to 8:59:59 interval ends.

**Interval Boundaries**

Counts that would typically match up for a day, such as CallsOffered and CallsHandled, might not always match up over specific intervals. This discrepancy occurs because the counts for some data elements might be increased across boundaries.

Consider this example: at 8:55, a call comes in to the contact center and is answered by an agent. The agent completes the call at 9:05.

- In the historical database, the call is counted as offered in the 8:30:00 to 8:59:59 interval.
- The call is counted as handled in the 9:00:00 to 9:29:59 interval.
- If you run a report for the 9:00:00 to 9:29:59 interval, it appears that tasks handled does not equal tasks offered for the interval.

You also might notice that tasks offered does not equal task abandoned + tasks handled for an interval. Tasks offered reflects the number of calls and tasks that were offered to agents in this interval, while tasks handled and tasks abandoned might include calls that were offered in the last interval and completed in this interval. Some historical report templates group statistics into "Completed Tasks" to indicate that the statistics represent all calls and tasks that completed in this interval.

In general, interval boundary issues are reduced if you run daily reports. However, if your contact center runs 24 hours a day, you might still notice discrepancies for intervals such as the 11:30:00 to 11:59:59 and 12:00:00 to 12:29:59 intervals.

**Reporting entities and concepts**

This section describes the Unified CCE entities for which reports are available. Each entity is defined and the information available about that entity is described.

**Agent States**

Agent states are determined from an agent's activity within a skill group. Agent state is recorded in numerous database tables and is presented in reports as both a number (Not Ready) and as a percentage (% Not Ready).

You can monitor agent states in real time to view current agent activity. You can also review past performance data to identify trends in agent states. For example, historical reports can show how much time an agent spends in Not Ready state, which indicates whether the agent is adhering to the schedule.

Information for some states is different for the chat Media Routing Domain (MRD). This table highlights these differences.
### Table 27: Agent states that appear in reports

<table>
<thead>
<tr>
<th>State in Skill Group</th>
<th>Description for all MRDs except chat</th>
<th>Description for chat MRD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active/Talking</strong></td>
<td>The agent is working on a task or a call in this skill group. For agents who handle non-voice tasks, this state is reported as <em>Active</em>. For agents who handle voice tasks, this state is reported as <em>Talking</em>.</td>
<td>The agent is performing wrap-up work for a task associated with this skill group. For these agents, the state is reported as <em>Active</em>.</td>
</tr>
<tr>
<td><strong>Work Ready</strong></td>
<td>The agent is performing wrap-up work for a call or task in this skill group. If the agent is handling a voice call, the agent enters Not Active state when wrap is complete. If the agent is handling a non-voice task, the agent might enter Not Active or Not Ready state when wrap up is complete.</td>
<td>The agent is performing wrap-up work for a task associated with this skill group. The agent enters Not Active state when wrap up is complete.</td>
</tr>
<tr>
<td><strong>Paused/Hold</strong></td>
<td>The agent is performing wrap-up work for a call in this skill group. The agent enters Not Ready state when wrap up is complete. For agents who handle voice tasks, the state is reported as <em>Hold</em>. For agents handling Outbound Option calls, the Hold state indicates that the agent has been reserved for a call because the Outbound Dialer puts the agent on hold while connecting the call.</td>
<td>The agent is performing wrap-up work for a call or task associated with this skill group. The agent enters Not Ready state when wrap up is complete.</td>
</tr>
<tr>
<td><strong>Reserved</strong></td>
<td>The agent has been offered a call or task associated with the skill group. For voice calls, agents are Reserved when their phones are ringing. Agents handling Outbound Option calls are never placed in Reserved state; the Outbound Option Dialer puts the agent on hold when reserving the agent for a call.</td>
<td>The agent is not in Active, Work Ready, or Paused state in this skill group. The agent has been offered one or more tasks associated with this skill group.</td>
</tr>
</tbody>
</table>
State in Skill Group | Description for all MRDs except chat | Description for chat MRD
--- | --- | ---
Busy Other | The agent is Active, Work Ready, Reserved, or on Hold/Paused in another skill group in the same MRD. | The agent is not in Active, Work Ready, Reserved, or Paused state with respect to a task associated with this skill group. The agent is in Active, Work Ready, Reserved, or Paused in another skill group in the same MRD. 

Busy Other is a state in which the agent handling calls is assigned to other skill groups during the interval. For example, an agent might be talking on an inbound call in one skill group while simultaneously logged on to, and ready to accept calls from, other skill groups.

The agent can be active (talking on or handling calls) in only one skill group at a time. Therefore, while active in one skill group, for the other skill group the agent is considered to be in the Busy Other state.

Not Active | The agent is not working on any task or call associated with this skill group. | The agent is not working on any task or call associated with this skill group.

Not Ready | The agent is not available to be assigned a task. If an agent is Not Ready in one skill group, the agent is Not Ready in all skill groups within the same MRD. | The agent is not available to be assigned a task. If an agent is Not Ready in one skill group, the agent is Not Ready in all skill groups within the same MRD.

How Agent States Are Calculated in Reports

Agent States are presented in many reports as percentages.

Table 28: Calculations for agent state percentages

<table>
<thead>
<tr>
<th>Table.Field</th>
<th>Calculation</th>
</tr>
</thead>
</table>
### Agent States, Skill Groups, and Precision Queues

Agents can belong to multiple skill groups or Precision Queues in an MRD. When an agent is handling a task that was routed to a skill group, the agent is Active in that skill group.

- For direct incoming calls or transferred routed calls that do not use the dialed number, the active skill group is the default or first skill group defined for the agent.
- For new outgoing calls (AgentOutCalls or InternalCalls) or transferred outbound calls, the active skill group is the first skill group defined for the agent.

If you are reporting on agents who handle chat tasks (and who can work on more than one task at a time), gather agent state information from both the Available in MRD and Agent State columns.

The agents' state in the active skill group or precision queue dictates their state in other skill groups or precision queues in the MRD to which they belong, as follows:

- If the agent is Active, Work Ready, Reserved, or Hold/Paused in one skill group or Precision Queue in the MRD, the agent state is Busy Other for all other skill groups or Precision Queues in the MRD.
- If the agent is Not Ready in one skill group or Precision Queue in the MRD, the agent is Not Ready in all skill groups or Precision Queues in the MRD.

#### Related Topics

- Reporting in a Multichannel Environment

### Agent State and Task State Relationship

Agent state times are reported on interval boundaries regardless of whether or not the call or task is finished. Call and task state times are reported only when the task ends. The call/task ends when wrap up is complete.
The following figure illustrates the correlation between agent state and call state for a voice call. The agent reserve time includes the time it took the call to arrive at the agent’s phone or desktop (network time) as well as the amount of time that the call rang on the agent’s phone or waited on the agent’s desktop (offer/ring time).

**Figure 61: Agent state and task state relationship**

If the interval boundary ends when the call is ringing on the agent's phone, the reserved time for the agent includes the network time and part of the ring time. At the next interval, the remaining ring time is reported in the reserved time of the agent. However, the call's time does not appear on a report until wrap up has been completed on the call.

### Reports that Show Agent States

These are some reports that show information on Agent State:

- Unified IC Agent Team State Counts Real Time
- Unified IC Agent Real Time All Fields
- Unified IC Agent Historical All Fields

### Agent Logout Reason Codes

Agent Logout Reason codes are defined in the agent desktop software and appear in historical reports as their numeric equivalent, with no text code. For example, if reason code 1 equals “end of shift” and the agent selects that reason for logging out, the report displays “1”.

In addition to the codes configured at the desktop, some codes are generated automatically when the agent is logged out by the software. The following table describes these predefined Logout Reason codes.

<table>
<thead>
<tr>
<th>Predefined Logout Reason Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>The agent reinitialized due to peripheral restart.</td>
</tr>
<tr>
<td>-2</td>
<td>The PG reset the agent, normally due to a PG failure.</td>
</tr>
<tr>
<td>-3</td>
<td>An administrator modified the agent's extension while the agent was logged in.</td>
</tr>
</tbody>
</table>
### Predefined Logout Reason Code

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>50002 A CTI OS component failed, causing the agent to be logged out. This could be due to closing the agent desktop application, heartbeat time out, a CTI OS Server failure, or a CTI OS failure.</td>
</tr>
<tr>
<td>50004 The agent was logged out due to agent inactivity as configured in agent desk settings.</td>
</tr>
<tr>
<td>50020 The agent was logged out when the agent's skill group assignment dynamically changed.</td>
</tr>
<tr>
<td>50030 The agent was logged out when the agent's skill group assignment dynamically changed on the Administration &amp; DataServer.</td>
</tr>
<tr>
<td>50040 The mobile agent was logged out because the call failed.</td>
</tr>
<tr>
<td>50042 The mobile agent was logged out because the phone line disconnected when using nailed connection mode.</td>
</tr>
</tbody>
</table>

### Agent Not Ready Reason Codes

There are reports that show the codes agents select when entering Not Ready state, that calculate the percentage of time spent in the Not Ready state, and that show specific Not Ready reasons based on the time range you specify.

These reports help you identify whether agents are taking the appropriate number of breaks and whether their breaks are the appropriate length.

Some reports display both the text of the reason code (if configured) and the corresponding number. For example, if an agent enters Not Ready state and selects “Break” as the reason code, and if you have configured text for this code in Configuration Manager, reports display “Break [1]”. Other reports display the numeric Not Ready reason code only.

If an agent's total login session is not included in the specified time range (for example, the agent was still logged in at the end of the time range), an asterisk (*) appears next to the agent's name in the report to indicate that data for that agent is not complete for the range.

For Unified CC, in addition to Not Ready reason codes that you define, there are predefined Not Ready reason codes for situations in which the agent is made Not Ready automatically by the software. The following table describes these predefined Not Ready reason codes.

---

**Reports that Show Agent Logout Reason Codes**

These are some reports that contain information on Agent Logout reason codes:

- Unified IC Agent Real Time All Fields
- Unified IC Agent Not Ready Detail
### Table 30: Predefined Not Ready reason codes for Unified CC

<table>
<thead>
<tr>
<th>Predefined Not Ready Reason Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>50001</td>
<td>The CTI OS client disconnected, logging the agent out. <strong>Note</strong>: This reason code is converted to a 50002, so 50001 does not display in the agent log out records.</td>
</tr>
<tr>
<td>50002</td>
<td>A CTI OS component failed, causing the agent to be logged out. This could be due to closing the agent desktop application, heartbeat time out, a CTI OS Server failure, or a CTI OS failure.</td>
</tr>
<tr>
<td>50003</td>
<td>Agent was logged out because Unified CM reported the device out of service.</td>
</tr>
<tr>
<td>50004</td>
<td>Agent was logged out due to agent inactivity as configured in agent desk settings.</td>
</tr>
<tr>
<td>50005</td>
<td>For a Packaged CCE deployment where the Multi-line Agent Control is enabled in the peripheral, and the Multi-line Agent Behavior is configured to impact agent state, the Agent will be set to Not Ready with this code while talking on a call on the Non-ACD line.</td>
</tr>
<tr>
<td>50010</td>
<td>The agent did not receive multiple consecutive routed calls. The system makes the agent Not Ready automatically so that additional calls are not routed to the agent. By default, the number of consecutive calls missed before the agent is made Not Ready is 2.</td>
</tr>
<tr>
<td>50020</td>
<td>Agent was logged out when the agent's skill group dynamically changed on the Administration &amp; Data Server.</td>
</tr>
<tr>
<td>50030</td>
<td>If an agent is logged in to a dynamic device target that is using the same dialed number (DN) as the PG static device target, the agent is logged out.</td>
</tr>
<tr>
<td>50040</td>
<td>Mobile agent was logged out because the call failed.</td>
</tr>
<tr>
<td>50041</td>
<td>Mobile agent state changed to Not Ready because the call failed when the mobile agent's phone line rang busy.</td>
</tr>
<tr>
<td>50042</td>
<td>Mobile agent was logged out because the phone line disconnected while using nailed connection mode.</td>
</tr>
<tr>
<td>32767</td>
<td>The agent's state was changed to Not Ready because the call fails when the agent's phone line rings busy.</td>
</tr>
<tr>
<td>-1</td>
<td>Agent reinitialized (used if peripheral restarts).</td>
</tr>
<tr>
<td>-2</td>
<td>PG reset the agent, normally due to a PG failure.</td>
</tr>
</tbody>
</table>
By default, predefined Not Ready reason codes do not have associated textual reason codes. They appear as numbers in reports. If you want to see a textual code for these Not Ready reason codes, enter the predefined Not Ready reason code into the Reason Code gadget with the related text. For example, you might want to label the 32767 Not Ready reason code "Redirection on No Answer".

### Reports that Show Agent Not Ready Reason Codes

These are some of the reports that contain information on Not Ready codes and the time spent as Not Ready:

- Unified IC (Intelligence Center) Agent Skill Group Real Time All Fields
- Unified IC Agent Skill Group Historical All Fields

### Agent Task Handling

Agents can receive and place many different types of tasks. There are reports that show you what kind of tasks agents are handling and how well they are handling them. For example, there are reports that display statistics for calls placed, received, transferred, and conferenced, and there are reports that indicate how many calls were rerouted when the agent failed to answer the call.

### Types of Tasks

Tasks can be internal or external, incoming or outgoing.

- **Internal tasks** are calls made to an agent from another person or from another agent on Packaged CCE.

- **External tasks** are calls that are placed off the Packaged CCE, tasks that come in via CVP, or tasks that are routed to an agent from a person outside Packaged CCE. For example, calls from the call center to customers are considered external.

- **Incoming tasks** are tasks that an agent receives. Multichannel tasks are always incoming.

- **Outgoing tasks** are calls that an agent places. For example, if a customer calls an agent, the call is incoming for the agent. If an agent calls a supervisor, the call is outgoing for the agent.

