Scripting and Media Routing Guide for Cisco Unified ICM/Contact Center Enterprise, Release 12.6(2)

First Published: 2023-04-28

Last Modified: 2023-04-28

CONTENTS

Preface

-
-
-
-
-
-
-
-

Change History

This table lists changes made to this guide. Most recent changes appear at the top.

Change	See	Date
Initial Release of Document for Release 12.6(2)		April 2023
Added new sections for the digital channel integration using Webex Connect	- Determination of Call Type for a Digital Routing Task - Digital Routing using Webex Connect - Example Digital Channel Interactions Using Webex Connect	

About This Guide

This manual describes how to use the Script Editor tool for Cisco Unified Intelligent Contact Management (Unified ICM) and Cisco Unified Contact Center Enterprise (Unified CCE) to create and maintain routing and administrative scripts.

Audience

This document is intended for system managers. A system manager must have a general understanding of contact center operations and management, and specific information about the contact centers and carrier networks connected to a Unified ICM/Unified CCE system.

Related Documents

Documentation for Cisco Unified Intelligent Contact Management/Cisco Unified Contact Center Enterprise, as well as related documentation, is accessible from Cisco.com at: https://www.cisco.com/cisco/web/psa/default.html.

Related documentation includes the documentation sets for Cisco Unified Contact Center Management Portal, Cisco Unified Customer Voice Portal (Unified CVP), Cisco Unified IP IVR, and Cisco Unified Intelligence Center. The following list provides for information:

- For documentation for these Cisco Unified Contact Center Products, go to https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/tsd-products-support-series-home.html

- For troubleshooting tips for the Cisco Unified Contact Center products, go to https://techzone.cisco.com/t5/Unified-Contact-Center/ct-p/ucc_ucce.

- You can access documentation for Cisco Unified Communications Manager from: https://www.cisco.com/c/en_in/products/unified-communications/unified-communications-manager-callmanager/index.html

- You can access technical Support documentation and tools from: https://www.cisco.com/en/US/support/index.html.

- You can access the Product Alert tool from (login required): https://www.cisco.com/c/en/us/support/web/tools/cns/notifications.html.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.

- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.

- To submit a service request, visit Cisco Support.

- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.

- To obtain general networking, training, and certification titles, visit Cisco Press.

- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Field Notice

Cisco publishes Field Notices to notify customers and partners about significant issues in Cisco products that typically require an upgrade, workaround, or other user action. For more information, see *Product Field Notice Summary* at https://www.cisco.com/c/en/us/support/web/tsd-products-field-notice-summary.html.

You can create custom subscriptions for Cisco products, series, or software to receive email alerts or consume RSS feeds when new announcements are released for the following notices:

- Cisco Security Advisories

- Field Notices

- End-of-Sale or Support Announcements

- Software Updates

- Updates to Known Bugs

For more information on creating custom subscriptions, see *My Notifications* at https://cway.cisco.com/mynotifications.

Documentation Feedback

To provide comments about this document, send an email message to the following address: contactcenterproducts_docfeedback@cisco.com

We appreciate your comments.

Conventions

This document uses the following conventions:

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

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

Common Task

Script Editor Online Help

This chapter contains information about some of the most common tasks you perform in Script Editor. This topic does not contain information about all the tasks you can perform. For more information on other Script Editor tasks, see the *Script Editor online help*.

Note If you are a department administrator for Packaged CCE deployments (Packaged CCE: CCE-PAC-M1 and Packaged CCE: CCE-PAC-M1 Lab Only), you do not have access to the Script Editor. Instead, use the Internet Script Editor client, unless restricted by the feature control of the client or by your role.

Toolbars

View Toolbars

You can control which toolbars appear in the application window by using the **View** menu.

Procedure

Step 1 Choose **Toolbar** from the **View** menu.

The submenu lists the four toolbars. Toolbars that are currently open, are checked.

Step 2 To open a toolbar from the submenu, select the unchecked toolbar.

Step 3 To close a toolbar from the submenu, clear the checked toolbar.

You can also control the size of the toolbar icons from the Toolbar submenu:

- To display large toolbar icons with text, check **Large Icons**.

- To display small toolbar icons without text, uncheck **Large Icons**.

Note Toolbar icons in this document are always shown as large icons with text.

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.

Related Topics

Use Palette to Build a Script

You use the Palette to build a script by moving icons representing different nodes into the script workspace. You can build a script in the following ways:

Procedure

- Place the cursor over an object and hold the left mouse button, then drag the object into the workspace and release the mouse button.
- Place the cursor over an object and click the left mouse button, then move the cursor into the workspace and click the mouse button again. To insert more of the same node, reposition cursor and click again.

General Tab

The following figure displays the General tab of the Palette:

Figure 2: General Tab of the Palette
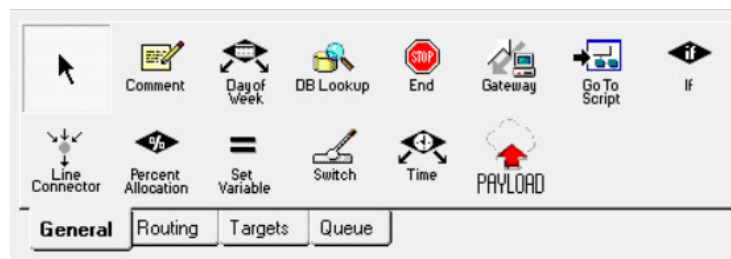
The General tab contains icons for the following scripting activities:

Related Topics

Routing Tab

The following figure displays the Routing tab of the Palette:

Figure 3: Routing Tab of the Palette
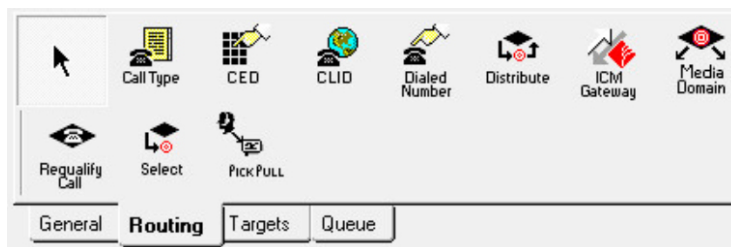
The Routing tab contains icons for the following scripting activities:

Related Topics

Targets Tab

The following figure displays the Targets tab of the Palette:

Figure 4: Targets Tab of the Palette

The Targets tab contains icons for the following scripting activities:

Related Topics

Queue Tab

The following figure displays the Queue tab of the Palette:

Figure 5: Queue Tab of the Palette

The Queue tab contains icons for the following scripting activities:

Related Topics

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 6: New Dialog Box

Choose the type of script to create:

Routing script: A routing script handles calls delivered by a routing client.

Administrative script: An administrative script runs at a specified interval and updates internal state.

Cancel
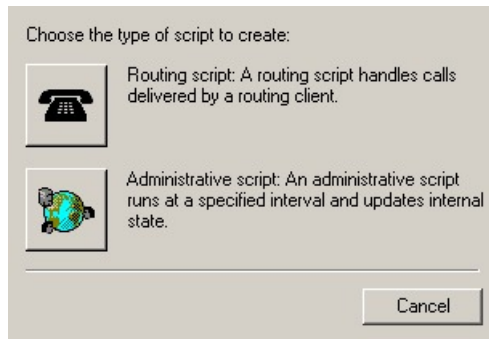
Step 2   Click the following icon.

Figure 7: Routing Script

The new script opens in the Edit window, with a Start node. (See Start Node, on page 207.

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

Figure 8: Comment icon

Comment

Most nodes have a Comment tab :

Figure 9: Comment Tab
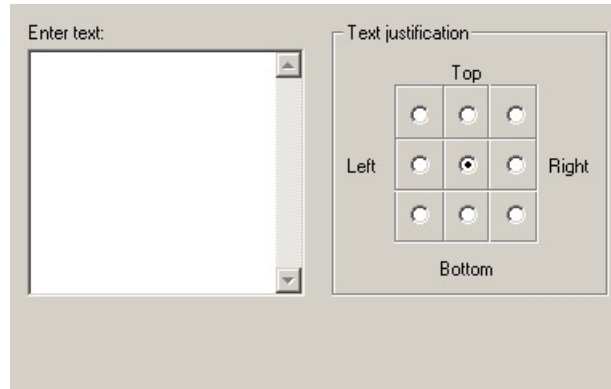
Procedure

Step 1   To add a comment, type in the **Enter text** field.

Step 2   To select the location where you want your comment to appear in the node, select a radio button in the **Text justification** area.

Specify a Connection Label Location for a Node

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

Figure 10: Connection Tab
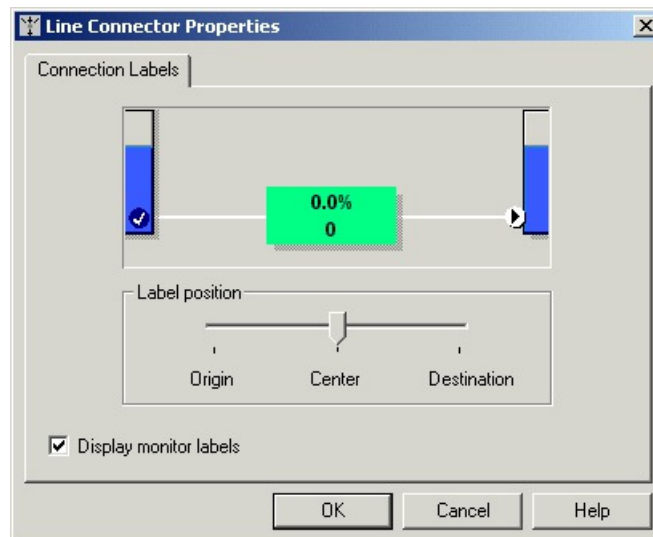
Procedure

Step 1 When viewing a script in monitor mode, you can specify the location of connection labels by moving the slider in the Label position area to one of the following locations:

  a) **Origin**, displays the connection label close to the node you are editing.
  b) **Destination**, displays the connection label close to the targeted node.
  c) **Center**, displays the connection label between the nodes.

Step 2 You can remove the connection label by clearing the **Display monitor labels** check box.

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 11: Validate All Query Dialog
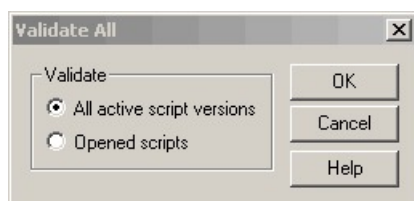
Step 3 Make the appropriate selection and click **OK**.

  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 and warnings. 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:
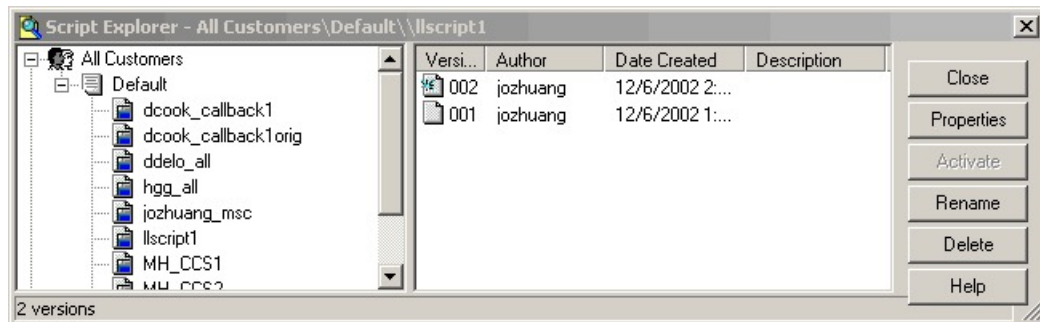
You can then set the active version of the script, view its properties, rename it, or delete it. For more information, see the *Script Editor Online Help*.

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 12: Call Type Manager Dialog Box—Call Directory Tab
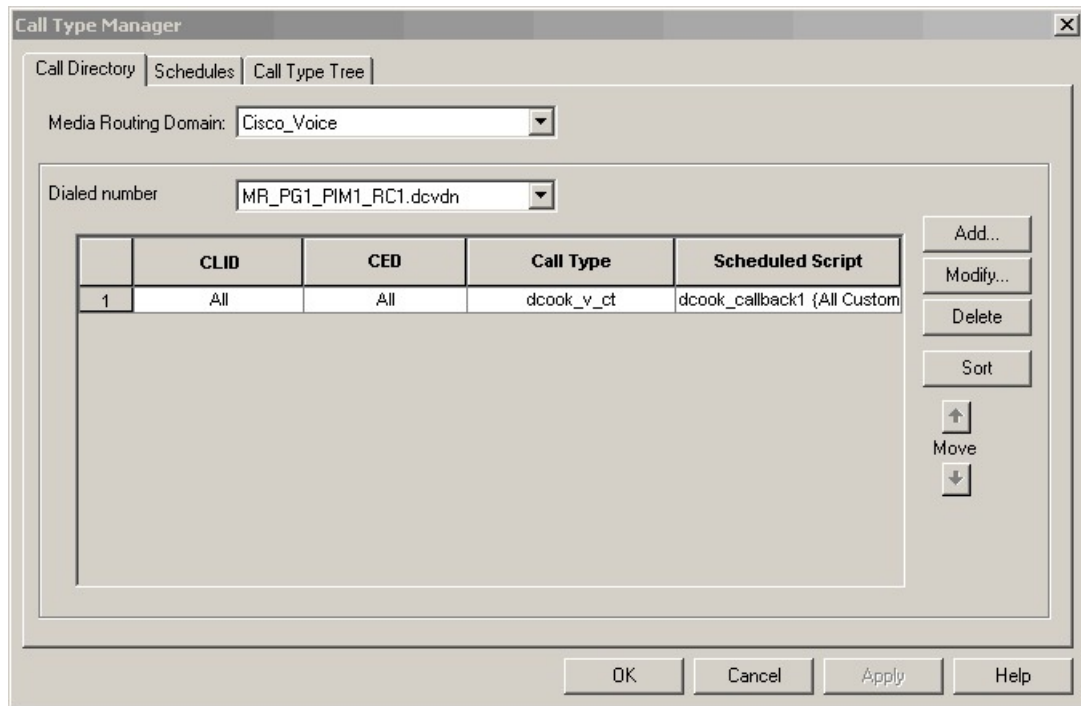
Step 2   Select the **Schedules** tab.

Figure 13: Call Type Manager Dialog Box - Schedules Tab
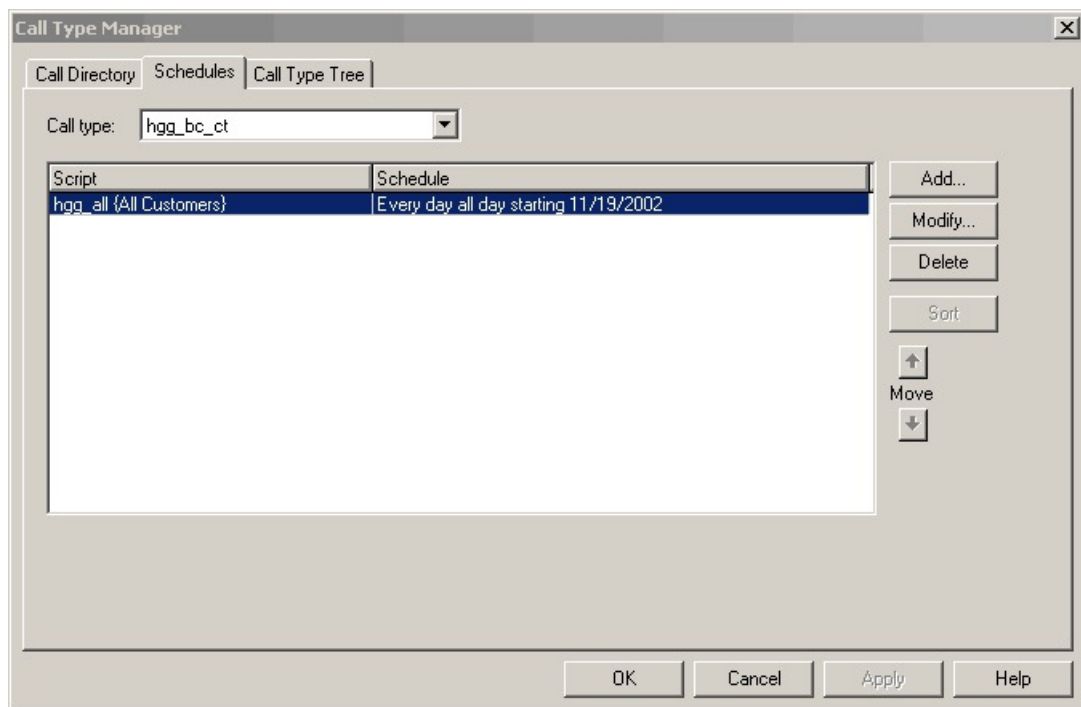
Step 3   Select the call type to associate with the script.

Step 4   Click **Add**. The Add Call Type Schedule dialog box opens.

Step 5   In the Script tab, select the script to schedule:

Figure 14: Add Call Type Dialog Box - Script Tab
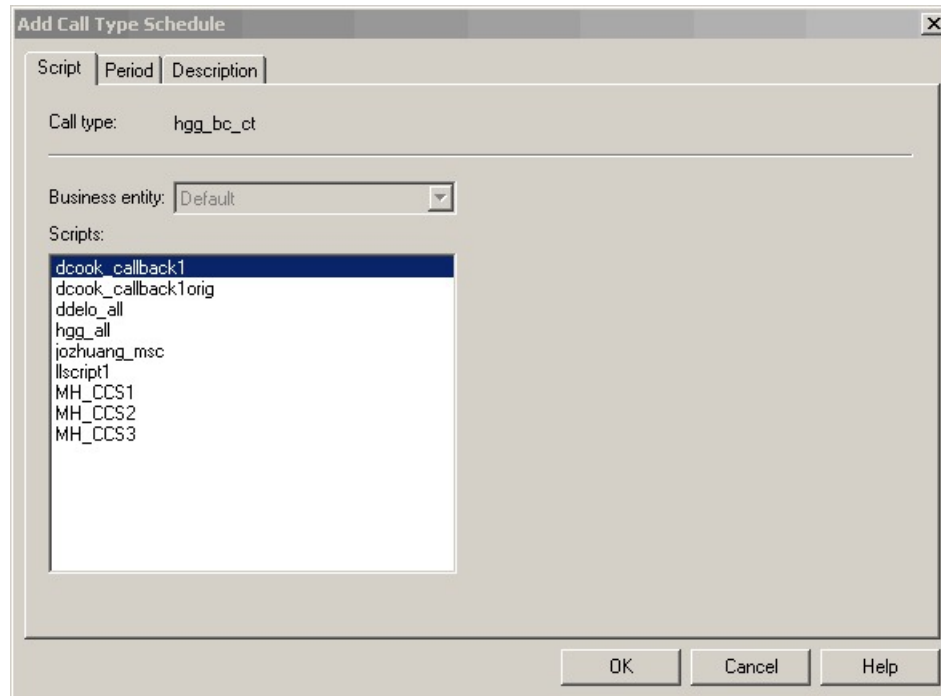
Step 6   In the **Period** tab, choose the information to define the period for which the schedule will be effective.

Figure 15: Add Call Type Schedule Dialog Box - Period Tab

Step 7   Optionally, in the **Description** tab, enter a description of the schedule.

Step 8   Click **OK** in the Add Call Type Schedule dialog box.

Step 9   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.

- **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.

Find Nodes Option

Unified ICM Script Editor is a tool used to create, modify, and schedule routing scripts. A script consists of a series of nodes. When managing large, complex scripts, the **Find Nodes** option makes it easier to identify/access the node you want to view.

Use the Find Nodes option to find specific script nodes by:

- **Node ID**

- **Object**

- **Node Type**

- **String**

Use the Find Nodes option to find script nodes in:

- **Current script**

- **All active script versions**

- **All open scripts**

Note For more information about using Unified ICM Script Editor's features, see the *Unified ICM Script Editor online help*.

Find Nodes and Nodes Found Dialog Boxes

Most properties of the "Find Nodes" dialog box are common and appear regardless of the "Find Nodes By" selection. However, as you select each of the options in the "Find Nodes By" section, the appearance of the "Find Nodes" dialog box changes.

The "Nodes Found" dialog box displays the results of your "Find" operation. The following tables list the "Find Node" dialog box and "Nodes Found" dialog box property descriptions.

Keyboard equivalents are indicated in parentheses after each field or button.

Use **Tab/Shift**+**Tab** in the "Find Nodes" dialog box to change focus and navigate to the sections, buttons, and input fields. You can reach each field or button within a section/group (Find Nodes By/In) by using the *Up/Down* keys.

Use **Tab/Shift**+**Tab** in the "Nodes Found" dialog box to change focus. You can reach each node by using the **Up/Down** keys.

Note Once the "Nodes Found" dialog box appears, click "Reload Configuration" or close "Nodes Found" dialog box to get the latest configuration. The change in configuration will appear in Script Editor.

Common find nodes dialog box properties

Find	Click to find nodes based on the search criteria set in the "Find Nodes" dialog box. The "Nodes Found" dialog appears displaying a list of the nodes found that matched the search criteria you select.
Close (Alt+F4)	Click to close the "Find Nodes" dialog box.
Help	Click to access the "Find Nodes" online help.

Find Nodes by Section

Node ID (Alt+N)	The default selection. Selects a node based on its node ID. Disabled if no script is open or the Current script option is not selected. When selected a "Please enter node ID:" field appears. Displays the node in the Current script based on the node ID.
Object (Alt+O)	Selects node(s) based on references to an object. When selected, the "Please select object type:" and "Please select object:" drop-down lists appear. Select the object type first, then the object. Displays a list of nodes in the Current script, All active script versions, or All opened scripts having a reference to the selected object.
Node Type (Alt+T)	Selects node(s) based on the node type. When selected, the "Please select node type" drop-down list appears. Displays a list of nodes of the selected type in the Current script, All active script versions, or All opened scripts.
String (Alt+S)	Selects node(s) based on a string that is entered. This string is case insensitive. When selected, the "Please enter string" field appears. Displays a list of nodes containing the substring of the input string in the Current script, All active script versions, or All opened scripts. **Note:** By default, this search criteria does not search the "Comment" field in each node. Check the "Search comment field option" (Alt+F) to search the "Comment" field as well.

Find Nodes in Section

Current script (Alt+C)	Select to find nodes in the current script based on any of the "Find Nodes By" properties. Disabled if no script is open.

All active script versions (Alt+A)	Select to find nodes in all active script versions based on the Object, Node Type, or String "Find Nodes By" properties. If no script is open, this is the only "Find Nodes In" option enabled.
All opened scripts (Alt+P)	Select to find nodes in all open scripts based on the Object, Node Type, or String "Find Nodes By" properties. Disabled if no script is open.

Common Nodes Found Dialog Box Properties

List field	A listing of the nodes found due to the **Find**.
Close (Alt+F4)	Click to close the "Nodes Found" dialog box.
Help	Click to access the "Nodes Found" online help.
Next Node (Alt+N)	Click to select the next node in the list.
Status bar	Displays "Finding nodes in script." as the search proceeds, then displays the number of nodes found matching the search criteria.

Find Nodes by Node ID

Procedure

Step 1 Open the Script Editor tool.

Step 2 Choose **Edit > Find Node (Ctrl+F)**. The Find Nodes dialog box appears.

Step 3 In the "Find Nodes By" section, choose **Node ID**.

Step 4 Enter the ID of the node you wish to view in the "Please enter node ID:" field.

Step 5 Click **Find**. The node with the ID matching your input is highlighted.

> **Note** If there is no node with a matching ID, a message appears stating: "Cannot find node with ID <xxx>." (where <xxx> is the node ID).

Step 6 Choose a node in the list and that node is highlighted in the script.

Find Nodes by Object

Procedure

Step 1 Open the Script Editor tool.

Step 2 Choose **Edit > Find Node (Ctrl+F)**. The "Find Nodes" dialog box appears.

Step 3   In the "Find Nodes By" section, choose **Object**. Two drop-down lists appear, one providing a list of object types ("Please select object type:"), the other providing a list of objects ("Please select object:").

Step 4   Select the desired object type.

Step 5   Select the desired object.

Step 6   Click **Find**. A "Nodes Found" dialog box appears displaying a list of nodes referencing the selected object.

> **Note**   If no node has a reference to the selected object, a message appears stating: "Cannot find any node with reference to <xxx> object: <Yay>." (where <xxx> is the object type selected and <Yay> is the object selected).

Step 7   Select a node in the list and that node is highlighted in the script.

Find Nodes by Node Type

Procedure

Step 1   Open the Script Editor tool.

Step 2   Choose **Edit > Find Node (Ctrl+F)**. The "Find Nodes" dialog box appears.

Step 3   In the "Find Nodes By" section, choose Node Type. A drop-down list appears providing a list of node types ("Please select node type:").

Step 4   Select the desired node type.

Step 5   Click **Find**. A "Nodes Found" dialog box appears displaying a list of nodes of the selected type.

> **Note**   If no node of the selected type is found, a message appears stating: "Cannot find any node of <xxx> type." (where <xxx> is the node type).

Step 6   Choose a node in the list and that node is highlighted in the script.

Find Nodes by String

Procedure

Step 1   Open the Script Editor tool.

Step 2   Choose **Edit > Find Node (Ctrl+F)**. The "Find Nodes" dialog box appears.

Step 3   In the "Find Nodes By" section, choose **String**. The "Please enter string:" field appears.

> **Note**   The Find By String "Please enter string:" field entry is case insensitive.

Step 4   Enter the appropriate string into this field.

Step 5   Click **Find**. A "Nodes Found" dialog box appears displaying a list of nodes containing a substring of the input string.

Note   By default, **Find** does not search the "Comment" field in each node. Check the **Search comment field** option **(Alt+F)** to search on the "Comment" field as well.

Note   If no node has reference to the input string, a message appears stating: "Cannot find any node with reference to string <xxx>." (where <xxx> is the string selected).

Step 6   Choose a node in the list and that node is highlighted in the script.

Call Types Contact Data Scripting

Call Types

When writing scripts to route contacts, you must understand call types and contact data.

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.

You create call types during Unified ICM configuration, through the Configuration Manager. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html and the *Cisco Unified Contact Center Enterprise Installation and Upgrade Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-guides-list.html

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 can also define a general default call type that is not specific to a routing client. See the *Configuration*

Guide for Cisco Unified ICM/Contact Center Enterprise at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html.

Relation Between Call Types and Scripts

Scripts are scheduled by **call type**. In other words, when the system 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 run.

Call Type Qualifiers

The following data determine the call type. This data is referred to as the call type qualifier.

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.

The differences between call type qualifiers for voice and other media are also explained in the following sections in this topic:

• Contact Data for Cisco Enterprise Chat and Email

Note You can also use the call type qualifiers for categorization within a script.

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, "NICClient.18005551212" might be a dialed number.

The **Calling line ID** and **Caller-entered digits** are used to further categorize the call and determine the call type.

Typically, a dialed number is associated with one or more call types.

Note The dialed number is referred to as the Script Selector for media other than voice.

Calling Line ID (CLID)

The Calling Line ID (CLID) is a string that represents the telephone number from where the call originated. The CLID is sometimes referred to as the ANI (Automatic Number Identification).

Typically, you would not use a CLID to define a Call Type. Rather, you would use a CLID prefix or CLID region.

For web requests, the CLID corresponds to the ApplicationString1 parameter.

E-mail requests do not use the CLID.

Use of CLID Prefixes

To define a Call Type based on the area code from where the call originated, you can use a CLID prefix.

For example, if you want to define a Call Type for all calls from the 508 area code, you specify 508 as the CLID prefix. You can further refine the Call Type to calls from a certain exchange within the area code. For example, you can specify 508486 as the CLID prefix. A routing script can then process the call based on the area code or local exchange of the caller.

Use of CLID Regions

You can define a Call Type that encompasses multiple CLID prefixes. For example, a useful Call Type could be defined as all calls from New York, which includes several area codes. To accomplish this, first define geographical regions through Unified ICM Configuration Manager. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*.

A routing script can then process the call based on the region of the caller.

Caller-Entered Digits (CED)

Caller-Entered Digits (CED) are numbers entered by the caller in response to prompts. For example, a caller might enter a number to indicate the type of service needed.

The caller can enter digits through the carrier network or the call center system. The Caller-Entered Digits can be used in defining the call's Call Type. A routing script can then process the call based on data entered by the caller.

Note You can differentiate between the case where the caller is not prompted for digits ("None Required") and the case where the caller is prompted but does not respond ("None Entered"). You can also choose "None" as the Caller-Entered Digits.

E-mail requests do not use the CED.

Data for Enterprise Chat and Email Web Requests

The Enterprise Chat and Email sends the following data to Unified ICM when requesting for routing a chat contact. These variables are mapped in the in.map.properties file to variables in the call form. For more

information about setting up ECE to send route requests to Unified ICM, see the administration documentation at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-email-interaction-manager/products-maintenance-guides-list.html.

- **dialednumber**—A string that determines which script to run on Unified ICM. The value of the script selector variable maps to the value of the Script Selector (which maps to the Dialed Number for voice contacts) in the Call Type created through Unified ICM Configuration Manager. Therefore, you must ensure that a Script Selector with the value that the ECE uses is set up in Unified ICM.

- **applicationstring1**—A optional string that you can use to select a Unified ICM routing script. The value of the ApplicationString1 variable corresponds to the Calling Line ID created through Unified ICM Configuration Manager. that determines which script to run on Unified ICM. The value of the script selector variable maps to the value of the Script Selector (which maps to the Dialed Number for voice contacts) in the Call Type created through Unified ICM Configuration Manager.

- **applicationstring2**—A optional string that you can use to select a Unified ICM routing script. The value of the ApplicationString2 variable corresponds to the Caller-Entered Digits created through Unified ICM Configuration Manager.

- **callvar1-10**—Unified ICM call variables, up to 10. These are optional fields that you can use to pass any application-specific information to Unified ICM.

- **eccvar1-2**—Expanded call context variables, used to pass additional information to the Unified ICM. You must set up ECC variables on the Unified ICM before the ECE can use them. For more information about creating ECC variables, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html.

Data for Enterprise Chat and Email E-mail Requests

The Enterprise Chat and Email sends the following data to Unified ICM when requesting that an e-mail message be routed:

- **Instance and skill group name** - A string that determines which script to run on Unified ICM. The value is the name of the ECE instance and the Unified ICM Routing skill group the message was assigned to, separated by a period; for example, "SupportInstance.techSupport". The string maps to the Script Selector value. Therefore, you must ensure that you set up the Script Selectors with the value of the ECE instance name and each Unified ICM Routing skill group in Unified ICM.

- **cisco.ece.Priority** - The priority of the message. The value of the Priority variable, which is "0" through "3"

  - "0" for Normal

Note The priority if set through the ECE rules. You can use the value of the variable to categorize the contact in a script.

  - "1" for High

  - "2" for Very High

- "3" for Urgent

- **cisco.ece.Category** - The categories of the message. The categories variable is an array containing up to 10 category values. The ECE administrator sets up the category values. You can use the value of the variable to categorize the contact in a script.

- **cisco.ece.MessageKey** - The unique identifier of the message that ECE is requesting Unified ICM to route. While you would not typically use the Message Key to categorize a contact in a script, it may be useful when Unified ICM is integrated with a CRM application; you could record the Message Key in the CRM database for future reference to e-mail correspondence with a customer.

Note These Expanded Call Context (ECC) variables are optional. Before you use these ECC variables, configure them for Unified CCE in the Configuration Manager's Expanded Call Variable List Tool and also in ECE.

For more information about Expanded Call Variables, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/cisco-enterprise-chat-email/products-maintenance-guides-list.html.

For more information about setting up ECE to send route requests to Unified ICM, see at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-guides-list.html.

Association of Contacts with Call Types

Following is the general process of how the system attempts to associate a contact with a call type:

1. If the dialed number, calling line ID, and caller-entered digits of the contact map to a defined call type, the system uses that call type:

2. If no call type matches the contact, the system uses the default call type for the routing client.

3. If no default call type is defined for the routing client, the system uses the general default call type.

   If no general default call type is defined, the system uses the default label defined for the dialed number.

   If no default label is defined for the dialed number, the system returns an error to the routing client.

Determination of Call Type for Voice Contact

The following example demonstrates how the system determines the call type for a voice contact and runs the appropriate script:

1. When configuring Unified ICM, you create a call type called "MASSACHUSETTS_SALES". This call type is defined as:

   - Having a dialed number of "NICClient1.8005551234".

   - Being from Massachusetts. This is determined by using a Calling Line ID Region, which consists of CLID Prefixes for all area codes in Massachusetts: 617, 508, 978, and 413.

- Being a Sales call. This is determined by a caller-entered digits value of "1", which is the number in the voice menu to indicate that the caller needs sales help.

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. A caller dials 1-800-555-1234, from the phone number 508-663-4958.

5. When prompted by a menu, the caller enters 1 to request sales help.

6. A route request is sent to Unified ICM.

7. Unified ICM examines the dialed number, which equals "18005551234".

8. Unified ICM evaluates the CLID value and determines that the CLID prefix is "508", which is an area code in Massachusetts.

9. Unified ICM examines the CED value, which is "1", which indicates that it is a Sales call.

10. Unified ICM determines that the call type is "MASSACHUSETTS_SALES" and runs the "MASSACHUSETTS_SALES_SCRIPT" script.

11. Unified ICM assigns the task to a particular agent.

Determination of Call Type for ECE Web Request

The following basic example demonstrates how the system determines the call type for a Enterprise Chat and Email chat web request:

1. When configuring the Unified ICM, you create a call type called "SSC_CT". This call type is defined as having a Script Selector (Dialed Number) of "SSC_DN".