For voice calls only, agents can also transfer calls, receive transferred calls, place consultative calls, and engage in conference calls.

The following table describes the tasks that an agent can receive and place and how those tasks are reported.
### Table 31: Types of tasks

<table>
<thead>
<tr>
<th>Type of task</th>
<th>Description</th>
<th>Reported As</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incoming direct/internal</td>
<td>Incoming Direct Tasks are tasks that come directly to the agent's extension. Examples of this kind of call include calls that are directly transferred by another agent without going through a script and calls that resulted from agent-to-agent calling. Data for these calls are stored in the InternalCallsRcvd fields of the Agent_Skill_Group_Interval historical database table.</td>
<td>Internal In</td>
</tr>
<tr>
<td>Outgoing external</td>
<td>These are calls initiated by agents from their extension that are placed outside the contact center. Outgoing External Tasks are always voice tasks. Consultative, conference out, and transfer out calls are counted as outgoing external calls if they are placed outside the contact center or to remote agent extensions at another site. Agent-to-Agent dialing is outgoing external for the agent initiating the call if the call has to be placed outside the contact center to get to the destination agent. Data for these calls are stored in the AgentOutCalls fields of the Agent_Skill_Group_Interval historical database table.</td>
<td>External Out Tasks</td>
</tr>
<tr>
<td>Outgoing internal</td>
<td>These are calls initiated by agents from their extension to another extension within the contact center. Outgoing Internal Tasks are always voice tasks. Consultative, conference out, and transfer out calls are counted as outgoing internal calls if they are placed to another CVP. Agent-to-Agent calls are outgoing internal for the agent initiating the call. Data for these calls are stored in the InternalCalls fields of the Agent_Skill_Group_Interval historical database table.</td>
<td>Internal Out Tasks</td>
</tr>
<tr>
<td>Type of task</td>
<td>Description</td>
<td>Reported As</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Packaged CCE–routed calls</td>
<td>All calls that are routed to the agent. Outbound Option calls are considered Packaged CCE–routed/incoming calls. Data for these calls are stored in the CallsHandled fields of the Agent_Skill_Group_Interval historical database table.</td>
<td>Tasks Handled Tasks Handled includes all calls, including calls that are transferred and conferenced, and consultative calls. Tasks Handled provides a high level view of routed tasks. Other report columns such as Transfer In and Conf Out provide more details about how the task was handled.</td>
</tr>
<tr>
<td>Transferred in</td>
<td>Calls transferred to an agent from another agent. Calls that are blind transferred by one agent to CVP for re-routing are counted in this column for the agent who receives the rerouted call. <strong>Note</strong> Data for these calls are stored in the TransferredIn fields of the Agent_Skill_Group_Interval historical database table.</td>
<td>Transfer In</td>
</tr>
<tr>
<td>Transferred out</td>
<td>Calls that are transferred from an agent. An agent can transfer both incoming and outgoing calls. Data for these calls are stored in the TransferredOut fields of the Agent_Skill_Group_Interval historical database table.</td>
<td>Transfer Out</td>
</tr>
<tr>
<td>Consultative</td>
<td>Calls in which an agent consulted with another agent or supervisor while having another call on hold. Data for these calls are stored in the ConsultativeCalls fields of the Agent_Skill_Group_Interval historical database table.</td>
<td>Cons Out</td>
</tr>
<tr>
<td>Conference in</td>
<td>Incoming calls that are conferenced. Data for these calls are stored in the ConferencedInCalls fields of the Agent_Skill_Group_Interval historical database table.</td>
<td>Conf In</td>
</tr>
<tr>
<td>Conference out</td>
<td>Outgoing calls that are conferenced. Data for these calls are stored in the ConferencedOutCalls fields of the Agent_Skill_Group_Interval historical database table.</td>
<td>Conf Out</td>
</tr>
</tbody>
</table>
Task Times

For each type of task that an agent can place, the amount of time that the agent spent working on that task is recorded in the Agent_Skill_Group_Interval database table, as follows:

- Packaged CCE–routed tasks - The time for these tasks begins when the agent answers the task and ends when the agent completes wrap up. The time is stored in the HandledCallsTime field.

- Incoming direct tasks - The time for these tasks begins when the agent answers the task and ends when the task disconnects. The time is stored in the InternalCallsRcvdTime field.

- External outgoing tasks - The time for these tasks begins when the agent initiates the task and ends when the task disconnects. The time is stored in the AgentOutCallsTime field.

- Internal outgoing tasks - The time for these tasks begins when the agent initiates the task and ends when the task disconnects. The time is stored in the InternalCallsTime field.

- Transferred-in tasks - The time for these tasks begins when the agent answers the transferred task and ends when the transfer is complete. The time is stored in the TransferredInCallsTime field.

- Transferred-out tasks - The time for these tasks begins when the agent activates the transfer button and ends when the transfer is complete. The time is stored in the TransferredOutCallsTime field.

- Consultative tasks - The time for these tasks begins when the agent activates the transfer button and ends when the target agent answers and the held task is restored (drop consultative call) or consult party drops. The time is stored in the ConsultativeCallsTime field.

- Conferenced-in tasks - The time for these tasks begins when the agent answers the task and ends when the task disconnects. The time is stored in the ConferenceInCallsTime field.

- Conferenced-out tasks - The time for these tasks begins when the agent activates the conference button and ends when the agent disconnects from the conference call and the supervisor drops out of the call. The time is stored in the ConferenceOutCallsTime field.

You might notice overlapping data in your reports for the amount of time for different types of calls. This happens because incoming tasks, such as routed tasks and calls directly to an agent, can be Transferred In and Conferenced In. Both incoming calls and outgoing calls placed by agents can be Transferred Out and Conferenced Out. The total time for the incoming or outgoing call includes transfer and conference time.

Agents can transfer and conference incoming calls both in and out. However, they can transfer and conference outgoing calls out only. This difference means that if an agent transfers an outgoing task to another agent, it is still considered an outgoing task.

Reports that Show Agent Task Handling

The Unified IC Agent Historical All Fields report contains information on Not Ready codes and the time spent as Not Ready.
Agent Utilization: Full-Time Equivalents and Percent Utilization

Because agents can work on multiple media and in multiple skill groups, they typically do not spend all of their time handling tasks for a single skill group. Determining staffing needs based on agents whose time is divided among skill groups and media can be difficult.

Report templates provide two types of statistics that give you a better view of how agents are being utilized and how many full-time agents would be required to handle the amount of work performed during an interval for a particular skill group.

These statistics are:

- % Utilization (percent utilization)
- FTE (full-time equivalent)

**Percent utilization** (% Utilization in reports) is computed in reports by dividing the total time agents spend handling calls in a skill group by the total time agents were ready to handle tasks. To calculate the time that an agent was ready, the system subtracts the Not Ready time from the total time that agents were logged on. Percent utilization shows you how well agents are being utilized within a skill group. For example, if the agent spent 20 minutes of the log on duration handling calls and was available to handle calls for 40 minutes, the percent utilization is 50%.

The **full-time equivalent** (FTE in reports) is the number of full-time agents that would be required to perform the work done during that interval for a skill group. To calculate the FTE, the system divides the total time that work was performed by the total time in the interval. For example, if agents spent a total of 3 hours (180 minutes) handling tasks during an interval (30 minutes), the FTE for task handling during the interval is 180 minutes / 30 minutes, which equals 6 full-time persons. This means that if all agents handled tasks full-time, the work could have been done by 6 agents.

Reports also provide FTE values based on an 8 hour shift calculation. It is assumed that agents work an 8-hour shift for the day. To calculate the FTE, the system divides the total time that work was performed by 8 hours. For example, if agents spent a total of 48 hours (2880 minutes) handling tasks during an 8 hour work shift (480 minutes), the FTE for task handling during the interval is 2880 minutes / 480 minutes, which equals 6 full-time persons. This means that if all agents handled tasks full-time, the work could have been done by 6 agents.

If you select a report interval that is less than 8 hours, the resulting value will be lower than expected.

**Reports that Show Percent Utilization and FTE Metrics**

These are some of the reports that contain operational information on Percent Utilization and FTE:

- Enterprise Skill Group Historical All Fields
- Peripheral Skill Group Historical All Fields Report
- Peripheral Skill Group Real Time All Fields Report
- Precision Queue Real Time All Fields
- Precision Queue Historical All Fields
Reports that Show Agent States

These are some reports that show information on Agent State:

- Unified IC Agent Team State Counts Real Time
- Unified IC Agent Real Time All Fields
- Unified IC Agent Historical All Fields

Reports that Show Agent Logout Reason Codes

These are some reports that contain information on Agent Logout reason codes:

- Unified IC Agent Real Time All Fields
- Unified IC Agent Not Ready Detail

Reporting on Agents

These are some Cisco Unified Intelligence Center reports relevant to agents.

Related Topics

- Agent Team Reports, on page 251
- Reports that Show Agent States, on page 240
- Supervisor Action Reports, on page 251
- Reports that Show Percent Utilization and FTE Metrics, on page 247

Skill Groups

A skill group is a collection of agents who share a common set of competencies that equip them to handle the same types of requests. Some examples of skill groups are a collection of agents who speak a specific language or who can assist callers with billing questions.

Each skill group belongs to a Media Routing Domain.

An agent can be a member of zero, one, or more skill groups.

To monitor agent performance, you can report on agents individually or you can report on all of the agents in one or more skill groups.

You can generate reports for skill groups that show agent activity (for example, the number of agents talking, available, or in wrap-up for a particular skill group).

In addition to generating Agent Skill Group reports, you can also use Skill Group reports to monitor operational performance. For example, you might want to see how a skill group is performing compared to other skill groups or to see if calls are being distributed evenly by your routing scripts and configuration.
Default Skill Group

The default skill group acts as a bucket to capture information about voice calls and non-voice tasks.

Role of Default Skill Group in Reporting

The default skill group acts as a bucket to capture information in these situations:

- For calls that are not routed by Packaged CCE routing script
- If a skill group is not specified in a routing script
- If the Agent to Agent node is used in a routing script for agent-to-agent dialing
- When the Queue to Agent node queues a task to an agent and the agent is not logged into the skill group specified in the Queue to Agent node

Using a default skill group helps to:

- Ensure the agent/skill group reports balance with the service and call type reports, since service and call type reports include only Packaged CCE-routed calls, and
- Isolate/identify non-Packaged CCE-routed calls within the agent and skill group report.

Statistics for the default skill group are affected by different types of calls, including new calls, agent-to-agent-dialing, and transferred and conferenced calls.

If you deploy Multichannel options in a Packaged CCE system, default skill groups are created for each Media Routing Domain that is configured.

How New Calls Increment Default Skill Group Statistics

Call statistics for all new outbound and incoming direct calls are incremented for the default skill group as follows:

- AgentOutCalls for external outbound calls

  Note  When an agent makes an outbound call as part of a consultative call, the call is not attributed to the Default skill group. It is attributed to the skill group for the consulting agent on the original call.

- InternalCalls for the internal outbound calls
- InternalCallRcvd for the direct incoming calls

  Note  CallsHandled is not incremented for the default skill group, since the default skill group can not be referenced in any script.
How Agent-to-Agent Dialing Increments Default Skill Group Statistics

Agent to agent dialing using the agent to agent node in the script also affects the default skill group. OutgoingExternal or OutgoingInternal are incremented for the default skill group of the agent initiating the agent to agent call. The default skill group InternalCallsReceived is incremented for the default skill group of the agent receiving the agent to agent call.

How Transferred and Conferenced Calls Increment Default Skill Group Statistics

The default skill group is also affected by transferred and conferenced calls. If agent A transfers or conferences a Packaged CCE/IPCC–routed call to another agent directly without using a script, OutgoingExternal or OutgoingInternal for agent A are incremented against the skill group of the Packaged CCE-routed call. However, IncomingDirect calls for agent B are incremented against the default skill group.

However, if the agent (agent A) transfers or conferences a Packaged CCE/IPCC–routed call to a dialed number that accesses a transfer or conference script that has an agent to agent node, OutgoingExternal or OutgoingInternal for the agent A is incremented for the skill group of the Packaged CCE/IPCC routed call. IncomingDirect calls for agent B are incremented for the default skill group.

The default skill group will also be incremented for emergency and supervisor assist calls when there is no existing call.

Reports on Skill Group Operations

Use the skill group templates to gain insight into operations, to see how one skill group is performing compared to other skill groups, and to track whether calls are being distributed evenly by your routing scripts and configuration.

- You can report on all skill groups in Packaged CCE.
- You can report on call statistics by agent skill group assignments

Note

For agents in more than one skill group, you can also use the agent by skill group templates as a tool for monitoring agent performance.

How Calls Offered is Calculated for Skill Group

The completed state for RouterCallsOffered at the skill group is calculated using these fields from the Skill_Group_Interval table:

- RouterCallsAbandToAgent
- CallsHandled
- RouterCallsDequeued
- RedirectNoAnsCalls
- RouterError
- ReserveCalls
• RouterCallsAbandQ
• RouterCallsAbandDequeued

Reports that Show Skill Group Operations

These are some of the reports that contain operational information on Skill Groups:

• Peripheral Skill Group Historical All Fields
• Peripheral Skill Group Real Time All Fields

Agent Teams and Supervisors

This section provides information about agent teams and supervisors.

Agent Team Reports

Supervisors can report on the agents in teams that they supervise to monitor the performance of a particular team.

You can select 0 or 1 primary supervisor for an agent team, and you can select multiple secondary supervisors for each team. Each supervisor can be a supervisor for multiple teams.

Supervisor Action Reports

Agent team supervisors can take advantage of supervisory features available on their desktops. These features include Supervisor Assist, Emergency Assist, Barge-In, and Intercept. There are two kinds of Supervisor and Emergency Assist: existing call and no call.

Note

These supervisory features are not available to agents using MRDs other than Voice.