2. When configuring ECE, you set the value of the script selector variable in the call form to "SSC_DN".

3. When configuring ECE, you set the dialednumber variable in the input map to equal the script selector variable in the call form.

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

5. You schedule the script to run for the "SSC_CT" call type.

6. A web user requests a chat session.

7. A route request is sent to Unified ICM.

8. Unified ICM determines that the Call Type is "SSC_CT" and runs the "SSC_SCRIPT" script.

9. Unified ICM instructs ECE to assign the task to a particular agent.

Determination of Call Type for ECE E-mail Contact

The following example demonstrates how Unified ICM determines the Call Type for a Enterprise Chat and Email e-mail contact:

1. When configuring Unified ICM, you create a Call Type called "EMAIL_CT". This Call Type is defined as having a Script Selector (Dialed Number).

2. When configuring ECE, you set the workflow rules to assign some messages to Unified ICM Routing queue "SalesQueue", which is mapped to an e-mail MRD.

3. You create a script called "EMAIL_SCRIPT". The e-mails are queued to skill group called Sales in the script.

4. You schedule the script to run for the "EMAIL_CT" Call Type.

5. A "Sales" message is assigned to the "SalesQueue" queue.

6. ECE sends a route request to Unified ICM.

7. Unified ICM determines that the Call Type is "EMAIL_CT" and runs the "EMAIL_SCRIPT" script.

8. Unified ICM instructs ECE to assign the task to a particular agent who is logged in to the Sales Skill group.

Determination of Call Type for a Digital Routing Task

This example is for a digital channel task from Webex Connect that uses the Digital Routing service. It demonstrates how the system determines the call type for the task and runs the appropriate script.

Note   Ensure that you have integrated Webex Connect with CCE and provisioned the digital channel capabilities. For more information, see the *Workflow for enabling and managing digital channel interactions* in the Cisco Unified Contact Center Enterprise Features Guide.

In this example, the task is a chat task.

*   When configuring CCE, you create a multichannel MRD called "Chat_Task_MRD". You create a call type called "Sales_Call_Type". You create a dialed number/script selector "Sales_Chat_DN", and associate it with "Chat_Task_MRD" and the "Sales_Call_Type" call type.

*   Map the media channel called "Chat" to the "Chat_Task_MRD" in the **Media Channel** page of the Unified CCE Administration portal. For instructions, see the *Set up media channels* section in the Cisco Unified Contact Center Enterprise Features Guide.

*   In Webex Connect, import the "Live_Chat_Inbound_Flow" template to set up a flow that presents a web form to the customer in a chat widget that is embedded in a website. Using the form the customer enters the details such as the Customer Name, Email Address, Phone Number and what is the query about, for example Sales. Set up the flow to pass the "Query" field of the response (in the flow template, this is defined as $(n38.inappmessaging.formFields.Query)) as the Script Selector value in the "CCE Create Task" node. Alternatively, use an evaluate node to map the user input to an appropriate Script Selector

/ Dialed Number in CCE, which would invoke the routing script as required. In the current example, the dialed number/script selector is "Sales_Chat_DN" in the "CCE Create Task" node. For more information about the "Live_Chat_Inbound_Flow" template, see Flow Configurations.

- You create a script called "Universal_Queue_script" that finds the longest available agent in the "Sales" skill group in the "Chat_Task_MRD" Media Routing Domain.

- You schedule the script to run for the "Sales_Call_Type" call type.

- A user requests a chat session from Webex Connect that triggers the "Live_Chat_Inbound_Flow", which then creates a task in CCE using the Digital Routing Service.

- Optionally, you can pass call variables and ECC variables while creating the task using the "CCE Create Task" node, which can then be used to either take routing decisions in the CCE script, or can be displayed to the agent on the Finesse Desktop.

- CCE uses the script selector to determine the call type, and runs the "Universal_Queue_script".

- CCE assigns the task to an agent who is logged into the "Sales" skill group in the "Chat_Task_MRD" Media Routing Domain.

Determination of Call Type for a Task Routing Task

This example is for a multichannel task from a third-party multichannel application that uses the Task Routing APIs. It demonstrates how the system determines the call type for the task and runs the appropriate script. In this example, the task is a chat task.

1. When configuring CCE, you create a multichannel MRD called "Chat_Task_MRD". You create a call type called "Chat". You create a dialed number/script selector "Chat_DN", and associate it with "Chat_Task_MRD" and the "Chat" call type.

2. You create a Customer Collaboration Platform Chat application from which a user can request to chat with an agent. You set the value of the script selector in the chat form to the "Chat_DN" dialed number.

3. You create a script called "Universal_Queue_script" that finds the longest available agent in the "Sales" skill group in the "Chat" MRD.

4. You schedule the script to run for the "Chat" call type.

5. A user requests a chat session from the Customer Collaboration Platform Chat application, using the Customer Collaboration Platform Task API.

6. Customer Collaboration Platform submits the task request to CCE, including the dialed number/script selector.

7. CCE uses the script selector to determine the call type, and runs the "Universal_Queue_script".

8. CCE assigns the task to an agent who is logged into the "Sales" skill group in the "Chat" MRD.

C H A P T E R 3

Contact Categorization

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.

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 solutions for different customer's needs.

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 run, 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 to static by using the Call Type node in a script. The Call Type node is in the Routing tab of the Palette.

Figure 16: The Call Type Icon

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

Figure 17: Call Type Properties Dialog Box - Static Call Type
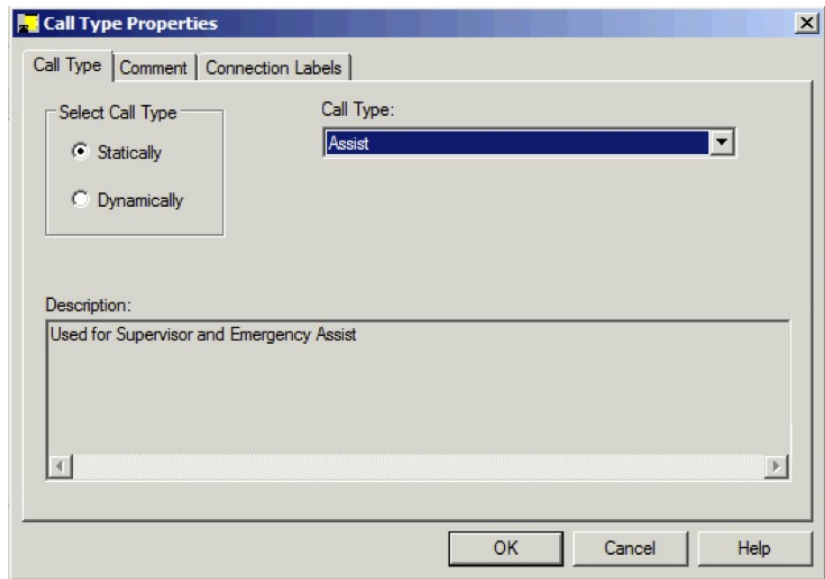
To define a static call type node, complete the following steps.

Procedure

Step 1	In the Call Type tab, click the **Statically** radio button.
Step 2	From the Call Type list, click the call type to assign to the contact.

What to do next

Warning	The Call Type node changes the call type and continues running the current script. The Requalify Call node stops running the current script and runs a new script associated with that call type.

Change Call Type to Dynamic

You can change the call type of a contact to dynamic by using the Call Type node in a script. The Call Type is on the Routing tab of the Palette.

Figure 18: Call Type Icon

Note 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 19: Call Type Properties Dialog Box - Dynamic Call Type
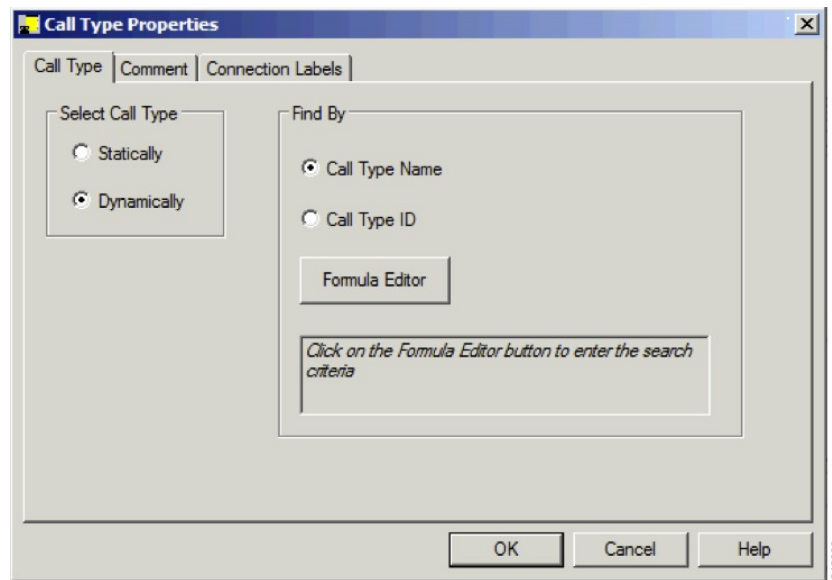
To define a dynamic call type node, complete the following steps.

Procedure

Step 1	On the call type tab, select the **Dynamically** radio button.
Step 2	To dynamically change the call type of a contact by call type name, In the Find By section, select the **Call Type Name** radio button.
Step 3	To dynamically change the call type of a contact by call type ID, In the Find By section, select the **Call Type ID** radio button.
Step 4	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.

What to do next

Warning The Call Type node changes the call type and continues the current script execution. The Requalify Call node terminates the current script execution and runs a new script associated with that call type.

Change Call Type and Run a New Script

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

Figure 20: The Requalify Call Icon

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

Figure 21: The Requalify Properties - Requalify Call Tab
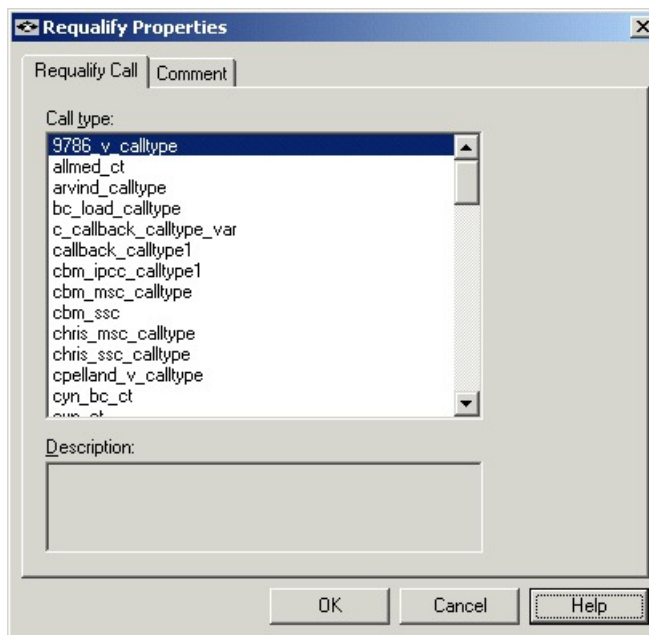
Define Requalify node properties as follows:

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

Warning The Call Type node changes the call type and continues running the current script. The Requalify Call node stops running the current script and runs 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 three call type qualifiers:

- Dialed number

- Calling line ID (CLID)

- Caller-entered digits (CED)

When the system determines a contact's call type based on these qualifiers, it runs the associated script.

However, after the script is run, you can further categorize the contact based on the values of the call type qualifiers.

For example, a call type may be defined as having a calling line ID that includes calls from all area codes in New England states. When the script associated with that call type is run, it examines the specific area code in the calling line ID and branch differently depending on the value and therefore, treating calls from different area codes in New England differently.

You can categorize a contact based on its call type qualifiers by using one or more of the following nodes:

- Dialed Number (DN) Node

- Calling Line ID (CLID) Node

- Caller Entered Digits (CED) Node

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).

Figure 22: The Dialed Number Icon

Following is the DN Properties dialog box of the Dialed Number node:

Figure 23: DN Properties - Dialed numbers Tab
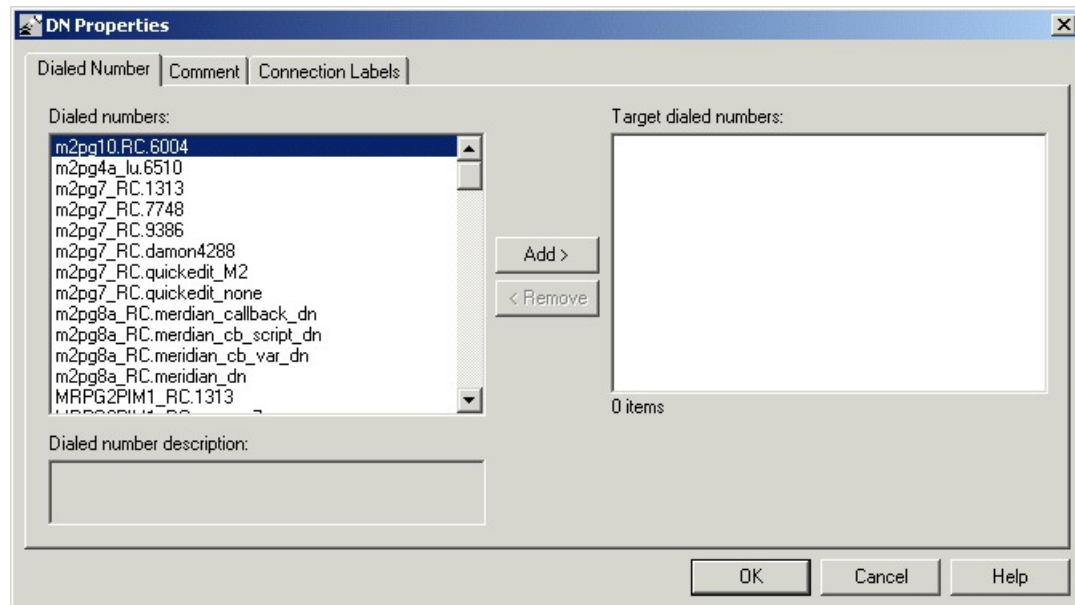
You can define the Dialed Number node properties as follows:

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.

Categorize Contact by Calling Line ID

You can categorize a contact based on its Calling Line ID by using the CLID node (in the Routing tab of the
Palette).

Figure 24: The CLID Icon

The CLID node tests the billing telephone number of the contact to see if it matches:

- A specific (CLID) that you provide (for example, a specific region, area code, or an area code plus a
  local exchange).
- A variable expression.

By default, Unified ICM compares the list of values against the calling line ID of the contact. However, you
can specify another value or expression to be used instead of the CLID.

![pencil icon]

| **Note** | For Web Collaboration requests, the CLID maps to the applicationstring1 variable. |

Before defining CLID node properties, insert one or more targets and connections from the CLID node.

Define CLID node properties as follows:

Procedure

Step 1 In the Calling Line ID tab:

Figure 25: CLID Properties - Calling Line ID
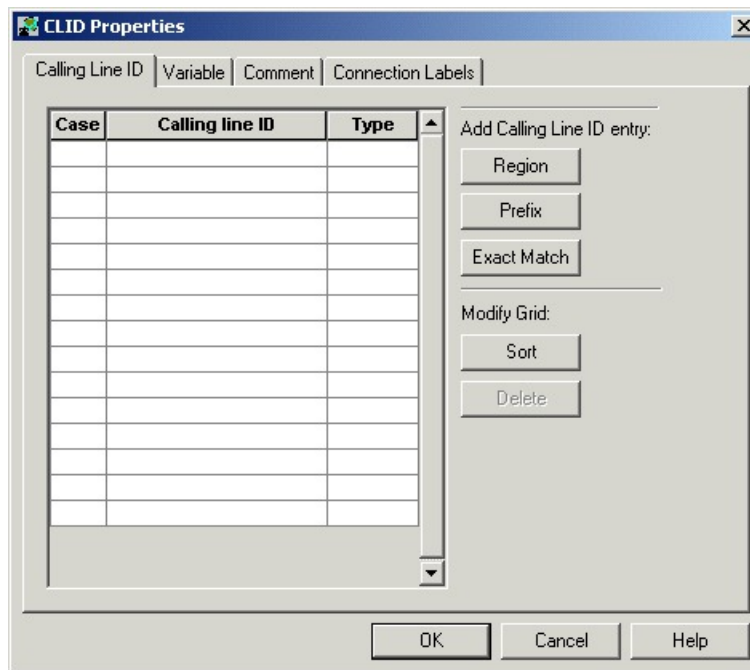
a) For each branch to a different target, select the Case (the number displayed on each success connection branch); for example, 1, 2, and 3.
b) For each Case, select the Type of match. Click **Region**, **Prefix**, or **Exact Match**.
c) For each Case, enter the Calling line ID to match. If you selected Region, choose a region from the drop-down list; for example, California. If you selected Prefix, enter the Prefix number. If you selected Exact Match, enter the Calling line ID to match.
d) To sort the list by case, click **Sort**.
e) To delete a row, select the row and click **Delete**.

Step 2 In the Variable tab:

Figure 26: CLID Properties - Variable Tab
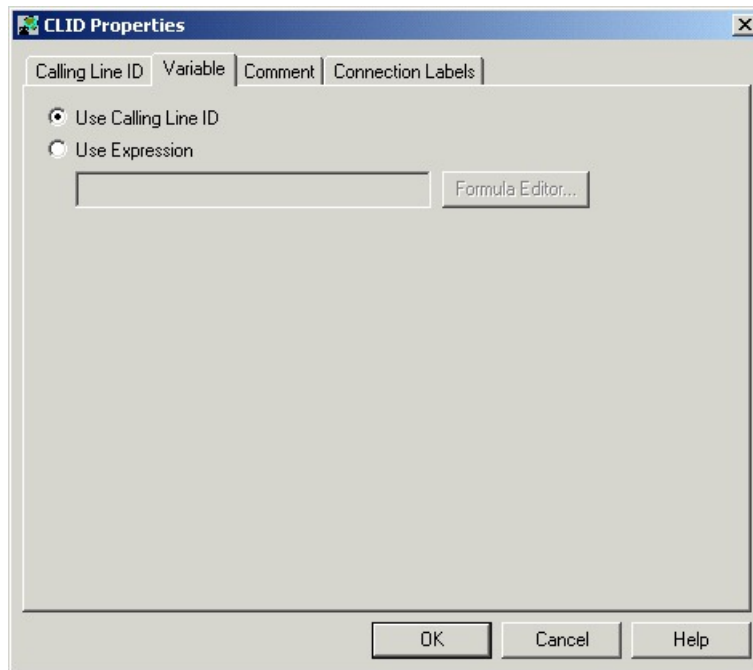
a) By default, **Use Calling Line ID** is selected to have Unified ICM compare the list of values you define in the Calling Line ID tab against the calling line ID of the contact.

b) To have Unified ICM use the value of an expression instead of the Calling Line ID, select **Use Expression** and enter the expression directly or click **Formula Editor** to use a formula to define the expression.

Step 3 Optionally, add comments and connection labels.

What to do next

Note If you delete a connection associated with a Case, the Case information you specified in the Properties dialog box is also deleted.

Categorize Contact by Caller-Entered Digits (CED)

You can categorize a contact based on the Caller-Entered Digits by using the CED Node (in the Routing tab of the Palette).

Figure 27: CED Properties Icon

For example, you can define the CED node to send contacts to:

• SkillGroupA if the Caller-Entered Digits match a specific string

• SkillGroupB if there are no Caller-Entered Digits

✎

Note   For Enterprise Chat and Email requests, the CED maps to the applicationstring2 variable.

Following is the CED Properties dialog box of the CED node:

Figure 28: CED Properties
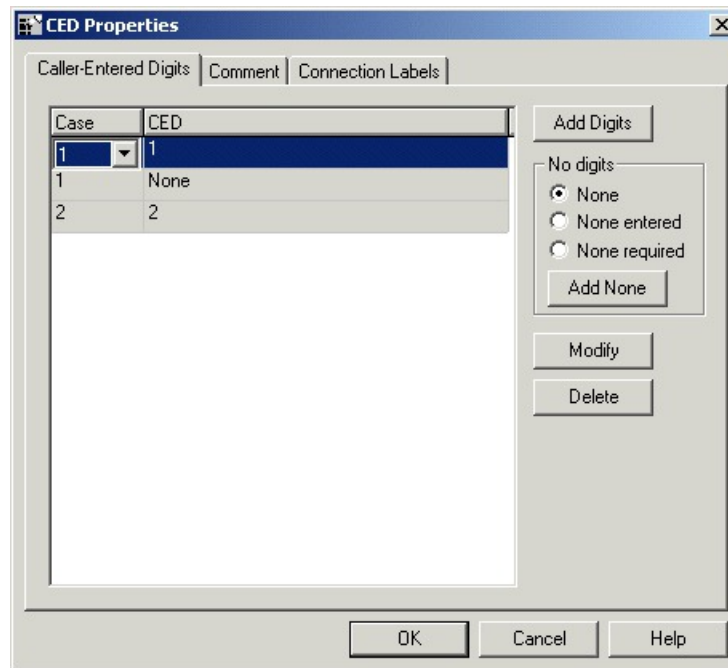
You must insert targets and connections from the Caller-Entered-Digits node before you can define the node's properties.

Then define CLID node properties as follows:

Procedure

Step 1   Click **Add Digits** to add a new CED value for a branch. In the new row, add a CED value and select the branch number. You can associate one or more CED values with each connection. Valid characters are: digits 0 through 9, asterisk (*), and number sign (#).

Step 2   Click **Add None** to add the value of None to a connection, specifying that it matches only those cases where no digits are entered or where no digits are required. When you specify **None**, you can select the following options:

a)   None, to include both None Entered and None Required situations.
b)   None Entered, to apply when the caller was prompted for digits but did not enter any.
c)   None Required, to apply when the caller was not prompted for digits.

What to do next

Note   If you delete a connection associated with a Case, the Case information you specified in the Properties dialog box is also deleted.

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 is run, 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 Central Controller VM.

For example, a call type named "CHAT_CT" may be defined to include all chat web requests. A script named "CHAT_SCRIPT" runs every time a contact with the call type "CHAT_CT" is received. Typically, this script instructs Enterprise Chat and Email 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 to report to improved weekend activity. Therefore, for chat web requests received on Saturday or Sunday, the script branches differently and instructs Enterprise Chat and Email 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 the 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 Unified ICM Central Controller. Following is the Time Properties dialog box of the Time node.

Figure 29: Time Icon

Following is the Properties dialog box of the Time node:

Figure 30: Time Properties
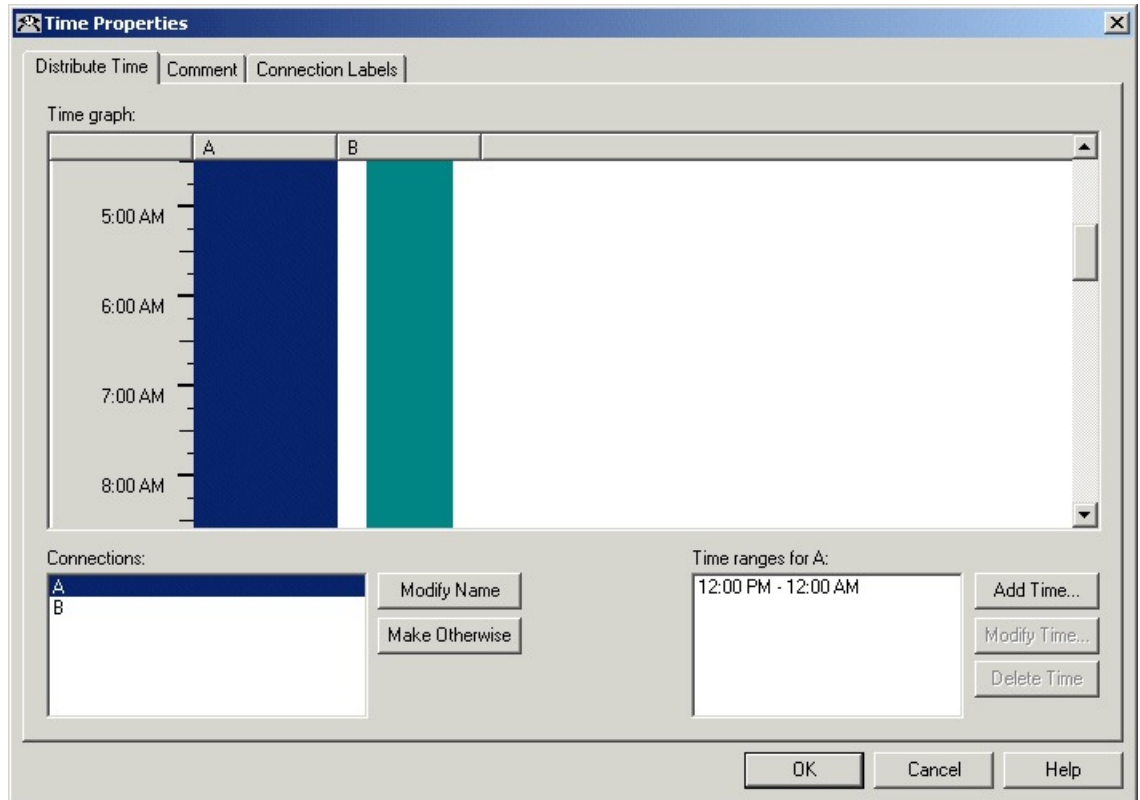
You must insert targets and connections from the Time node before you can define the node's properties.

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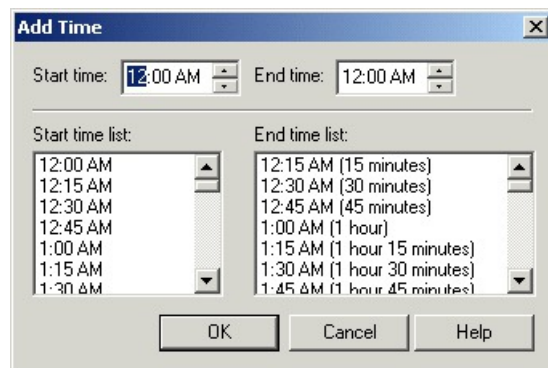
Step 2 To delete a time associated with the branch, select the time and click **Delete Time**.

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

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

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.).

Figure 32: Day of Week Icon

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

Figure 33: Day of Week Properties

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.

What to do next

Note   If you delete a connection, the day of the week information you specified in the Properties dialog box is also deleted.

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.

Run 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 Unified ICM encounters a Go To Script node, it stops running the current script and starts the script indicated in the node.

Figure 34: Go To Script Icon

For example, you might have several scripts that check for exception conditions and, if none are found, run 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 35: Go To Script Properties
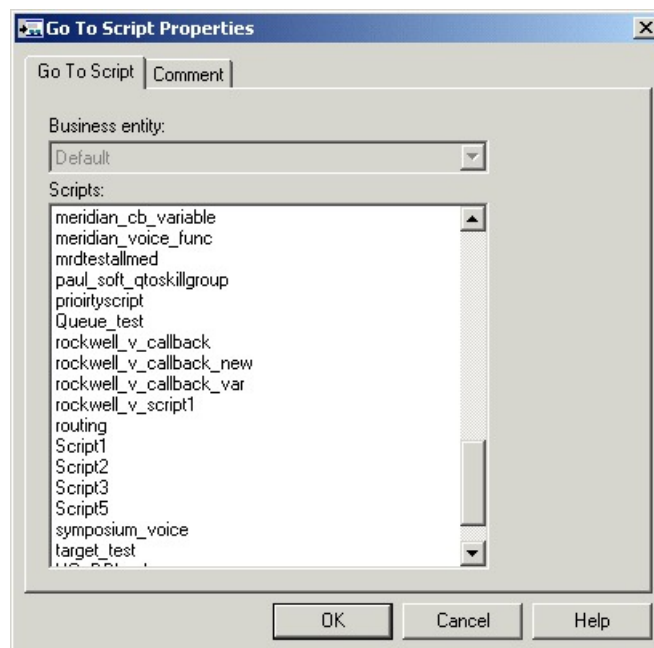
Define "Go to Script" node properties as follows:

Procedure

Step 1 Select the Business entity that owns the script that the node should run. By default, Unified ICM consists of one business entity. Multiple business entities are allowed only if you enable partitioning. For more information about partitioning, see the *Security Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html.

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 a Specific Branch

You use the Switch node (in the General tab of the Palette) to direct script execution to its active output connection. You can define multiple output connections from the Switch node; you must specify one of them as the active connection.

Figure 36: Switch Icon

Following is the Properties dialog box of the Switch node:

Figure 37: Switch Properties
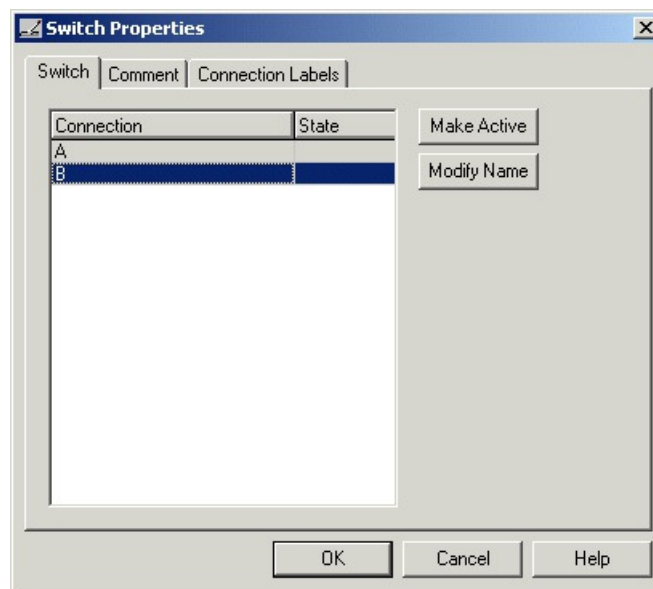
You must insert targets and connections from the Switch node before you can define the node's properties.

Then define Switch node properties as follows:

Procedure

Step 1   By default, Connections are labeled "A", "B", etc. To re-label a Connection, click **Modify Name** and make changes to the name.

Step 2   To make a connection active, select a connection and click **Make Active**.

Note   Only one connection can be active at any time. To change the active branch, you must re-edit the script and create a new script version.

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 38: 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.

Warning   Unlike selecting targets by rules or distributing contacts to targets, distributing contacts does not consider real-time contact center conditions and therefore may lead to load imbalances.

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

Figure 39: % Allocation Properties
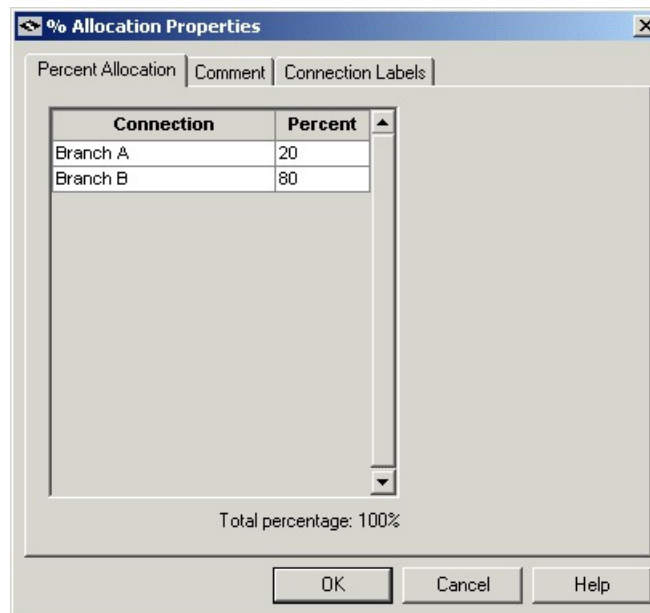
Define Percent Allocation node properties as follows:

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 40: If Icon

When the system runs an If node, it first evaluates the condition specified in the node Properties dialog box Define condition field. If the system 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 41: If Properties
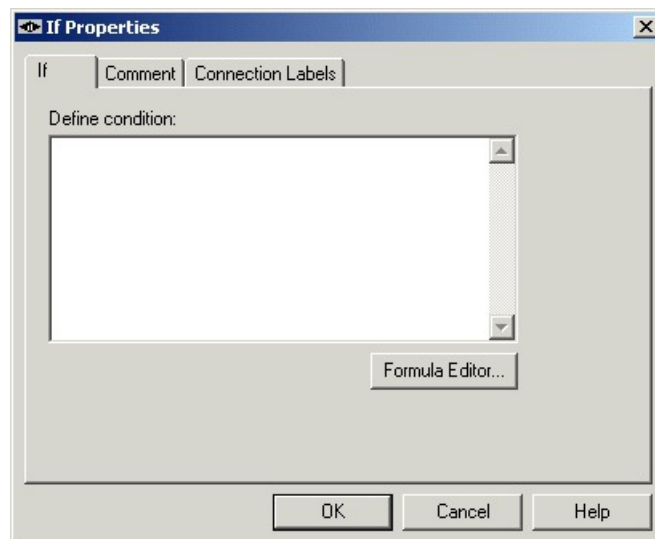
Define If node properties as follows:

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 Queue section of this document.

Categorization by External Data

Scripts can categorize a contact based on data stored in a database that is not part of Unified ICM, such as a Customer Relationship Management (CRM) system.

For example, a script that processes incoming phone calls can query a CRM using the CLID to determine if the customer should receive Premium or Standard phone support. The records for the customer with the phone number matching the CLID are retrieved, and the value of the Support column is returned to the script. If the value indicates that the customer has paid for Premium support, script processing continues down one branch that assigns the phone call to a skill group dedicated to responding to Premium customers; otherwise, script processing continues down another branch that assigns the phone call to a more general skill group, where the wait time is expected to be longer.

Note   You must use Unified ICM Configuration Manager to define the external database table and the columns that are to be referenced in scripts. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html.

You can categorize a contact by retrieving external data with the DB Lookup node, then referencing the retrieved external data on the DB Lookup node success branch.

Modify CallRouter Registry to Provide for Database Lookup Authentication

By default, when attempting to access a remote database, the CallRouter authenticates itself as:

To access the remote database, the call router authenticates using external database lookup value in SQLLogin registry key located at HKEY_LOCAL_MACHINE\SOFTWARE\Cisco.System,INC.\ICM\<instancename>\RouterA\Router\CurrentVersion\Configuration\Database.

Update the External DBLookUp registry value using the CCEDataProtect Tool. For more information, see **Configure External DBLookUp Registry Value using CCEDataProtect Tool** procedure in the *Administration Guide for Cisco Unified Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-maintenance-guides-list.html.

You must specify the following registry value in the CCEDataProtect Tool on the CallRouter for SQLLogin registry key.

```
\\<DBMachine>\<DBName>=(<DB_username>,<DB_password>)
```

Use a comma as the delimiter to separate data for separate databases.

The following example shows login credentials for two external databases:

```
\\MachineA\DatabaseA=(username,password),\\MachineB\DatabaseB=(username,password).
```

☞

Important The SQLLogin key does not support passwords that contains the following characters:

'/', '\', '=', '(', ')', or ','

You must configure the database paths to Side A and Side B in the DB Lookup Explorer and the CallRouter Registry for the Database Lookup Authentication.

✏️

Note The database paths in the DB Lookup tool and the CallRouter Registry must be the same. Use either hostname or IP addresses in the DB Lookup Explorer and the CallRouter Registry. If you use hostname in the DB Lookup Explorer tool, use hostname in CallRouter Registry as well. If you use IP Address in the DB Lookup Explorer tool, use IP Address in CallRouter Registry as well.

✏️

Note If the external remote database is on SQL Server 2017 version, you have to install the ODBC Driver 17 manually on the server hosting the external database. Download the ODBC Driver 17 from Microsoft.

Categorize Contact by External Data

You use the DB Lookup node (in the General tab of the Palette) to query a specific row of data from an external database. You can then reference columns from that row in other nodes like an If node.

Figure 42: DB Lookup Icon

Following is the Properties dialog box of the DB Lookup node:

Figure 43: DB Lookup Properties
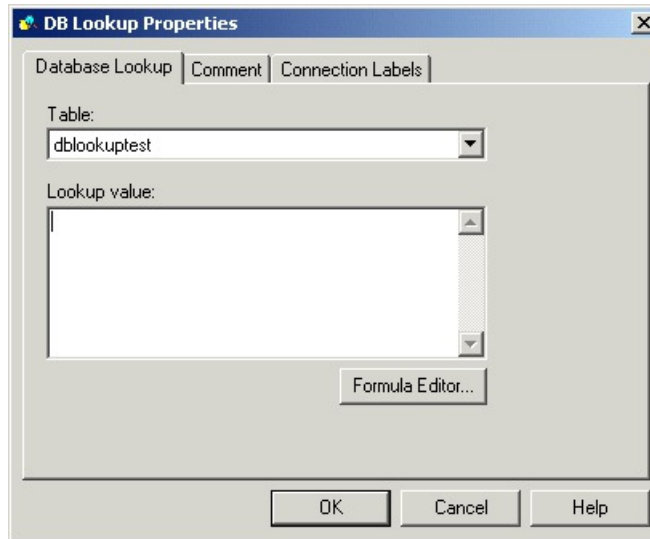
Define the DB Lookup node properties as follows:

Procedure

Step 1   Select the database table you want to query. (The drop-down list contains the enterprise names of all lookup tables defined in Unified ICM.)

Note	Define all integer fields in tables accessed by a DBLookup node as NOT NULL. Only the following data types are supported for SQL databases: SQLINT1 (tinyint), SQLINT2 (smallint), SQLINT4 (int), SQLCHAR (char), SQLVARCHAR (varchar), SQLFLT4DBFLT4 (real), SQLFLT8DBFLT8 (float), and SQLDATETIME (datetime). You must define all fields except SQLDATETIME, SQLVARCHAR, and SQLCHAR as NOT NULL fields. You can define these three fields as NULL.

Step 2   In the Lookup value field, define constant or expression to match the primary key value in the row you want to retrieve. The lookup value must be of the appropriate data type to match the primary key field in the table. You can use formulas to define the expression. For example, if phone_number is the primary key field in the database table, you might use Call.CallingLineID as the Lookup Value.

Step 3   Optionally, add comments and connection labels.

Reference Retrieved External Data

To access retrieved data, you use the following syntax in nodes that reference a column:

```
Database.table-name.column-name
```

Where:

Procedure

- `table-name` is the enterprise name of the table as defined through the Configuration Manager.
- `column-name` is the name of the column from the table, which is also defined through the Unified ICM Configuration Manager.

What to do next

For example, if the table Customers contains a column named Priority, you would reference that column in an If expression as follows:

```
Database.Customers.Priority = 1
```

For more information, see the chapter Use of Formulas.

Categorize by External Applications

You can categorize a contact based on data returned from an application external to Unified ICM by using the Application Gateway node (in the General tab of the Palette).

Figure 44: Gateway Icon

For example, a script that processes incoming phone calls can send the caller's account number to an external application, which returns to the script the caller's account balance. The script can then branch on the value of the account balance, providing premium service to callers with higher account balances.

Note   You must use Unified ICM Configuration Manager to define the external application. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*.

Define the Application Gateway properties as follows:

Procedure

Step 1   In the Send tab:

Figure 45: App Gateway Properties - Send
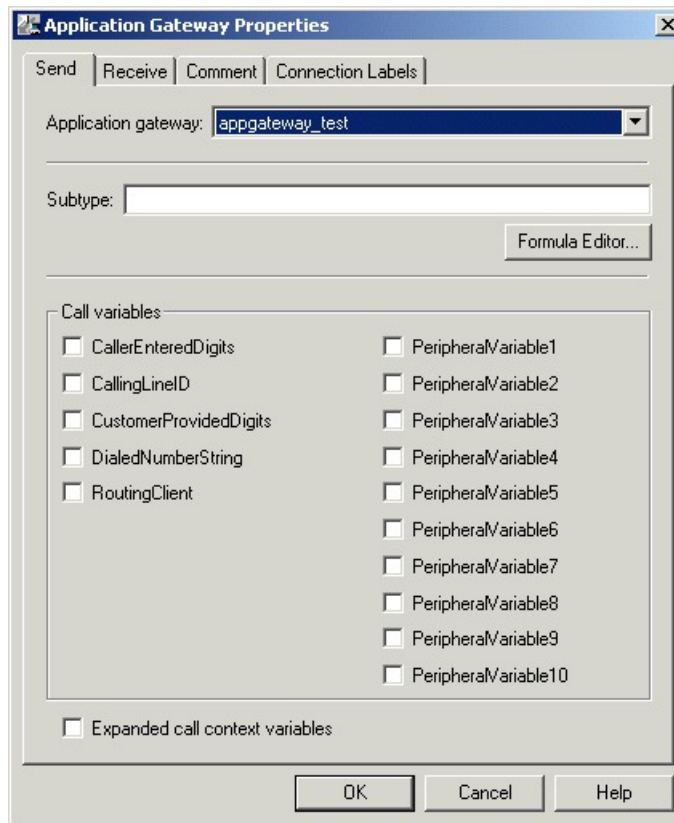
a) Choose the gateway from the Application Gateway drop-down list.

b) In the Subtype field, enter the string that is to be sent to the external application, or use the Formula Editor to write an expression that evaluates to a string.

c) In the Call variables list, check the call variables to send to the external application.

d) To send expanded call variables to the external application, check **Expanded call context variables**.

Step 2   In the Receive tab:

Figure 46: App Gateway Properties - Receive
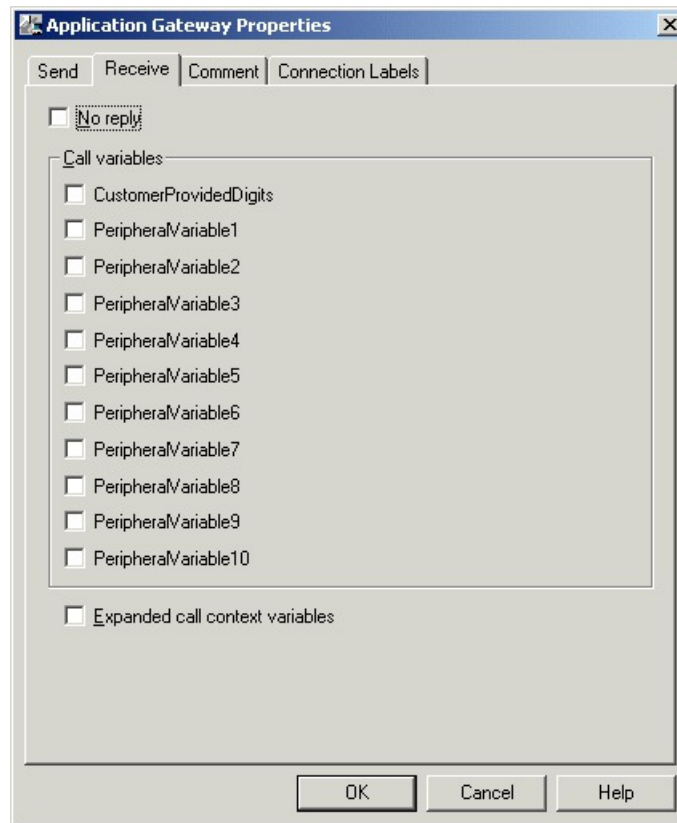
a) Select **No Reply** if the external application is not to return data to the script.

Note   If you select this option, Unified ICM cannot retrieve any data from the external application.

b) In the Call variables list, check variables that the external application may modify.

c) Select Expanded call context variables if the external application modifies and returns values for the expanded call variables.

Step 3   Optionally, add comments and connection labels.

Selection of Routing Targets

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.

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

Peripheral-level skill targets include:

- Agents

- Skill groups

There are two types of routing targets: Skill targets and Network targets.

Routes

A value returned by a routing script that maps to a target at a peripheral, such as a service, skill group, agent, or translation route to a label.

> ✎
>
> **Note** You create routes through Unified ICM Configuration Manager before writing routing scripts. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html.

Translation Routes

A translation route is a target at a peripheral that does not map to a specific service, skill group, or agent. When a contact arrives with the trunk group and DNIS that correspond to a translation route, the Peripheral Gateway (PG) is responsible for determining the ultimate target.

When Unified ICM routes a call to a translation route, it sends a message to the PG. In this message the PG receives the call context from the Router, including the ultimate target and further instructions for the PG. This enables the PG to seamlessly pass information set or retrieved in the Routing script to the PIM/peripheral as well as to CTI clients. For example, the PG might be instructed to coordinate with a host computer so that the caller's account number appears on the teleset of the agent who picks up the call.

> ✎
>
> **Note** You create translation routes through Unified ICM Configuration Manager before writing routing scripts. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html.

Target Sets

A target set is a list of possible targets. During script processing, the actual target is chosen from the set by the preceding node on the script branch, a Select Node or Distribute Node.

Skill Targets

A skill target is an entity at a peripheral or in the enterprise to which Unified ICM can route a contact. There are two types of skill targets: **Peripheral-level skill targets** and **Enterprise-level skill targets**.

Peripheral-level skill targets include:

- Agents

- Skill groups

- Services

Enterprise-level skill targets include:

- Enterprise skill groups

- Enterprise services

Agent Routing Nodes

The following nodes available for agent routing:

1. Queue to Agent Node. For more information, see

2. Agent to Agent Node. For more information, see

3. Agent Node.

Define Agent Node Properties

You define a set of agents who can receive the contact by using the Agent node in the Targets tab of the Palette.

Note Do not use this node in Unified CCE environments. It is intended for use with ACDs.

Figure 47: The Agent Icon

The script can determine the target agent from the set by one of the following methods:

- Selecting the target by rules (Select node)
- Distributing contacts to targets in the set (Distribute node)
- A combination of selecting the target and distributing contacts (Route Select node)

Following is the Properties dialog box of the Agent node:
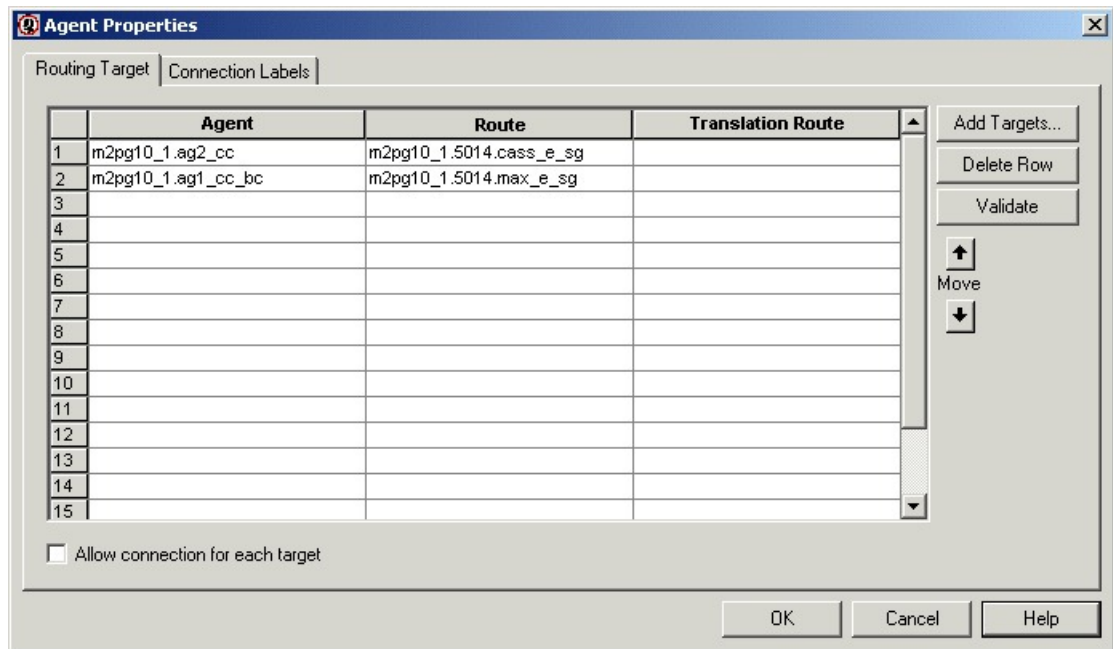
Define Agent node properties as follows:

Procedure

Step 1   For each agent in the target set the following:

a)   In the Agent column, for each row used, select the agent to which the contact can be routed. You can use the drop-down list for each table cell, or select multiple agents by clicking **Add Targets** and using the dialog box that opens to select multiple agents.

b)   In the Route column, select the route that maps to a specific target at the peripheral.

c)   Optionally, in the Translation Route column, select a translation route.

Step 2   Optionally, check Allow connection for each target to have an output terminal appear to the right of each individual target defined in the node. Control passes through this terminal when the associated target is chosen. When the script terminates, the route for the selected agent is still used.

Step 3   Click **Validate** to check whether the targets you defined are valid. Correct any errors that are flagged.

Step 4   Optionally, add connection labels.

Define Set of Skill Groups to Receive the Contact

You define a set of skill groups that can receive the contact using the Skill Group node in the Targets tab of the Palette.

Figure 49: The Skill Group Icon

You can specify the skill groups in the set in two ways:

- Select **Explicit target references** to specify static skill group configurations, including the Skill Group, Route, and, optionally, a Translation Route. See Define Skill Groups by Explicit Target Reference, on page 55.

- Select **Lookup target references by expression** to enter formulas that are resolved to skill group references at runtime. See Define Skill Groups Using an Expression, on page 57.

From the defined set, your script can determine the target skill group by one of the following methods:

- Selecting the target by rules (Select node)
- Distributing contacts to targets in the set (Distribute node)
- A combination of selecting the target and distributing contacts (Route Select node)

Define Skill Groups by Explicit Target Reference

On the Skill Group node in the Targets tab of the Palette, select **Explicit target references** to specify the skill groups that can receive the contact:

Figure 50: Skill Group Properties - Explicit Target References

Define Skill Group node properties as follows:

Procedure

Step 1 For each skill group in the target set the following:
   a) In the Skill Group column, for each row used, select the skill group to which the contact can be routed. You can use the drop-down list for each table cell, or select multiple skill groups by clicking **Add Targets** and using the dialog box that opens to select multiple skill groups.
   b) In the Route column, select the route that maps to a specific target at the peripheral.
   c) Optionally, in the Translation Route column, select a translation route.

Step 2 Optionally, check **Allow connection for each target** to have an output terminal appear to the right of each individual target defined in the node. Control passes through this terminal when the associated target is chosen. When the script terminates, the route for the selected skill group is still used.

Step 3 Click **Validate** to check whether the targets you defined are valid. Correct any errors that are flagged.

Step 4 Optionally, add connection labels.

Define Skill Groups Using an Expression

On the Skill Group node in the Targets tab of the Palette, select **Look up target references by expression** to enter expressions to select Skill Groups. The formulas are evaluated dynamically at run-time to determine the actual skill groups.

Figure 51: Skill Group Properties - Look up Target References by Expression
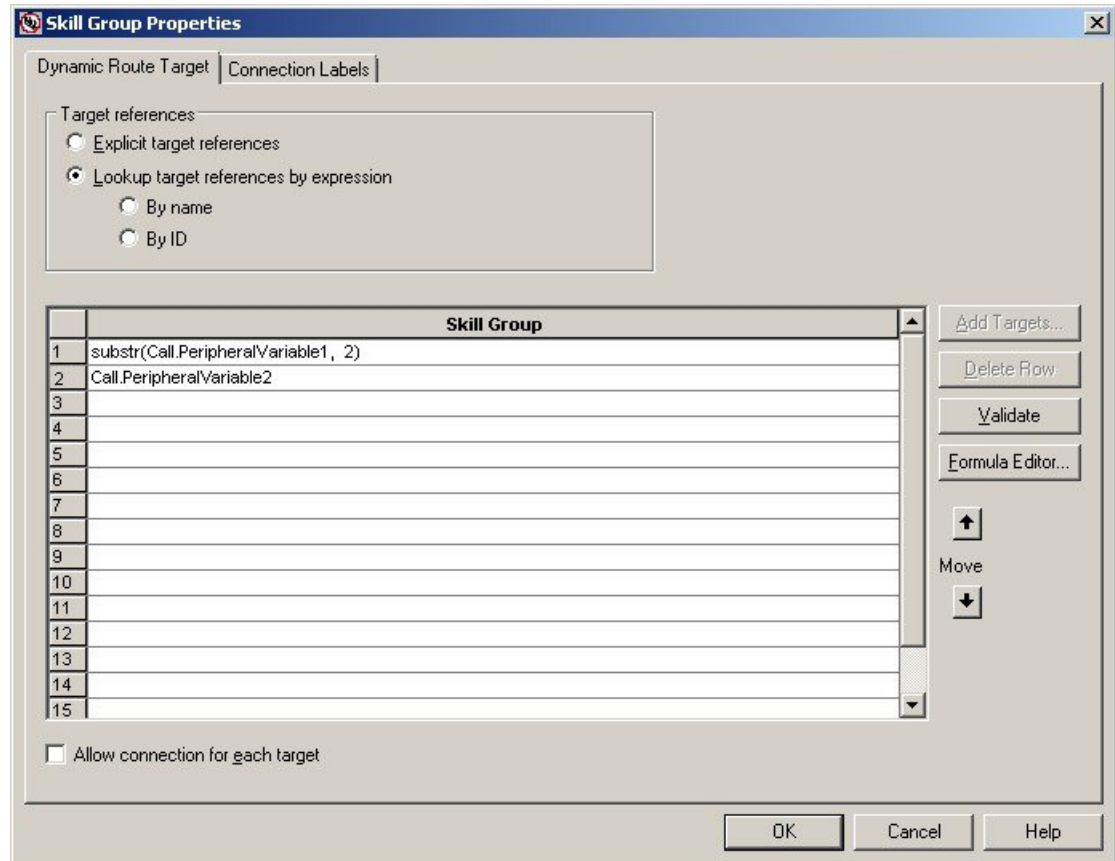
Procedure

Step 1	Select **Look up target references by expression**
Step 2	Select whether to select the skill groups by name or ID:
	• Select **By name** to select a skill group by the skill group enterprise name.
	• Select **By ID** to select the skill group by the skill group database ID.
Step 3	Write your formulas to select skill groups on the Skill Group page.
	Click **Formula Editor** for help building a formula.
Step 4	Optionally, click **Validate** to check for errors in the expression.
Step 5	Click **OK** to save and close the dialog.

When you save the dialog, the configured expressions are validated. If errors are found, the dialog displays a warning and highlights the rows with errors.

When you use a ScriptTarget node with dynamic skill group configuration as a target node for a ScriptSelect or ScriptDistribute node, the router evaluates all the targets of the ScriptTarget node and then returns only valid targets. If the skill group expression cannot be evaluated to a valid skill group object, the node continues on the failure branch and the router logs an error message.

> **Note** **Look up Target References by Expression** only supports skill groups that target Unified CCE peripherals. Expressions that evaluate to skill groups on any other type of peripheral are not considered valid targets.

Define a Set of Queue to Skill Groups to Receive the Contact

You define a set of queue to skill groups that can receive the contact using the Queue node in the Queue tab of the Palette.

Figure 52: The Queue Icon

You can specify the queue to skill groups in the set in two ways:

- Select **Explicit target references** to specify static queue to skill group configurations, including the Skill Group, Consider If, Route, and, optionally, a Translation Route.

- Select **Lookup target references by expression** to enter formulas that are resolved to queue to skill group references at runtime.

Related Topics

Define Queue to Skill Groups by Explicit Target Reference

You can select explicit target references to specify the skill groups that can receive the contact.

Define Queue to Skill Group node properties as follows:

Procedure

Step 1 From the **Queue** tab of the Palette, click **Queue** to create a **Queue to Skill Group** node.

Step 2 From the node, right-click and select **Properties**.

Step 3 Select the **Explicit target references** option.

Step 4 For each skill group in the target, set the following:

a) In the Skill Group column, for each row used, select the skill group to which the contact can be routed. You can use the drop-down list for each table cell, or select multiple skill groups by clicking **Add Targets** and using the dialog box that opens to select multiple skill groups.

b) In the Consider If column, add an expression to one or more rows. If a Consider If expression returns True, the expression considers the skill group and queues there.

c) In the Route column, select the route that maps to a specific target at the peripheral.

d) Optionally, in the Translation Route column, select a translation route.

Step 5 Click **Validate** to check whether the targets you defined are valid. Correct any errors that are flagged.

Define Queue to Skill Groups Using an Expression

You can look up target references by expression to enter expressions to select Queue to Skill Groups. The formulas are evaluated dynamically at run-time to determine the actual skill groups.

Procedure

Step 1 From the **Queue** tab of the Palette, click **Queue** to create a **Queue to Skill Group** node.

Step 2 From the node, right-click and select **Properties**.

Step 3 Select **Lookup target references by expression**

Step 4 Select whether to choose the skill groups by name or ID:

- Select **By name** to choose a skill group by the skill group name.

- Select **By ID** to choose the skill group by the skill group database ID.

Step 5 Write your formulas to select skill groups on the Skill Group page.

Click **Formula Editor** for help building a formula.

Step 6 Optionally, click **Validate** to check for errors in the expression.

Step 7 Click **OK** to save and close the dialog.
When you save the dialog, the configured expressions are validated. If errors are found, the dialog displays a warning and highlights the rows with errors.

When you use a queue to skill group node with dynamic queue to skill group configuration as a target, the router evaluates all the targets of the queue to skill group node and then returns only valid targets. If the queue to skill group expression cannot be evaluated to a valid queue to skill group object, the node continues on the failure branch and the router logs an error message.

Note **Look up Target References by Expression** only supports skill groups that target Unified CCE peripherals. Expressions that evaluate to skill groups on any other type of peripheral are not considered valid targets.

Define Set of Services to Receive the Contact

You define a set of services that can receive the contact by using the Service node in the Targets tab of the Palette.

Figure 53: The Service Icon

The script can determine the target service from the set by one of the following methods:

- Selecting the target by rules (Select node)

- Distributing contacts to targets in the set (Distribute node)

- A combination of selecting the target and distributing contacts (Route Select node)

Following is the Properties dialog box of the Service node:

Figure 54: Service Properties - Routing Target Tab

Define Service node properties as follows:

Procedure

Step 1   For each service in the target set the following:

a) In the Service column, for each row used, select the service to which the contact can be routed. You can use the drop-down list for each table cell, or select multiple services by clicking **Add Targets** and using the dialog box that opens to select multiple services.

b) In the Route column, select the route that maps to a specific target at the peripheral.

c) Optionally, in the Translation Route column, select a translation route.

Step 2   Optionally, check **Allow connection for each target** to have an output terminal appear to the right of each individual target defined in the node. Control passes through this terminal when the associated target is chosen. When the script terminates, the route for the selected service is still used.

Step 3   Click **Validate** to check whether the targets you defined are valid. Correct any errors that are flagged.

Step 4   Optionally, add connection labels.

Related Topics

Targets Tab, on page 4

Define Set of Enterprise Skill Groups to Receive the Contact

You define a set of enterprise skill groups that can receive the contact by using the Enterprise Skill Group node in the Targets tab of the Palette.

Figure 55: The Enterprise Skill Group Icon

The script can determine the target enterprise skill group from the set by one of the following methods:

• Selecting the target by rules (Select node)

• Distributing contacts to targets in the set (Distribute node)

• A combination of selecting the target and distributing contacts (Route Select node)

Following is the Properties dialog box of the Enterprise Skill Group node:

Figure 56: Enterprise Skill Group Properties - Routing Tab

Define Enterprise Skill Group node properties as follows:

Procedure

Step 1   From the Business Entity drop-down list, select the business entity for the enterprise skill groups.

Step 2   From the Enterprise target drop-down list, select the enterprise target for the enterprise skill groups.

Step 3   For each enterprise skill group in the target set the following:

   a)   In the Skill Group column, for each row used, select the enterprise skill group to which the contact can be routed.

   b)   In the Route column, select the route that maps to a specific target at the peripheral.

   c)   Optionally, in the Translation Route column, select a translation route.

Step 4   Optionally, check **Allow connection for each target** to have an output terminal appear to the right of each individual target defined in the node. Control passes through this terminal when the associated target is chosen. When the script terminates, the route for the selected enterprise skill group is still used.

Step 5   Click **Validate** to check whether the targets you defined are valid. Correct any errors that are flagged.

Step 6   Optionally, add connection labels.

Define Set of Enterprise Services to Receive the Contact

You define a set of enterprise services that can receive the contact by using the Enterprise Service node in the Targets tab of the Palette.

Figure 57: The Enterprise Service Icon

The script can determine the target enterprise service from the set by one of the following methods:

- Selecting the target by rules (Select node)
- Distributing contacts to targets in the set (Distribute node)
- A combination of selecting the target and distributing contacts (Route Select node)

Following is the Properties dialog box of the Enterprise Service node:

Figure 58: Enterprise Service Properties - Routing Target tab

Define Enterprise Service node properties as follows:

Procedure

Step 1 From the Business Entity drop-down list, select the business entity for the enterprise services.

Step 2   Choose the enterprise target for the enterprise services from the Enterprise target drop-down list.

Step 3   For each enterprise service in the target set the following:

a) In the Service column, for each row used, select the enterprise service to which the contact can be routed.

b) In the Route column, select the route that maps to a specific target at the peripheral.

c) Optionally, in the Translation Route column, select a translation route.

Step 4   Optionally, check **Allow connection for each target** to have an output terminal appear to the right of each individual target defined in the node. Control passes through this terminal when the associated target is chosen. When the script terminates, the route for the selected enterprise service is still used.

Step 5   Click **Validate** to check whether the targets you defined are valid. Correct any errors that are flagged.

Step 6   Optionally, add connection labels.

Network Targets

A network target is an end point on the network to which a script can send a contact. Following are the supported network targets:

• Announcement

• Scheduled Select

• Ring

• Busy

• Label

• Divert Label

Route Call to an Announcement

You can route a call to an announcement on the network by using the Announcement node in the Targets tab of the Palette.

Figure 59: Announcement Icon

When executing the Announcement Node, Unified ICM returns the label associated with the announcement to the routing client. The Announcement Node terminates the script.

You might use the Announcement Node to:

• Explain why the call cannot currently be handled.

• Direct the caller to another phone number or to another way of contacting the company.

Note You must configure Announcements and associate them with labels using Unified ICM Configuration Manager. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*.

Following is the Properties dialog box of the Announcement node:

Figure 60: Announcement Properties - Announcement Tab
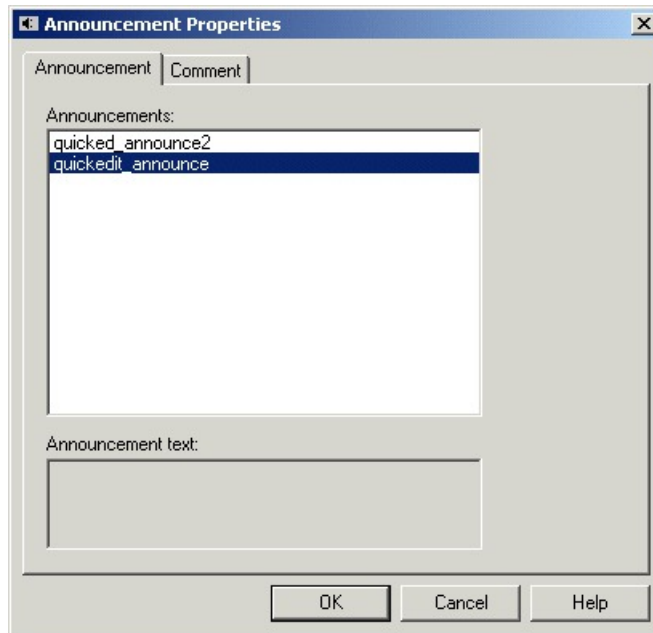
Define Announcement node properties as follows:

Procedure

Step 1	Choose an announcement from the Announcements list.
Step 2	Optionally, add comments.

Route Call to a Scheduled Target

You can send a call to a target that is configured to handle a specific number of simultaneous contacts based on a periodic schedule by using the Scheduled Select node in the Targets tab of the Palette.

Figure 61: Schedule Select Icon

Unified ICM keeps track of the schedule and the number of calls sent to the target. The routing client informs Unified ICM when a call at the target ends, so it always knows how many calls are currently in progress at the target and whether it can handle an additional call.

When Unified ICM runs the Scheduled Select Node, it searches the list of selected targets for one that is capable of handling the contact, based on its current schedule. If a target is found, the routing script ends and returns a label associated with the target to the routing client. The routing client then translates the label to a peripheral target.

✎

Note   You must configure Scheduled targets and associated them with labels using Unified ICM Configuration Manager. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*.

Not all routing clients support the Scheduled Select node.

Following is the Properties dialog box of the Announcement node:

Figure 62: Schedule Select Properties
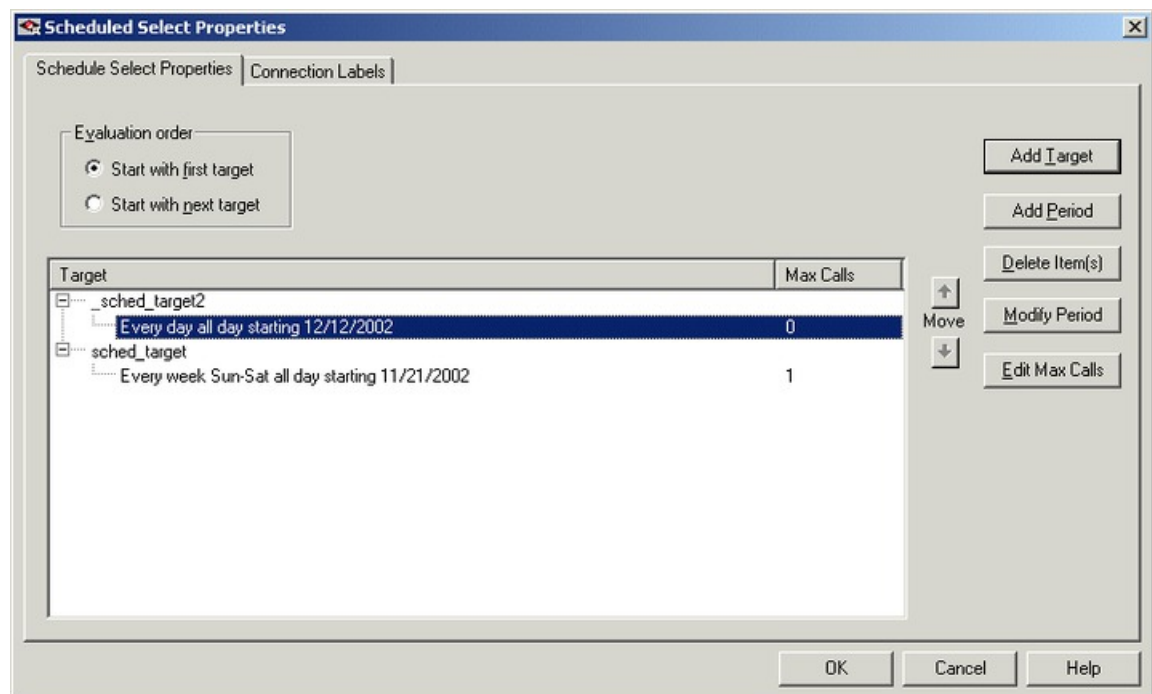
Define Scheduled Select node properties as follows:

Procedure

Step 1   In the Evaluation Order field select:
a)   **Start with first target** to have Unified ICM always start the search from the first target in the list.
b)   **Start with next target** to have Unified ICM start the search from the first target after the last chosen target.