Supervisor Assist and Emergency Assist for Existing Call

For Unified Intelligence Center, you can enable Supervisor Assist and Emergency Assist.

Agents can activate supervisor assist or emergency assist buttons on their desktop when they need special assistance from the primary or secondary supervisor assigned to their team.

Follow these guidelines to ensure that you can obtain accurate and useful data from these features:

• Plan to configure skill groups for supervisors handling Supervisor Assist and Emergency Assist requests. For example, you might configure one skill group for the primary and secondary supervisors of each agent team. This way, you can direct requests to these skill groups and report on Supervisor and Emergency Assist call activity for these skill groups.
• Plan to create call types and configure dialed numbers that map to the created call type.
• Run scripts that direct the requests to the appropriate supervisor skill group. In the script, first target the primary supervisor and then, if you have configured secondary supervisors, queue to secondary supervisors.

If consult is selected as an option on the agent desktop settings for supervisor or emergency assist. If the agent is on a call when the agent activates either the Supervisor or Emergency Assist desktop feature, the CTI software activates the conference key on behalf of the agent's phone and calls the supervisor using the Supervisor or Emergency Assist script. (This example assumes the emergency or supervisor assist script has an Agent-to-Agent node to find a supervisor. See Configuration and Scripting Considerations for Reporting on Supervisor Action.) The supervisor answers the call and consults privately with the agent. The following fields are incremented within the Agent Skill Group and Skill group tables.

**Table 32: Existing call: Consultative**

<table>
<thead>
<tr>
<th>Fields incremented for Agent’s skill group to which the call was routed</th>
<th>Fields incremented for Supervisor’s default skill group</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallsHandled, InternalCall, SupervisorAssistCalls/EmergencyAssist</td>
<td>InternalCallsRcvd</td>
</tr>
</tbody>
</table>

For the agent, the call is reported in Tasks Handled and either Sup Assist or Emergency report fields. For the supervisor, the call is reported in Tasks Handled report fields.

**Note**

During the consultation, the supervisor can decide to barge-in to the call using the supervisor desktop Barge-In feature.

**Barge-In**

When the supervisor activates the Barge-In desktop feature, the agent’s desktop completes a conference to the supervisor so that the supervisor can join into the conversation with the call. The following fields are incremented for both the agent and the supervisor when the Barge-In feature is activated in the agent skill group and skill group tables.

**Table 33: Supervisor Barge-In**

<table>
<thead>
<tr>
<th>Fields incremented for Agent’s skill group to which the call was routed</th>
<th>Fields incremented for Supervisor’s default skill group</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallsHandled, InternalCalls, BargeInCalls</td>
<td>BargeInCalls, InternalCallsRcvd</td>
</tr>
</tbody>
</table>

For the agent, the call is reported in Tasks Handled and Barge-In report fields. For the supervisor, the call is reported in Tasks Handled and Barge-In report fields.
Intercept

If the supervisor decides to intercept (take over) the call, the supervisor activates the Intercept desktop button. This causes the agent to be dropped out of the conference, thereby allowing the supervisor to take over the call. The following fields are incremented during the intercept operation for both the agent skill group and skill group tables.

Table 34: Supervisor intercept

<table>
<thead>
<tr>
<th>Fields incremented for Agent's skill group to which the call was routed</th>
<th>Fields incremented for Supervisor's default skill group</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterceptCalls</td>
<td>InterceptCalls</td>
</tr>
</tbody>
</table>

For the agent, the call is reported in the Intercept report field. For the supervisor, the call is reported in the Intercept report field.

**Reports that Show Information on Agent Teams**

These are some of the reports that contain information on Agent Teams

- Unified IC Agent Team Real Time All Fields
- Unified IC Agent Team Historical All Fields
- Agent Team State Counts Real Time

**Reporting on Supervisors**

These are some Cisco Unified Intelligence Center reports relevant to supervisors.

**Related Topics**

- Agent Team Reports, on page 251
- Supervisor Action Reports, on page 251
- Reports that Show Percent Utilization and FTE Metrics, on page 247
- Reports that Show Agent Not Ready Reason Codes, on page 243
- Reports that Show Agent Logout Reason Codes, on page 241
- Reports on agent activity in skill groups

**Reports that Show Information on Agent Teams**

These are some of the reports that contain information on Agent Teams

- Unified IC Agent Team Real Time All Fields
- Unified IC Agent Team Historical All Fields
- Agent Team State Counts Real Time
**Average Speed of Answer**

Average Speed of Answer (ASA) is the average of the sum of the time that all incoming tasks to the service waited before being answered. This includes delay time, queue time, and ring time.

ASA starts when the call enters the queue and is set at these levels:

- Agent
- Skill Group
- Call Type
- Precision Queue

At the agent and skill group levels, the ASA metric is useful for monitoring agent and skill group performance. At the Call Type level, the ASA metric provides insight into how callers experience the system and how quickly calls are being answered.

**For Agents:** The agent's average speed of answer in HH:MM:SS (hours, minutes, seconds) calculated by dividing the total time that callers spent in queue and while the call was ringing at the agent's desktop before the task is answered by the number of calls that the agent answered.

**For Skill Groups:** The skill group's average speed of answer in HH:MM:SS (hour, minutes, seconds) calculated from the time spent by callers when placed in queue and ringing at the agent's desktop before the task is answered divided by the number of tasks answered.

**For Call Types:** The average answer wait time from when first queue to skill group or longest available agent (LAA) select node was executed for this call to when this call was answered. This is an important measure of service quality because the time can vary, even over the course of one day, due to call volumes and staff levels.

**For Precision Queues:** The Precision Queue's average speed of answer in HH:MM:SS (hour, minutes, seconds) calculated from the time spent by callers when placed in queue and ringing at the agent's desktop before the task is answered divided by the number of tasks answered.

**How ASA is Calculated**

Calculations for ASA differ based on the type of system associated with the reporting object.

### Table 35: Calculating ASA

<table>
<thead>
<tr>
<th>Table.Field</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call_Type_Interval</td>
<td>Call_Type_Interval.AnswerWaitTime / Call_Type_Interval.CallsHandled</td>
</tr>
<tr>
<td>Call_Type_Skill_Group_Interval</td>
<td>Call_Type_Skill_Group_Interval.AnswerWaitTime / Call_Type_Skill_Group_Interval.CallsAnswered</td>
</tr>
</tbody>
</table>
ASA for Agent and Skill group

**Agent.** The ASA is calculated for the agent at the PG level. The internal queuing time is sent to the PG by Packaged CCE when an agent becomes available for the call. The PG adds up the internal queue time, ring time, and network time and adds that into AnswerWaitTime in the agent skill group table. AnswerWaitTime is then divided by the CallsAnswered for the agent.

**Skill group.** The ASA is calculated for the skill group at the PG level.

Consider this example:

- A call is queued at skill group X.
- At Time T, the call is then queued at skill group Y at time T+30 seconds.
- An additional 10 seconds transpire before the call is answered by an agent at skill group Y.

In this case, the internal queuing time will be 40 seconds. This is the total length that the call has been queued even though it was only queued at skill group Y for 10 seconds.

The agent's PG adds the internal queue time, ring time, and network time to create the total AnswerWaitTime for the call and adds it to AnswerWaitTime in the skill group table. AnswerWaitTime is then divided by CallsAnswered within the skill group table to arrive at the ASA for the skill group.

**Precision Queue.** The ASA is calculated for the Precision Queue by summing Skill Groups across PGs which are associated with the Precision Queue.

Reports That Show ASA for Agents

These are some reports that show ASA statistics for Agents and Skill Groups:

- Unified IC Peripheral Skill Group Real Time All Fields
- Unified IC Peripheral Skill Group Historical All Fields

ASA for Call Type

The Call Type ASA is calculated as AnswerWaitTime divided by CallsAnswered.

Call Type ASA is applicable only when calls are translation routed and includes time spent in the Enterprise Queue as well as time spent in the ACD queue.

Reports That Show ASA for Call Type

The following report contains ASA statistics:

- Unified IC Call Type Historical All Fields
Requery

When a call is delivered to an agent that does not answer the call, Packaged CCE manages the call using a method called **requery**.

With requery, supervision of the call occurs from the start. There are timers for the call, including the time it went to an agent, came back, and got routed to another agent.

To use the Requery feature, enable the Target Requery setting in the Queue node. Requery is configured using a time set in CVP and CUCM. The CallType does not change and the call remains in the same script. CVP keeps control of the call during the requery process.

The following steps show a typical requery process.

1. A script connects a call to an agent by sending a connect message to CVP with requery enabled, and the requery timer begins.
2. The agent phone rings.
3. After the requery timeout expires on CVP, the call is pulled back from the agent and a message is sent to the router, indicating that the call is being requeried. The PG marks the agent as unavailable; the router then picks another target according to the routing script and directs CVP to connect the call to the new destination. The destination could be another agent or an IVR to requeue the call.

**Best Practices for Requery**

We advise you to use requery as the preferred method. Follow these guidelines to obtain accurate and useful data from requery.

- Within the routing script, enable the Target Requery option in the routing script. Target Requery is available from the Queue, Queue to Agent, Precision Queue, Label, Select, and RouteSelect nodes.
- Decide how long a call is to ring before being redirected to a new agent or skill group. Consider how requeried calls can affect the service level. In other words, make sure the service level threshold time is properly aligned with the CVP requery timeout.
- Plan to create a separate call type for requery situations. Using a separate call type enables you to report on activity for the requeried call type. Viewing data for this call type helps you determine the number of calls requeried and see how the calls are finally handled.
- If requery occurs from a queue node, the script execution takes the error path of the node that corresponds to the selected agent. In the script's error path of the queue node, change the call type.

**Call Types**

This section provides information about call types.

**Call Type Reports**

Key statistics provided by service and call type reports include:

- Average Speed of Answer (ASA)
- Number of calls received, handled, and abandoned
• How long callers waited in queue
• Number of calls queued for an available agent
• Whether Service Level objectives are being met
• Whether the caller had to be transferred
• Number of callers who heard a busy signal
• Number of calls who encountered an error

Although skill group and agent reports provide many of these same metrics—such as ASA, Avg. Handle Time, abandons, redirects, and calls handled—the call type reports show these metrics in a format that gives a more complete picture of the customer experience and helps you review statistics organized by application.

## Call Types

A call type is a category of incoming call. Based on the call type, the CallRouter selects the routing script that ultimately sends the call to an appropriate agent. Each call type has a schedule that determines which routing script or scripts are active for that call type at any time.

Call types are also the highest level reporting entity and are peripheral-independent.

There are two classes of call types: voice (phone calls) and non-voice (for example, email and text chat).

- Voice call types are categorized initially by the dialed number (DN) and, optionally, by the the calling line ID (CLID).
- Non-voice call types are categorized initially by the Script Type Selector and, optionally, Application String 1 and Application String 2.

Creating call types that correlate to the type of service the caller wants and changing call types during the script result in reporting statistics that reflect the customer's experience.

---

**Note**

Configuring a separate call type for each type of call treatment that you offer can eliminate the need for most custom reporting.

---

**Note**

The software allows routing that can offer calls simultaneously to multiple skill groups. The Call_Type_Skill_Group_Interval table records details for call types associated with specific skill groups. Reports generated from this table show how scripts routed the calls, as well as the calls associated with a skill group that encountered errors, that abandoned, that were redirected on no answer, and so forth.

---

### Best Practices for Call Types

Consider the call types you need to create to meet your reporting needs and configure a separate call type for each type of call treatment that you want to offer.

Based on the deployment model, scripting, queuing, and on whether or not calls are translation-routed, you can define call types to:
• Provide enterprise-wide routing statistics for the call center, such as the number of calls to be routed to
different peripherals or the number of calls that encounter routing errors.

• Group calls to report on certain types of activity that occur within the contact center. For example, you
might create separate call types for calls that redirect on no answer or calls that are transferred to another
agent.

• Report on statistics for a self-service CVP application.

**Do you want to configure a separate call type associated with call transfers and conferences?**

Doing so enables you to direct the transfer to a different routing script.

**Do you plan to report on individual transactions within Network CVP Self-Service or Information
Gathering applications?**

If so, you might configure a separate call type for each transaction.

**Do you want to separate Information Gathering CVP metrics from queue metrics?**

If so, you might configure a separate call type for queuing.

**Do you want to configure a separate call type associated with RONA situations?**

This enables you to direct calls that Ring No Answer to a routing script designed for this situation and to
report on this Redirection on No Answer call type to see how calls that redirect on no answer are eventually
handled.

**Do you want to configure a separate call type associated with the Supervisor and Emergency Assist
script for each agent team?**

This enables you to direct the assistance request to the Supervisor and Emergency Assist routing script which
can assign the request to the primary or secondary supervisor for that agent's team. You can use Call Type
reports to view data for supervisor assistance calls.

**Do you want to determine the Service Level for call types?**

Service Level indicates how well you are meeting your goal for answering calls.

You can configure the Service Level setting individually for each call type or set a global Service Level for
all call types.

**Do you want to configure abandoned short calls to filter out calls that abandon very quickly?**

If you want to use abandoned short calls, configure the call type Abandon Wait Time in Configuration Manager.
Calls that abandon within the Abandon Wait Time are reported as short calls.

If you do not want to use abandoned short calls, leave the Abandon Wait Time field blank.

**Do you want to define “bucket intervals” for reporting on answered and abandoned calls for the call
type?**

These “bucket intervals” appear in call type reports that display the number of calls answered and abandoned
for each interval and are useful for monitoring when calls are abandoning or being answered.

**Changing Call Types**

Call type can be changed throughout the life of a call to direct the call to a new routing script or to gather
report metrics for different legs or transactions.

Reasons for changing the call type within a routing script include the following:
In a self-service CVP script, you might change the call type at specific points in the script to indicate that a transaction has been completed.

For example, if the customer is calling a bank and successfully checks an account balance using a Self-Service script, you might want to change the call type to indicate that the account balance transaction has completed and a new transaction has begun. In this case, you would create a call type for each transaction on which you want to report.

You might change the call type when a call enters a queue at the end of an Information Gathering CVP script in order to separate Information Gathering and queuing metrics. In this case, you would create call types associated with the Information Gathering applications and call types associated with queuing.

The service level threshold timer at the call type starts as soon as the call enters the call type that has a service level defined. When the service level timer expires, the service level is applied to the current call type associated with the call.

If a call type is changed using the Requalify or Call Type nodes, then the service threshold timer is reset.

Service levels are defined only for call types associated with scripts that use the Queue To and LAA Select nodes.

---

**Note**

The call type changes depending on the following factors:

- TCD record for each leg of the call is associated with the last call type.
- When you use the Capture (CAP) micro-application, different TCD rows with multiple call types are populated.
- When a call is abandoned in a queue, the call type is not changed.

---

**Call Type Reporting**

The use of Call Type reports is based on the business need for your enterprise and is determined by how you plan to use the functionality provided by Packaged CCE software.

Call Type reporting provides full customer experience in Unified CC, similar to Service reporting in Packaged CCE.

Call Type reports can be used for the following purposes:

- Calls answered by agents
- Calls abandoned at the CVP
- Calls that abandon while en-route to an agent or while being offered to an agent's phone
- Short calls
- Calls that are given the busy, ring, default-routed or network-routed treatment
- Calls that go to another call type within a routing script using the Call Type or Requalify node
- Calls that abandon en-route to the VRU
- Calls that have a bad label
- Calls that re-route on no answer from the agent's phone
• Calls that terminate the script using the Label node to a non-monitored device, such as voice mail
• Cradle-to-grave reporting for call-handling statistics when calls are translation routed
• Reporting on calls grouped for the purposes of global call treatment
• Providing enterprise-wide routing statistics for your call center, such as the number of calls routed to different peripherals and the number of calls that encountered routing errors
• Reporting on statistics for a self-service CVP application
• Reporting on certain activities such as calls that are transferred, provided Call Types are configured for those activities

How Calls Offered is Calculated for Call Type

The completed state for CallsOffered at the Call Type is calculated using these fields from the Call_Type_Interval table:

• CallsHandled
• ErrorCount
• ICRDefaultRouted
• NetworkDefaultRouted
• ReturnBusy
• ReturnRing
• NetworkAnnouncement
• OverflowOut
• IncompleteCalls
• ShortCalls
• CallsRoutedNonAgent
• CallsRONA
• ReturnRelease
• AgentErrorCount
• TotalCallsAband

How Call Errors Affect Call Type Reporting

The way call errors increment the database depends on the following conditions:

• Calls that abandon en-route to the CCE/CVP scripts are calls that abandon in the network while they are being sent to the VRU. An example of this is if a call abandons while it is being sent to the VRU from a CTI Route point in Communications Manager. These calls increment the ErrorCount column in the Call_Type tables.
If the caller abandons within the Abandon Wait Time set for the call type, calls that abandon en-route to CVP might be counted as short calls, instead of as errors. See the next section for more information on abandoned short calls

- Calls that abandon en-route to agents are calls that encounter an error when the call is at the agent desktop. This call is counted as part of the AgentErrorCount in the Call_Type tables.

The Calls Error field in call type reports is a calculated field that combines both error columns. For example, the Calls Error field in the Call Type Historical All Fields report is derived from Call_Type_Interval.IncompleteCalls + Call_Type_Interval.AgentErrorCount.

**How Calls with a Bad Label Affect Call Type Reporting**

A bad label refers to an incorrectly configured label or missing label. It is always good practice to define a default label, so that calls that do encounter an incorrectly configured label can at least go to the default label and get handled as well as get accounted for in the call type report.

Labels might be configured incorrectly in the following ways:

- The label specified in the script node might not exist on the routing client: In this case, label specified in the script node might not exist on the routing client.

- The label points to the wrong agent: In this case, the pre-call message is sent to one agent, but the actual call is sent to a different agent. This call is reported as an incomplete call.

If the node does not define a label, the call encounters error conditions and is reported as an error.

**How Calls that Experience CVP Ring No Answer Affect Call Type Reporting**

You can view a count of CVP Ring No Answer calls in agent and skill group reports.

The CVP Ring No Answer feature ensures that when an agent does not answer a call, the call is taken away from the agent after a specified number of seconds and re-assigned to another agent or requeued. Ring No Answer is also used to change the agent state to Not Ready when a call is rerouted from the agent's phone. The CVP Ring No Answer feature makes the agent unavailable for routing requests. When the Packaged CCE CVP Ring No Answer timeout expires, the call is re-queried for routing to a different skill group or agent.

---

**Note**

The CVP Ring No Answer timeout must be less than 30 seconds because the Central Controller waits up to 30 seconds for a response from the CVP. If the response is not received within 30 seconds, the call fails.

You can configure the routing script to handle Ring No Answer situations in two ways: the script can change the call type when the calls is requeried, or the script can continue to use the same call type.

The manner in which you script for Ring No Answer affects the report data that you see, as follows:

- If you change the call type, CallsOffered, CallsRequeried, and OverflowOut is updated for the initial call type. CallsOffered and fields related to the completion of the call, such as CallsHandled, are incremented for the second call type.

Using two call types enables you to identify Ring No Answer occurrences in call type reports. For example, if you create a specific call type for use in Ring No Answer situations, then you can see whether calls are redirecting by monitoring the calls offered to that call type. You can also see whether the Flow Out field is incremented for other call types.
• If you do not change the call type, CallsOffered and fields related to the completion of the call, such as CallsHandled, are incremented. FlowOut is not incremented. You will not be able to tell without looking at agent or skill group reports whether calls are redirecting on no answer. (You could write a custom report to see values for CallsRequeried.)

**Note**

Because the Unified CVP application performs a query to redirect the call to a different agent or skill group instead of branching to another script, the CallsRONA field is not incremented for the call type.

How Calls that Terminate Label Node and Route to Non-Monitored Devices Affect Reporting

The Label node is used to divert a call to voice mail or web attendant or some other device that is not monitored by Unified ICM/CC because of digits collected by the caller during a voice menu or due to some other conditions. These calls are counted as RoutedNonAgent and appear in the “Other” column of call type reports.

**Call Type Reports**

The following reports display call type data:

- Unified IC Call Type Abandon/Answer Distribution Historical
- Unified IC Call Type Historical All Fields
- Unified IC Call Type Real Time All Fields

**Reporting on Calls**

**Bucket Intervals**

Bucket Intervals allow you to track data for calls abandoned or answered within specific time increments (for example, between 0 and 8 seconds, or under 60 seconds).

Bucket Intervals are associated with the following:

- Call Types
- Skill Groups
- Precision Queues

They can be set for the system as a whole and for individual call types. Local settings override those set at the system level.

Service Level tells you what percentage of calls are being answered within a certain time, but does not tell you how closely to the Service Level calls are being answered or abandoned. Bucket intervals provide additional insight into how long callers are waiting before their calls are answered or before they abandon.

For example, if your Service Level is two minutes, you might want to set up intervals for 30 seconds, one minute, 90 seconds, 120 seconds, 180 seconds, 210 seconds, and 240 seconds. Using these intervals, you can see whether calls are being answered in the thirty seconds after the Service Level Threshold of 180 seconds or if most are waiting a full minute longer to be answered.
The intervals also give you insight into how long callers are willing to wait before abandoning. Perhaps many callers do not abandon until two minutes past the Service Level. This might indicate that your Service Level goal can be modified.

To avoid reporting inconsistencies, modify Bucket Interval settings only at specific time boundaries (that is, end of day, week, or month). Ensure that no one is running reports for the intervals that you are changing when you modify the boundaries.

Packaged CCE ships with a single System default Bucket Interval whose boundaries (increments) are: 8, 30, 60, 90, 120, 180, 300, 600, and 1200 (in seconds).

Bucket Interval Reports

The following reports display Bucket Interval data:

- Unified Intelligence Center: Call Type Abandon/Answer Distribution Historical
- Unified Intelligence Center: Call Type Historical All Fields Report
- Skill Group Abandon/Answer Distribution
- Precision Queue Abandon/Answer Distribution

Service Levels

Service levels help you set and measure goals for answering calls. Service levels are configurable; you can define them in different ways, depending on the kind of information you want them to provide.

About Service Levels

All calls that have a service level event within a specified period are considered to be service level calls offered for that period. This designation differs from a call's offered value, which counts each call at the time it is first offered to the service.

Note

Service level is not affected for calls that are neither answered nor abandoned within the service level time. For example, calls that encounter an error condition or are sent to non-monitored devices (using the label node) within the service level threshold do not affect the service level.

Two important configuration parameters contribute to the calculation of service level:

- Service level threshold - the number of seconds you set as a goal to treat a call. To calculate the service level for a period of time, Packaged CCE software determines the number of calls that have had a service level event within that interval.
- Service level type - the manner in which calls that abandon affect the service level.

Service Level Threshold

The service level threshold is the number of seconds you set as a goal for connecting a call with an agent.
For example, your goal might be to answer 80% of calls within two minutes. In this case, you would set the service level threshold to 120 seconds. Reports show you the percentage of calls that are answered within that time threshold, enabling you to see whether you are meeting your goal.

A service level threshold of 0 seconds means that no service level event will be set for the call; it will not be treated as a service level call.

**Service Level Type**

Service level type determines how calls that abandon before the service level threshold impact the service level calculation.

**Service level type is presented as three options** in the Configuration Manager: positively, negatively, or not at all.

- **Abandoned calls positively impact**
  
  Some contact centers want abandoned calls to positively impact the service level. These contact centers consider a call abandoned within the service level threshold time a treated call. Abandoned calls are considered to positively impact the service level.

- **Abandoned calls negatively impact**
  
  Other contact centers might consider only those calls answered within the service level threshold time as treated calls. For these contact centers, the service level is detrimentally affected by calls that abandon within the service level time. Abandoned calls negatively impact the service level.

- **Ignore abandoned calls**
  
  Others might choose to exclude the abandoned calls from the service level calculation (abandoned calls ignored).

The calculations for service level are based on the service level type defined for the service level configuration. They are described in the following table.

**Table 36: Formulas for service level type**

<table>
<thead>
<tr>
<th>Service level type</th>
<th>Formula used to determine service level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore abandoned calls</td>
<td>For Call Type and service: ServiceLevelCalls / (ServiceLevelCallsOffered – ServiceLevelAband)</td>
</tr>
<tr>
<td>Negative impact of abandoned calls</td>
<td>For Call Type and service: ServiceLevelCalls / (ServiceLevelCallsOffered)</td>
</tr>
<tr>
<td>Positive impact of abandoned calls</td>
<td>For Call Type and service (ServiceLevelCalls + ServiceLevelAband) / (ServiceLevelCallsOffered)</td>
</tr>
</tbody>
</table>

For an example of how service level type is calculated, consider the following call counts:

- Answered within service level threshold (ServiceLevelCalls) = 70 calls
- Abandoned within service level threshold (ServiceLevelAband) = 10 calls
• Exceeded service level threshold \((\text{ServiceLevelCallsOffered} - (\text{ServiceLevelCalls} + \text{ServiceLevelAband}))\) = 20 calls

• Total service level events \((\text{ServiceLevelCallsOffered})\) = 100 calls

For these call counts, the service level is calculated for each type as follows:

### Table 37: Calculations for service level

<table>
<thead>
<tr>
<th>For this service level type:</th>
<th>The service level calculation is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandoned calls ignored</td>
<td>(\frac{70}{(100-10)} = 77%)</td>
</tr>
<tr>
<td>Abandoned calls negatively impact</td>
<td>(\frac{70}{100} = 70%)</td>
</tr>
<tr>
<td>Abandoned calls positively impact</td>
<td>(\frac{(70 + 10)}{100} = 80%)</td>
</tr>
</tbody>
</table>

Leave the Abandon Wait Time field blank if you prefer to not track abandoned calls.

**Service Level at Call Type**

For measuring overall customer experience, the call type provides the most insight into overall call treatment and how callers are experiencing the system.

The service level threshold timer at the call type starts as soon as the call enters the call type that has a service level defined. When the service level timer expires, the service level is applied to the current call type associated with the call.

Only call types that are associated with scripts that use the Queue To nodes define service levels. If a call type is changed using the Requalify or Call Type nodes, then the service threshold timer is reset.

There are four *service level events* that can occur for the call type:

- The call is answered by an agent before the service level threshold expires. In this case, the ServiceLevelsCallsOffered and ServiceLevelCalls database fields are incremented.

- The call abandons while in the CVP or at the agent's phone before the service level threshold expires. In this case, the ServiceLevelCallsOffered and ServiceLevelAband database fields are incremented.

- The call redirects on no answer before the service level threshold expires. In this case, the ServiceLevelCallsOffered database field is incremented.

- The service level threshold timer expires. Example: the call reaches the service level threshold without being answered by an agent or abandoned. In this case, the ServiceLevelCallsOffered database field is incremented.

If calls encounter an error before the Service level threshold expires, the ServiceLevelError database field is incremented, but ServiceLevelOffered is not incremented. If the call encounters an error after the service level threshold expires, ServiceLevelOffered is incremented.

To exclude errors from your service level calculation:

- Adjust the ServiceLevelCallsOffered by excluding error calls.

\[
\text{Adjusted SL Offered calls} = \text{SL Offered calls} - (\text{Total Error calls} - \text{ServiceLevelError})
\]
In this example, abandoned calls have a negative impact.

\[
\text{Service Level} = \frac{\text{Service Level Calls Offered}}{\text{(Service Level Calls Offered}} - \text{Agent Error Count} + \text{Error Count} - \text{Service Level Error})
\]

**Service Level at Skill Group**

At the skill group level, the service level metric is useful for monitoring agent and skill group performance. The service level threshold timer at the skill group starts as soon as the call is queued to a skill group.