Step 2   Click **Add Target** to add a new scheduled target. In the Add Schedule Targets dialog box, select targets to add from the Available targets list, and click **Add>** to move them to the Add targets list. When finished, click **OK**. The targets are added to the list.

Step 3   To add a time period for the scheduled target, select the target and click **Add Period**. The Add Periodic Schedule dialog box opens. Define the time period and click **OK**.

Step 4   To modify a time period for the scheduled target, select the time period and click **Modify Period**. The Modify Periodic Schedule dialog box opens. Modify the time period and click **OK**.

Step 5   To edit Max Calls for a time period, select the time period and click **Edit Max Calls**. The number in the Max Calls column is now editable. Modify the value as needed

Step 6   Optionally, add connection labels.

Route Call to an Unanswered Ring

You can route a call to an unanswered ring by using the Ring node in the Targets tab of the Palette.

Figure 63: Ring Icon

When Unified ICM runs a Ring node, it returns the first Ring label associated with the routing client; the routing client then plays an unanswered ring for the caller.

Note: You must define a Ring label using Unified ICM Configuration Manager. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*.

Note: Not all routing clients support the Ring Node.

To define the Ring node properties, you simply add comments to the node.

Route Call to a Busy Signal

You can route a call to a busy signal by using the Busy node in the Targets tab of the Palette.

Figure 64: Busy Icon

When Unified ICM runs a Busy node, it returns the first Busy label associated with the routing client; the routing client then plays a busy signal for the caller.

Note: You must define a Busy label using Unified ICM Configuration Manager. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*.

Note: Not all routing clients support the Busy node.

To define the Busy node properties, you simply add comments to the node.

Return Label to Routing Client

You can return a label to a routing client by using the Label node in the Targets tab of the Palette.

Figure 65: Label Icon

When Unified ICM runs a Label node, it returns the first valid label for the routing client.

This differs from the Divert Label node, which returns all the values in the Selected Labels list to the routing client.

If Unified ICM finds no valid label, it returns the default label for the Dialed Number. In either case, the Label node terminates execution of the script

When you define a Label Node, you can select Configured Labels and Dynamic Labels. Configured labels are static, defined through the Unified ICM Configuration Manager. Dynamic Labels are expressions the CallRouter processes in real time, converting an expression into a character string that is then returned to the routing client as a label. You use formulas to create a Dynamic Label.

The following registry setting in the router controls where to send the dynamic labels for network transferred calls. This registry setting controls ALL of the network transferred calls:

`HKLM\Cisco Systems,Inc\ICM\<inst>`

`\Router<side>\Router\CurrentVersion\Configuration`

`\Global\NetworkXferDynLabelDestination`

`<setting>`

If <setting> =	Then the label is sent to
0	The network transfer routing client
1	The network routing client
2	Either the network transfer routing client or the network routing client, based on the NetworkTransferPreferred setting of the network routing client

Note The Label node supports Target Requery.

Following is the Properties dialog box of the Label node:

Figure 66: Label Properties - Label Tab
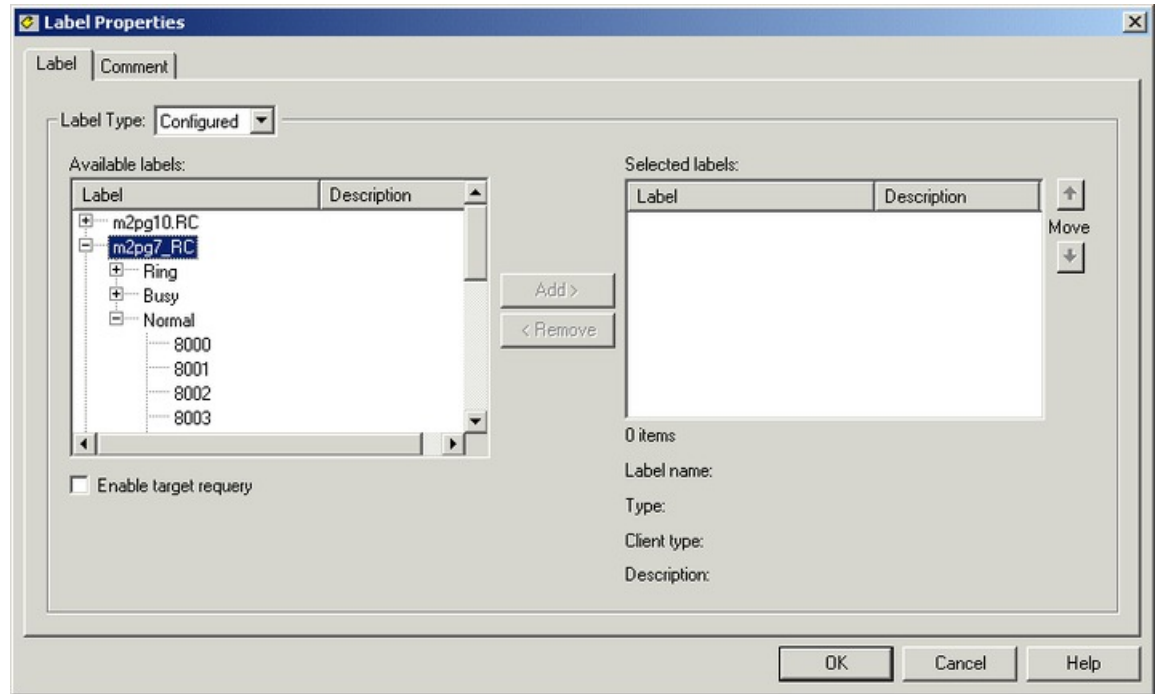
Define Label node properties as follows:

Procedure

Step 1   Choose the one of the following from Label Type:

a)   Configured, to select from a list of configured labels.

b)   Dynamic, to define an expression that is to be returned as a label.

Warning	The incorrect use of dynamic labels in scripts can result in call surges. The router does not do extrapolation for dynamic labels. So if many calls come in at the same time, the router may send these calls to the same label, and the Available and LongestAvailable will not be extrapolated. To avoid a site becoming flooded with calls, use static labels assigned to skill groups and services rather than dynamic labels. The router does extrapolation for labels assigned to skill groups and services.

Step 2   If you select Configured, select labels from the Available labels list and click **Add>** to add them to the Selected labels list.

Step 3   If you select Dynamic, enter a Label Expression, optionally using the **Formula Editor**.

Step 4   Optionally, check **Enable target requery**.

Step 5   Optionally, add comments.

Related Topics

Return Multiple Labels to Routing Client

You can return multiple labels to a routing client by using the Divert Label node in the Targets tab of the Palette.

Figure 67: Divert Label Icon

When Unified ICM runs a Divert Label node, it returns all the values in the Selected Labels list to the routing client.

This differs from the Label node, which returns the first valid label for the current routing client.

Unified ICM then tries each label until it finds one that does not produce a Busy or Ring Tone No Response.

In most cases, you can specify up to ten labels in the Divert Label node. The exception is in NAM/CICM configurations, where you can specify up to nine labels; you must reserve one label for use by the NAM.

When you define a Divert Label node, you can select Configured Labels and Dynamic Labels. Configured labels are static, defined through the Unified ICM Configuration Manager. Dynamic Labels are expressions the CallRouter processes in real time, converting an expression into a character string that is then returned to the routing client as a label. You use formulas to create a Dynamic Label.

Notes:

- With the Divert Label Node, you can only select labels for routing clients with client types that support DivertOnBusy. Unified Contact Center client type does not support DivertOnBusy. The following client types do support DivertOnBusy: CRSP, AUCS INAP NIC, SS7IN NIC, NTL NIC and MCI NIC.
- When using a Divert Label node, arrange the labels so that a label defined with a type Busy or Ring the last label in the Divert Label Selected labels list. The reason for this is that a Busy or Ring label terminates the call; any label appearing after Ring or Busy is never used. Also, never use a Divert Label node when only one label exists in the list; use the Label node, instead.
- Never use a Divert Label node when only one label exists in the list; use the Label node instead.

Define Divert Label node properties as follows:

Procedure

Step 1   Select the Label Type:
a)   Configured, to select from a list of configured labels.
b)   Dynamic, to define an expression that is to be returned as a label.

Step 2   If you select Configured, select labels from the Available labels list and click **Add>** to add them to the Selected labels list.

Step 3   If you select Dynamic, enter a Label Expression, optionally using the Formula Editor.

Step 4   Optionally, add comments.

Selection of Targets by Rules

You can create a script to select a specific target from set of targets based on the rule you select or define.

For example, you can create a script that selects as a target the skill group with longest available agent (LAA) from a set of skill groups.

When defining scripts to select targets by rules, you must be aware of the following:

- Types of Target Searches
- Standard Selection Rules
- Custom Selection Rules

Types of Target Searches

When you use a Select node, you specify if the script is to search for the target that matches the criteria you define starting with the first target in the list, or the next target after the previously chosen target:

Start with first target - Also referred to as a homing search, this option has the script search for the target that meets the criteria you define starting with the first item in the list. For example, you can define a script to select the longest available agent from either the Technical Support skill group (the first skill group in the list) or the backup skill group (the second, and last, item in the list). When you choose this option, whenever the script can find an available agent in the Technical support skill group, the contact is routed to that agent. Agents from the Backup skill group are only selected as targets when the script fails to find an available agent in the Technical support skill group. When you select this option, targets towards the top of the list typically receive a higher percentage of the contacts.

Start with next target - Also referred to as a rotary search, this option has the script search for the target that meets the criteria you define starting with the item in the list after the target previously selected. For example, you can define a script to select the longest available agent from one of three separate technical support skill groups. When you select this option, after a script selects an agent from the first skill group, for the next contact, the script starts looking for an agent in the second skill group. When you select this option, contacts are distributed more evenly among the potential targets.

Standard Selection Rules

When you use the Select node, you can choose from one of the following standard selection rules:

Table 1: Select Node Standard Selection Rules

Selection Rule	Applicable Targets	Formula	Description
Always Select	Any target		Selects the first target that passes the specified acceptance rule.

Selection Rule	Applicable Targets	Formula	Description
Longest Available Agent (LAA)	Skill Groups and Enterprise Skill Groups	Consider if: *.AgentsAvail > 0 Evaluate: MAX (*.LongestAvailable)	Selects the target with the agent who has been available for the longest time. This selection rule helps to ensure that all agents in the skill group set are kept equally busy. It does not ensure that a particular agent is assigned the contact. If the target set includes a skill group that has subgroups (.pri, .sec, etc.), only agents logged in to the base group are considered. Because agents do not usually log in to the base group, specify the base groups you want to consider.
Next Available Agent (NAA)	Skill Groups and Enterprise Skill Groups	Consider if: *.AgentsAvail > 0 Evaluate: MAX (*. AgentsAvail/*. AgentsSignedOn)	Selects the target with the highest percentage of available agents.
Minimum Average Speed Answer (Min ASA)	Services and Enterprise Services	Evaluate: MIN (*.AvgSpeedAnswerTo5)	Selects the target in the set that is, on average, answering contacts most quickly. Because this selection rule evaluates the historical average, it does not select a target based on the current or expected future state of the contact center. Therefore, unexpected load imbalances may occur when you use this rule. To avoid this potential problem, you can use the Minimum Expected Delay selection rule instead.
Minimum Calls in Queue Per Position (Min C/Q)	Services and Enterprise Services	Evaluate: MIN (*.CallsQNow/*.AgentsReady)	Selects the target in the set with the lowest ratio of calls waiting and staffed stations. If agents are equally efficient at each target in the set, this rule tends to lead to the shortest average hold times. However, if agents are not equally efficient, some customers might wait longer than necessary at the less efficient target. To avoid this potential problem, you can use the Minimum Expected Delay selection rule instead.
Minimum Average Queue Delay (Min AvgQD)	Services and Enterprise Services	Evaluate: MIN (*.AvgDelayQTo5)	Selects the target in the set with shortest average hold times, assuming that agents at each target are equally efficient. Because this selection rule evaluates the historical average, it does not select a target based on the current or expected future state of the contact center. Therefore, unexpected load imbalances may occur when you use this rule. To avoid this potential problem, you can use the Minimum Expected Delay selection rule instead.
Minimum Longest Delayed Call (Min Delay)	Services and Enterprise Services	Evaluate: MIN (*.LongestCallQ)	Selects the target with the shortest longest delayed call. Note: This selection rule evaluates the historical average, not the current or expected future state of the contact center. Routing contacts to the target with the shortest longest delayed call does not immediately change the longest delay value. Therefore, this selection rule may route a disproportionately large number of calls to a single target. To avoid this potential problem, you can use the Minimum Expected Delay selection rule instead.

Selection Rule	Applicable Targets	Formula	Description
Minimum Expected Delay (MED)	Services and Enterprise Services	Evaluate: MIN (*.ExpectedDelay)	Selects the target with the shortest expected delay. In making this evaluation, this selection rule considers the average handle time, the number of contacts in queue, and the number of positions staffed. This rule is usually the most effective rule for keeping queue times to a minimum. The MED algorithm is not supported on Unified CCE.

Caution: Values used by the standard selection rules Minimum Average Speed Answer, Minimum Average Queue Delay, and Minimum Longest Delayed Call change slowly. Because the averages in these rules only consider what has happened in the past rather than what is currently happening, using these rules inappropriately can lead to load imbalances as newly routed contacts have little immediate effect on the values used to route later contacts. In contrast, the standard selection rule Minimum Expected Delay takes into account each contact as it is routed. Selecting the service using the Minimum Expected Delay rule usually provides the best balance among the services in the target set.

Custom Selection Rules

Instead of using one of the standard selection rules, you can create a custom selection rule. To write custom selection rules, you must be familiar with using formulas. For more information on formulas, see the chapter Use of Formulas.

A custom selection rule has three parts:

- `Consider if` **statement** 0. This expression ensures that only skill groups with at least one available agents are considered as targets; if a skill group has no available agents, the expression returns False and the skill group is not considered.

- **Selection criteria**. A numeric expression plus an evaluation rule. The numeric expression determines a certain value of each target for which the Consider if statement returned True. The evaluation rule determines if the target with the minimum or maximum value returned is selected. If you do not define selection criteria, the first target to return True for the Consider if statement is selected. For example, the Selection criteria for the predefined Longest Available Agent selection rule is MAX (*.LongestAvailable). This expression determines the highest LongestAvailable value of all the skill groups in the target set for which the Consider if statement returned True.

- `Accept if` **statement**. A Boolean expression that returns True or False for the target selected by the selection criteria. If the Accept if statement returns True, the target is selected; if it returns False, no target is selected by the Select node.

Selection of Targets by Rules

You can select targets by rules by using the Select node in the Routing tab of the Palette.

Figure 68: Select Node Icon

The Select node sets up a rule by which the node chooses from a set of routing targets for the contact. You can select a standard rule or define your own custom rule. This node also has target requery capabilities.

Notes:

- You can follow the instructions in this section to select targets by rules using the Select node. You can also use the Route Select node to select targets as well as distribute contacts to targets.

- You must add a skill target and create a connection from the Select node's success terminal, and define at least one target in the target set, before defining the Select node.

- The Select Node supports Target Requery.

Following is the Properties dialog box of the Select node:

Figure 69: Select Properties - Select Tab
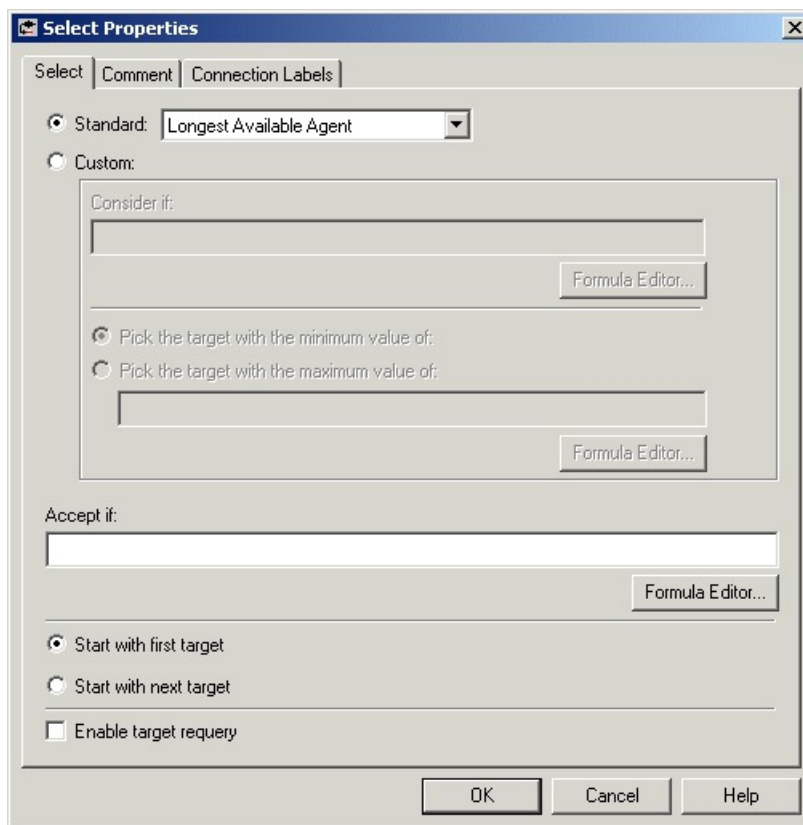
Define Select node properties as follows:

1. Select from the list of Standard rules, or select Custom and:

    - In the **Consider if** field, enter a Boolean expression. A target is considered for selection only if the Consider If expression is true for that target. Optionally, use the Formula Editor.

    - Select **Pick the target with the minimum value of** or **Pick the target with the maximum value of** and enter an expression to select the target.

2. Optionally, in the **Accept if** field, enter a Boolean expression that must evaluate to true for the target to be selected. It determines whether the target must be returned or the process must end. The Boolean

expression does not permit wild cards and evaluates various conditions to determine if any target must be returned. For example, you can use a peripheral user variable such as **Peripheral.PG_CCM.userRouteToCustomerService==1** in the **Accept if** field of a Select node to determine whether the script is routed to any customer service skill groups at a particular time.

3. Select **Start with first target** or **Start with next target** to indicate how Unified ICM looks for targets. For more information, see the section Types of Target Searches.

4. Optionally, check **Enable target requery**.

5. Optionally, add comments and connection labels.

Related Topics

Target Requery, on page 84

Distribute Contacts to Targets

You can create a script to allocate contacts among the targets in the set based on current information about each target by using the Distribute node in the Routing tab of the Palette.

Figure 70: Distribute Icon

For example, you can distribute contacts among a set of services based on the number of agents in the service in the Ready state. Services with more agents in the ready state are routed more contacts than services with fewer agents in the Ready state, thus keeping the load balanced.

You can distribute contacts to any of the following types of target sets: Agent, Skill Group, Service, Enterprise Skill Group, or Enterprise Service.

When creating a script to distribute contacts to targets, you must define the following for the Distribute node using formulas. For more information on using formulas, see the chapter Use of Formulas.

- `Consider if` **statement**. A Boolean expression that returns True or False for each target in the target set. Only targets in the set for which the expression is True are eligible to be distributed contacts. If you do not define an expression for the Consider if statement, all targets in the target set are considered. For example,
- `Distribute by` **statement**. A formula used to distribute contacts to targets

Notes

- You can follow the instructions in this section to distribute contacts to targets using the Distribute node. You can also use the Route Select node to select targets as well as distribute contacts to targets.
- You must add a skill target and create a connection from the Distribute node's success terminal, and define at least one target in the target set, before defining the Distribute Enter node.

Following is the Properties dialog box of the Distribute node:

Figure 71: Distribute Properties - Distribute Tab
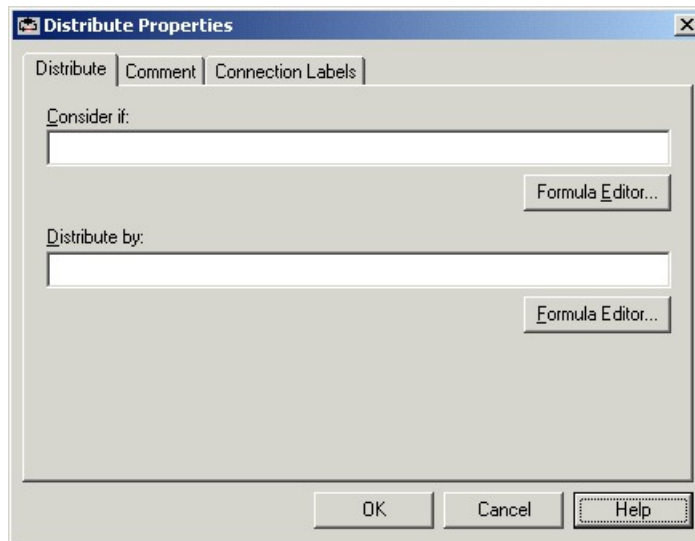
Define Distribute node properties as follows:

Procedure

Step 1   Enter a condition in the Consider if field to test potential targets against.

Step 2   Enter a formula by which to distribute contacts in the Distribute by field.

Step 3   Optionally, add comments and connection labels.

Select Targets and Distribute Contacts with One Node

You can combine the functionality of selecting targets by rules with the Select node and distributing contacts to targets with the Distribute node by using the Route Select node in the Targets tab of the Palette.

Figure 72: Route Select Icon

Note   The Route Select Node supports Target Requery.

Following is the Properties dialog box for the Route Select node:

Figure 73: Route Select Properties - Route Tab
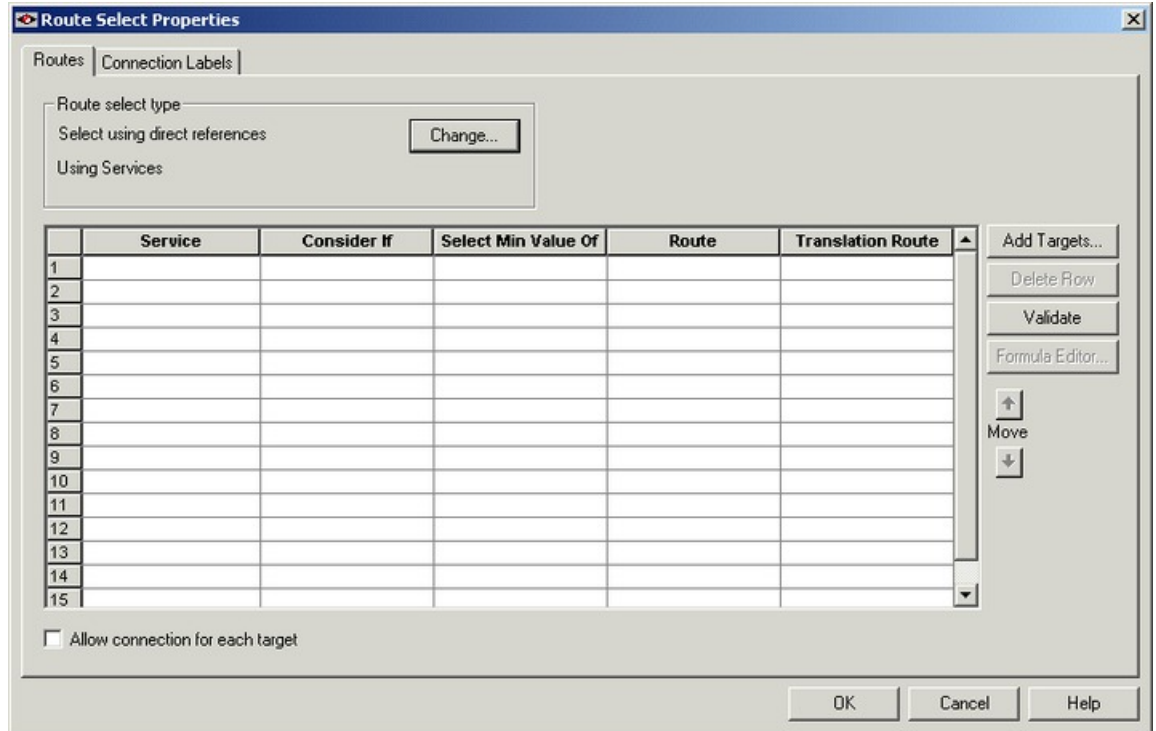
Define Distribute node properties as follows:

Procedure

Step 1   To select the Route select type, click **Change**. The Route Select Type dialog box opens:

Figure 74: Route Select Type

a) For Target Type, select **Agent**, **Enterprise Service**, **Enterprise Skill Group**, **Service**, **Service Array**, or **Skill Group**.

b) If you selected Enterprise Service or Enterprise Skill Group, select a **Business Entity** and **Enterprise target**.

c) Select **Distribute among targets** or **Select most eligible targets**.

d) If you selected **Select most eligible targets**, select **Pick the target with the minimum value** or **Pick the target with the maximum value**. In the Accept target if field, enter a condition that the target must meet to be selected. Select **Start with first target** or **Start with next target**. For more information, see Types of Target Searches.

e) In the Target references field, select **Explicit target references** to use direct references to targets, or **Lookup target references by expression** to use expressions that evaluate to names of targets. **Note Lookup target references by expression** is not available when an External Authorization server is used with Internet Script Editor and will be grayed out in the interface.

f) Optionally, check **Enable target requery**.

g) Click **OK**.

Step 2   The fields in the Route Select Properties dialog box change depending on your route select type selections. Enter and select data appropriate for the type you selected.

Step 3   Click **Validate** to check whether the targets you defined are valid. Correct any errors that are flagged.

Step 4   Optionally, add connection labels.

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.

Figure 75: Agent to Agent Icon

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 run.

Following is the Properties dialog box for the Agent to Agent node:

Figure 76: Agent to Agent Properties
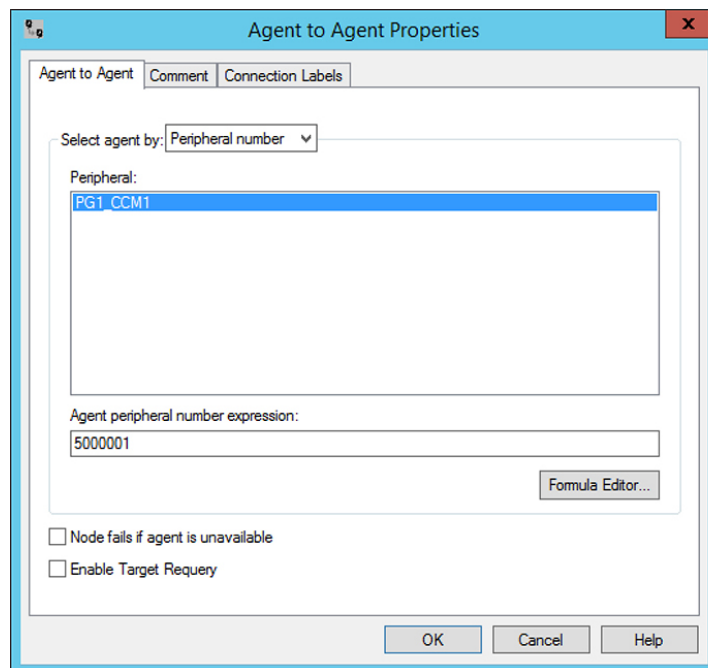
Define Agent to Agent node properties as follows:

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 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 run 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 run 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 run 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 run if the Router does not find a valid label for the agent.

Step 4 Optionally, add comments and connection labels.

Step 5 Optionally, check **Enable Target Requery**.

Related Topics

Target Requery, on page 84

Send Contact to a Different Unified ICM System

A script may determine that the contact should be routed by a different Unified ICM system.

You send a contact to a different system by using the ICM Gateway node in the Routing tab of the Palette.

Figure 77: ICM Gateway Icon

The ICM Gateway node passes a routing request to the selected Unified ICM system.

Note Before you can successfully use an ICM Gateway node in a script, you must use the Unified ICM Configuration Manager to configure a gateway to the Unified ICM system to which you send the request. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-guides-list.html.

Define the ICM Gateway node properties as follows:

Procedure

Step 1 In the Send tab:

    a) Select the gateway to Unified ICM system (and hence the specific Unified ICM instance) from the ICM Gateways list to which you want to send the request.

    b) Check **Validate** returned labels to have Unified ICM validate the returned labels.

c) Specify whether Calling Line ID masking instructions should be applied before the request is passed to the other Unified ICM system. The Calling Line ID masking refers to when the caller's phone number is modified so that Unified ICM application does not display all of the digits; this is used in a NAM environment, where NAM sends the call to a customer Unified ICM. Choose one of the following:

  • 1)**Do not apply masking rule** - If selected, masking instructions are ignored.

  • 2) **Apply masking rule if call is presentation restricted** - If selected, applies masking instructions if the call variable CLIDRestricted is set to 1.

  • 3) **Always apply masking rule** - If selected, masking instructions are always applied.

Note   The Calling line ID masking rule is set through the Unified ICM Configuration Manager's System Information dialog box.

Step 2   In the Default Label tab:

Figure 78: ICM Gateway Properties
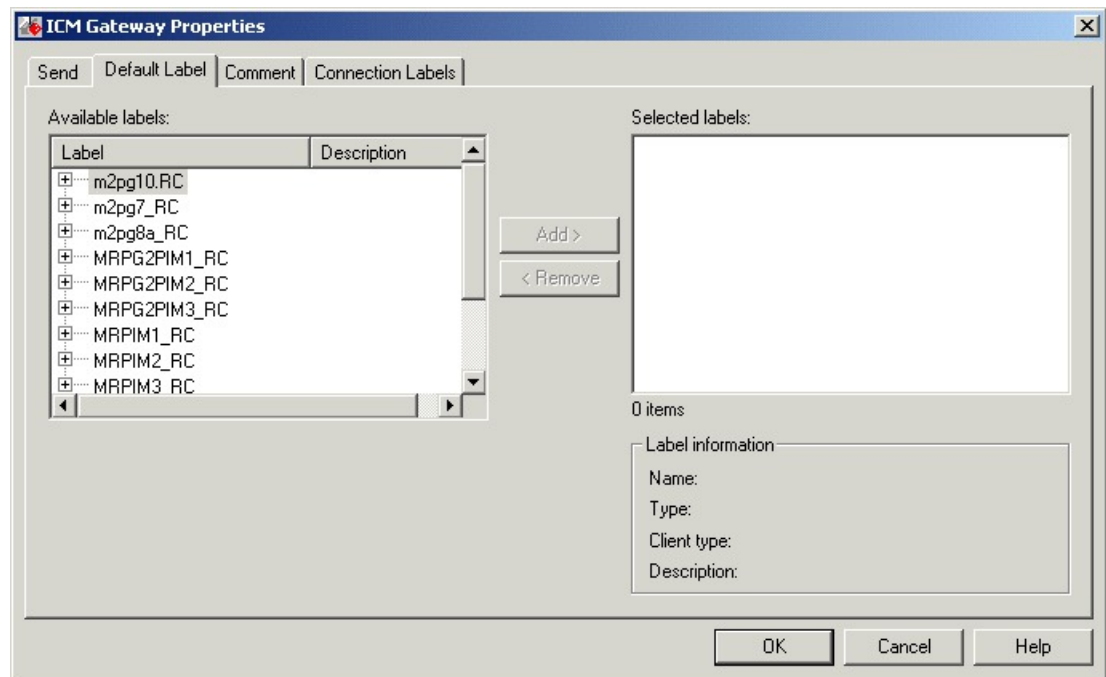
a) In the Available Labels list, select one default label for each routing client to be used by the targeted Unified ICM system.

b) Click **Add** to move the selected label to the Selected labels list.

Step 3   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 79: End Icon

If the script reaches the End node, it has failed to find a target for the contact. Unified ICM 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.

Termination Node

You can terminate the script and specify how to handle the contact by using the Termination node in the Targets tab of the Palette.

Figure 80: Termination Icon

The Termination node includes the following options to invoke a default contact processing action or route for the Dialed Number:

• Default Label - Unified ICM uses the default label configured for the Dialed Number.

• Network Default - The routing client uses its own default processing.

• Ring - Makes the Termination node equivalent to a Ring node.

• Busy - Makes the Termination node equivalent to a Busy node.

Following is the Properties dialog box for the Termination node:

Figure 81: Termination Properties
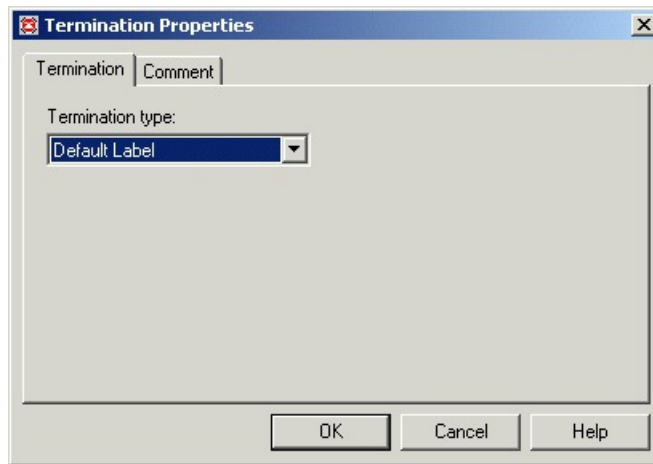
Related Topics

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 82: Release Call Icon

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

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

Service Requested

ServiceRequested is a call variable available in Script Editor. It provides more details regarding the routing request. The field is currently only set for multichannel routing (Task Routing), voice callback (Agent Request) and Pick/ Pull routing, otherwise it is set to 0. Based on the value of this field, the script can take different actions.

ServiceRequested Variable

Service Requested Variable	Description
0 = ROUTE_SERVICE_REQUEST_NONE	No service requested.
1 = ROUTE_SERVICE_REQUEST_VOICE_CALLBACK	Caller is requesting a voice callback.

2 = ROUTE_SERVICE_REQUEST_TRANSFER	Transfer a task that is already assigned to an agent back to a queue.
3 = ROUTE_SERVICE_REQUEST_RONA	A task is being rerouted on no answer (RONA).
4 = ROUTE_SERVICE_REQUEST_PICK_EXT_QUEUE	Pick a task that is associated external application queue for a specified agent.
5 = ROUTE_SERVICE_REQUEST_PULL_EXT_QUEUE	Pull a task that is associated external application queue for a specified agent.
6 = ROUTE_SERVICE_REQUEST_PICK_UCCE_AGENT	Pick a task that is already assigned to another agent for a specified agent.
7 = ROUTE_SERVICE_REQUEST_TRANSFER_PICK	Transfer a task that is already assigned to an agent directly to another agent, instead of back to a queue.

Target Requery

Target Requery is a script node feature that you can use to handle routing failures, for example due to No Answer or Busy responses, or for unreachable targets caused by 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 ACD agent.

- Failure to deliver a call to an individual Enterprise Agent (EA).

- 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.

> **Note** You can enable the Target Requery feature for CVP, ICM to ICM gateway, and a subset of the supported carrier NICs. You cannot use the requery feature with any of the multimedia requests because the MR PG does not support requery mechanism.

Target Requery Functionality

In the 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 in a Label, Route Select, or Select node, if the router fails to route to a 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. If the router fails to route to the target node, the failure path of the node is taken immediately. To implement requery in a Queue node, 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:

- Is compatible with translation routes.

- Does not require different definitions for different failure cases. However, you can choose to handle different failures differently.

- Assigns the SERVICE_DIVERT_ON_BUSY service type for calls that use target requery. The Event Select in the connect message includes a REROUTE_REQUESTED_MASK. When a destination cannot be reached, the NIC queries the CallRouter for an alternative destination label.

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 2: RequeryStatus Variables

Requery Status Variable	Description
REQUERY_ANSWER (0)	Script ends. The call was successfully sent to the chosen target. **Note** This variable is used internally by the CallRouter. You cannot test for this variable in an IF node.
REQUERY_ROUTE_SELECT_FAILURE (1)	Routing client generated an error code from ReRouteReq msg indicating a Route Select failure.
REQUERY_CALLED_PARTY_BUSY (2)	Routing client generated error code from ReRouteReq msg indicating the called party is busy.
REQUERY_NO_ANSWER (3)	Routing client generated an error code from ReRouteReq msg indicating no answer.
REQUERY_ERROR (4)	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).

Requery Status Variable	Description
REQUERY_ABORTED (5)	The attempt to send the call to a target failed because the caller abandoned before the call could be delivered to its destination. In the case of ABANDON and DISCONNECT, the CallRouter assumes the call has ended and ends the script. The RequeryStatus value is set to 5, indicating REQUERY_ABORTED. **Note** This variable is used internally by the CallRouter. You cannot test this variable in an IF node.
REQUERY_TIMED_OUT (6)	The call attempt failed because the routing client did not respond to the router within the DivertOnBusyCallTimeout period (180 seconds by default). If the target node has Router Requery enabled, when DivertOnBusyCallTimeout period expires, the Router closes the Router Requery with REQUERY_TIMED_OUT . **Note** This variable is used internally by the CallRouter. You cannot test this variable in an IF node.

Nodes That Support Target Requery

The following nodes support Target Requery:

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

Note Scheduled Targets do not support Target Requery.

Target Requery with Unified CCE and Unified IP IVR

Target Requery is not supported when Unified IP IVR is used as the queue point in a Unified CCE system because Unified IP IVR cannot requeue the call. The call is requeued from Cisco Communications Manager instead. Setting the Target Requery option on the script node has no effect on the requeuing of the call, and may lead to timeouts and other script errors for that particular call.

Do not enable Target Requery in script nodes when you use Unified IP IVR.

Use Target Requery

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

Procedure

Step 1	Open the node properties.
Step 2	Click **Change**. A dialog box opens.
Step 3	Check **Enable target requery**.
Step 4	Click **OK** to close the dialog box.
Step 5	Click **OK** to close the properties dialog box.
Step 6	For the Label, Select and Precision Queue nodes:

For the Label and Select nodes:

a) Open the node properties.
b) Check **Enable Target Requery**.
c) Click **OK** to close the properties dialog box.

Target checking

About Call Tracer

You can use the Call Tracer utility from within Script Editor to test and debug a routing script and to confirm that the Unified ICM is selecting targets as you expect. Call Tracer simulates a contact to the CallRouter and generates a text-based description of how the contact was handled. The test contact is processed by the active CallRouter, using all the real-time data of Unified ICM as it exists at that moment, but interactions with any peripheral is simulated.

Check Targets

To check targets for a voice calls:

Before you begin

Start the Call Tracer by double-clicking the Call Tracer icon in the Administration & Data Server group. The Script Editor and Call Tracer open. To open Call Tracer if the Script Editor is already open, double-click the Call Tracer icon in either the Administration & Data Server group or within the Script Editor toolbar.

Figure 83: Call Tracer Icon

Procedure

Step 1	In the Call Tracer window, choose a Media Domain as voice, a Routing Client, and a Dialed Number.
Step 2	Optionally, enter an ANI (callers telephone number).
Step 3	If you want to test a response from a VRU routing script, enter values in Caller-Entered-Digits (CED) and VRU Responses for External Script.
Step 4	To use Network Transfer Call, a feature that integrates Unified ICM Post-Routing function with a carrier network's call control ability so that a call can be transferred anywhere in the network without the use of transfer/connect services or inter-site tie lines:
	a) Select Use Network Transfer.
	b) Select a routing client and a Dialed Number value.
Step 5	Click **Send Call** to submit the request. The results appear in the Call Trace Results field.
Step 6	To send additional calls, optionally change any of the call parameters and then click **Send Call** again. The Call Trace Results field is updated.
Step 7	Unified ICM runs only installed scripts in response to requests from the Call Tracer. Any uninstalled changes in scripts are not reflected in the Call Tracer results.

Procedure

To check targets for a non-voice calls:

Procedure

Step 1	In the Call Tracer window, choose a Media Domain other than a voice call, a Routing Client, and a Script Selector.
Step 2	Optionally, enter an Application String 1 and Application String 2.
Step 3	For tracing **PICK/PULL** request.
	a) Select a **Service Type**, and choose a **Queue Type**
	b) Enter a **Queue ID** and select the CCUM/system Peripherals with a **Preferred Agent**.
Step 4	Click **Send Call** to submit the request. The results appear in the Call Trace Results field.
Step 5	To send additional calls, optionally change any of the call parameters and then click **Send Call** again. The Call Trace Results field is updated.
Step 6	Unified ICM runs only installed scripts in response to requests from the Call Tracer. Any uninstalled changes in scripts are not reflected in the Call Tracer results.

Example Results

In the following example results from Call Tracer, Unified ICM invokes a script called sales. It begins with the Start node and then runs a Percent Allocation node that has three branches allocated for 42%, 16%, and 42%, respectively. The asterisk indicates that for this call, Unified ICM chooses the first branch. This branch

leads to a target that maps to the service Scranton.Sales. Had this been an actual call, the call would be delivered to that service.

As you continue to send calls to the Unified ICM, the numbers for each node change to indicate how calls have been distributed. Each time a call arrives at a Percent Allocation node, the Unified ICM calculates the percentage of calls previously sent to each branch:

```
> DialedNumber (8005551212) using CallType (General_Sales - 6)

Start #3

__Sales\Sales__General (version 2)

Percent allocation #54

__*42% = 0

__ 16% = 0

__ 42% = 0

Target #64

__Service: Scranton.Sales R

Route: Scranton.Sales

Peripheral target: DNIS: 1111, Trunk group: Scranton.Incoming800

Label: 2010000000
```

In the next example, for each branch the new call is counted in the denominator. Therefore, the numbers add up to less than 1.0 and gradually, as you send more calls, the total increases up to 1.0. Unified ICM picks the branch that is below its allocation value. For the example allocation, Unified ICM picked the third branch as 0.375 is below the allocation of 42% or 0.42 and sent the call to the Gary.Sales service.

Traces are linked to live scripts. For example, clicking on the trace details of a Percent Allocation node immediately, if it is not already open, opens the script and locates that node in the Script Editor window. Previously, you needed to manually decode script, script version and node ID information, and then visually scan the script for the desired node. Node titles in the trace window support localization.

The Call Tracer results refer to script nodes by their internal integer identifiers. To find which node maps to each identifier, open the script and choose Display Node IDs from the Script menu:

```
> DialedNumber (8005551212) using CallType (General_Sales - 6)

Start #3

__Sales\Sales__General (version 2)

Percent allocation #54

__42% = 0.416667

__16% = 0.166667

__*42% = 0.375

Target #58

__Service: Gary.Sales

Route: Gary.Sales

Peripheral target: DNIS: 1111, Trunk group: Gary.Incoming800
```

```
Label: 3010000000
```

Check VRU Scripts

To check VRU scripts:

Before you begin

Call Tracer allows you to simulate responses from VRU routing scripts. For each Script node Unified ICM encounters, you can specify one of the following responses:

- Fail. Unified ICM treats the External Script nodes as though it had failed.
- No Response. Unified ICM assumes it received no response from the Run External Script node.
- Response. You can specify a CED value and values for Variable1 through Variable10. Unified ICM assumes these values were returned by the External Script node.

Procedure

Step 1 In the Call Tracer dialog box, click **Add**. The VRU Response dialog box opens.

Step 2 Select the type of response (Fail, No Response, Response) the Call Tracer is to simulate.

Step 3 If you chose Response in Step 2, optionally specify values to be returned for CED and Variable1 through Variable10.

Step 4 Click **OK** to apply changes and close the dialog box.

What to do next

Note You can also set the initial condition of the 10 call variables for the first run of the External Script.

Network VRUs

VRU Functionality

You can use routing scripts to divert a call to a Network VRU for additional call processing.

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 Unified Customer 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.

Network VRU Functionality

A Network VRU supports the Unified ICM's service control interface. A Unified ICM routing script can divert a call to a Network VRU and instruct the VRU to perform specific processing before the Unified ICM determines the final destination for the call.

Related Topics

ECC Payloads in Scripts, on page 144

VRU Configuration

You configure Network VRUs through Unified ICM Configuration Manager. Following are the general steps you must take to configure VRUs before you can write routing scripts. There are different Network VRU types. Each type represents a different architecture and call flow. These are summarized below and described in detail in the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*. Before you can write routing scripts, you must do the following through the Unified ICM Configuration Manager:

Procedure

Step 1 Configure the Network VRU.

Step 2 Associate the Network VRU with a Network VRU Label.

Note This step is required for Type 3, Type 5, and Type 7 VRUs.

Step 3 Depending on the type of Network VRU configured, you need to configure various additional items.

Step 4 Configure Network VRU scripts.

Note You create and maintain VRU scripts on the VRU; however, you must define these scripts through the Unified ICM Configuration Manager so that Unified ICM is aware of these scripts so they can be called from Unified ICM routing scripts.

Related Topics

ECC Payloads in Scripts, on page 144

Network VRU Types and Script Nodes

Before a routing script can run a Network VRU script or queue a call at a Network VRU, it may require an explicit node to force the routing client to connect the call to the Network VRU. The node used to send the call to the Network VRU depends on the Network VRU type, as shown in the following table:

Table 3: Network VRU Types

Type	Description	Routing Script Node to Use to Send Call to VRU
2	A Network VRU at the customer premises. In a NAM environment, a Type 2 VRU is connected at Unified ICM and not at the NAM: Unified IP IVR in a Unified CCE solution.	Translation Route to VRU
3	A Network VRU connected to a NAM where there is a separate routing client that controls the connection to the VRU. Use this type rather than Type 7 when the routing client can automatically take back the call from the VRU when Unified ICM returns a destination.	Send to VRU
5	A Network VRU that is controlled by the routing client. Use this type (rather than a Type 3 or Type 6) when the routing client has access to multiple VRUs.	Send to VRU
6	A Network VRU that receives the call and then sends a route request to Unified ICM.	Send to VRU
7	A Network VRU controlled by the routing client. Note: Use this type (instead of Type 3) when the routing client cannot take back the call from the VRU. That is, Unified ICM automatically instructs the VRU to release when it sends a route response to the routing client: Cisco Internet Service Node using a Cisco IOS-based Voice Browser.	Send to VRU
8	Similar to Type 2, but a Type 8 VRU is used when the NAM has a routing client that controls the call to the VRU.	Translation Route to VRU
9	Used for self-service / queuing at IP-IVR when using a System PG, typically in a Parent-Child deployment.	Send To VRU
10	Simplifies configuration requirements in Unified CVP Comprehensive Model deployments. Use this type for calls that need to be transferred to Cisco Unified Customer Voice Portal (Unified CVP) for self-service or queuing.	Send To VRU Translation Route to VRU

Access to VRU Scripts in Unified ICM Scripts

After you set up VRUs through the 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 a VRU for further processing by using the Send to VRU node (in the Queue tab of the Palette).

Figure 84: The Send to VRU Icon

The Send to VRU node is also valid for use with Type 3, Type 5, Type 6, Type 7, and Type 10 VRUs.

When Unified ICM runs 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.

- If the Network VRU is of Type 3 or Type 7, the router sends an instruction to the routing client to connect the call to the Network VRU. The instruction contains the label configured for this Network VRU as well as a correlation ID to allow the router to uniquely identify the dialog with the routing client and the dialog with the VRU PG. The router waits until it receives a RequestInstruction message from one of the connected VRU PGs with the same correlationID as a sign that the call is now connected to the VRU.

- If the Network VRU is of Type 5 the router sends an instruction to the routing client to connect the call to a VRU resource. The instruction contains the label configured for this Network VRU. The router waits for a response back from the routing client that the call is properly connected to the VRU resource.

- If the Network VRU is of Type 10 (which is typically used only with Unified CVP), the Router uses a two-step transfer. The first transfer is to move the call to the Unified CVP, followed by a second automatic transfer to setup a VRU leg for the call that employs the VXML Gateway or Cisco Virtualized Voice Browser (VVB) for IVR treatment. The purpose of Type 10 VRU is to simplify the configuration needed in the Comprehensive Call flow model.

  Important points to note while you are using the Type 10 network VRU are:

  - For calls originating from Unified CM, the Correlation ID is automatically added to the end of the transfer label defined in the Type 10 Network VRU configuration.

  - For calls that are front-ended or originating from Unified CVP, only a single automatic transfer from the switch leg to the VRU leg occurs, using the transfer label defined for the Type 10 Network VRU.

  - For calls originating from all other routing clients like TDM ACD's, other VRU's or NIC's, a "Translation Route to VRU" node should be employed for the first phase of transfer, instead of the "SendToVRU" node without which the calls would fail.

There are two failure cases:

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

- If Unified ICM 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. For Network VRUs of Types 3, 5, 6, or 7 you do not have to use the Send To VRU node. However, include the node in routing scripts as a visual aid if you ever need to troubleshoot the script.

- If the call is delivered to the VRU 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.

Send Call to a VRU with Translation Route to VRU

For Type 2 and Type 8 VRUs, you can send a call to a VRU by using the Translation Route to VRU node (in the Queue tab of the Palette).

Figure 85: The Translation Route to VRU Icon

When Unified ICM uses the translation route to VRU method to route a call to a Type 2 SCI VRU, the VRU becomes the new routing client (dynamic routing client) for the call. This means that the next Connect message is sent to the VRU from the CallRouter to connect the call (to an agent, for example).

To transfer a call to CVP using the Translation Route scheme, you must use the CVP DNIS API to configure DNIS numbers in the CVP call server. For more information, see the *CVP Dialed Number Identification Service API* chapter in the *Cisco Packaged Contact Center Enterprise Developer Reference guide*.

Note For the dynamic routing client feature to work, the routing target (network VRU, services, and so on) needs to have labels configured for the type 2 VRU as a routing client.

This feature is applicable for following call scenarios:

- Receiving a call on a converged enterprise network after a carrier pre-route using Unified CVP to control the call.

- Supporting Outbound Option for Unified CCE where a shared Network VRU is used for providing announcements to targeted phones/answer machines.

- Using pre-routing for load balancing purposes before the call is sent to a Unified CVP controlled network, with call context and cradle to grave reporting capabilities.

In an environment where the routing client is not capable of Network Transfer all the pre-route calls must be transferred to Unified CVP by a Translation Route To VRU to take advantage of network transfer feature.

Typical Call Flow Example:

1. A call is pre-routed by Unified ICM using a carrier NIC to a Type 2 premise VRU using the TranslationRouteToVRU mechanism. The VRU becomes the new routing client.

2. When the VRU is done with the call, the script connects the call to an agent on a TDM ACD (using translation routing again) and the VRU connects the call to the agent.

3. The agent answers the call and talks to the caller.

4. The agent initiates a transfer and the script and configuration initiate a network transfer. The Connect message is sent to the Type 2 VRU, and the VRU connects the call to the new destination.

Following is the Properties dialog box for the Translation Route to VRU node:

Figure 86: Translation Route to VRU Properties
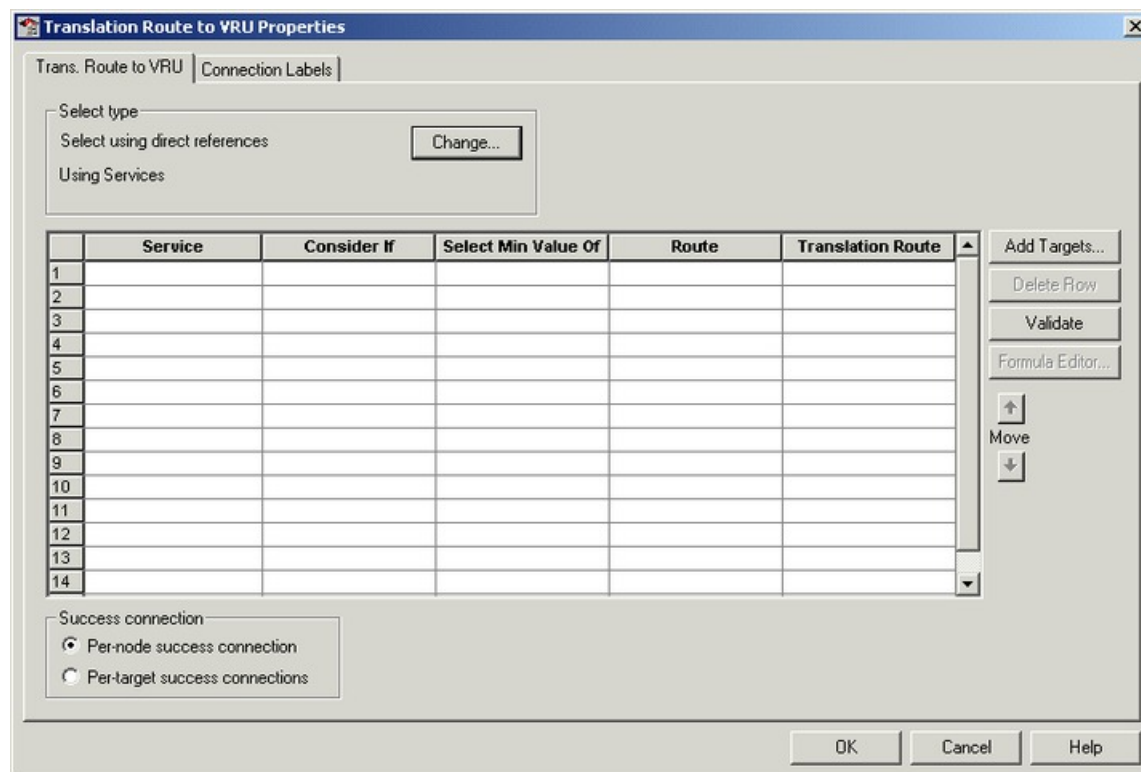
Define Translation Route to VRU node properties as follows:

Procedure

Step 1 To change the type of target:

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

b) Choose the Target Type (Enterprise Service, Service, or Service Array).

c) If you selected Enterprise Service, select a **Business Entity** and **Enterprise target**.

d) Specify whether the Translation Route to VRU node is to act like a Select or Distribute node.

• Distribute Among Targets. The Translation Route to VRU node is to act like a Distribute node, distributing calls among the targets based on the relative values.

• Select Most Eligible Target. (Radio button.) The Translation Route to VRU node is to act like a Select node.

If you select this option, you:

• Define whether to pick the target with the maximum value or the minimum value.

• Define a formula that determines which target is to be accepted.

• Define the type of target search.

Step 2   To add targets, click **Add Targets**. The Add Targets dialog box opens. Use the Available Targets list and the Add button to select targets.

> **Note**   If you choose Enterprise Service as a target type, you can select just one item from the list. If you choose Service or Service Array, you can select one or more items from the list.

Step 3   Click **OK** to close the Add Targets dialog box. The target members you selected appear in the Properties dialog box.

Step 4   Continue defining Target information for each target:

• **Consider If** (Optional.) A formula that must evaluate to true for the target when the Unified ICM initiates the Translation Route to VRU node, or that target is not considered.

• **Select Max/Select Min Value of** A formula that determines which of the targets is selected.

• (Drop-down list.) The route on which to send the call if you select this target. (The list contains all routes associated with the target.)

• **Translation Route** (Drop-down list.) The route to send the call for initial VRU processing if you select this target. (The list contains all translation routes associated with the same peripheral as the target.)

> **Note**   You must specify a value for this field. When a call is sent to a translation route, the PG retrieves the final route from the Unified ICM and coordinates the other processing with the VRU.

• **Per-node success connection**(Radio button.) Select this option to attach one success output terminal to the node. This terminal is used regardless of which target you select.

• **Per-target success connection** (Radio button.) Select this option to attach a success output terminal to each target in the node.

> **Note**   This option is useful in situations where you want to use different scripts depending on the selected target for a call.

Step 5   Optionally, click **Validate** to validate the node properties.

Step 6   Optionally, add connection labels.

Send Call to a VRU after Translation Route to VRU

A SendToVRU node first checks for the Dialed Number. If the Dialed Number is not configured, the node immediately fails, interrupting the dialog. If the Dialed Number is configured, the node checks for the Customer. If the Customer is not defined, the node tries to send the call to Default Network VRU.

If the Customer is defined, the SendToVRU node sends the call to the Customer's Network VRU. If the node cannot identify the Customer Network VRU, the node sends the call to the Default Network VRU.

An explicit SendToVRU node only exits if the call is at the specified VRU, for example, the dialed number's customer Network VRU. If the call is at the wrong VRU, the node transfers the call to the correct VRU.

Example Call Flows:

1. A call is moved to a Type 2 VRU via a Translation Route to VRU. The SendToVRU node moves the call to a Type 3 VRU if the Type 3 VRU is configured to be the network VRU for this call.

2. A call arrives from a Type 6 VRU. SendToVRU attempts to send the call to the configured network VRU if the network VRU is different from the routing client's VRU.

3. SendToVRU fails the node if the transfer to the network VRU fails.

   • If the configuration is wrong or missing, (for example, if a network VRU is not defined, or proper label is not defined) then the fail path is run and the call remains at the VRU.

   • If the CallRouter succeeds in sending a Connect message, the CallRouter resets the previous VRU information and waits for RequestInstruction to set up the new VRU connection. If the operation times out, (RequestInstruction never arrives), then the fail path of the SendToVru node is taken and the call is assumed not to be at a VRU. The same behavior applies if CallRouter receives a ReRoute instead of a RequestInstruction command.

4. A registry flag enables/disables this feature. The registry key is called *NetworkVRUCheckEnabled* and is located at ...*Router\CurrentVersion\Configuration\Global*. The default value is 1 (enabled), setting it to 0 disables it.

Run External Scripts

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

Figure 87: The Run External Script icon

You can use multiple Run External Script nodes to run a series of scripts on the VRU.

The Run External Script node is valid for use with all the VRU types.

Note   When you integrate the ECE with the Unified ICM, you can also use the Run External Script node to push a URL to the caller's web browser. To do this, an entry in the Network VRU list must point to the URL map file on the ECE. For more information, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*.

The execution of Unified ICM 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.

Note
- If the current call is not at a VRU when the Run External Script node is run, Unified ICM sends the call to the associated Network VRU, as run 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.

Note
When an uninterruptible script is used in a Run External Script node, the CallRouter waits for the script result from the VRU. It then runs 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 88: Run External Script Properties
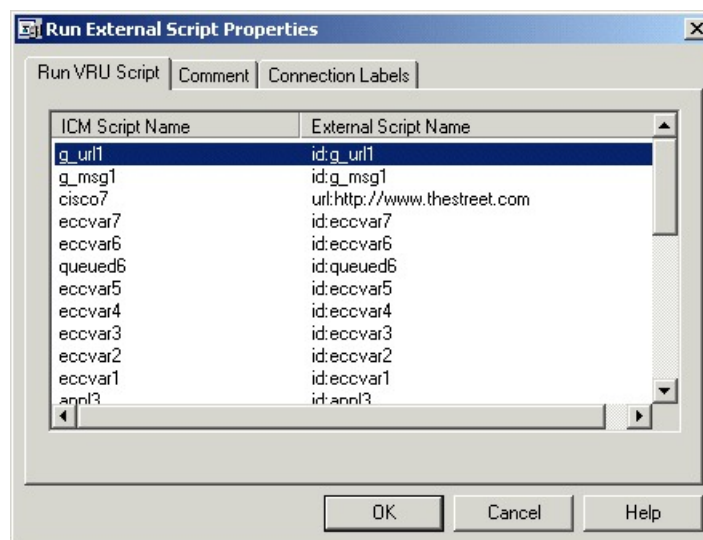
Define Run External Script node properties as follows:

Procedure

Step 1 Select the Unified ICM Script/External Script Name you want to run.

Step 2 Optionally, add comments and connection labels.

VRU Errors

The following table lists the possible values for the VruStatus variable:

Table 4: VruStatus Variable Codes

Value	Meaning	Description
0	VRU_SUCCESS	The last VRU node was successful.
1	VRU_ERROR	The last VRU node failed because of a routing or configuration error.
2	VRU_TIMEOUT	The last Send To VRU or Translation Route to VRU node 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.
3	VRU_ABORTED	The last VRU node did not complete because the caller ended the call or stopped responding. (Because this causes the routing script to terminate immediately, this value is never seen.)
4	VRU_DIALOG_ FAILED	The last VRU node failed because communication with the VRU ended unexpectedly.
5	VRU_SCRIPT_ NOT_FOUND	The VRU failed because the referenced VRU script was not found in the Unified ICM configuration.
6	STATUS_MAX_QUEUE_LIMIT_EXCEEDED	The last node failed because the maximum call queuing limit was exceeded.
7	STATUS_NO_VALID_EXPRESSION	The last node failed because no valid expression was found.
8	STATUS_NO_VALID_TARGET	The last node failed because no valid target was found.
10	STATUS_NO_MRD_MATCH	The last node failed because no targets matched with the Media Routing Domain on the call.

Value	Meaning	Description
11	STATUS_CONSIDER_IF_FAILED	The last node failed because the Consider If expression failed on all targets.
12	STATUS_NO_VALID_PERIPHERAL	The last node failed because none of the targets were configured on the supported peripheral.
13	STATUS_NO_ONLINE_PERIPHERAL	The last node failed because all targets are on peripherals that are offline.

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, an enterprise skill group, a precision queue, or one or more scheduled targets. 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 one or more skill groups, an enterprise skill group, or one or more scheduled targets.

- Adjust the priority of call in a queue for one or more skill groups or scheduled targets.

- Remove the current call from any queues to which it is assigned.

Call Flow:

1. The call is first sent to the Network VRU. This step is required before you queue the call.

2. The call is queued for three skill groups.

3. 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 a VRU . Routing script execution ends and the call is delivered to the target.

   You could use other nodes like Queue to Skill Group or Queue to Precision Queue to queue the calls to different targets.

⚠

Caution   Do not use the nodes like Route Select to queue the calls when the script cycles between a Wait node and a Run External Script node.

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 VRU for one or more skill groups, enterprise skill group, or one or more scheduled targets using the Queue node (in the **Queue** tab of the Palette).

Figure 89: The Queue Icon

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

Note   If the current call is not at a VRU when the Queue node is run, the Unified ICM sends the call to the associated Network VRU. (This does not apply to Type 2 or Type 8 VRUs, which are VRUs at customer premises.)

You cannot reference two types of targets (for example, skill groups and scheduled targets) within a single Queue node. However, you can run multiple Queue nodes sequentially to queue a call to different target types.

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 20 (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 to 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 90: Queue to Skill Group Properties
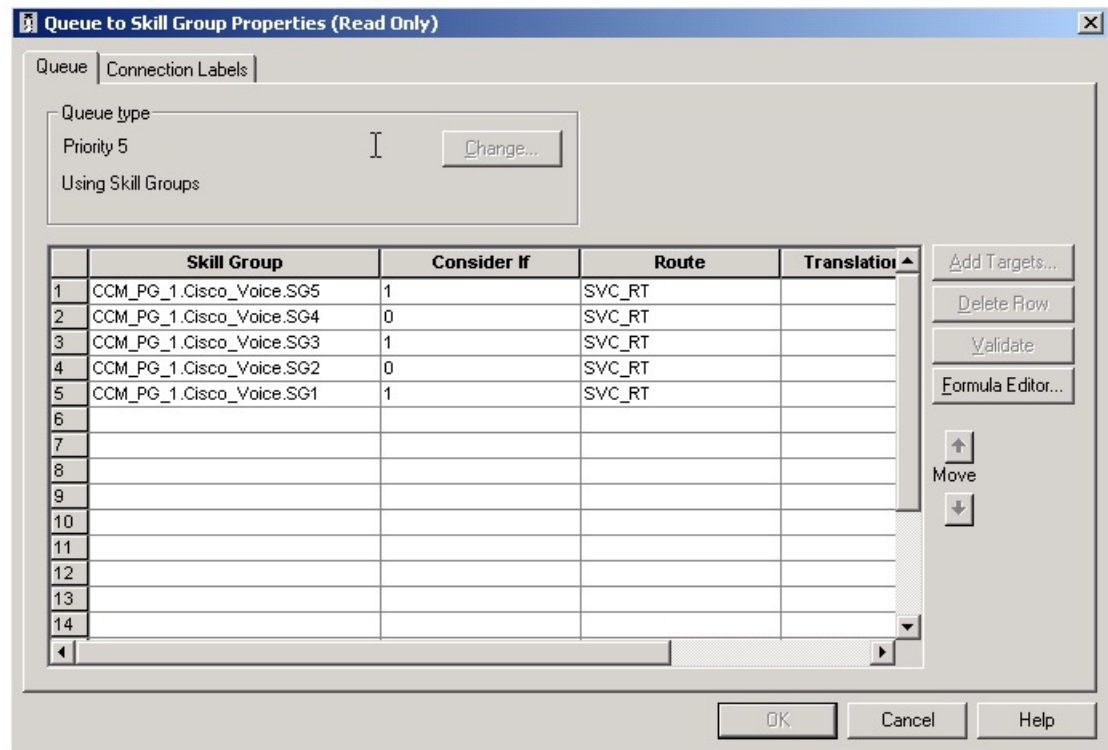
Define Queue node properties as follows:

Procedure

Step 1 To change the queue type:

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

b) Select a **Target Type** (Enterprise Skill Group, Scheduled Target, or 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, run multiple Queue nodes sequentially.

c) Optionally, select a Business Entity and Enterprise Target.

d) 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 20 for least priority. (The default is 5.)

e) 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.

f) 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 foreach target member:

a) Route. (Drop-down list.) 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.)

b) Translation Route. (Drop-down list.) The route to send the call for initial VRU processing if you select this target. (The list contains all translation routes associated with the same peripheral as the target.)

c) Scheduled Target. Individual targets to which the call is queued, if the Target Type is Scheduled Target.

Step 4   Optionally, add connection labels.

What to do next

Note   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.

Related Topics

Precision Queue Script Node

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

Figure 91: Precision Queue Script Node

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 run 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 20 (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 92: Precision Queue Properties Dialog Box—Static Precision Queue
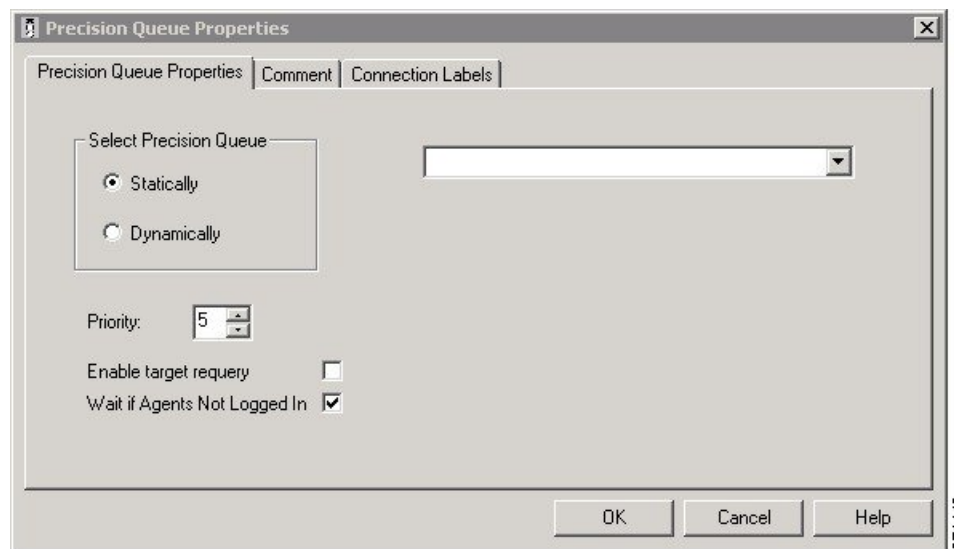
The following property is unique to static precision queues:

- **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** 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 20. 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.