You can use Precision Queues to enhance or replace skill groups. For more information about Precision Queues, see Create Precision Queue.

**Note**

By default, the service level threshold for a skill group is set to the default value of the agent peripheral.

There are five service level events that can occur for the skill group:

- The call is answered by an agent before the service level threshold expires. In this case, the Service Level Calls Offered and Service Level Calls database fields are incremented for the skill group that answered the call. If the call is queued to more than one skill group, then the Service Level Calls Offered and Service Level Calls Dequeued database fields are incremented for the other skill groups.

- The call is dequeued from a skill group before the service level threshold expires. In this case, the Service Level Calls Offered and Service Level Calls Dequeued database fields are incremented. Calls may be dequeued using the Cancel Queue node, when they are de-queued from the skill group to be routed to a different skill group.

- The call abandons while in the VRU (queue) or at the agent's phone before the service level threshold expires. In this case, the Service Level Calls Offered and Service Level Aband database fields are incremented.

- The call redirects on no answer before the service level threshold expires. In this case, the Service Level Calls Offered database field is incremented.

- The service level threshold timer expires. Example: the call reaches the service level threshold without being answered by an agent or abandoned. In this case, the Service Level Calls Offered database field is incremented.

Calls can queue to more than one skill group depending on your scripting, and service level metrics are updated for each skill group to which a single call queues.

It is important to understand how service levels are impacted in such cases.

- If a call is queued to more than one skill group and then the call is answered before the service level threshold expires, Service Level Calls Offered and Service Level Calls database fields are incremented for the skill group that answered the call. For the other skill groups, Service Level Calls Offered and Service Level Calls Dequeued database fields are incremented.

- If a call is queued to more than one skill group and the call abandons in queue before the service level threshold expires, Service Level Calls Offered and Service Level Calls Aband database fields are incremented for all the skill groups. This result will have a negative or positive impact on service levels in all the skill groups depending on how you have decided to treat abandoned calls for service level calculations in your configuration for the individual skill groups.
• If a call is queued to more than one skill group and the call abandons in queue after the service level threshold expires, ServiceLevelsCallsOffered database field is incremented for all the skill groups. This result will adversely affect your service level.

• If a call is queued to more than one skill group and the call abandons after it was routed to a skill group (example: abandon while ringing at the agent) before the Service level threshold expires, ServiceLevelCallsOffered and ServiceLevelCallsAband database fields are incremented for the skill group that had the abandon, while other skill groups have ServiceLevelCallsOffered and ServiceLevelCallsDequeued database fields incremented.

If you want to remove errors from ServiceLevelCallsOffered, you can use this formula in a custom report: ServiceLevelCallsOffered – (Errors – SLErrors).

**Best Practices for Service Levels**

Consider these guidelines when configuring and scripting service level:

• Service level time begins as soon as the call enters a Call Type. Set up Call Type scripts specifically to collect queue and agent statistics such that service level time begins once a call is queued to a skill group. Define service levels only for Call Types that point to a script that includes a Queue to Skill Group node.

• Set up one Call Type to collect statistics before the queue (that is, the initial Call Type designated for the script via Call Type mapping).

• Set up other Call Types used specifically to collect queue and agent statistics.

• In your routing scripts, include the Requalify or Call Type nodes to submit the call to the Call Type used to collect queuing information.

• Skill group/precision queue and service level metrics are updated for each skill group/precision queue to which a single call queues. Service Levels could be adversely affected if calls abandon within or outside the service level threshold in such cases. Consider queuing to a single skill group/precision queue if you include abandons in your Service Level calculations and do not want abandons to affect Service Levels adversely.

If you follow these guidelines, the first Call Type to which the call was initially mapped will gather statistics before the call is queued to the skill group. The script will then pass the call to the Call Type set up specifically to collect information after the call is queued to the skill group/precision queue.

**Service Level Relationships: MRD, Skill Groups, and Precision Queues**

Service level setting for MRDs, skill groups, and Precision Queues are hierarchical and are interpreted as follows:

The MRD is the highest level. The default settings for the MRD are Service level threshold = 30 seconds and Service level type = Ignore Abandoned Calls. Ignore Abandoned Calls is the only value, and it is protected. Skill groups and Precision Queues inherit their values from the MRD, unless they are explicitly set. The default settings for both skill groups and Precision Queues are taken from the MRD, but you can override them.

**Short Calls, Abandoned Calls, and Overflow Calls**

This section provides information about short calls, abandoned calls, and overflow calls.
**Short Calls**

A short call is a call that is either abandoned very quickly or answered and terminated very quickly. By defining what you believe to be a short call, you can filter out from reporting metrics those calls that did not stay in the system long enough to be considered and counted as events.

The Abandoned Call Wait timer defines the threshold under which the abandoned call will not be counted. If the abandoned threshold is lower than the service level threshold, the call will not affect the service level. If call wait time is higher than this threshold, the call is counted as Offered.

The Answered Short Call threshold defines the time under which the call will not be counted as answered and will not impact agent performance.

If you plan to use short calls to filter out false abandon or to detect when calls are answered and terminated too quickly to be considered handled, consider the following:

- You can configure abandoned short calls for the peripheral. These calls are tracked for the services that are configured for that peripheral.
- You can choose not to count any abandoned calls as short calls regardless of how quickly they abandon.
- You can choose how abandoned calls affect the service level—negatively, positively, or not at all.
- You cannot configure answered short calls for call type.
- You can choose not to count any answered calls as short calls regardless of how quickly they terminate.

---

**Note**

The concept of short calls applies to the Voice media class only.

To access these short call capabilities, refer to the following section.

**Using Short Calls as Filters and Detection Devices**

Perform these general steps:

1. Access the AW (DataServer).
2. Go into the **Configuration Manager > Tools > Explorer Tools > PG Explorer**.
3. Click **Retrieve**.
4. Expand **Generic PG**.
5. Click **CUCM_PG#**.

On the right side of the screen there is a group of tabs:

- **Peripheral**
- **Advanced**
- **Agent Distribution**
- **Peripheral Monitor**
- **Default route**
- **Routing client**
Abandoned Short Calls

A call is considered abandoned if it abandons after the value set for the Abandon Call Wait time threshold. This is set globally.

If the call abandons before the Abandon Call Wait Time threshold, the call is reported as a short call.

Abandoned short calls affect reporting because they update the CallsOffered field but not the CallsAbandon field.

Answered Short Calls

Answered short calls reflect when a caller hangs up quickly if no agent answers the phone.

Answered short calls are reported for skill groups and agent skill groups.

The short call timer starts when the agent answers the call, and the CallsAnswered metric is updated for these calls.

The ShortCalls fields within the Skill_Group_Interval and Agent_Skill_Group_Interval tables are incremented if the Talk Time is less than the Answered short call threshold configured for the peripheral. The call is reported both as handled and as a short call.

If auto-answer is enabled for the agent, and if there are a high number of short calls within a certain interval, reporting on short calls can be used to determine which agents were not at their stations when a call was automatically answered. This conclusion assumes that the caller hangs up quickly when there is no agent on the phone.

Short Call Reports

A number of All Fields Reports contain a Short Tasks column to enable you to track calls that are offered but are neither handled nor abandoned.

The following reports display operational information on short calls:

- Unified Intelligence Center Agent Historical All Fields Report
- Unified Intelligence Center Call Type Historical All Fields Report
- Unified Intelligence Center Agent Skill Group Historical All Fields
- Precision Queue Historical All fields

Abandoned Calls

A call is considered abandoned if the caller hangs up before being connected to an agent. This includes situations where the caller hangs up while queued and waiting at the CVP. A high number of abandoned calls might be an indication that callers are waiting in the queue for too long.

Service reports provide cumulative statistics for all abandoned calls. Call Type reports provide additional visibility on where calls are abandoning.
If a call abandons before the Abandon Call Wait Time threshold, it is considered a short call. For example, if you configure the abandoned call wait time for 10 seconds, and a caller disconnects at nine seconds, that call is a short call—it is not considered offered or abandoned.

To configure the Abandon Call Wait Time, access the Peripheral tab on the xyz screen. For more information about this procedure, go to Short Calls, on page 268.

How Abandoned Calls Affect Reporting

There are three types of abandon metrics: abandon at the VRU (prompt or self service), abandon in queue, and abandon at the agent.

Packaged CCE/CC tracks the abandon counts for each of these abandon types separately. The time spent by these abandoned calls before abandoning is also tracked.

The value represented by the Aban column on the Call Type reports provides total abandon count for the call type, which includes calls that abandoned while at the VRU (prompting or self service), calls that abandon in queue, and calls that abandoned while ringing at the agent's phone or en route to the agent's phone. This value is derived from the TotalCallsAband database field.

Reports also provide average time spent by these abandoned calls in the Avg Aban Delay Time field. This field represents the average delay time of all abandoned calls that ended in this call type during the current interval. This value is derived from Call_Type_Interval.CallDelayAbandTime / Call_Type_Interval.TotalCallsAband.

To separate information gathering and queuing statistics, you can also determine the time spent by a call only in the call type where the call abandoned. This value is tracked in the CTDelayTotalAbanTime database field. It includes only the time spent in the call type where the call abandoned and not all call types.

Consider this example:

- A call spends 30 seconds in the information gathering call type, "Info_Call_Type".
- The script then changes the call type to the queuing call type—say Queue_Call_Type—and the call is queued.
- After 15 seconds waiting in queue the call is abandoned.

In this case the total time spent by the call before abandoning will be 45 seconds. However, the time spent by the call in the "Queue_Call_Type" where the call abandoned will be 15 seconds. The call type statistics for the "Queue_Call_Type" will be updated as follows:

Queue_Call_Type

- CallDelayAbandTime = 45 seconds
- CTDelayTotalAbanTime = 15 seconds.

You could write custom reports to able to report on the different abandons and the time spent by these abandons. To determine the counts and the time associated with the abandoned calls, for calls in the script, or at the VRU (prompt or self service), subtract Agent Abandons and Queue Abandons from Total Abandons.
How Abandoned Short Calls Affect Reporting

A short call at the call type is a call that abandons within the call type's Abandon Wait Time threshold. By defining what you believe to be a short call, you can filter out those calls that you believe did not stay in the system long enough to be counted as a real call. You can define short calls for call types and services.

Note

Short calls are configured globally for all call types.

The short call timer starts as soon as the route request is received for the call. The CallsOffered field is updated when the route request is received. If the call abandons within the Abandon Wait Time threshold, the ShortCalls field is updated, but the number of calls abandoned is not updated. Since the call type is the highest level reporting entity, calls that abandon at the VRU or at the agent's phone can also be considered short calls at the call type if they abandon within the call type's Abandon Wait Time threshold.

If you do not want to count any abandoned calls as short calls regardless of how quickly they abandon, you can disable abandoned short calls by leaving the Abandon Wait Time field for the call type blank.

Abandoned Call Reports

The following reports display Abandon statistics for call types and services:

- CallType Abandon/Answer Distribution Historical
- CallType Historical All Fields
- CallType Real Time
- CallType SkillGroup Historical All Fields

Transfers and Conferencing

Voice calls can be transferred or conferenced. Non-voice tasks, such as email, chat, and Blended Collaboration tasks cannot be transferred and conferenced.

Transfers are blind. In a blind transfer, an agent transfers a call to another agent without first ensuring that another agent is available.

About Transfers and Conferences

Packaged CCE supports transfers and conferences directly to agents and also to skill groups. Follow these guidelines to obtain accurate and useful data from transfers and conferences:

- Configure the dialed numbers with associated route points for transfer and conference to agents and skill groups.
- Plan to create a separate script for transfers that use the dialed numbers you configured. In the initial script, change the call type such that when the call is transferred, it is directed to the transfer script. Having a separate script allows you to track data across call types and skill groups, instead of the agent's default skill group.
The agent can manually transfer the call to another agent or can conference in another agent by dialing that agent's extension directly on the ACD.

The agent can use the ACD number to access a routing script on the ACD or can optionally post-route the call to Packaged CCE.

The latter method is preferable, since Packaged CCE will be able to track the transferred calls and how they are handled across the enterprise. This also provides the ability to transfer the call to another ACD site.

It is best to discourage direct agent-to-agent transfers, as one has to rely on the ACD for skill group and service assignment for the purposes of reporting. In situations where you are handling transfers and conferences on the ACD itself, use the ACD number to control how transfers are reported.

Note Packaged CCE will be unaware of a direct transfer if the agent extension is not configured.

If you are planning to provide for transfers and conference calls using post-routing, follow these guidelines to obtain accurate and useful data:

- Consider creating separate scripts for transfer and conference scenarios if you are post-routing all transfers and conferences.
- Plan to create a separate script for transfers on Packaged CCE that uses the dialed numbers you configured. Having a separate script on the ACD or on Packaged CCE allows you to track data across all known skill groups.

Note If you do not do this, the skill group affected by reporting statistics is unknown, and results are unpredictable.

**Configuration and Scripting for Transfers and Conferences**

Follow these guidelines when configuring and scripting for transfers and conferences to skill groups:

- Configure dialed numbers.
- Create new call types or identify existing call types and associate the call types with the Dialed Numbers.
- Create a routing script for transferring to skill groups that includes a Queue to Skill Group node. This script ensures that transferred and conferenced calls are queued to the correct skill group.
- Associate the call type with the routing script.

**Operational Reporting**

Operational reporting includes reports on translation routes, trunks, and trunk groups.
Trunks and Trunk Groups

Every peripheral has one or more associated trunk groups, with each trunk group containing one or more physical trunks.

You can report on data such as the number of trunks in service, number of trunks idle, and the time during which all trunks in a trunk group were simultaneously busy (All Trunks Busy).

The following report contains operational information on trunk groups:

- Unified Intelligence Center IVR Ports Performance Historical Report

IVR/VRU Self-Service

This section provides information about self service using Customer Voice Portal (CVP) for Interactive Voice Response (IVR).

About CVP

CVP, a voice response unit, also called an Interactive Voice Response Unit (IVR), is a telecommunications device that plays recorded announcements and responds to caller-entered touch-tone digits. CVP can also be equipped with Automatic Speech Recognition (ASR) or Text-to-Speech (TTS) capabilities.

In Packaged CCE terms, CVP is a device that corresponds to a peripheral and is integrated by means of a PG.

Uses for CVP

Your enterprise might implement one or more types of CVP applications to provide initial call treatment and enterprise queuing.

- In Self-Service applications, the customer can obtain information through a series of CVP prompts, and the entire transaction occurs within the CVP. For example, if the customer calls a bank, the Self-Service application might prompt the user for an account number and password and then provide abilities to check account balance, review recent payments, modify PIN numbers, and so forth.

- In Information Gathering applications, the CVP prompts the caller for certain information, such as which department the caller wants to reach, and then uses the information in the routing decision and might pass the information to the agent desktop.

- CVP is also used to enterprise-queue calls while a customer waits for an available agent. During queuing, the CVP might be configured to play music on hold or perform a CVP application.

CVP Application Reporting

You can use a CVP for a number of different purposes, including queuing, customer self-service, and information gathering.

Impact of CVP type on report data

The types of CVP applications that you use in your enterprise determine what report data you should monitor.

For example:
• If your CVP performs queuing only, you might want to see how long callers waited in queue and the number of callers who abandoned while queued.

• If your CVP is used for Self-Service, you might want to see how many successful transactions occurred in the Self-Service application and whether the caller was transferred to an agent from the application.

• If you are using an Information Gathering application, you might want to see how many callers opted out of the digit collection to be transferred directly to an agent.

**Self-Service, Information Gathering, and Queuing CVP Applications**

Information gathering CVP applications are used to decide what skill group to queue the call to by walking the caller through a series of voice prompts. The Caller Entered Digits (CED) are passed back from the CVP to be used within the routing script, to decide the optimal skill group to answer the call.

You must be able to determine the following from an IVR service used for information gathering:

- How many calls traversed the application
- How long each call remained in the information gathering application
- How many calls disconnected before being routed to an agent
- How many calls were eventually routed to agents

If there is IVR treatment before the call is queued, best practice is to change the call type just before queue node. Changing the call type will reset the service level timer and not include the IVR time. If you don't change call types prior to the Queue to Skill Group node, the IVR treatment time will be included in the calculation of service level, having a negative impact on your service level. See the following figure.

**Figure 62: Changing a call type prior to the Queue to Skill Group node**

The following illustration shows how a call moves from the Information Gathering application to the queuing applications.
In this example, 20 seconds will be used to calculate ASA and decide the service level instead of 50 seconds (30 + 20 seconds).

**Figure 63: Call Type data for calls that abandon after Call Type is changed**

If the call abandons before being requalified to the Call Type that handles queuing, the Call Abandon Wait time is not reset. Therefore, the Abandon Wait time for the information gathering Call Type starts when the call enters the first Call Type, and ends when the call abandons, as illustrated below:

**Figure 64: Call Type for calls that abandon before Call Type is changed**

The following table illustrates how some basic metrics are defined at the Call Type and the IVR Service.

**Table 38: Self-service and information gathering application metrics**

<table>
<thead>
<tr>
<th>Report metric</th>
<th>Call type</th>
<th>VRU service</th>
<th>Skill group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandon Wait Time</td>
<td>Starts when a call first enters a Call Type and ends when it abandons.</td>
<td>Starts when the call enters the service.</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Average Speed of Answer (ASA)</td>
<td>Starts at the first Queue to Skill Group node in the routing script.</td>
<td>Starts at the first Queue to Skill Group node in the routing script.</td>
<td></td>
</tr>
</tbody>
</table>
### Monitoring Self-Service and Information Gathering Application Progress

You might determine the effectiveness of a Self-Service application in several ways:

- Monitoring the effectiveness of the application as a whole. For example, you might only want to monitor whether a customer's need was satisfied through the CVP application and that the caller did not need to be transferred to an agent.

- Monitoring the effectiveness of individual transactions within the application. For example, in a banking application a customer might have the ability to perform multiple transactions, such as account lookup, obtaining balance information, and learning about recent payments. You might want to see which of these transactions was used and whether the caller successfully completed the transaction.

- Monitoring failure cases in which a system error, such as a failed database lookup, caused the caller to be transferred by an agent instead of continuing through the CVP application.

Similarly, you might determine the effectiveness of an Information Gathering application in several ways:

- Monitoring whether the caller used the system prompts to be routed to an appropriate resource or used a failout path, such as pressing "0", to be routed directly to an agent.

- Monitoring failure cases in which system errors, such as a failed database lookup, caused the caller to be transferred to an agent instead of continuing through the digit collection prompts for more appropriate routing.

### VRU Progress Variable

CVP applications are unique among call center applications in that the reports need to describe events that are application-specific, using application-specific terminology. Such reports vary widely from one customer to another and from one VRU application to another. Some customers only need to know how many CVP calls were satisfactorily handled by their CVP applications; others would like to track the usage and success rates of specific transactions within their CVP applications. Still others are interested in the actual series of activities performed by a specific caller and even the content of data collected or delivered.

The definition of a successfully handled call varies as well. In some cases a single transaction constitutes success. In others, each individual transaction has its own success criteria, and there might be several gradations of success. For example, some customers want to differentiate between calls in which no transactions were completed before transferring to an agent and calls in which one or more transactions were completed before transferring to an agent.

The system provides tools that customers can use as needed to meet these requirements:

- The VRUProgress variable in the CallRouter call object
- Seven VRUProgress rollup buckets in the Call_Type_Interval table
- Reporting templates for reporting on Call_Type_Interval VRUProgress statistics
- A VRUProgress field in the Route_Call_Detail table

The following table describes the VRUProgress variables that you can use in your CVP script applications and how they map to report columns.

These VRUProgress variables map to columns that appear in CVP Activity reports, enabling you to see how many calls were counted for each variable per Call Type. You can use this data to modify applications if needed. For example, if you see that many callers are experiencing error conditions that cause a forced transfer you could correct the function of that node. If you see that many callers are opting to be transferred to an agent before being handled by the application, you might want to add functionality to the application.

**Table 39: VRUProgress variable**

<table>
<thead>
<tr>
<th>Variable Setting in Script</th>
<th>Show in Reports as</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Not a VRU call—does not appear in reports</td>
<td>Indicates that this call is not a VRU call. It is the default value.</td>
</tr>
<tr>
<td>1</td>
<td>VRU Unhandled</td>
<td>Indicates that the caller's needs have not been met at this point in the application.</td>
</tr>
<tr>
<td>2</td>
<td>VRU Handled</td>
<td>Indicates that the caller's needs have been met by this point in the application. For example, the caller successfully received an account balance.</td>
</tr>
<tr>
<td>3</td>
<td>VRU Assisted</td>
<td>Indicates that this call was transferred to an agent after the caller's needs were met with the application. For example, the caller successfully received account information and then requested to speak to an agent for a different reason or for additional information not available through automatic means.</td>
</tr>
<tr>
<td>4</td>
<td>VRU Opt Out Unhandled</td>
<td>Indicates that the call was transferred to an agent at the caller's request before the caller's needs were met by the application. For example, the caller pressed &quot;0&quot; to be transferred to an agent before performing automated transactions or while in the process of completing a transaction.</td>
</tr>
<tr>
<td>5</td>
<td>VRU Scripted Transfer</td>
<td>Indicates that the call was transferred to an agent as part of the application design. For example, after the caller checked an account balance the application transferred the caller to agent to discuss new account options. Another example is that after a caller entered digits to request a particular type of service the call was transferred to an available agent to handle the request.</td>
</tr>
<tr>
<td>Variable Setting in Script</td>
<td>Show in Reports as</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>6</td>
<td>VRU Forced Transfer</td>
<td>Indicates that the caller was transferred to an agent because of a system error. For example, a failure at a particular node in the application could lead to the call being transferred to the agent.</td>
</tr>
<tr>
<td>7</td>
<td>VRU Other</td>
<td>Indicates that the call disposition does not match any of the other VRU Progress variables.</td>
</tr>
</tbody>
</table>

You can use the VRU Progress variable to indicate the final VRU status at the end of the transaction or to indicate changes in VRU status through the different transactions in the application.

**Note**

While you can change the VRU Progress variable throughout the application, only the final status is reported for the Call Type. The value of the VRU Progress variable is written to the database when the routing script terminates. You can report on the VRU status of the application as a whole using the Call Type VRU Activity reports by monitoring statistics for the Call Type associated with the script.

If you want to report on individual transactions within the application, change the VRU Progress variable and then the Call Type at the end of each transaction. You should have a different Call Type for each transaction with a related VRU Progress variable. This action ensures that the value of the VRU Progress variable is captured for that particular transaction, not just at the end of the routing script. The value is written to the database for the Call Type associated with that transaction when the Call Types changes. You can report on individual transactions using the Call Type VRU Progress reports by monitoring statistics for the Call Types associated with those transactions.

**Reports that Show VRU Metrics**

This report shows metrics for VRU applications:

- Unified Intelligence Center IVR Ports Performance Historical Report

**Data Loss and Component Failover**

Packaged CCE uses sophisticated techniques in gathering and storing data. Due to the complexity of the system, the amount of data being gathered, the diversity of devices and peripherals, and the complex processes involved, it is possible that historical reports might present inconsistent data.

Although these reporting inconsistencies are confusing, most can be traced to temporary effects caused by time lags between the processes involved and nature of the data itself.
This section identifies and explains common conditions that lead to temporary and permanent inconsistencies in historical data. It discusses the potential effect of system failover on data that appears in reports and offers guidance on how to guard against data loss.

**Data flow for Reporting**

Data is sent from the Call Router to the Logger. By default, reporting occurs directly from the Logger. A Historical Data Server (HDS) is optional. If the HDS is configured, the Logger forwards historical data to the Historical Database Server in summary intervals.

**Recovery and Replication**

**Recovery Keys**

The recovery key is the base key for all historical data tables. This key is always incremented by 1 before a new record is inserted into any historical table.

In a duplex configuration, the Logger that finishes initializing first is designated the primary Logger (although both the Loggers are always active and working in parallel). The recovery key is always initialized by the primary Logger. The recovery key is based on the current GMT date and time and always has a value greater than any previous value generated. This helps the recovery process to keep the Loggers in sync.

The replication process may have a latency of about one to five minutes because the Logger replicates data table-by-table on the HDS.

**Replication**

The Replication process is responsible for replicating data that has been committed to the Logger Central database to the HDS database.

**Note**

The external HDS is an optional component. By default, reporting is done from the Logger.

The Replication mechanism consists of two processes: the Replication Server Process that runs on the Logger and the Replication Client Process that runs on the Distributor on which HDS has also been installed.

The Replication Client sends a request to the Replication Server requesting historical data that have associated Recovery Keys higher than those currently on corresponding historical tables. The Replication server sends the requested data back as a set of 2000 records each time.

The Replication server reads the historical data from the actual tables on the Logger and sends it to the Replication Client, which writes the historical data to the actual corresponding tables in the HDS database. Temporary tables are not used to replicate the data from the Logger to the HDS.

**Possible Points of Delay or Inconsistency**

If the Logger connected to the HDS goes offline, the HDS does not connect to a different Logger. For example, if the HDS is connected to Logger B and Logger B fails, the HDS does not connect to Logger A. When Logger B comes back up, it recovers data from Logger A and begins to receive current historical information. Once the Logger has recovered all of the data from Logger A, it begins to replicate this data to the HDS.
If reports are run from this HDS for recent intervals while the Logger is offline or while the Logger is in the process of recovering or replicating data, you might not see data for those intervals in reports. This situation is temporary, and you will see the data once the replication process for the tables used by the reports is complete. If you are using a fault-tolerant system with two HDS Administration & Data Servers, you can run reports using the backup HDS while the primary HDS is not receiving data.

If the HDS goes offline and you are using a fault-tolerant system with two HDS Administration & Data Servers, you can run reports using the backup HDS. When the HDS comes back up, it recovers data from the last HDS data backup and also replicates data from the Logger for the most recent data not available in the backup.

The recovery data replication is faster than regular Logger-HDS data replication. Once the HDS has recovered to its typical Logger-HDS latency of one to five minutes, data replication proceeds as usual.

### Methods to Prevent Data Loss from Logger and HDS Failure

Data loss manifests as *data holes*, which are one or more missing records in an historical database table. There are two types of data loss: temporary and permanent:

- A temporary data hole can happen during the Logger recovery process. For example, Logger A goes down, then comes back up and contacts Logger B to synchronize and recover historical data that was written while it was down. While this recovery process is going on, the reporting database on Logger A may have temporary data holes, which will be filled when the recovery process completes.

- A permanent data hole can happen during an Emergency Purge. For example, there can be permanent data loss if an emergency purge deletes records on one Logger that have not been sent to the other Logger or to the HDS.

### Data Retention and Backups

When the HDS recovers after going offline, it retrieves all of the data on the Logger for the interval for which data is missing from the backup. You must manually restore the rest of the data from the last HDS backup.

### CPU Utilization

It is possible that the process on one of the Loggers is slow because of space issues or an overload of the SQL Server. In this situation, the data on the Logger with the slower SQL Server will lag in persistence of the historical data with respect to the other Logger. This causes the HDS on the corresponding side to lag as well.

As a consequence, if both sides have an HDS set up and the same reports are run from both HDSs, the reports might differ. This is usually a temporary inconsistency, since the condition that causes the SQL server process to slow might be remedied. Autogrowing of the database and load conditions often remediate. The Loggers and the HDSs eventually catch up and are in sync. Running the reports later will result in consistent reports.