- **Wait if Agents Not Logged In check box** — When this check box is selected and the agents who are associated with a step are not logged in, then the router will wait for the time that is configured for that step. When this check box is not selected, the router will not wait on any step. However, on the last step, the router will wait indefinitely irrespective of the selection.

Related Topics

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 93: Precision Queue Properties Dialog Box—Dynamic Precision Queue
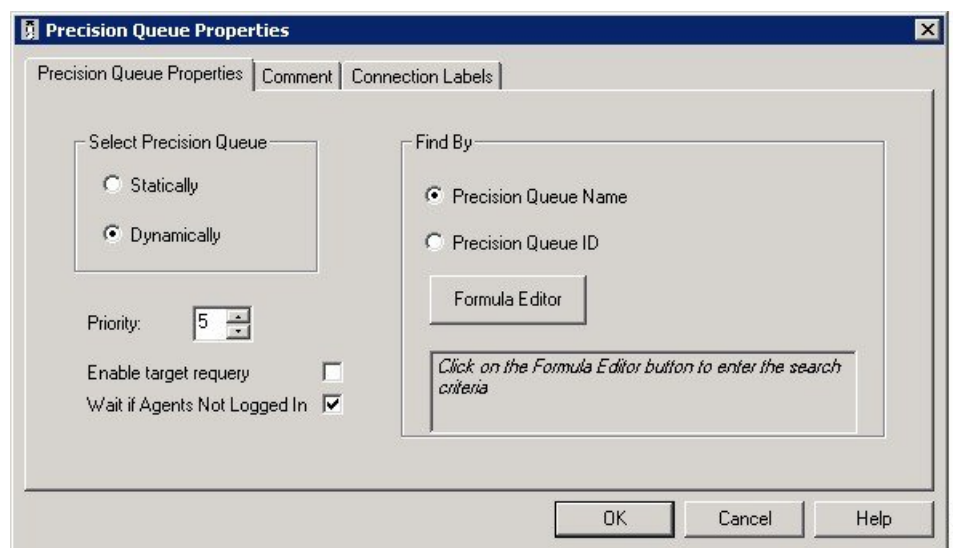
Note Dynamic Precision Queue selection is not available when an External Authorization server is used with Internet Script Editor and will be grayed out in the interface.

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 VRU call processing.

Note The section on static precision queues describes the properties that are common to static and dynamic precision queues.

Queuing Behavior of the 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 perform, 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.

If there is an update to the precision queue definition, then all queued calls in the precision queue are re-evaluated and the execution begins again from the first step.

For example, consider the wait time for an ongoing call at step 1 to be 1080 seconds, of which 1000 seconds has already elapsed. Now, suppose the wait time is changed to 900 seconds, then the wait time for this call is also reset to 900 seconds, even though only 80 more seconds are left to move to the next step.

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 94: 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 is run 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. As the delay for Platinum customers is greater than 20 seconds, no Standard customers are handled ever. 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 95: Queue Priority Properties
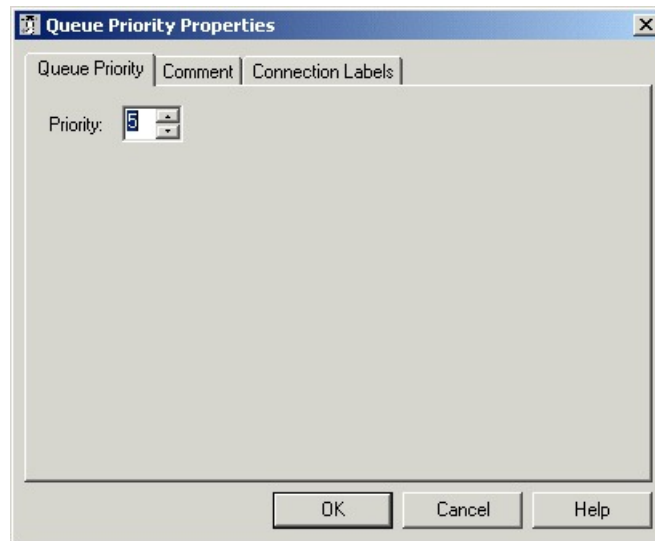
Remove Call from a Queue

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

Figure 96: The Cancel Queuing Icon

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

VRU MicroApp Nodes

Note   You can only use these nodes if supported by your VRU.

The three VRU MicroApp nodes (Collect Data, Menu, Play) are essentially specialized Run External Script nodes. They allow you to specify all details of the interaction in the script node, rather than using the Unified ICM Configuration Manager to specify the Network VRU Script.

The three MicroApp nodes each represent a specific sort of VRU interaction:

• Collect Data instructs the VRU to collect data from a caller after playing a prompt. The prompt can be played using a recorded announcement or using TTS. The data collected can be collected using touch-tone or using Automatic Speech Recognition (ASR).

• Menu is a simplified form of Collect Data that instructs the VRU to build a menu. It prompts the caller (either using a recorded announcement or TTS) and collects a single digit (either using touch tone or ASR). At the Menu node the script branches, depending on the input from the caller.

• Play instructs the VRU to speak a recorded announcement or a data element, such as a number or date. This can be done either by using recorded announcements or through Text-to-Speech (TTS) capabilities.

Collect Data From Caller

You can have a script play a prompt and instruct the caller to enter information by using the Collect Data node (in the Queue tab of the Palette) . The caller-entered data can then be used to redirect the call to the appropriate destination.

Figure 97: Collect icon

Following is the Properties dialog box for the Collect Data node:

Figure 98: Collect Data Properties

Define Collect Data node properties as follows:

Procedure

Step 1   In the File Name field, enter the name of the media file to be played to the caller.

Step 2   In the Library drop-down list, select the location of the file. You can select:

a)   System

b)   Application (default)

c)   None

Step 3   In the Protocols drop-down list, select the data transmission convention to use for the media file contents. You can select:

a)   HTTP: (Hypertext Transfer Protocol, the default)

b)   RSTP: (streaming)

c)   file:

d)   other

Step 4   Specify the number of digits a caller can enter:

a)   In the **Minimum number of digits** field, select a number from 1 to 32 to indicate the minimum number of digits the caller must enter. The default is 1.

b)   In the **Maximum number of digits** field, select a number from 1 to 32 to indicate the maximum number of digits the caller must enter. The default is 1.

c)   In the **Termination key** field, enter the key that the caller presses to signify the end of digit entry. Value options are the digits 0-9, # (pound, the default) or * (asterisk). For variable-length data entry, only. A key that the user presses to signify the end of digit entry. Valid options: The digits 0-9, # (pound, the default) or * (asterisk).

Note   If the minimum number value equals the maximum number value, this field is grayed out.

Step 5   Optionally, check **Use ASR** to use automatic speech recognition. If you check this option:

a)   Caller-entered information is obtained from spoken input as well at DTMF entry.

b)   In the **Automatic speech recognition grammar** text box, enter a grammar against which caller spoken-input is matched.

Step 6   Optionally , check **Allow barge-in** to have any digit entry by the caller interrupt the media playback. Unified CVP deals with barge-in as follows: If barge-in is not allowed (not checked), the Voice Browser continues to prompt play when a caller starts entering digits. If barge-in is allowed (checked), the Voice Browser discontinues prompt play when the caller starts entering digits.

Step 7   Optionally, click **Advanced Properties** to specify how the micro-application should handle invalid or timed-out entries.The Advanced Properties override the VRU Default settings configured in the Unified ICM configuration database. The values you set apply only to the current node; other Collect Data nodes are not affected. Initial values for timeouts and number of tries on this dialog box show the current values, obtained either from the database defaults or from a VRU Settings node. Invalid entry and No entry Media file names may be defined on the VRU device and invisible to Unified ICM environment. If this is the first micro-application operation in the script, the value for these fields is Default; it remains this value unless a VRU Settings micro-application changes it.

a)   Optionally, clear Unified ICM may interrupt. When checked (the default), the operation of the node can be interrupted by the router.

b)   Optionally, clear Inter-digit timeout. When checked (the default), enter the number of seconds allowed between entering digits before the system assumes the caller is finished. Valid options: 1-99, default: 3.

c) In both the Invalid Entry Media Properties and No Entry Media Properties sections, enter the file names of the files to play if the caller enters invalid data or if the caller enters no data.

Note   To use the default settings on the VRU device, enter the word Default in these fields.

d) Select the locations of the files in the Library drop-down lists. Valid options: System, Application (default), None.

Note   If **None**, include the full path file name in the File name value; for example: http://www.xyzcorp.com/Media_Folder/File.wav.

e) Select the data transmission convention to use in the Protocol drop-down lists. Valid options: HTTP: (Hypertext Transfer Protocol, the default), RSTP: (streaming), file:, other.

f) Enter a number in the Number of tries field to indicate the number of times the Collect Data or Menu MicroApp queries the user for data when the user enters invalid data or does not enter data. Valid options: 1-9 (default: 3).

Note   Optionally, you can clear the check box to have the default value defined in the database used.

g) In the **No Entry Media Properties** section, enter a number in the Timeout field to indicate the number of seconds a caller is allowed to begin entering digits. If exceeded, the system times-out. Valid options: 1-99, default: 5.

Note   If not checked, the default value defined in the database is used.

h) When you have finished specifying advanced properties, click **OK**.

Step 8   Optionally, add comments or connection labels.

Prompt Caller to Select from a Set of Options

You can have a script play a prompt and instruct the caller to select from a list of options using the Menu node (in the Queue tab of the Palette).

Figure 99: The Menu Icon

You can then use the caller-entered data to redirect the call to the appropriate destination.

Following is the Properties dialog box for the Menu node:

Figure 100: Menu Properties

Note　You must define one or more success connections from the Menu node before you can define Menu node properties.

Define Menu node properties as follows:

Procedure

Step 1　In the **File Name** text box, enter the name of the media file to be played to the caller.

Step 2　In the Library drop-down list, select the location of the file. You can select:

   a)　System
   b)　Application (default)
   c)　None

Step 3　In the Protocols drop-down list, select the data transmission convention to use for the media file contents. You can select:

   a)　HTTP: (Hypertext Transfer Protocol, the default)
   b)　RSTP: (streaming)
   c)　file:

    d) other

Step 4 Define menu choices. For each row beginning with the possible caller entered digit (which you cannot edit):

    a) In the **Case** column, select a number from the drop-down list. The available numbers correspond to the success branches from the Menu node. The number indicates that if the caller enters the digit in that row, the call processing continues down that success branch.

    b) Enter a description from the selected menu choice.

Step 5 Optionally, check **Use ASR** to use automatic speech recognition. If you check this option:

    a) Caller-entered information is obtained from spoken input as well at DTMF entry.

    b) In the **Automatic speech recognition grammar** text box, enter a grammar against which caller spoken-input is matched.

Step 6 Optionally , check **Allow barge-in** to have any digit entry by the caller interrupt the media playback.

> **Note** The Unified CVP deals with barge-in as follows: If barge-in is not allowed (not checked), the Voice Browser continues to prompt play when a caller starts entering digits. If barge-in is allowed (checked), the Voice Browser discontinues prompt play when the caller starts entering digits.

Step 7 Optionally, click **Advanced Properties** to specify how the micro-application should handle invalid or timed-out entries. The Advanced Properties override the VRU Default settings configured in the Unified ICM configuration database. The values you set apply only to the current node; other Collect Data nodes are not affected. Initial values for timeouts and number of tries on this dialog box show the current values, obtained either from the database defaults or from a VRU Settings node. Invalid entry and No entry Media file names may be defined on the VRU device and invisible to Unified ICM environment. If this is the first micro-application operation in the script, the value for these fields is Default; it remains this value unless a VRU Settings micro-application changes it.

    a) Optionally, clear Unified ICM may interrupt. When checked (the default), the operation of the node can be interrupted by the router.

    b) Optionally, clear Inter-digit timeout. When checked (the default), enter the number of seconds allowed between entering digits before the system assumes the caller is finished. Valid options: 1-99, default: 3.

    c) In both the Invalid Entry Media Properties and No Entry Media Properties sections, enter the file names of the files to play if the caller enters invalid data or if the caller enters no data.

> **Note** To use the default settings on the VRU device, enter the word Default in these fields.

    d) Select the locations of the files in the Library drop-down lists. Valid options: System, Application (default), None.

> **Note** If **None**, include the full path file name in the File name value; for example: http://www.xyzcorp.com/Media_Folder/File.wav.

    e) Select the data transmission convention to use in the Protocol drop-down lists. Valid options: HTTP: (Hypertext Transfer Protocol, the default), RSTP: (streaming), file:, other.

    f) Enter a number in the Number of tries field to indicate the number of times the Collect Data or Menu MicroApp queries the user for data when the user enters invalid data or does not enter data. Valid options: 1-9 (default: 3).

> **Note** Optionally, you can clear the check box to have the default value defined in the database used.

    g) In the No Entry Media Properties section, enter a number in the Timeout field to indicate the number of seconds a caller is allowed to begin entering digits. If exceeded, the system times-out. Valid options: 1-99, default: 5.

Note   If not selected, the default value defined in the database is used.

    h) When you have finished specifying advanced properties, click **OK**.

Step 8   Optionally, add comments or connection labels.

Play Specific Recordings to Caller

You can instruct the VRU to play a series of media files and/or data to the caller by using the Play node (in the Queue tab of the Palette).

Figure 101: The Play Icon

Data can be a literal string or a formula that evaluates to a string. The data, its type, and the format it is to be played in is sent to VRU as part of the play request.

Note   The CallRouter does not verify the format setting to see if it is valid for the specific data type or micro-application. If the data format is invalid, the micro-application's result code indicates such an error.

Following is the Properties dialog box for the Play node:

Figure 102: Play Properties
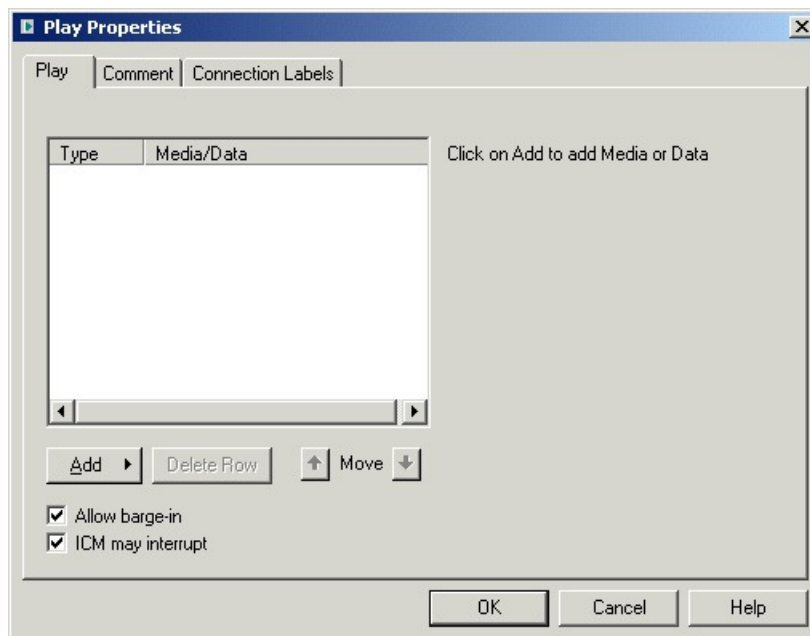
Define Play node properties as follows:

Procedure

Step 1   Click **Add** to add a new file or data element to play to the caller.

Step 2   In the pop-up menu, choose Media or Data.

Step 3   If you selected Media in Step 3:

a) Enter the name of the file to play in the File name field.

b) In the Library drop-down list, select the location of the file. You can choose:

- System

- Application (default)

- None

c) In the **Protocols** drop-down list, select the data transmission convention to use for the media file contents. You can choose:

- HTTP: (Hypertext Transfer Protocol, the default)

- RSTP: (streaming)

- file:

- other

Step 4   If you selected Data in Step 3:

a) In the Data field, enter a string (or a formula that evaluates to a string) to be played by the VRU.

b) In the Data type drop-down list, choose the type of data to be played: Type of data to be played by the VRU. Valid options are:

- Number - Numeric

- Char - Character

- Etime - Elapsed timeTOD - Time of day (12 hr)

- TOD - Time of day

- 24TOD - Time of day (24 hr)

- DOW - Day of week

- Date - Entire date

- Currency - Money units

- Text - text

c) If you have selected the Etime, TOD, or 24TOD options in Step b (in the Time format drop-down list), then you should select the time format. The valid options are:

- HHMM (default) - Hours and minutes

- HHMMSS - Hours, minutes, and seconds

- HHMMAP (TOD format, only) - Hours and minutes, A.M. or P.M.

Note   If the minimum number value equals the maximum number value, this field is grayed out.

Step 5   Optionally, clear the **Allow barge-in** check box to not have any digit entry by the caller interrupt the media playback. Unified CVP deals with barge-in as follows: If barge-in is not allowed (not checked), the Voice Browser continues to prompt play when a caller starts entering digits. If barge-in is allowed (checked), the Voice Browser discontinues prompt play when the caller starts entering digits.

Step 6   Optionally, clear the **ICM may interrupt** check box. When checked (the default), the operation of the node can be interrupted by the router.

Step 7   Optionally, add comments or connection labels.

Override VRU Settings

You can override a default VRU setting on a call-by-call basis by using the VRU Settings node (in the Queue tab of the Palette).

Figure 103: The VRU Settings Icon

You can only override one VRU setting with the VRU Settings node; you must use additional nodes to override additional settings.

Following is the Properties dialog box for the VRU Settings node:

Figure 104: VRU Settings Properties
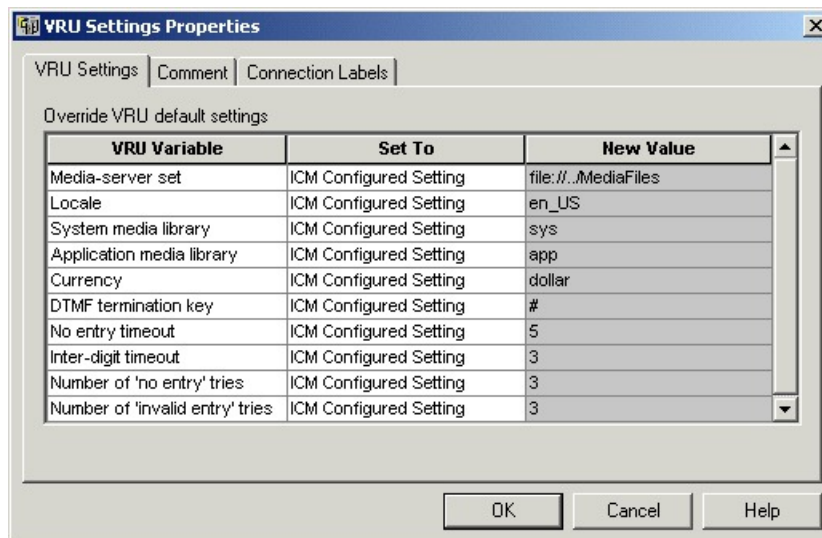
Define VRU Settings node properties as follows:

Procedure

Step 1	In the VRU Variable list, select the VRU variable to override.
Step 2	Select an option in the Set To field:

a) Select ICM Configured Setting to have Script Editor set the variable to the value stored in the VRU_Defaults table, at runtime.

b) Select VRU Peripheral Setting to have Script Editor, at runtime, leave the variable's value as is.

> **Note** This value might be from the VRU_Defaults table, or consist of a value defined through a previous VRU Settings node.

c) Select Customer Defined Value to have Script Editor, at runtime, set the variable to the value specified in the New Value field. If you select this setting, the New Value field opens for editing. Specify a value in this field. (The variable type determines the type of data you can enter.)

Step 3	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 105: 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 106: Wait Properties
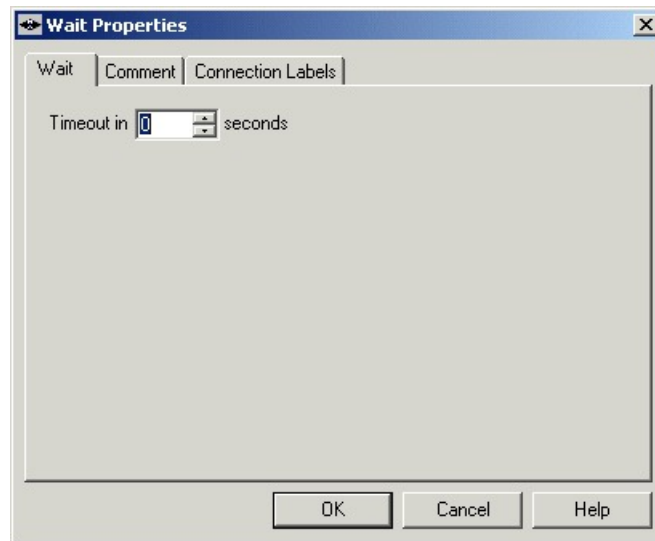
Define Wait node properties as follows:

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 your system is integrated with multichannel features, you write routing scripts to route contacts that these features handle. Multichannel features include Enterprise Chat and Email, and third-party multichannel applications that use the Task Routing APIs.

Enterprise Chat and Email

You can configure CCE deployments with Enterprise Chat and Email to use independent media queues, in which an agent can handle tasks for a single media channel. You can also useTask Routing, in which an agent can handle tasks for several media channels.

Supported Route Requests for ECE Web Request

Unified CCE supports the following types of multichannel route requests when integrated with ECE:

- **Web callback** - A web callback request is a request that does not involve ECE. You click a button on a website that says, "Call me back." Then the caller and agent simply talk on the phone.

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

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

Application Request Routing with ECE Web Request

The ECE routes requests to a 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 Router, which runs a routing script to determine how best to handle the request.

Unified CCE uses a media class ID to identify the type of media or channel. A media class is a communication channel that correlates to an application. Following are the predefined media classes in Unified CCE:

- **Cisco_Chat** - chat requests

- **Cisco_Voice** - web and delayed callbacks requests, and basic Unified CCE inbound and outbound voice calls

- **Cisco_Email** - email requests

Each media class has at least one Media Routing Domain (MRD), which is a collection of skill groups and services that are associated with a medium. The Unified 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 Unified CCE script, but it is possible to route requests from different MRDs using one script.

Synchronized Agents and Skill Groups for ECE

Agents are common across the multichannel software, but skill groups are application-specific. You can create agents using ECE or in contact center enterprise solutions and share the agents across applications. When agents or skill groups are created in ECE, they are simultaneously created in contact center enterprise solutions. If an agent is created in contact center enterprise solutions, you must enable the agent in ECE before the agent can work on those applications.

Only create, modify, or delete ECE skill groups in ECE. Skill groups are application-specific. When you create a skill group in ECE, the skill group is simultaneously created in the contact center enterprise solutions. But, you cannot enable that skill group in the core contact center enterprise applications.

Independent Media Queues for ECE

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

You can use independent media queues in both the Unified CCE and ACD environments.

Task Routing

Task Routing describes the system's ability to route requests from different media channels to agents who work with customer contacts in multiple media. Routing scripts can send requests to agents based on business rules regardless of the media channel from which the request came. For example, based on an agent's skills and current tasks, Unified CCE can route phone, chat, and email message requests to an agent who works with all these media. The agent can switch media on a task-by-task basis.

You can set up routing scripts so that multichannel tasks are assigned to the longest available agent in a skill group or precision queue in same Media Routing Domain as the task. You can also prioritize multichannel tasks using skill group and precision queue routing, as you would voice calls.

Because Task Routing must coordinate an agent's work on multiple tasks across different channels, Task Routing requires that Unified ICM have complete control over how agents are assigned tasks. Therefore, Task Routing requires that you use the Unified ICM in a Unified CCE environment.Task Routing is not supported in legacy ACD environments.

Digital Routing using Webex Connect

You can configure the Contact Center Enterprise (CCE) deployments with Webex Connect using the Digital Routing service, in which agents can handle tasks for several digital channels such as SMS, chat, and email.

Digital Routing Task Requests

The Digital Routing service that is installed on Cloud Connect routes the digital channel tasks to the router through a 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 Router, which runs a routing script to determine how best to handle the request. You must have certain mandatory ECC variables defined in CCE for the Digital Routing tasks to be routed successfully. For the list of mandatory ECC Variables that must be configured for the Digital Routing Tasks, see ECC Variables for Digital Routing Tasks, on page 124. For instructions about how to add the ECC variables in CCE, see the *Define ECC variables* section in the Cisco Unified Contact Center Enterprise Features Guide.

If you are using ECC payloads, you must associate the ECC payloads that you created for the Digital Routing service to the Network VRU using the Network VRU Explorer tool in the Unified ICM Configuration Manager. You must then associate the Network VRU to the MR-PG using the PG Explorer tool in the Unified ICM Configuration Manager. For instructions about how to associate ECC payloads to Network VRU and the Network VRU to the MR-PG, see the *Configure MR PG* section in the Configuration Guide for Cisco Unified ICM Enterprise.

In case you use different ECC payloads, that is one for receiving task context data from MR-PG to the Router and another to route the tasks to Finesse through Agent PG, ensure that the ECC variables defined for Digital Routing tasks are part of both the ECC payloads. If you use the system's "Default" payload, ensure that the ECC variables are part of the "Default" payload. These ECC variables carry task metadata that the Agent Desktop uses to identify a task as the digital channel task and to interact with Webex Engage to load the media pane. If you do not define these variables and add them in the Unified CCE Administration portal, the tasks do not appear in the Agent Desktop correctly. You can create and modify ECC payloads in the Configuration Manager > List Tools > Expanded Call Variable Payload List tool. For more information about *ECC variables and payloads*, see Expanded Call Context (ECC) Variables.

ECC Variables for Digital Routing Tasks

This section provides the list of ECC Variables that you must configure for the Digital Channel interaction. All ECC variables are of type scalar and they are case-sensitive.

Table 5: ECC Variables

Variable	Table Size	Length	Mandatory / Optional
user_DR_Primary	Used to hold the FQDN of either the publisher node or the subscriber node of Cloud Connect. The FQDN denotes the Digital Routing service node that injected the task into CCE.	[1]1 character + the length of the Cloud Connect FQDN.	Mandatory
user_DR_Backup	Used to hold the FQDN of either the publisher node or the subscriber node of Cloud Connect. The FQDN denotes the peer node of the Digital Routing service, which can be used as a backup, in case the Primary node is unavailable.	1 character + the length of the Cloud Connect FQDN.	Mandatory
user_DR_MediaResourceID	Used to hold the Conversation ID of Webex Engage, which the Webex Connect injects. The Cisco Finesse uses this conversation ID to load the media in the Manage Digital Channels gadget.	16 characters	Mandatory
user_DR_CustomerName	Used to display the customer name in the pop-over notification that an agent views when a task is received in the Manage Digital Channels gadget.	Up to a maximum of 210 characters.	Optional

Variable	Table Size	Length	Mandatory / Optional
`user_DR_MediaChannelName`	Used to populate the media channel name for the task, if you configure and add this ECC variable in the **Unified Contact Center Enterprise Management** portal. See the *Define ECC variables* section in the Cisco Unified Contact Center Enterprise Features Guide. This is typically required if you have mapped more than one Digital Routing media channel to the same MRD but still want unique task icons to appear on the agent desktop. For example, in CCE, the system-defined media channels, chat and SMS are both mapped to the same MRD. If there is a 1:1 mapping between the media channel and MRD, the system automatically identifies the icon to be displayed on the desktop based on the media channel with which the task was associated.	8 character	Optional

[1] If there are lengthy FQDNs, the Digital Routing service compresses the FQDN such that it is below 100 characters, and thereby allow ECC variable some space out of the 2000 bytes limit per call / task to be saved by not having to send the long name. In such cases, the variable can be defined as 100 characters long in CCE. The Digital Routing service automatically fills in the first character to indicate whether the following data is compressed or uncompressed. A prefix of 1 denotes compressed data, while a prefix of 0 denotes uncompressed data.

Media Routing Domains

Media Routing Domains (MRDs) organize how requests for each communication medium, such as voice and email, are routed to agents. For example, the Enterprise Chat and Email uses a Unified CCE MRD to route a task to an agent who is associated with a skill group or precision queue and a particular channel.

For Enterprise Chat and Email, configure MRDs in Configuration Manager. For custom multichannel applications that use the Task Routing APIs, configure MRDs in Unified CCE Administration.

For Digital Channel tasks that are initiated from Webex Connect, you can create MRDs either using the Media Routing Domain List tool in Configuration Manager, or through Unified CCE Administration tool. If you create the MRDs through the Unified CCE Administration portal, the MRDs get automatically associated to the system-defined application paths. If you have created the MRDs using the Media Routing Domain List tool in Configuration Manager, you must explicitly associate the MRD with the application path.

For every agent peripheral gateway, there is a system-defined application path that gets created with a suffix "UQ.Desktop". There is also an associated system-defined application called UQ.Desktop that automatically gets created in the sytem and identifies the Cisco Finesse server as a client to the Agent PG, to control Agent states in MRDs created for digital channels. The first number in the application path identifies the Logical Controller ID of the Agent PG. An example of the system-defined application path is `5000.UQ.Desktop`.

Media Routing Domains and Interruptibility

When you configure MRDs, you indicate whether tasks for the MRD are interruptible. If the MRD is not interruptible, an 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, tasks in which the agent and customer interact synchronously, such as voice calls and chats, are not interruptible. Email messages are typically interruptible because contact with the customer is asynchronous. Therefore, an agent responding to an email 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 of the route request.

For example, you can have different MRDs for email and chat. You can have a single script for both types of requests that branches so that it routes email messages and chats to different targets.

Note For multichannel tasks submitted by applications using the Task Routing APIs, Unified CCE determines the MRD based on the dialed number/script selector in the task request.

Use the **Media Routing Domain** node (in the Routing tab of the Palette).

Figure 107: MRD Domain Icon

Insert targets and connections from the MRD node before you define the node's properties.

Note A branch can include multiple MRDs, but you can associate a single MRD with only one branch.

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

Figure 108: Media Routing Domain Properties
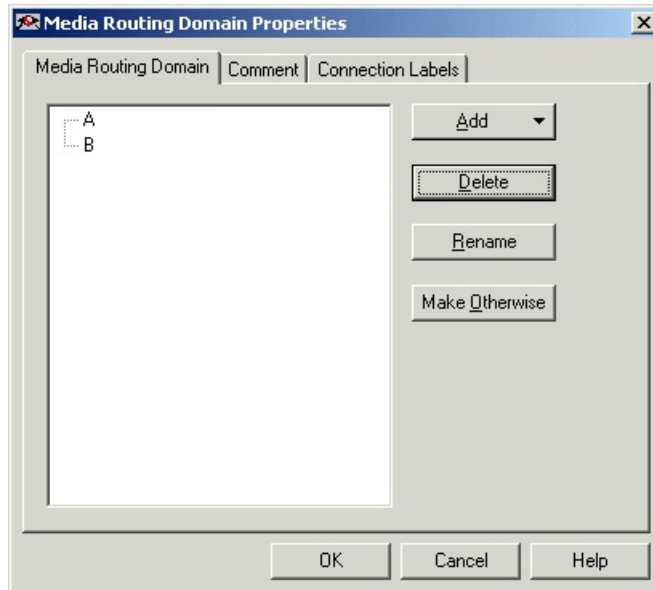
Define Media Routing Domain node properties as follows:

Procedure

Step 1   To associate an MRD with a branch, select the branch:
a) Click **Add**.
b) Choose an 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.

Pick / Pull Node

The **Pick / Pull** node is available from the **Routing** tab **Object Palette**.

Pick / Pull node verifies that the route request is a valid pick or pull request for the specified MRD. The node checks that *serviceRequested* is one of the following:

- 4 = ROUTE_SERVICE_REQUEST_PICK_EXT_QUEUE

- 5 = ROUTE_SERVICE_REQUEST_PULL_EXT_QUEUE

- 6 = ROUTE_SERVICE_REQUEST_PICK_UCCE_AGENT

• 7 = ROUTE_SERVICE_REQUEST_TRANSFER_PICK

For these values, the script proceeds and terminates after the pick or pull is completed. For any other value of *serviceRequested*, the request flows through the node's failure branch. Check the error code in *VRUStatus*.

The node provides real-time monitoring of the following:

- Pick successes.
- Pull successes.
- Pick errors.
- Pull errors.

The following meters are applicable for the Pick/Pull node.

- Pick meter - Displays the number of nonvoice task pick requests that were successful.
- Pull meter - Displays the number of nonvoice task pull requests that were successful.
- Pick error meter - Displays the number of nonvoice task pick requests that failed.
- Pull error meter - Displays the number of nonvoice task pull requests that failed.