However, if the database server runs of disk space, the situation is quite serious and might cause data to be out of sync for a longer duration until the problem is remedied. A permanent loss of data can occur when data is purged from the peer Logger and never replicated on the slower side.

### Scheduled Purge and Retention Settings on Loggers

The goal of the scheduled purge is to free up database space by purging the oldest data.
**Emergency Purge**

The Packaged CCE logger retention settings and database size are preconfigured to be able to ensure sufficient database space for the data, in the unlikely event that the Logger Central Database becomes full or reaches a configured threshold size. Its objective is to free up space by purging data from the historical tables so that the database has more free space than the allowed minimum.

The emergency purge goes through each historical table in a predefined order one at a time and purges one hour's worth of data from the table. As data is purged from each historical table, a check is made to verify if the free space is more than the minimum threshold value. Once adequate space has been recovered, the emergency purge procedure stops. Otherwise, it continues through to the next historical table and keeps looping as necessary.

Permanent loss of historical data can occur if the emergency purge removes historical data that has not yet made it to an HDS and has also not been replicated to the peer Logger that is "down" or in the recovery process.

Database used percentage is displayed as a normal status message in the replication process every few minutes. You can occasionally monitor this value to make sure that is does not grow too often or too fast. Emergency purge occurs when the percentage used is greater than the configured value (usually 90%).

**Data Loss from PIM Failure and Reporting**

Here are some reporting considerations when you experience data loss from UCM or CVP VRU PIM failure.

The Peripheral Interface Manager (PIM) is the process on the Peripheral Gateway responsible for the actual connection to the peripheral and for normalizing the CTI interface on behalf of Packaged CCE.

If a PIM fails, if the link between the PIM and the UCM or CVP goes down, or if the UCM or CVP goes down, then all of the reporting data that has been gathered for the peripheral associated with the PIM is deleted.

When the PIM failures occur, the peripheral is marked offline to the central controller.

For UCCE agent PG's, the state of all agents on that peripheral is set to *logged out* and is reported as such to the CallRouter.

When the PG is duplexed, either the Side A or Side B PIM is active for each peripheral. If one side loses connection, the other comes up and activates.

**Other Possible Points of Failover**

**Peripheral Gateway / CTI Manager Service Failover**

If the agent's PG shuts down or the CTI Manager service shuts down, the agent is momentarily logged out. The agent might be logged in again automatically once the backup PG or CTI Manager comes into service. The agent Media Logout Status reports for the agent, agent skill group, agent team, and agent peripheral show a logout reason code of 50002.

**Table 40: Agent State before and after Peripheral Gateway/CTI Manager Service failover**

<table>
<thead>
<tr>
<th>Agent State at Fail-Over</th>
<th>Agent State after Fail-over</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available</td>
<td>Available</td>
</tr>
</tbody>
</table>
Agent Desktop / CTI OS Server Failover

If the agent desktop (CTI OS desktop) shuts down or loses communication with CTI OS Server, or if CTI OS Server shuts down, the agent is logged out of all MRDs supported by the peripheral that has lost communication with Packaged CCE software.

The agent is logged in again automatically when one of the following occurs:

- The agent desktop comes back up or resumes communication with CTI OS Server
- The agent is connected to the backup CTI OS server

The agent Media Logout Status reports for the agent, agent skill group, agent team, and agent peripheral show a logout reason code of 50002.

The state to which the agent reverts after failover depends on the agent's state when the failover occurred, as described in the following table.

<table>
<thead>
<tr>
<th>Agent State at Fail-Over</th>
<th>Agent State after Fail-over</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Ready</td>
<td>Not Ready</td>
</tr>
<tr>
<td>Wrap-up</td>
<td>Available, if in Available state before the call. Otherwise, the agent reverts to Not Ready.</td>
</tr>
</tbody>
</table>

Table 41: Agent State before and after Agent Desktop/CTI OS Server failover

Call type and Skill Group Metrics

This section provides information about call type and skill group metrics.

Call Type Real Time Table Metrics

The following table shows the fields in the Call_Type_Real_Time table that affect reporting metrics by metric category:
Table 42: Call_Type_Real_Time table and reporting metrics

<table>
<thead>
<tr>
<th>Queued Metrics</th>
<th>At VRU Metrics/ Answered Metrics</th>
<th>Service Level Metrics</th>
<th>Abandoned Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvgRouterDelayQ</td>
<td>CallsAtVRUNow</td>
<td>ServiceLevelAbandHalf</td>
<td>CallDelayAbandTime</td>
</tr>
<tr>
<td>AvgRouterDelayQ</td>
<td>AnsweredWaitTimeHalf</td>
<td>ServiceLevelAbandTo5</td>
<td>CallDelayAbandTime</td>
</tr>
<tr>
<td>AvgRouterDelayQ</td>
<td>AnsweredWaitTimeToday</td>
<td>ServiceLevelAbandToday</td>
<td>CallDelayAbandTime</td>
</tr>
<tr>
<td>AvgRouterDelayQTo5</td>
<td>CallsAnsweredHalf</td>
<td>ServiceLevelCallsTo5</td>
<td>CTDelayAbandTime</td>
</tr>
<tr>
<td>AvgRouterDelayQToday</td>
<td>CallsAnsweredTo5</td>
<td>ServiceLevelCallsToday</td>
<td>CTDelayAbandTime</td>
</tr>
<tr>
<td>CallsLeftQ</td>
<td>CallsAtAgentNow</td>
<td>ServiceLevelCallsOfferedHalf</td>
<td>DelayAgentAbandTime</td>
</tr>
<tr>
<td>CallsAtVRUNow</td>
<td>RouterQueueCallsHalf</td>
<td>ServiceLevelCallsOfferedTo5</td>
<td>DelayAgentAbandTime</td>
</tr>
<tr>
<td>RouterCallsQ</td>
<td>RouterQueueCallsToday</td>
<td>ServiceLevelHalf</td>
<td>DelayAgentAbandTime</td>
</tr>
<tr>
<td>RouterQueueQ</td>
<td>RouterQueueWaitTimeHalf</td>
<td>ServiceLevelTo5</td>
<td>DelayQAbandTime</td>
</tr>
<tr>
<td>RouterQueueQ</td>
<td>RouterQueueWaitTimeTo5</td>
<td>ServiceLevelToday</td>
<td>DelayQAbandTime</td>
</tr>
<tr>
<td>RouterQueueQ</td>
<td>RouterQueueWaitTimeToday</td>
<td></td>
<td>DelayQAbandTime</td>
</tr>
<tr>
<td>RouterQueueQ</td>
<td>ServiceLevelCallsQHeld</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following table shows the fields (by metric category) in the Call_Type_Interval table that affect reporting metrics:
### Table 43: Call_Type_Interval table and reporting metrics

<table>
<thead>
<tr>
<th>Queued Metrics</th>
<th>At VRU Metrics/ Answered Metrics</th>
<th>Service Level Metrics</th>
<th>Abandoned Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvgRouterDelayQ</td>
<td></td>
<td>ServiceLevelAband</td>
<td>AbandInterval1 - AbandInterval10</td>
</tr>
<tr>
<td>CallsQHandled</td>
<td></td>
<td>ServiceLevelCalls</td>
<td>CallDelayAbandTime</td>
</tr>
<tr>
<td>RouterQueueCalls</td>
<td></td>
<td>ServieLevelCallsOffered</td>
<td>CTDelayAbandTime</td>
</tr>
<tr>
<td>RouterQueueCallType Limit</td>
<td></td>
<td>ServiceLevel</td>
<td>DelayAgentAbandTime</td>
</tr>
<tr>
<td>RouterQueueGlobalLimit</td>
<td></td>
<td></td>
<td>DelayQAbandTime</td>
</tr>
<tr>
<td>RouterQueueWaitTime</td>
<td></td>
<td></td>
<td>RouterCallsAbandQ</td>
</tr>
<tr>
<td></td>
<td>At CVP :</td>
<td></td>
<td>RouterCallsAbandToAgent</td>
</tr>
<tr>
<td></td>
<td>CTVRUTime</td>
<td></td>
<td>TotalCallsAband</td>
</tr>
<tr>
<td></td>
<td>VRUTime</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Answered:</strong></td>
<td>AnsInterval1 - AnsInterval10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AnswerWaitTime</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CallsAnswered</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Network VRU and Skill Group Metrics**

The Answer Wait Time and ASA metrics do not include the time spent in the network queue, while the Service Level metrics do.

The Skill Group abandoned metrics allow you to determine the number of calls that abandoned while queued, but they do not allow you to determine the number of calls that abandoned after they left the CVP and before an agent answered them.

None of the Skill Group metrics include time spent in self-service or calls that ended during self-service because a call is not associated with a skill group until it is queued, and a call is queued after self-service is complete.

The following table shows the fields (by metric category) in the Skill_Group_Real_Time table that affect reporting metrics:

### Table 44: Skill_Group_Real_Time table and reporting metrics

<table>
<thead>
<tr>
<th>Queued Metrics</th>
<th>At CVP Metrics/ Answered Metrics</th>
<th>Service Level Metrics</th>
<th>Abandoned Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallsQueuedNow</td>
<td></td>
<td>ServiceLevelTo5</td>
<td>RouterCallsAbandQTo5</td>
</tr>
<tr>
<td>LongestCallQ</td>
<td></td>
<td>ServiceLevelCallsTo5</td>
<td>RouterCallsAbandToAgentTo5</td>
</tr>
<tr>
<td>RouterCallsQNow</td>
<td></td>
<td>ServiceLevelCallsAbandTo5</td>
<td></td>
</tr>
<tr>
<td>RouterLongestCallInQ</td>
<td></td>
<td>ServiceLevelCallsDequeuedTo5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>At CVP:</td>
<td>ServiceLevelRonaTo5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>None.</td>
<td>ServiceLevelCallsOfferedTo5</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Answered:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AnswerWaitTimeTo5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CallsAnsweredTo5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The following table shows the fields (by metric category) in the Skill_Group_Interval table that affect reporting metrics:

Table 45: Skill_Group_Interval table and reporting metrics

<table>
<thead>
<tr>
<th>Queued Metrics</th>
<th>At CVP Metrics/ Answered Metrics</th>
<th>Service Level Metrics</th>
<th>Abandoned Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallsQueued</td>
<td>At CVP:</td>
<td>ServiceLevel</td>
<td>AbandonRingCalls</td>
</tr>
<tr>
<td>RouterQueueCalls</td>
<td>None.</td>
<td>ServiceLevelCalls</td>
<td>AbandonRingTime</td>
</tr>
<tr>
<td>Answered:</td>
<td>AnswerWaitTime</td>
<td>ServiceLevelCallsAband</td>
<td>RouterCallsAbandQ</td>
</tr>
<tr>
<td>CallsAnswered</td>
<td></td>
<td>ServiceLevelCallsDequeued</td>
<td>RouterCallsAbandToAgent</td>
</tr>
</tbody>
</table>

**Reporting on Skill Groups**

These are some Cisco Unified Intelligence Center reports relevant to skill groups.

- Peripheral Skill Group Historical All Fields
- Peripheral Skill Group Real Time All Fields

**Reporting on Precision Queues**

These Cisco Unified Intelligence Center reporting templates are new for Precision Routing:

- **Agent Precision Queue Membership** - This report displays selected agents, the media routing domain into which the agent is logged, and the active precision queue with up to five associated attributes.

- **Precision Queue Real Time All Fields** - This report displays the current status of selected precision queues.

- **Precision Queue Historical All Fields** - This report displays interval data for consolidated call and precision queue statistics.

- **Agent Precision Queue Historical All Fields** - This report displays activity for selected agents for a selected interval, sorted by precision queue.

These Cisco Unified Intelligence Center reporting templates contain updates for Precision Routing:

- **Agent Real Time** - This report displays, for each agent, the active skill group or active precision queue, the state, and the call direction within each media routing domain into which the agent is logged.

- **Agent Team Real Time** - This report displays the current status for each selected agent team and displays the current state and the active skill group or active precision queue for each agent in the selected agent teams.

- **Call Type Queue Historical All Fields** - This report displays the summary statistics for skill groups and precision queues within Call Type ID.
Reporting on Precision Queues
INDEX

15-minute data 232

A

abandoned calls 269
abandoned short calls 257, 268, 269, 271
accessing micro-application by specifying Run VRU Script Node 170
adding 16, 20, 22, 29, 31, 34, 36, 44, 50, 51, 53, 56, 59, 72
  agents 16
  attributes 34
  bulk jobs 72
  call types 51, 56
  desk settings 22
  dialed numbers 53
  ECC variables 56
  Network VRU Scripts 59
  precision queues 36
  reason codes 29
  skill groups 44
  supervisors 20
  teams 31
admin scripts 229
administrators 5, 6, 8
  Manage menu 8
  sign in 5
  user interface 6
agent 247
  utilization 247
agent desktops 24, 282
  agent desktop failover 282
agent destops 24
agent logout reason codes 240
  predefined 240
agent not ready reason codes 241
agent reporting 256
  agent RONA statistics 256
agent states 236, 239
  and skill groups 239
  and tasks 239
agent states (continued)
  list of 236
  reporting on 236
  agent task handling, reporting on 243
  agent trace 10, 77
  agents 10, 16, 19, 71, 248
    adding 16
    agent trace 10
    bulk jobs 71
    deleting 19
    editing 19
    reports 19, 248
  and deployment type 77
  answered short calls 268, 269
ASA 255
  agent 255
  reporting 255
  attributes 33, 34, 35
    adding 34
    deleting 35
    editing 34

B

Barge-in 252
bucket intervals 49, 50
  adding 50
  reporting 49
built-in functions 154
  date and time functions 154
bulk jobs 16, 67, 68, 71, 72, 73, 78
  adding 72
  agents 16, 71
  content files 68
  dialed numbers 71
  logs 78
  review details 73
calculated fields 235

call detail data 234

Call Studio 223, 225, 228, 230

about 223
developing scripts 230
integrating scripts (simplified with Unified ICME scripts) 228
integrating scripts with Unified ICME scripts 228
using ReqICMLabel 225
call type 229
defined 229

Call Type 260

Calls Offered calculation 260
call type reporting 51
call type reports 259
call type, abandon 270
call type, abandoned short calls 271
call type, abandons en-route to VRU 260
call type, calls with bad label 261
call type, non-monitored devices 262
call type, RONA 261
call types 51, 56, 104, 257, 258
adding 51, 56
and scripts 51
categorization 51
call type qualifiers 104
changing 258
default call type 104
planning 257

Calls Offered 250, 260

Call Type 260

Skill Group 250
capacity info 76
Capacity Info 7

capacity info tab 76
capacity limits 76

Capture micro-application error code settings 173
categorizing contacts 106, 110, 111, 112, 114, 115

  by branching 115
  by call type qualifiers 110
  by dialed number 110
  by the day of week 114
  by time 112
  by time and date 111
  through scheduling scripts by call type 106

CCE Data Server 1
Cisco Customer Voice Portal (CVP) 273
Cisco Finesse 1
Cisco IP-IVR 273
Cisco Unified Communications Manager (CUCM) 1
Cisco Unified Contact Center Enterprise (Unified CCE) 1
Cisco Unified Intelligence Center (CUIC) 1
Communications Manager 81

Conferences 271, 272
configuration 272
planning 271
configuration manager 13, 85
minimum password length 85
configuration parameters 178, 182
Play Media micro-application 182
configure deployment 10, 75
configuring 178

Network VRU Script 178
creating 177

Unified CCE applications 177

CTI Manager Service fail-over 281
CTI OS desktop 24
CTI OS Server failover 282
custom functions 159, 160, 161

  adding custom functions 159
  exporting custom functions 161
  importing custom functions 160
custom reports 234

Customer Voice Portal (CVP) 1
CVP scripting 166, 167, 172, 173, 181, 182, 188, 211, 216, 220, 222
about 167
Capture micro-application 181
considerations for ICM Enterprise 166
data handling 172
error checking 173
Get Digits (GD) micro-application 211
Get Speech (GS) micro-application 220
information exchange 172
Menu (M) micro-application 216
micro-applications 167
Play Data (PD) micro-application 188
Play Media (PM) micro-application 182

with Call Studio 222
with ICM Enterprise 167

data 231, 232, 233, 234
15-minute 232
call detail 234
configuration 233
half 231
historical 232
interval 232
now 231
real time 231
To5 231
today 231
data handling in scripting 172
data holes 280
data retention 280
data storage for Play Data micro-application 188
Database Schema Handbook 235
database tables 232, 235, 279
  Database Schema Handbook 235
  interval tables 232
  reporting 235
  temporary tables 279
Default Media Server 180
  for Micro-applications 180
default skill group data, agent to agent dialing 250
default skill group data, conferences 250
default skill group data, new calls 249
default skill group data, transfers 250
default skill group reporting 249
Default Skill Group Reporting 249
deleting 19, 21, 24, 30, 32, 35, 42, 47, 87
  agents 19
  attributes 35
  desk settings 24
  permanently 87
  precision queues 42
  reason codes 30
  skill groups 47
  supervisors 21
  teams 32
deploying scripts 230
  with Call Studio 230
  with Operations Console 230
deployment 1, 7, 10, 75, 77
  configure deployment 10, 75
  Deployment Info 7
  Packaged CCE 1
  system validation 77
Deployment Info 7
desk settings 22, 23, 24, 82
  adding 22
  deleting 24
  editing 23
  locate and change default 82
dialed number 105
daialed numbers 53, 71
  adding 53
  bulk jobs 71
digit entry completion 215, 219
  Get Digits micro-application 215
  Menu micro-application 219
dynamic audio file capability 179

E
ECC variables 56, 58
  adding 56
  sizing 58
  editing 19, 21, 23, 30, 32, 34, 42, 47
  agents 19
  attributes 34
  desk settings 23
  reason codes 30
  skill groups 42, 47
  supervisors 21
  teams 32
Emergency Assist 251
end node 121
error checking 173
  in scripting 173
error code settings 173
  Capture micro-application 173
  Get Digits micro-application 173
  Get Speech micro-application 173
  Menu micro-application 173
  Play Data micro-application 173
  Play Media micro-application 173
  user.microapp.error_code ECC variable for non-video 173
  user.microapp.error_code ECC variable for video 173
error codes 173
  user.microapp.error_code ECC 173
examples 179, 187, 210, 213, 218
  Get Digits micro-application configuration 213
  Menu micro-application configuration 218
  Play Data micro-application configuration 210
  Play Media micro-application configuration 187
  using a dynamic audio file 179
external VoiceXML 214, 221, 222
  error handling, Get Speech micro-application 214
  Get Speech micro-application 214
  passing data back to Unified ICME 222
  passing information 221

F
fail-over 278, 280, 281
  fail-over, effect on reporting 278
  logger and HDS 280
  PGs 281
  PIM failure 281
failover 282
  agent desktop failover 282
fault tolerance 280
Finese desktop 24
Finesse desktop 24
formulas \textbf{142} 
  formula example \textbf{142} 
  using formulas \textbf{142} 
  what is a formula \textbf{142} 
full-time equivalents reporting \textbf{247} 

\textbf{G} 
Get Digits micro-application \textbf{173, 213, 215} 
  configuration examples \textbf{213} 
  digit entry completion \textbf{215} 
  error code settings \textbf{173} 
  if no entry timeout occurs \textbf{215} 
Get Speech micro-application \textbf{173, 214} 
  error code settings \textbf{173} 
  external VoiceXML \textbf{214} 

\textbf{H} 
half data (real time) \textbf{231} 
half-hours boundaries, impact on reporting \textbf{236} 
HDS \textbf{279, 280} 
  and historical data \textbf{279} 
  HDS fail-over \textbf{280} 
Helix Server \textbf{186} 
  streaming ringtones \textbf{186} 
high-level configuration steps for call flow models \textbf{224} 
  historical data \textbf{232, 279} 
Home menu \textbf{7} 
how to \textbf{138, 140, 141, 146} 
  change the queue to agent type \textbf{140} 
  set variable values with the set variable node \textbf{146} 
  specify an agent directly \textbf{141} 
  use MRD to categorize contacts \textbf{138} 

\textbf{I} 
independent media queues \textbf{137} 
information exchange \textbf{172} 
  in scripting \textbf{172} 
integrating scripts \textbf{228} 
  Call Studio and ICME \textbf{228} 
integrating scripts - traditional \textbf{228} 
  Call Studio and ICME \textbf{228} 
Intercept \textbf{253} 
i nternationalization \textbf{7} 
interval data \textbf{232} 
intervals \textbf{232} 
  database tables \textbf{232} 

\textbf{K} 
keyboard shortcuts \textbf{12} 

\textbf{L} 
List windows \textbf{10} 
logger \textbf{279, 280} 
  and historical data \textbf{279} 
  logger fail-over \textbf{280} 
logs \textbf{78} 
  bulk jobs \textbf{78} 
  system validation \textbf{78} 

\textbf{M} 
Manage menu \textbf{6, 8} 
  administrators \textbf{8} 
  supervisor permission \textbf{6} 
  supervisors \textbf{8} 
mathematical Functions \textbf{157} 
media routing domain \textbf{137} 
Media Server \textbf{180} 
  specify in Operations Console \textbf{180} 
Menu micro-application \textbf{173, 218, 219} 
  configuration examples \textbf{218} 
  digit entry completion \textbf{219} 
  error code settings \textbf{173} 
  if no entry timeout occurs \textbf{219} 
micro-applications \textbf{167, 177, 179, 181, 182, 188, 211, 216, 220} 
  accessing \textbf{177} 
  Capture \textbf{181} 
  dynamic audio file support \textbf{179} 
  Get Digits (GD) \textbf{211} 
  Get Speech (GS) \textbf{220} 
  Menu (M) \textbf{216} 
  Play Data (PD) \textbf{188} 
  Play Media (PM) \textbf{182} 
  specifying Run External Script Node to access \textbf{179} 
  using for scripting \textbf{179} 
Micro-applications \textbf{180} 
  Default media server \textbf{180} 
  minimum password length \textbf{85} 
  miscellaneous functions \textbf{157} 
  multichannel routing \textbf{136} 
  multichannel scripting \textbf{137}
network queuing and reporting 284
Skill Group metrics 284
Network VRU Script 178
configuring 178
Network VRU Scripts 59, 60, 62, 66
adding 59
configuration parameters 62
sample configuration values 66
sample VRU Script Names 62
VRU Script Name Parameters 60
network VRUs 124
nodes 121
end 121
release call 121
Now data 231

Operations Console 230
deploying Call Studio scripts 230
operators 149, 150, 151, 152, 153
arithmetic operators 150
bitwise operators 152
equality operators 151
logical operators 152
miscellaneous operators 153
operator precedence 149
prefix operators 150
relational operators 151
Outbound Option 228
scripting 228

Play Data micro-application (continued)
data storage 188
error code settings 173
play back types for voice data 190
Play Media micro-application 173, 187
configuration example, play welcome message 187
error code settings 173
precision queues 33, 35, 36, 39, 41, 42, 43
adding 36
attributes 33, 35
Consider If 35, 39, 41
deleting 42
expressions 35, 39
or skill groups 43
steps 39
terms 35
Wait for 39
attributes 33, 35
precision queues 33
terms 33
predefined Not Ready reason codes 241
purges 280

Real time and historical data, comparing records 235
real time data 231
reason code 29
adding 29
reason codes 29, 30
deleting 30
editing 30
recovery keys 279
replication 279
report comparison 235, 236
half-hour boundary issues 236
real time and historical reports 235
reporting 19, 248
agents 19, 248
reporting data 235, 278, 279, 280
comparing 235
data purges and retention 280
database tables 235
delay and inconsistency 279
flow from logger to HDS 279
in fail-overs 278
ReqICMLLabel 225
using to pass data 225
requirements 222
VoiceXML 222
reservation scripts 229
RONA 256
route call detail data (RCD) 234
routing scripts 233, 258
routing target 118
running external scripts 125

S
Script Execution 115
scripting 228, 229
   Outbound Option 228
   Outbound Option admin scripts 229
   Outbound Option reservation scripts 229
scripting for CVP 166, 167, 172, 173, 181, 182, 188, 211, 216, 220, 222
   about 167
   Capture micro-application 181
   considerations for ICM Enterprise 166
   data handling 172
   error checking 173
   Get Digits (GD) micro-application 211
   Get Speech (GS) micro-application 220
   information exchange 172
   Menu (M) micro-application 216
   micro-applications 167
   Play Data (PD) micro-application 188
   Play Media (PM) micro-application 182
   with Call Studio 222
   with ICM Enterprise 167
selecting routing targets 118
service creation environments (script editors) 167
   Call Studio 167
   ICM Script Editor 167
service level 263, 264
   service level threshold 263
   service level type 263, 264
Service Level 257
service level length 85
service level reporting 263
service level type 85
Service Level, call type 265
Service Level, skill group 266
Settings menu 10
short calls 268
   configuring 268
   short calls, abandoned 268
   short calls, answered 268
   sign in 5
   administrators 5
   supervisors 5
Skill Group 250
   Calls Offered calculation 250
skill groups 42, 43, 44, 47, 239, 248, 250
   adding 44
   skill groups (continued)
   and Cisco Unified Intelligence Center reporting 44
deleting 47
   editing 42, 47
   or precision queues 43
   agent activity in 248
   and agent states 239
   reporting on skill group operations 250
   skill group agent reporting 248
specifying Run External Script Node to access Unified CVP
   micro-applications 179
   stopping script processing 121
   streaming ringtones 186
   Helix Server 186
supervisor action 251
supervisor assist 30
Supervisor Assist 251
supervisors 5, 6, 8, 20, 21, 30
   adding 20
   and teams 30
deleting 21
   editing 21
   Manage menu 6, 8
   sign in 5
   user interface 6
system bucket interval 85
system validation 75, 77, 78
   configure deployment 75
   logs 78
System Validation 7
T
target query 122
tasks 239, 243, 246
   task states 239
   task times 246
   task types 243
teams 30, 31, 32
   adding 31
deleting 32
   editing 32
termination call detail data (TCD) 234
To5 data 231
transfers 271, 272
   configuration 272
   planning 271
trunks and trunk groups 273
**U**

Unified CCE 166, 177
- creating applications to access Unified CVP
  - micro-applications 177
- scripting 166

Unified IC 232
- and 15-minute data 232

Unified ICME 222
- passing data back with external VoiceXML 222

universal queue 137
- configuration overview 137

user interface 3, 5, 6, 7, 10, 12
- keyboard shortcuts 12
- languages 7
- List windows 3, 10
- popup selections 10
- searching 10
- sign in 5
- sorting 10
- supervisors 6
- administrators 6

using 221
- param elements 221

utility nodes 163, 164
- comment 164
- line connector 164
- start 163

**V**

variables 142, 143, 145, 149
- call control variables 143
- closed variables 149
- ECC variables 145
- multi-target variables 143
- single-target variables 143
- user variables 145
- variable syntax 142
- what is a variable 142

videos 173
- error code settings for Full Video 173

Virtual Machines 1

voice data, playback types 190

VoiceXML 166, 214, 221, 222
- error handling, Get Speech micro-application 214
- external, Get Speech micro-application 214
- passing data back to Unified ICME with external 222
- passing information to the external 221
- requirements 222
- scripting 166

VRU 179, 184
- Run VRU Script node 179, 184

VRU Progress variables 276

VRUs 124, 126, 127, 273, 274
- accessing VRU scripts 124
- checking for VRU errors 126
- information gathering 274
- planning 273
- queuing 274
- queuing calls at VRUs 127
- self-service 274
- VRU application reporting 273