Note	All **Pick / Pull** routing works for agents in the **Ready** or **Available** state. To be able to do this in **Not Ready** state, enable the Registry key. The Registry key that needs to be enabled on the CCE Router is EnablePickPullWhileInNotReady. This is located under the following registry hive on the CCE Router VM

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco
Systems,Inc.\ICM\ucce\RouterA\Router\CurrentVersion\Configuration\Config.
```

The Registry key is dynamic and doesn't require a restart of the router, for changes to take effect. The default value is 0 or disabled. To enable the Registry key, set the value to 1.

Skill Group and Precision Queue Routing for Nonvoice Tasks

Routing to skill groups and precision queues is largely the same for voice calls and nonvoice tasks. However, the way that contact center enterprise distributes tasks has the following implications for agents who can handle multiple concurrent tasks:

- **Precision queues**—In precision queue routing, Unified CCE assigns tasks to agents in order of the precision queue steps. Unified CCE assigns tasks to agents who match the attributes for step one, up to their task limit, until all those agents are busy. Unified CCE then assigns tasks to agents who match attributes for step two, and so on. If you configure agents to handle three concurrent tasks, Unified CCE assigns three tasks to each agent in the first step. It then moves on to the second step and assigns any remaining tasks to those agents.

- **Overflow skill groups**—Routing scripts can specify a preferred skill group and an overflow skill group. Unified CCE assigns tasks to all agents in the preferred skill group, up to their task limit, before assigning any tasks in the overflow skill group. If you configure agents to handle three concurrent tasks, Unified CCE assigns three tasks to each agent in the preferred skill group. It then moves on to the overflow skill group and assigns any remaining tasks to those agents.

Note	The number of available slots is an important factor in the Longest Available Agent (LAA) calculation.

The number of available slots = The maximum concurrent task limit for the MRD that an Agent has logged into **-** Current tasks being handled by the Agent or routed to the Agent.

If there are multiple skill groups that are part of the queue node, then the skill group that has the higher LAA is picked. Then, the agents within the picked skill group (or the Precision Queue) who have the highest number of available slots for non-voice tasks get prioritised.

Agents with the same number of available slots get prioritized based on the time in the available state or the LAA mechanism. |

Related Topics

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

Related Topics

Change Queue to Agent Type

Procedure

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

Figure 109: Queue Agent Type
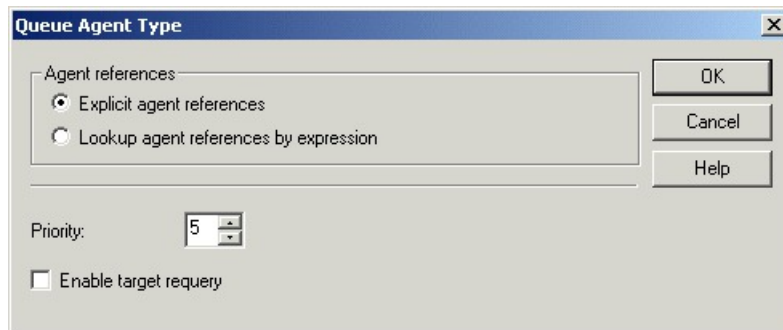
Step 2	To select a specific agent, select **Explicit agent references**.
Step 3	To select and agent by an expression, select **Lookup agent references by expression**.
Step 4	Select a **Priority** between 1 (the highest) and 20 (the lowest).
Step 5	Optionally, select **Enable target requery**.

Specify an Agent Directly

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

Figure 110: Agent Direct Properties
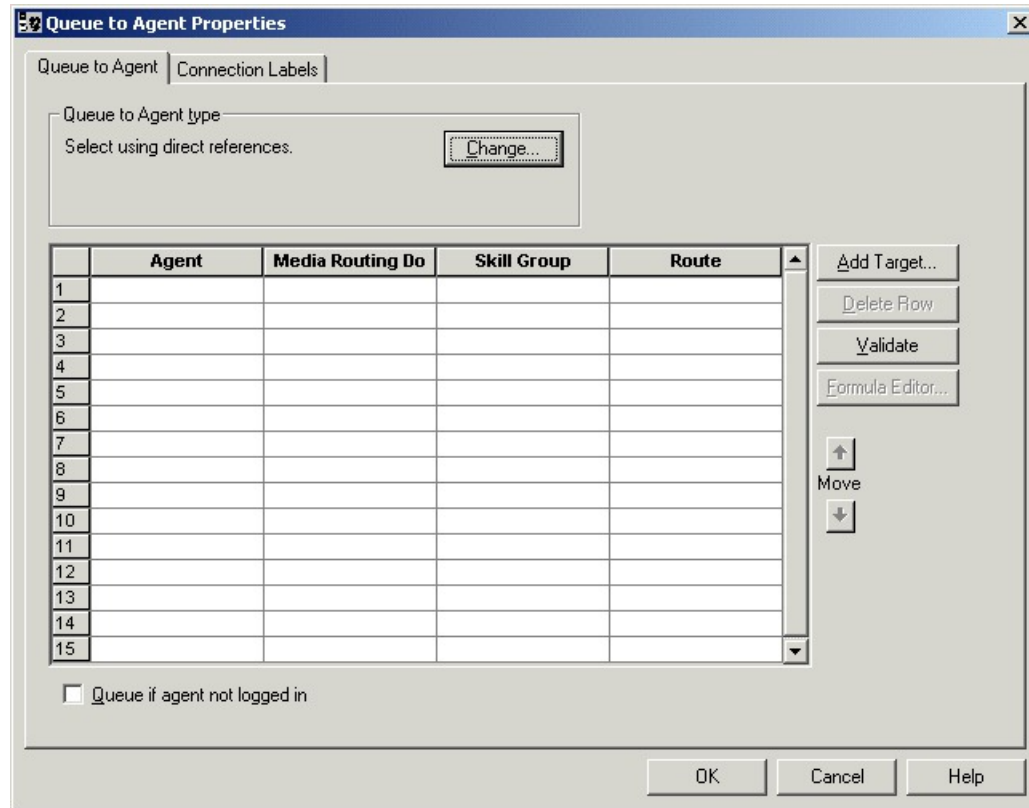
To specify agents directly:

Procedure

Step 1   If necessary, change the Queue to Agent type to **Explicit agent references**.

Step 2   In the **Agent** column, select an agent.

Step 3   In the **Media Routing Domain** column, select the media routing domain for the selected agent.

Step 4   In the **Skill Group** column, select the skill group for the selected agent and media routing domain.

Step 5   In the **Route** column, select the route for the selected agent and media routing domain.

Step 6   Optionally, select **Queue if agent not logged in**, to have the contact queued to the agent even if the agent is not currently logged in.

Step 7   To test the data you entered, click **Validate**.

Step 8   Optionally, modify **Connection Labels**.

Select an Agent by an Expression

Following is the properties dialog box of the **Queue to Agent** node when you select to use an expression:

Figure 111: Queue to Agent Properties

To specify agents by expression:

Procedure

Step 1   If necessary, change the **Queue to Agent** type to **Lookup agent references by expression**.

Step 2   In the **Peripheral** column, to choose an agent by Peripheral number, choose a peripheral and provide a formula in the Agent Expression that translates to the Agent's Peripheral number. If no peripheral is chosen, the agent expression should translate to the SkillTargetID or Enterprise Name of the agent.

Step 3   In the **Agent Expression** column, provide a formula using the formula editor that translates to the agent's Peripheral number or SkillTargetID or agent Enterprise Name. To choose a peripheral (in Peripheral) and provide a formula that translates to an agent's Peripheral number.

> **Note**   The Peripheral column controls how the AgentExpression column is evaluated as an ICM ID. However, if you select a Peripheral from the Peripheral column, then the Agent Expression column is evaluated as an Agent Peripheral number.

Step 4   Optionally, in the **Consider if** column, enter the formula that evaluates to true for the target when the ICM system runs the Queue to Agent node, or that target will not be selected. For help in creating a formula, put the cursor in this field and then click the **Formula Editor** button.

Step 5   Select the **Enterprise Skill Group** that includes the appropriate skill groups to cover all media routing domain cases for the selected Agent.

Step 6   Select the **Enterprise Route** that has an appropriate collection of routes, or the **Route**, matching the agent and media routing domain.

If you do not select either an **Enterprise Route** or a **Route**, the following error message appears:

Figure 112: No Enterprise Route or Route Chosen
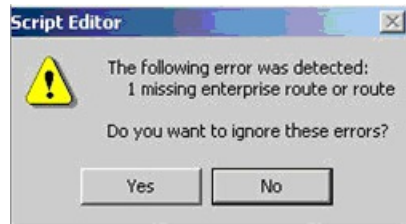
If you select both **Enterprise Route** and **Route**, the following error message appears:

Figure 113: Both an Enterprise Route and a Route Chosen
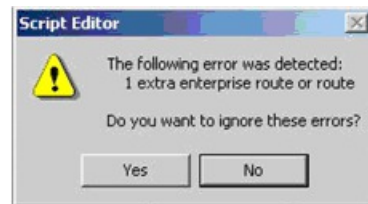
The specified **Enterprise Route** or **Route** is used to send the call to an agent.

Step 7   Optionally, select **Queue if agent not logged in**, to have the contact queued to the agent even if the agent is not currently logged in.

Step 8   To test the data you entered, click **Validate**.

Step 9   Optionally, modify connection labels.

RONA and Transfer Scripting for Task Routing and Digital Routing

Note   This section applies to the following:

> • Tasks submitted by third-party multichannel applications that use the Task Routing APIs.
>
> • Tasks submitted by Webex Connect that use the Digital Routing APIs.

The *ServiceRequested* call variable is set when tasks are transferred or RONA. You can determine the value of the *ServiceRequested* call variable in an **If** node in the routing script. Based on the value of this field, the script can take different actions. For example, the script can raise the priority of the task so that it goes to the front of the queue.

The relevant *ServiceRequested* values are:

> • 2: This value identifies a transferred task.
>
> • 3: This value identifies a RONA task.

Related Topics

Estimated Wait Time Scripting for Task Routing and Digital Routing

> **Note** This section applies to the following:
>
> • Tasks submitted by third-party multichannel applications that use the Task Routing APIs.
>
> • Tasks submitted by Webex Connect that use the Digital Routing APIs.

Customers submitting a task request might want to know approximately how long they will wait for an agent. You can configure the routing script to provide the customer with an estimate of the wait time. The estimated wait time is calculated once, when the task enters the queue. The time is not updated as the position in the queue changes.

The default estimated wait time algorithm is based on a running five minute window of the rate of tasks leaving the queue. Any tasks that are routed or abandoned during the previous 5 minutes are considered as part of the rate leaving queue. For Precision Queues, the rate leaving the queue represents the rate at which tasks are delivered or abandoned from the entire precision queue, not any individual Precision Queue steps. The algorithm computes the wait time for each of the queues against which the task is queued (Skill Groups or Precision Queues) and then returns the minimum estimated wait time.

> **Note** Queue to Agent is not supported.

While the queue builds, the small number of tasks in the queue makes the estimated wait time less accurate and the value fluctuates rapidly. As the queue operates with more tasks over time, the estimated wait time is more accurate and consistent.

Scripts for estimated wait time include:

• A **Set Variable** node, `Call.EstimatedWaitTime` to set the estimated wait time.

• A **Run External Script** node to apply a Network VRU script that returns the estimated wait time to the customer.

Related Topics

Use of Formulas

Formula Usage

A formula consists of one or more expressions that the scripting environment evaluates to produce a value that it can use for subsequent script processing. You define expressions—made up of variables, constants, operators, and functions—as part of custom selection rules or distribution criteria in scripts. (See the sections on variable usage, selection of targets by rules, and distribution of contacts to targets.)

Formula Example

This is an example of a simple formula:

```
CallerEnteredDigits == 1
```

In this example:

- The left value, *CallerEnteredDigits*, is a variable. More specifically, it is a call control variable.

- The operator is the "Equal To" 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:

object-type.object-name.variable-name

Where:

- The object-type is an object category, such as Service.

- The object-name is the name of an object contained in Unified ICM database, such as the name of a service (for example, BosSales).

- The variable-name is the name of an object that can hold a value, such as a call control variable (for example, (CallerEnteredDigits).

- Each component in the variable is separated by a period (.).

Note Passing of internationalized characters through Media Routing interface is not supported. The application that interacts with ICM through the Media Routing interface must send any call related data in English only.

Single-Target Variables

A single-target variable examines data for one specified routing target. For example, the variable:

Service.BosSales.ExpectedDelay: Examines the expected delay for the BosSales service.

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.

Business Hours Variables

Business Hours variables indicate whether a business hour is open or closed along with the ReasonCode. They can be used to decide the routing logic for an incoming contact. As a team lead or business user, you can open, close, or manage Business Hours from a simplified web interface.

Business Hours Variables

The following two variables are available in Script Editor for BusinessHours feature.

1. BusinessHourStatus
2. Reason Code

BusinessHourStatus

BusinessHourStatus variable indicates whether the BusinessHours is open or closed. The variable value 0 indicates the BusinessHours is closed and 1 indicates it is open.

You can use to take the routing decision for incoming call. The Syntax to use the variable is:

```
BusinessHours.<BusinessHoursName>.BusinessHourStatus == 1
```

The above expression will be successful if the BusinessHours status is open.

Reason Code

The Reason Code indicates the reason for the current status of Business Hours. You can use the **ReasonCode** to provide customized treatments for callers when the Business Hours are closed. The syntax to use this variable:

```
BusinessHours.<BusinessHoursName>.ReasonCode. == 1001
```

The above expression will be successful if the Reason code is 1001.

For more information on configuring the ReasonCode and associating with the Business Hours, see the following guides:

at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-feature-guides-list.html

and

at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-technical-reference-list.html.

The BusinessHours variables are available only in the IF node of Script Editor.

Business Hours in Scripts

Using unified ICM script editor, you can configure BusinessHours in ICM script to control call treatment. You can create a simple expression in **IF** node which checks the Configured BusinessHours real time status and return true or false based on expression.

510447
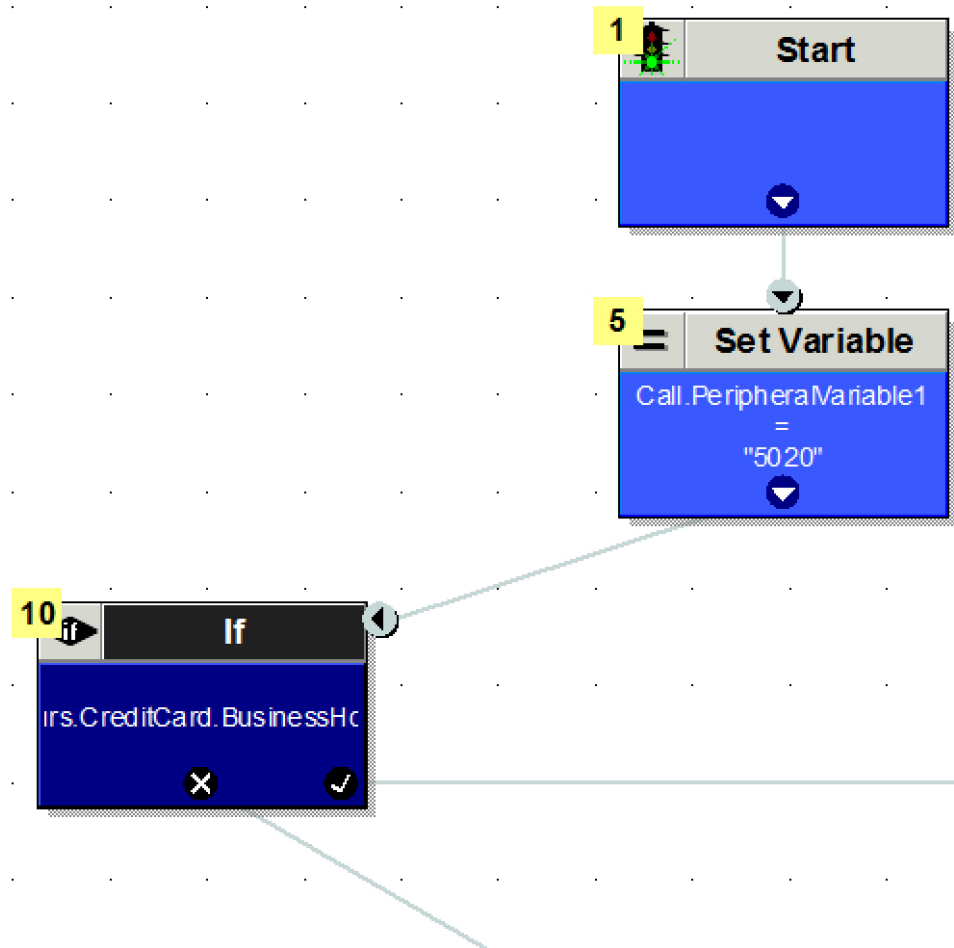
```
BusinessHours.CreditCard.BusinessHourStatus==1
```

The formula given above checks that the real time status of **BusinessHours** named **CreditCard** and returns through Success path of **IF node** when the **Business Hours** are open.

You can create above formula using **Formula Editor** in Script editor.

```
┌──────────────────────────────────────────────────────────────┐
│  Formula Editor                                          [X]   │
│                                                               │
│  Formula:                                          ┌───────┐  │
│  ┌──────────────────────────────────────────┐      │  OK   │  │
│  │ BusinessHours.CreditCard.BusinessHourStatus==1  └───────┘  │
│  │                                          │      ┌───────┐  │
│  │                                          │      │Cancel │  │
│  │                                          │      └───────┘  │
│  │                                          │      ┌───────┐  │
│  │                                          │      │ Help  │  │
│  └──────────────────────────────────────────┘      └───────┘  │
│  Variables │ Built-In Functions │ Custom Functions │          │
│  ┌──────────┐                                                 │
│  │ Paste ▲  │                                                 │
│  └──────────┘                                                 │
│  Object types:         Objects:              Variables:       │
│  ApplicationGateway    {}                                     │
│  ApplicationPath       CreditCard                             │
│  BusinessHours         DebitCard                              │
│  Call                  Technical_Support                      │
│  CallType                                                     │
│  Database                                                     │
│  NetworkTrunkGroup                                            │
│  Peripheral                                                   │
└──────────────────────────────────────────────────────────────┘
```
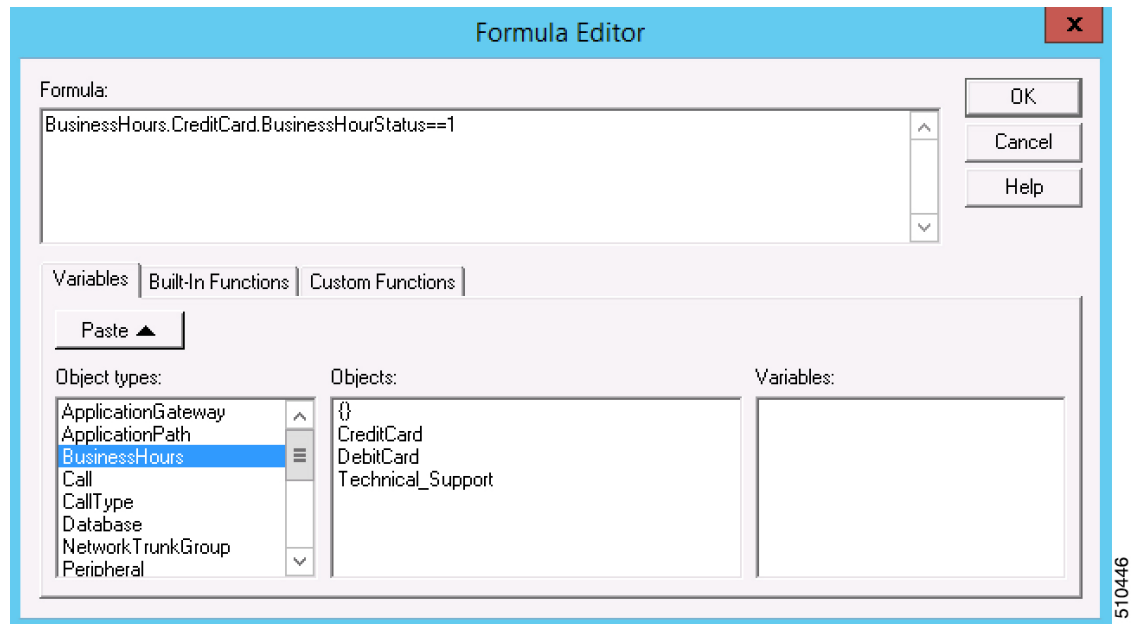
> **Note** Business hour object type is available for selection only when there is at least one Business Hour that is configured in system using CCE Administration web page.

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.

Variable	Data Type	Description	Can be Set by the User
CallerEnteredDigits	String	Digits caller entered in response to prompts.	Yes
CallingLineID	String	Billing telephone number of the caller.	No
CLIDRestricted	Integer	If 1, CLID presentation should be restricted. If 0, CLID presentation should not be restricted.	Set in Unified ICM Configuration Manager. Open Tools > Miscellaneous Tools > System Information. Check Enabled in the CLID Masking section of the screen to turn on.
CustomerProvidedDigits	String	Digits to be passed to the routing client for forwarding to the call recipient.	Yes
DialedNumberString	String	Telephone number dialed by the caller.	No

Variable	Data Type	Description	Can be Set by the User
ExpCallVarName	String	Expanded Call Context (ECC) variable value assigned in scripts and passed with contact.	Yes
NetworkTransferEnabled	Integer	If 1, network transfer is enabled. If 0, network transfer is not enabled.	Yes
PeripheralVariable1- PeripheralVariable10	String	Values passed to and from the peripheral.	Yes
RequeryStatus	Integer	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.	No
RouterCallDay	Integer	An encoded value that indicates the date on which Unified ICM processes the call.	No
RouterCallKey	Integer	A value that is unique among all calls Unified ICM has processed since midnight. RouterCallDay and RouterCallKey combine to form a unique call identifier.	No
RoutingClient	String	The name of the routing client that made the route request.	No
TimeInQueue	Integer	Number of seconds a call has been queued.	No
UserToUserInfo	String	ISDN private network User to User information	Yes
VruStatus	Integer	Indicates the result of a previous VRU node.	No
CallGUID	varchar (32)	Globally unique call identifier.	No
LocationParamName	varchar(50)	Location name.	No
PstnTrunkGroupID	varchar(32)	The Trunk Group ID on which the call arrived on IOS Gateway.	No
PstnTrunkGroupChannelNumber	Integer	The Trunk Group Channel Number on which the call arrived on IOS Gateway.	No
SIPHeader	varchar(255)	Specific header information extracted from a SIP call that arrives at Unified CVP (or VRU).	Yes

Note For a Post-Routing® request from an Aspect ACD, PeripheralVariable1 through PeripheralVariable5 map to the Aspect variables A through E. The Aspect routing client passes these variables to the Unified ICM as part of the request and the Unified ICM returns them with the response. Other routing clients might use some of these variables for other purposes. The values of these variables are also stored in the Route_Call_Detail table of the database.

Note The Call Variables can be used in a "SET" node 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 themselves will have a new set of CallVariables created for the Route Request.

Note 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. Example: value(Call.PeripheralVariable1)>=value(Call.PeripheralVariable2) where Call.PeripheralVariable1 and Call.PeripheralVariable2 are given as Numeric string.

Expanded Call Context (ECC) Variables

Expanded call context (ECC) variables store values associated with the contact.

ECC values are written to Termination Call Detail records only if, and when, an ECC value is explicitly set. You can set the variables in several ways, such as using a script, a VRU, a NIC, CTI, and so on. This applies to null values as well as non-null values.

If an ECC 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.

The Latin 1 Character set for expanded call context variables and peripheral call variables is supported when used with Unified CVP, Cisco Finesse, and Cisco Customer Collaboration Platform, among others.

The use of multi-byte character sets in limited usage for ECC and peripheral call variables is also supported, when:

- Setting them in the Script Editor using double quotes.

- Stored in Termination Call Variables with an appropriate SQL collation.

- Setting and receiving them through agent desktops.

ECC values are generally 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. If it does not involve translation routes or the Unified CCE, and is outside the original peripheral, then the solution cannot propagate ECC variables, like all call variables.

The solution comes with some predefined ECC variables. You can create others through the Configuration Manager.

ECC Payloads

You can define as many ECC variables as necessary. But, you can only pass 2000 bytes of ECC variables on a specific interface at any one time. To aid you in organizing ECC variables for specific purposes, the solution has *ECC payloads*.

An ECC payload is a defined set of ECC variables with a maximum size of 2000 bytes. You can create ECC payloads to suit the necessary information for a given operation. You can include a specific ECC variable in multiple ECC payloads.The particular ECC variables in a given ECC payload are called its *members*.

Note For ECC payloads to a CTI client, the size limit is 2000 bytes plus an extra 500 bytes for the ECC variable names. Unlike other interfaces, the CTI message includes ECC variable names.

In certain cases, mainly when using APIs, you might create an ECC payload that exceeds the CTI Server message size limit. If you use such an ECC payload in a client request, the CTI Server rejects the request. For an OPC message with such an ECC payload, the CTI Server sends the message without the ECC data. In this case, the following event is logged, "CTI Server was unable to forward ECC variables due to an overflow condition."

You can use several ECC payloads in the same call flow, but only one ECC payload has scope at a given moment. TCDs and RCDs record the ID of the ECC payload that had scope during that leg of the call. The *Call.ECCPayloadID* variable contains the ID of the ECC payload which currently has scope.

In solutions that only use the default ECC payload, the system does not create an ECC variable that exceeds the 2000-byte limit for an ECC payload or the 2500-byte CTI Message Size limit. The system does this because it automatically adds all ECC variables to the default ECC payload if that is the only ECC payload.

If you create another ECC payload, the system no longer checks the 2000-byte limit when creating ECC variables. The system creates the ECC variables without assigning them to an ECC payload. Assign the new ECC variable to an appropriate ECC payload yourself through the ECC Payload Tool.

You can create and modify ECC payloads in the **Configuration Manager** > **List Tools** > **Expanded Call Variable Payload List** tool.

Default ECC Payload

The solution includes an ECC payload named "Default" for backward compatibility. If your solution does not require more ECC variable space, you only need the Default payload. The solution uses the Default payload unless you override it.

If your solution only has the Default payload, the solution automatically adds any new ECC variables to the Default payload until it reaches the 2000-byte limit.

Note You cannot delete the Default payload. But, you can change its members.

Important During upgrades, when the system first migrates your existing ECC variables to the Default payload, it does not check the CTI message size limit. The member names might exceed the extra 500 bytes that is allocated for ECC payloads to a CTI client. Manually check the **CTI Message Size** counter in the **Expanded Call Variable Payload List** tool to ensure that the Default payload does not exceed the limit. If the Default payload exceeds the limit, modify it to meet the limit.

In a fresh install, the Default payload includes the predefined system ECC variables. In an upgrade, the Default payload's contents depend on whether the starting release supports ECC payloads:

  • **ECC payloads not supported**—During the upgrade, a script adds your existing ECC variables to the Default payload.

  • **ECC payloads are supported**—The upgrade brings forward the existing definition of your Default payload.

Note   If your solution includes PGs from a previous release that does not support ECC payloads, the Router always sends the Default payload to those PGs. Those PGs can properly handle the Default payload.

ECC Payload Node

The **ECC Payload** node is available from the **General** tab on the **Object Palette**:

Figure 114: Payload icon

Use this node to change the ECC payload that has scope for the following part of your script. Once you select an ECC payload, it has scope for all non-VRU operations until changed. You can select the ECC payload either statically or dynamically by the payload's EnterpriseName or ID.

ECC Payload Use by Interface

This table summarizes the use of ECC payloads in various operations:

Condition	ECC Payload That Is Used
Routing to VRU	Default payload If an ECC payload is specified in the configuration of that VRU, it overrides the Default payload.
Routing to Application Gateway	ECC payload that currently has scope in the script
Routing to Agent PG (including the Unified CM PG and Avaya PG)	ECC payload that currently has scope in the script
Routing to Media Routing PG	Default payload If an ECC payload is specified in the configuration of the VRU for that MR PG, it overrides the Default payload.
Routing to pre-12.0 PG	Always Default payload
Routing to System PG (Agent or VRU)	Always Default payload
Routing to Avaya Aura Symposium PG	Always Default payload
Routing to Aspect PG	Always Default payload

Condition	ECC Payload That Is Used
Contact Director to target Unified CCE	ECC payload that currently has scope in the script
Routing to INCRP NIC	ECC payload that currently has scope in the script
Pre-route to Gateway PG on Parent in Parent/Child	Always Default payload

Note   If you do not create another ECC payload, the solution uses the Default payload for everything.

ECC Payloads in Scripts

The solution uses the Default payload unless you override it.

When a call or task begins a routing script, it always starts with the Default payload. If the Default payload contains the ECC variables that the call or task requires, there is no need to change ECC payloads. The ECC Payload node sets which ECC payload currently has scope in the scripting environment. When you set an ECC payload to have scope, it retains scope until you change to another ECC payload. Any scripting node that sends messages externally uses the ECC payload that currently has scope.

However, you can specify an ECC payload in the configuration of a VRU. The VRU configuration includes a field for specifying an ECC payload to use with that VRU. If that field is set, the scripting environment always uses the specified ECC payload for that VRU. If you did not set the field for a particular VRU, the scripting environment always uses the Default payload.

Note   The same behavior occurs with the Type 2 VRU that is associated with the MR PG.

For example, take the following script. When the routing script begins, the Default payload has scope. Before sending a request to the Application Gateway, you use the ECC Payload node to change to the "SalesAgents_DB" payload. That ECC payload's members contain the data that the Application Gateway uses in a database lookup for details that the agent needs for this call. On returning, you use the ECC Payload node to give scope to the "SalesAgents" payload. This ECC payload contains the ECC variables that the Agent PG in the PG1_CCM1.0001.SG skill group needs. If the call queues at the VRU, the scope switches to the specified ECC payload for the VRU, if configured, or else to the Default payload.

The *Call.ECCPayloadID* variable contains the ID of the ECC payload which currently has scope. When the script follows the success path from the ECC Payload node, the node updates *Call.ECCPayloadID*. In case of any error while setting the ECC payload, this script takes the failure path from the ECC Payload node. The ECC payload that last had scope continues to have scope in the failure path case.
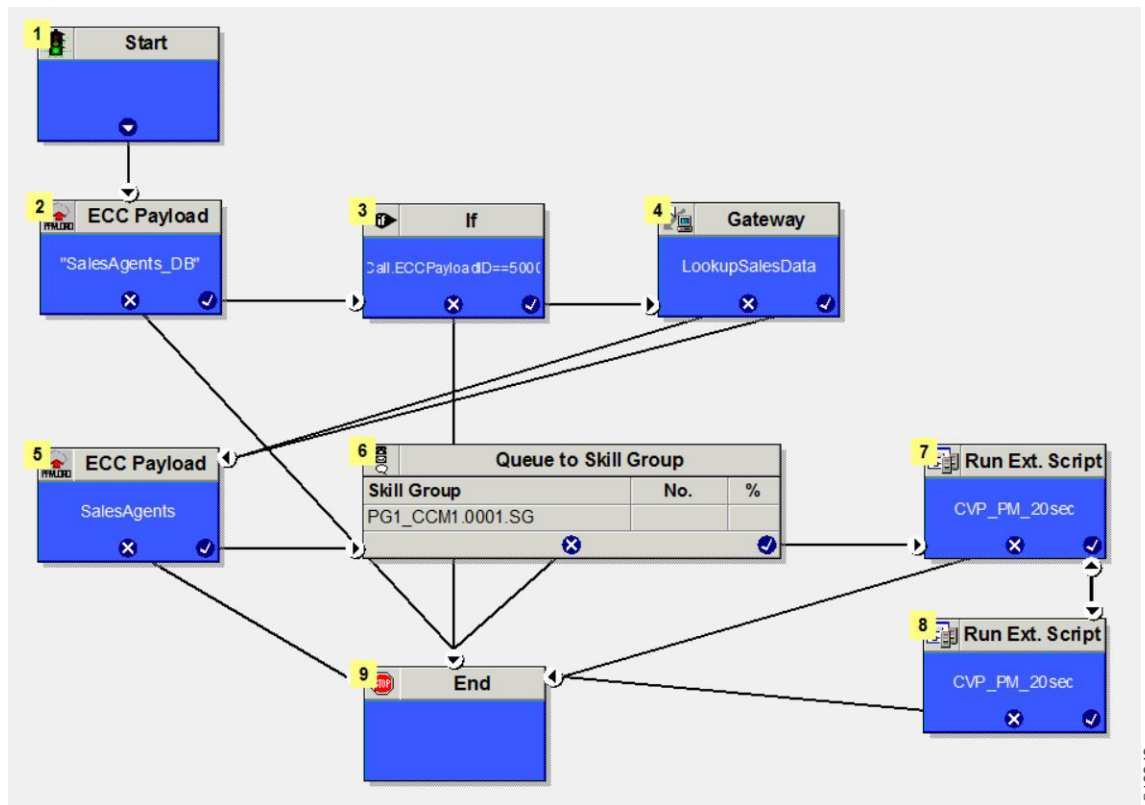
Transfers and Conferences with ECC Payloads

In releases before the introduction of ECC payloads, the solution merged the ECC variables from all the calls during a transfer or conference. Now, with ECC payloads, the retained data depends on the ECC payload that you use for each call.

Using the same ECC payload for all the transfer or conference is simplest. This method ensures that all members are available for reporting and on the desktop. When the merge occurs, the system keeps the value from the latest call that has a value for that member.

If your call flow swaps ECC payloads during a conference or transfer, only ECC variables from the final ECC payload appear in the reporting and on the desktop. Ensure that you include any member that you need for reporting or on the desktop in all the ECC payloads that the call flow uses.

ECC Payloads with Contact Director or ICM-to-ICM

The Contact Director and its targets (and ICM-to-ICM solutions) can use different payloads across their interface. When the Contact Director sends a message to the target Unified CCE, the script that runs on the Contact Director Router governs the ECC payload that is sent. When the target Unified CCE sends a message to the Contact Director, the script that runs on the target Unified CCE Router governs the ECC payload that is sent.

If you translation route the call to a VRU on the target Unified CCE, the initial connect message goes up to the Contact Director. The script that runs on the target Unified CCE Router governs the ECC payload that is sent to the Contact Director. By comparison, when the target Router exchanges a RunScript message with its own VRU, it uses the ECC payload that is configured for that VRU.

During initialization, the INCRP NIC maps all ECC variables that the Contact Director and target Unified CCE use between the two systems. The INCRP NIC then merges the ECC variables that are sent by each system into the destination environment by the target system.

ECC Payloads with Parent/Child

In general, Parent/Child solutions use the Default payload for all call flows. The Router always uses the Default payload for these operations:

- Local calls on a System PG.

- Calls queued on a Type 9 IP-IVR. (Ignores any ECC payload that is configured on the Network VRU.)

- Calls pre-routed on the Gateway PG on the Parent.

The CVP at the Parent uses the ECC payload that is configured for the corresponding Network VRU.

Persistent vs. Non-persistent Call Variables

When the Unified CCE/Unified ICM 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 (in the Configuration Manager), 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.

Expanded Call Context Variables for Web Callback

You must create several ECC variables if you intend to use Enterprise Chat and Email and/or Voice Media Routing Domains to route Delayed Callback requests. The ECC variables are:

- user.ece.activity.id - used to send the CIM activity ID to Unified ICM.

- user.ece.customer.name - used to pass the customer name from the entry point to Unified ICM.

⚠️
Warning   Do not overwrite reserved ECC variables in a script. Overwriting these variables may cause the application to route tasks to agents in an incorrect manner.

For more information about configuring ECC variables, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*.

Expanded Call Context Variables for Cx KPI

ECC variables for the CX KPI, should be configured in the **Expanded Call Variable List** tool.

For CX KPI, the ECC variable is:

- user.Cx_Survey_Info

> ☞
>
> **Important** Use the **Expanded Call Variable List** tool in the **Configuration Manager** to limit the *user.Cx_Survey_Info* variable to 50 bytes, by default the size is 40 characters.

> ✎
>
> **Note** If your using custom payloads, add the ECC variable in the **ECC Payload List** tool.

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 Configuration Manager. For more information, see *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*.

Each user variable must:

- Have a name that begins with user.

- Be associated with an object type, for example, service. (This enables the Unified ICM to maintain an instance of that variable for each object of that type in the system.)

- Be checked as persistent. A persistent variable maintains its value between script invocations. This allows you to set the variable in one script and later reference it in another script.

- 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 115: 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 the Unified ICM 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:

Case	SkillGroup.Avail	SkillGroup.ICMAvailable
Only voice domain is used	Number of agents in the Available state.	Same
Multiple Domains are used	Number of agents in the Available state, regardless of what they may be doing in this or other domains.	Number of agents who can actually handle an additional task or call in the domain.

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 Enterprise Chat and Email 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 a chat domain. This table shows the voice skill group states and the number of agents available in that state.

Case	SkillGroup.Avail	SkillGroup.ICMAvailable
Initial state	3	3
First agent handles a call	2	2
Second agent handles a chat session	2 (because there are two agents doing nothing in the domain)	1 (because there is only one agent left to handle voice calls)
Voice call ends	3	2
Chat ends	3	3

If a routing script needs to check the number of available agents, using SkillGroup.Avail produces effective results as it uses an extrapolation mechanism in determining the available agent.

Following is another example showing agents handling non-interruptible chat tasks. Assume three agents are logged in to a chat skill group, each allowed to handle two chats. This table shows states for the chat skill group.

Case	SkillGroup.Avail	SkillGroup.TalkingIn	SkillGroup.ICMAvailable
Initial state	3	0	3
First agent handles a chat session	2 (because the agent is now in the talking state)	1	3 (because all three agents can still handle additional chats)
Second agent handles a chat session	1	2	3
Third agent handles a chat session	0	3	3
First agent handles second chat session	0	3 (even though a total of 4 chats are in progress, only 3 agents are doing the work)	2 (because only the second and third agents can handle an additional chat)

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.

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.

Priority	Operator type	Operators
1	Prefix (unary)	+ - ! ~
2	Multiplication and division	* /
3	Addition and subtraction	+ -
4	Shift right and shift left	>> <<
5	Relational	< > <= >=
6	Equality	== !=
7	Bitwise And	&
8	Bitwise exclusive Or	^
9	Bitwise inclusive Or	|
10	And	&&
11	Or	||
12	Conditional	?
13	Sequential	,

Operators

Prefix Operators

The Prefix Operators in the following table take a single operand:

Operator	Meaning	Comments/Examples
+	Positive	Numeric values are positive by default, so the positive operator (+) is optional. Example: 2 and +2 represent the same value.
-	Negative	The negative operator (-) changes the sign of a value. Example: 2 represents a positive value; -2 represents a negative value.
!	Logical negation	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 > 0 is true and (SkillGroup.Sales.Avail > 0) is false.
~	One's complement	Operates on a bit value, changing each 1 bit to 0 and each 0 bit to 1. Note: This operator is rarely used.

Arithmetic Operators

The Arithmetic Operators in the following table take two operands:

Operator	Meaning	Comments/Examples
*	Multiplication	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.
/	Division	
+	Addition	
-	Subtraction	

Equality Operators

The Equality Operators in the following table take two operands:

Operator	Meaning	Comments/Examples
==	Equal to	Equality operators allow you to determine whether two values are equivalent or not.
!=	Not Equal To	

Relational Operators

The Relational Operators in the following table take two operands:

Operator	Meaning	Comments/Examples
>	Greater than	Relational operators allow you to perform a more sophisticated comparison than the equality operators.
<	Less than	
>=	Greater Than or Equal To	
<=	Less Than or Equal To	

Logical Operators

The Logical Operators in the following table take two operands. Logical operators examine the values of different logical expressions:

Operator	Meaning	Comments/Examples
&&	And	The expression is true if both of the operands are true. If either is false, the overall expression is false.
||	Or	The expression is true if either or both of the operands is true. If both are false, the overall expression is false.

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.

Operator	Meaning	Comments/Examples
&	And	The & Bitwise Operator turns specific bits in a value on or off.
|	Inclusive Or	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 "one or the other, but not both").
^	Exclusive Or	

Miscellaneous Operators

The following table lists miscellaneous operators:

Operator	Meaning	Comments/Examples
?	Conditional	The conditional operator (?) takes three operands and its syntax is as follows: The Unified ICM 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.
&	Concatenation	The concatenation operator (&) joins two strings end-to-end. returns the value.
,	Sequential	The sequential or comma operator (,) takes two operands, each of which is an expression. Unified ICM 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.
<< >>	Shift left Shift right	The shift left (<<) and shift right (>>) operators shift the bits within a value.

Built-in functions

Date and Time Functions

The following table lists the built-in date and time functions:

Function	Data Type	Return Value/Example
date [(date)]	Integer	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. For example, = date (2001, 7, 15) tests whether the current date is July 15, 2001. **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.
day [(date)]	Integer	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.
hour [(time)]	Integer	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.

Function	Data Type	Return Value/Example
minute [(time)]	Integer	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.
month [(date)]	Integer	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.
now	Float	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.
second [(time)]	Integer	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.
time [(time)]	Float	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. For example, = time (20:05:00) tests whether the current time is 08:05:00
weekday [(date)]	Integer	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.
year [(date)]	And	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.

Mathematical Functions

The following table lists the built-in mathematical functions:

Function	Data Type	Return Value/Example
abs(n)	Floating Point or Integer	Returns the absolute value of n (the number with no sign).
max(n1, n2 [,n3] . . .)	Floating Point or Integer	Returns the largest of the operands. Each operand must be numeric.
min(n1, n2 [,n3] . . .)	Integer	Returns the smallest of the operands. Each operand must be numeric.
mod(n1,n2)	Floating Point or Integer	Returns the integer remainder of n1 divided by n2.
random()	Floating Point or Integer	Returns a random value between 0 and 1.

Function	Data Type	Return Value/Example
sqrt(n)	Floating Point or Integer	Returns the square root of n. (The operand n must be numeric and non-negative).
trunc(n)	Floating Point or Integer	Returns the value of n truncated to an integer.

Miscellaneous Functions

The following table lists the built-in miscellaneous functions:

Function	Data Type	Return Value/Example
after(string1,string2)	String	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.
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.
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.
concatenate(string1,string2, . . .)	String	Returns the concatenation of the arguments. The function takes up to eight arguments.
EstimatedWaitTime	Integer	Returns the minimum estimated wait time for each of the queues against which the call is queued (skill group(s) or precision queue(s)). Queue to Agent(s) is not supported. If no data is available, returns -1. The estimated wait time is calculated once, when the call enters the queue. The default estimated wait time algorithm is based on a running five minute window of the rate of calls leaving the queue. Any calls which are routed or abandoned during the previous 5 minutes are taken into account as part of the rate leaving queue. For precision queues, the rate leaving queue represents the rate at which calls are delivered or abandoned from the entire precision queue, not any individual precision queue steps.
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.

Function	Data Type	Return Value/Example
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.
isPickPullRequest()	Logical	Whether the current service requested is for pick or pull type.
isPickPullRequest()	Logical	Whether the current service requested is for pick or pull type.
left(string,n)	String	Returns the left-most n characters of the string.
len(string)	Integer	Returns the number of characters in the string.
mid(string,start,length)	String	Returns a substring of the string, beginning with the specified start character and continuing for the specified number of characters.
result	Floating Point or Integer	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, the result function returns the number of seconds the selected agent has been available.
right(string,n)	String	Returns the right-most n characters of the string.
substr(string,start [, length])	String	Returns a substring of the string, beginning with the specified start character and continuing for the specified number of characters.
text(n)	String	Converts a numeric value into a string.
valid(variable)	Logical	Returns whether the variable has a valid value.

ValidValue(variable,value)	String	If the variable has a valid value, returns that value; otherwise, returns "value". Returns either a name from the database or the string value None.
		
		Note Use the following formula for the Enterprise Skillgroup that may not contain any Skillgroups:
		
		`ValidValue(EntSkill.`
		`Default\EnterpriseSkill`
		`groupPri.Loggedon,0)`

value(string)	Floating Point or Integer	Converts a string into a numeric value.

Custom Functions

Custom functions are those functions you create for use within scripts, as opposed to built-in functions.

Add 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 **Add** to open the Add Custom Function dialog box.

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.

Note   When you import the custom functions, the pattern #[0-9]+# is replaced with corresponding object name. The digits between "#" characters represent object ID. The # characters which are not the part of this pattern are not removed or replaced.

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.

Note	You cannot change the file type; you can save the script only in .ICMF format.

| 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. |

Dynamic Formula for PQ

You can pass the PQ name or ID dynamically while creating a formula in the IF node. Whenever a call encounters the IF node after you create a formula, the router evaluates the formula based on the PQ name or ID that is given in the Peripheral or ECC variable. This feature is implemented to check the real time statistics of the PQ before the call is queued to that PQ.

For example: Consider a formula `PQ.{Call.PeripheralVariable1}.LoggedOn > 0.` In this formula, `{Call.PeripheralVariable1}` is the variable that picks the PQ based on the PQ Name or ID dynamically. When a call comes in, the router evaluates the formula and determines the numbers of agents that are logged in the particular PQ that is given in the Peripheral or ECC variable . After the formula is evaluated, and if an agent is logged in, the call is routed to the agent.

Note Dynamic formula for PQ is only supported in the IF node.

Note Only Peripheral Variables and User defined ECC Variables are allowed within the curly brackets of a Dynamic Expression.

Note Evaluation of dynamic expression is done first by using name and then by ID of the PQ. If no PQ is found in system based on name specified in **Set** variable, then PQ is searched based on ID.

Ensure PQ names are unique and doesn't match with other PQ's ID configured in system. In case ID is used for dynamic expression and it matched with any other name, then wrong PQ is picked up during expression evaluation.

Procedure

Step 1	Place an **If object** in the workspace; right-click to open the Properties dialog box.
Step 2	Click **Formula Editor**.
Step 3	Select the **Variables** tab, and in the Object types list, select **PQ**. The variables that can be selected for the object are listed.
Step 4	In the Objects list, select **{}**. The variables that can be selected for the object are listed.
Step 5	From the **Variables** list, select the variable that you want to include in the formula.
Step 6	Click **Paste** to view the selected elements in the Formula box.
Step 7	To add a Peripheral or an ECC variable, place the cursor inside the curly brackets.

The options in the Object types and Variables lists change.

Step 8   In the Object types list, select **Call**

Step 9   In the Variables list, select the variable that you want to include in the formula.

Step 10   Optional: Add a relational operator to complete the expression

Step 11   Click **OK** to close the Formula Editor dialog box.

Step 12   Click **OK** to close the IF Properties dialog box.

Step 13   Click **Save**.

Dynamic Formula for Business Hours

You can pass the Business Hours name or ID dynamically while creating a formula in the **IF** node. Whenever a call encounters the **IF** node after you create a formula, the router evaluates the formula based on the Business Hours name or ID that is given in the Peripheral or ECC variable. This feature is implemented to check the real-time statistics of the Business Hours.

Note   Dynamic formula for Business Hours is only supported in the **IF** node.

Note   Only Peripheral and User defined ECC Variables are allowed within the curly brackets of a Dynamic Expression.

You cannot create multiple formulas using dynamic expressions.

For example:

```
BusinessHours.{Call.PeripheralVariable1}.BusinessHourStatus==1 &&
BusinessHours.{Call.PeripheralVariable1}.BusinessHourStatus==1
BusinessHours.{Call.PeripheralVariable1}.BusinessHourStatus==1 || BusinessHours.test
BusinessHours.BusinessHourStatus==0
```

.

You can pass the Business Hours name or ID dynamically while creating a formula in the **IF** node. Whenever a call encounters the **IF** node after you create a formula, the router evaluates the formula based on the Business Hours name/ID that is given in the Peripheral/ECC variable.

For example:

```
BusinessHours.{Call.PeripheralVariable1}.BusinessHourStatus==1.
```

In this formula, `{Call.PeripheralVariable1}` is the variable that dynamically picks the Business Hours based on the Business Hours Name or ID. When a call comes, the router evaluates the formula and determines the real-time status of that Business Hours that is given in the Peripheral or ECC variable. After the formula is evaluated, and if BusinessHours's status is 1, the call takes the success path of IF node.

Note Dynamic expression is first evaluated by using name and then by ID of the BusinessHours. If no BusinessHours is found in system based on name specified in **Set** variable, then BusinessHours is searched based on ID.

Ensure BusinessHours' names are unique and doesn't match with other BusinessHours' ID configured in system. In case ID is used for dynamic expression and it matched with any other BusinessHours' name, then wrong BusinessHours is picked up during expression evaluation.

Procedure

Step 1	Place an **If object** in the workspace; right-click to open the Properties dialog box.
Step 2	Click **Formula Editor**.
Step 3	Select the **Variables** tab, and in the Object types list, select **Business Hours**. The variables that can be selected for the object are listed.
Step 4	In the Objects list, select **{}**. The variables that can be selected for the object are listed.
Step 5	From the **Variables** list, select the variable that you want to include in the formula.
Step 6	Click **Paste** to view the selected elements in the Formula box.
Step 7	To add a Peripheral or an ECC variable, place the cursor inside the curly brackets. The options in the Object types and Variables lists change.
Step 8	In the Object types list, select **Call**
Step 9	In the Variables list, select the variable that you want to include in the formula.
Step 10	Optional: Add a relational operator to complete the expression
Step 11	Click **OK** to close the Formula Editor dialog box.
Step 12	Click **OK** to close the IF Properties dialog box.
Step 13	Click **Save**.

Script Administration

Check Script Routes

After you save a script, you should check that all routes referenced have valid labels for the routing clients and dialed numbers for which you have scheduled the script.

Note You must have saved the script to the Unified ICM database before you can check the routes.

For the Unified ICM to route calls through a script, you must have defined a routing label for each route referenced in the script. Each label is valid only for specific routing clients and, optionally, for specific dialed numbers.

To check script routes:

Procedure

Step 1 Start Check Routes from the Administration & Data Server group. The Check Routes window opens.

Step 2 Using the drop-down lists, choose the following:

• Name of the routing client to send calls through the script.

• Dialed number for the call to be sent through the script.

- The routing script name.

- The version number of the routing script. (The default is the active version.)

Step 3   To validate the route of a network transfer call target, check Use Network Transfer specify a routing client and a Dialed Number.

Step 4   Use the drop-down list to select Routes Used Directly by Script (the default) or Translation Route Used in Script. The routes referenced in the script appear in the left column. (If any of these routes do not have an associated label that is valid for the routing client and Dialed Number you have chosen, an error message appears in the Errors field.)

Note   To see the configuration details for a translation route, select the route name and click **View Translation Route**. (This button becomes enabled after you choose a specific translation route.) The Translation Route dialog box opens.

Step 5   To see the specific peripheral targets associated with a route, select the route name in the left column. The associated peripheral targets appear in the center column.

The routes referenced in the script appear in the left column. If any of these routes does not have an associated label that is valid for the routing client and Dialed Number you have chosen, an error message appears in the Errors field at the bottom of the window.

Step 6   To see the specific labels associated with any of these peripheral targets, select the peripheral target. The associated peripheral targets appear in the right column.

Symbols might appear next to a label, explaining the following:

- The label is not valid for the specified routing client.

- The label is not valid for the specified Dialed Number.

Step 7   To check configuration information for a route or peripheral target, double-click a route or peripheral target name. Check Routes to display the configuration information for that route or peripheral target.

Note   From the Peripheral Target dialog box, you can access information about the route by clicking the **Route** button.

Step 8   To correct any problems you find through Check Routes, make and save changes within the Script Editor or the Unified ICM Configuration Manager.

Step 9   To update the changes, click **Reload** in the Check Routes window. (If you have created a new version of the script, be sure to update the Version field.) Check Routes reads the latest version of scripts and configuration data from the local database.

Active Scripts

Although the Script Editor may contain several versions of a script, only one version of a particular script can be active at one time: This is the version that Unified ICM runs if the script is currently scheduled.

You can use one of three methods to set the active version of a script:

- Preferences

- Make Active Version command

• Script Explorer

Use Preferences to Set an Active Script

Procedure

Step 1   Within Script Editor, select **Options** > **Preferences**. The Script Editor Preferences dialog box opens.

Step 2   Optionally, select the **Automatically Make a Script Active When Saved** option.

> • When you select this option, whenever you save a valid script, the Unified ICM makes that new version the active version.

> • When you clear this check box, you must manually activate the script after saving it.

Use Script Make Active Version Command

Procedure

In an open script in edit mode, select **Script** > **Make Active Version** or click **Make Active Version**. The system makes the open script version the current active version.

View Enabled Scripts

After you save and schedule a script, it is considered enabled. Use the Enabled Scripts dialog box to examine all scripts currently enabled in the system.

To view all enabled routing and administrative scripts:

Procedure

Step 1   Within Script Editor, choose **Scripts** > **Enabled Scripts**. The Enabled Scripts dialog box opens listing all routing scripts that are scheduled for the current date and time.

The dialog box lists all call types and the script that is currently scheduled for each. The Call Count column lists the number of calls the script has processed.

Step 2   Optionally, use the Display Count From section of the dialog box to choose how to display the number of calls processed. Whichever option you choose, the counts are updated every 15 seconds. This allows you to see which scripts are currently handling calls.

| Note | The Call Count values are associated with scripts, not necessarily with call types. If the script has been scheduled for more than one call type, the Call Count value includes all calls processed by the script regardless of call type. |

Step 3 To see all administrative scripts scheduled for the current date and time, click the **Administrative Scripts** tab. This lists all administrative scripts.

Schedule Administrative Script

To schedule an administrative script:

Procedure

Step 1 Within Script Editor, choose **Script** > **Administrative Manager**. The Administrative Manager dialog box opens, listing any administrative scripts that are currently scheduled.

Step 2 Click **Add**. The Add Administrative Schedule dialog box appears, opening at the Script tab.

Step 3 Select the script you want to schedule and click the **Period** tab.

Step 4 Specify when you want the script to be active:

   a) In the Date Range, Recurrence Pattern, and Duration sections, specify the range of times when this script may run.

   b) In the Frequency section, specify how often the script should run during the specified time range.

Step 5 Optionally, click the **Description** tab and add a descriptive text about this schedule.

Step 6 Click **OK** to submit the schedule and return to the Administration Manager.

Step 7 Click **OK** to save your changes to the Unified ICM database and close the Administrative Manager.

| Note | • The first execution occurs at the start time of the schedule.

• The last execution occurs before or at the end time of the schedule.

• The timing of script execution might not be exact. Typically, scripts run within a few seconds after the scheduled time. The last script execution might occur slightly after the scheduled end time. |

Script Monitoring

After you save a routing or administrative script, you can observe how it runs. Watching routing requests moving through a script in real-time helps ensure that the routing script is operating as expected.

When you monitor a script, that is, view the script in Monitor mode, labels appear on each connection in the script.

Note A script can have a maximum of 900 monitored labels. If a script exceeds that limit some of the real-time monitors stop working. In this case, you see periodic messages in the Router log and Event report that the script exceeds the monitor limit. If you edit a script that is over the limit, a warning displays when you attempt to save the script.

Monitor Labels

Most monitor labels display the raw number and percentage of route requests that have passed through the connection since the start of the monitoring period.

Each target set also lists the number and percentage of calls routed to each of the targets in that set. Each statistic is updated automatically as new real-time data become available about every 15 seconds.

When you edit a script, position nodes so that there is enough space for the monitor labels to display. (Because you cannot make any changes to a script while in Monitor mode, you cannot rearrange the nodes at that time, unless you have enabled Quick Edit from Monitor mode as described in the following section). Use the **Script > Display Monitor Labels** command while in edit mode to display blank monitor labels on each connection of the script.

Enable Quick Edit from Monitor Mode

To enable Quick Edit from Monitor mode:

Procedure

	Command or Action	**Purpose**
Step 1	From the Options menu, choose **Preferences**.	
Step 2	Check **Allow quick edit from monitor mode**.	
Step 3	Click **OK**.	

What to do next

You can now perform quick edits when monitoring a script.

List of Monitor Labels

The following nodes have monitor labels that appear beside them while the scripts are monitored.

- Wait

- Queue

- Precision Queue

- Translation Route to VRU

- DB Lookup

- Gateway

- ICM Gateway

- Queue to Agent

- Collect Data (displays only the Abort meter).

- Menu (displays only the Abort meter).

- Play (displays only the Abort meter).

- Run External Script (Displays In Progress and Abort meter).

The monitor labels can display two or more of the following meters:

- InQueue meter - Displays the number of calls in thequeue.

- InProgress meter - Displays the number of calls in progress in the node.

- Abort meter - Displays the number of calls aborted.

- Routed meter - Displays the number of calls that are routed to the Precision Queue.

The InProgress meter appears in the top row of all the monitor labels for all the listed nodes, except the Queue and Precision Queue nodes. The top row of the Queue node monitor label displays the InQueue meter and the top row of the Precision Queue monitor label displays the Routed meter.

The following meters are applicable only for Queue to Skill Group and Precision Queue node

- Picked by another Skill Group/Precision Queue meter - Displays the number of non-voice task requests that are picked from a Skill Groups/Precision Queue other than the targets specified in the node.

- Pulled by another Skill Group/Precision Queue meter - Displays the number of non-voice task requests that are pulled from a Skill Groups/Precision Queue other than the targets specified in the node.

The following meters are applicable only for the Pick/Pull node.

- Pick meter - Displays the number of non-voice task pick requests that were successfull.

- Pull meter - Displays the number of non-voice task pull requests that were successfull.

- Pick error meter - Displays the number of non-voice task pick requests that failed.

- Pull error meter - Displays the number of non-voice task pull requests that failed.

Note Since **Picked by another Skill Group/Precision Queue and Pulled by another Skill Group/Precision Queue** meters are specific to non-voice task routing, these are not populated in case of voice call routing.

Adjust Monitor Label Location

To adjust the Monitor Label Location:

Procedure

Step 1   In the Properties dialog box of a node within Script Editor, click the Labels tab.

Step 2   Specify the following:

a) Label position (Slider): Choose if you want the connection labels to appear close to the node (Origin), close to the targeted node (Destination), or half way in-between (Center).

b) Display monitor labels (check box): If you select this check box, then when the Script Editor is in Monitor mode, labels display for each connection from the node. If you clear the check box, no labels display for connections from the node.

Step 3   Click **OK** to apply changes and to close the **Select Properties** dialog box.

Access Monitor Mode

Within Script Editor, do one of the following:

• Choose **File** > **Open** to access the Open dialog box and choose the Monitor option from the **Open Mode** drop-down list.

• Within an open script in the Script Editor workspace, choose **Script > Monitor** Script or click **Monitor Script** in the toolbar.

• Within the Enabled Scripts dialog box, choose the **Monitor radio button** in the Open Script in Mode section and click **Open**.

What to do next

The script displays in Monitor mode in the Script Editor workspace.

Set Monitor Mode Options

To set Monitor mode options:

Procedure

Step 1   In Script Editor, do one of the following:

• Choose **Script > Monitor Options**.

• Click **Monitor** on the status bar. A drop-down list appears.

Step 2   Choose one of the following options:

• Start of Day. Initial monitor values are totals since midnight. New values are added to these totals every 15 seconds.

• Starting Now. Initial monitor values are all zeros. New values are added to these totals every 15 seconds.

- For Each Interval. Initial monitor values are the values reported for the most recent 15-second interval. New values are added to these numbers every 15 seconds.

- Percentages for Each Node. The percentages for each connection from a node are calculated by dividing the number of executions that passed though the connection by the number of executions that entered the node. All connections coming from each node add up to 100%.

- Percentages for Entire Script. The percentages for each connection are calculated by dividing the number of executions that passed through that connection by the total number of executions handled by the script.

View Real-Time Data

In viewing the number of contacts that pass through each branch of the script, you can also view real-time data about activity at the contact centers, including statistics for each skill group, service, call type, or scheduled target referenced in the script.

For more information about the real-time data, see the descriptions of the Service Real Time and Skill Group Real Time tables in the *Database Schema Handbook for Cisco Unified Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-technical-reference-list.html.

To view real-time service, skill group, or precision queue data:

Procedure

Step 1 While in Monitor mode in Script Editor, choose **Script > Display Real-Time Data**.

The Real-Time Data window appears. By default, the Real-Time Data window displays data about the Services referenced in the current script. (Scroll to the right to see additional columns.) The values in the screen are updated continuously as new real-time data arrives at the Administration & Data Server.

Note When an External Authorization server is used with Internet Script Editor, you can only select from the list of available Call Types or Precision Queues as identified by the Authorization server.

Step 2 To display information about skill groups or scheduled targets, rather than services, change the value in the field in the upper-left corner of the Real-Time Data window. The window contents automatically change to show skill group data.

Step 3 When finished, click **Close**.

View Dynamic Real-Time Data

You can view dynamic call types, dynamic precision queues, or dynamic skill groups.

Procedure

Step 1 While in Monitor Mode in Script Editor, open an active script and choose **Script** > **Display Real-Time Data**. The **Real-Time Data** window appears.

Step 2   To display dynamic call types, dynamic precision queues, or dynamic skill groups from the drop-down list, choose **Call Type**, **Precision Queue**, or **Skill Group** in the upper-left corner.

Step 3   Click **Add Dynamic**.
The **Select Dynamic Call Type Objects**, **Select Dynamic Precision Queue Objects**, or **Select Dynamic Skill Group Objects** dialog box appears, depending on your list choice.

Step 4   In the **Available dynamic objects** list, select the objects to monitor and then click **Add**.
You can monitor up to 20 dynamic call types, up to 20 dynamic precision queues, or up to 20 skill groups. You cannot monitor dynamic call types, dynamic precision queues, and dynamic skill groups in the same Real-Time Data window.

Step 5   Click **OK**.
The selected dynamic objects appear in the **Real-Time Data** window. Data for the selected objects appear when the data next refreshes.

Change Real-Time Data Configuration

To change the real-time data configuration:

Procedure

Step 1   Click **Configure** in the real-time data window or choose **Script > Configure Real-Time Data**. The Configure real-time data dialog box opens.Use this dialog box to add new columns, remove existing columns, reorder the columns, or change the column labels.

Step 2   Choose a Routing target type (Service, Scheduled Target, or Skill groups). This updates the other fields:

- The Real-Time Columns list contains all the real-time columns available in the database for the target type.

- The Routing Target Columns list contains the columns to appear in the Real-Time Data window.

- If you select a column in the Routing Target Columns list, the heading for that column appears in the Column Header field.

Step 3   Use the Add and Remove buttons to move columns between the lists. Use the Move buttons to change the order of the selected columns. (To change back to the default column order, click **Default Columns**.)

Step 4   Optionally, to edit a Routing Target Columns heading, make changes within the Column Header field. (To change back to the default header, highlight the Routing Target Column name and click the Default Header button.)

Step 5   Optionally, select the **Short Headers**/ **Long Headers** radio buttons to change between the full and abbreviated forms of the default column headers. The abbreviated forms are typically three to four letters. (For example, AHT is used for Average Handle Time.)

Step 6   To save the changes, click **OK**. The settings apply to the current and future Script Editor sessions.

View Router Logs

You can view CallRouter log messages to determine how contacts are routed and to see any errors the Unified ICM encounters in processing routing requests.

You start the Router Log Viewer from the Administration & Data Server group. The Router Log Viewer window opens.

The top field of the Router Log Viewer window displays information about each call the Unified ICM routes, including:

- The time that routing request was received

- The Dialed Number (DN), or script selector, and the caller's billing telephone number (ANI)

- Any Caller-Entered Digits (CED)

- The label that Unified ICM returned to the routing client

The bottom field of the window displays any errors that the Unified ICM encounters in routing calls, including:

- The time the error occurred

- Text describing the error

Export a Script

To export a script:

Procedure

Step 1 Choose **File > Export Script**. The Export Script dialog box opens with the name of the current script and version number in the File name field.

Step 2 Optionally, change the file name. You cannot change the file type; you can only save the script in .ICMS format.

Step 3 Click **Save**. If the file name already exists, the system prompts you to confirm the save.

Step 4 If prompted, click **OK**.

Note ICM Scripts, which are created on newer versions of the ICM Script editor, will be unable to import the newer scripts into the older ICM versions.

Import a Script

To import a script:

Procedure

Step 1   Choose **File > Import Script**. The Select Script to Import dialog box opens.

Step 2   Choose a file name with an Unified ICM Script extension (.ICMS) and click **Open**. 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 partially mapped script is opened as a new script in Import mode and 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 the imported object.

>> • Imported Object. The name of the imported object.

>> • Mapped To. What this imported object will be mapped to.

Step 3   Click an Imported Object value. The Mapped To column's drop-down list shows all the valid objects on the target system. (The Script Editor window also highlights the script nodes that refer to this object.)

Step 4   Optionally, choose an object from the Mapped To drop-down list on the target system that you want to map the imported object to.

> **Note**   Choose an object from the Mapped To drop-down list on the target system that you want to map the imported object to.

Step 5   When the mapping is complete, click **Apply** and Finish.

✎

Note   ICM Scripts, which are created on newer versions of the ICM Script editor, will be unable to import the newer scripts into the older ICM versions.

Modify Script Version and Schedule System Information

To modify script version and schedule system information:

Procedure

Step 1   Within the Unified ICM Configuration Manager, choose **Tools > Miscellaneous Tools > System Information**. The System Information window opens.

Step 2   Enter the Minimum Script Schedule Time.

Step 3   Set the number of script versions to be retained in the Retain Script Versions field.

Step 4   Click **Save** to apply your changes.

CHAPTER 9

Script Editor Feature Control

Script Feature Control Methods

Administrators can use Script Editor Feature Control to restrict users, or classes of users, from some or all of the functionality of the Unified ICM Script Editor software. In a possible deployment scenario, a Unified ICM administrator can restrict certain people from doing specific types of script editing.

An administrator has two means to restrict access to the editing features of Script Editor and Internet Script Editor:

- Edit Options

- Script Node Control

Administrators control script editing by creating feature sets that they assign to users. The feature set controls which script nodes are accessible to the user, and which edit modes are available to the user.

✏️

Note   When Unified ICM runs on a partitioned system, users need at least reference access to objects to edit scripts that contain references to those objects. For example, a user needs at least reference access to skill groups to edit a script in which those skill groups are included in the Skill Group node.

For more information about Feature Control, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html.

Edit Options

The administrator can assign one of two editing options:

- Full Edit (Includes Quick Edit)

- Quick Edit Only

You can only access the Full Edit mode or the Quick Edit mode from the Monitor or the Browse modes. The Full Edit and the Quick Edit modes cannot be accessed from each other.

Full Edit (Includes Quick Edit)

Full Edit mode allows you to use Full Edit mode or Quick Edit mode when working with scripts. Both Full Edit and Quick Edit are enabled on the Script Editor toolbar. When changing from Monitor or Browse mode to Full Edit mode, the Script Editor workspace background goes from gray to white.

Your Full Edit mode editing capabilities are determined by whether or not you are assigned to a feature control set:

- If you are a full-edit user and are not assigned to a feature control set, you can add, edit, or delete any script or node.

- If you are a full-edit user and are assigned to a feature control set, you can add, edit, or delete any script. You only have Full Edit permissions for those scripts that do not contain nodes that are marked as unavailable to you in a feature set.

Full Edit gives you restricted editing capabilities if you are assigned a feature control set. You can still edit any script, however, your ability to edit specific script nodes is determined by which nodes are selected in the Node Control table (located on the Script Editor Feature Control dialog) of the assigned feature control set. This allows you to edit the structure of a script or to create, edit, and delete scripts containing the selected nodes.

If a script is opened that contains a disabled node, you can browse or monitor the script but you cannot put the script into edit mode. If you attempt to put this script into edit mode a message indicating you are not authorized to enter edit mode appears. However, you can still Quick Edit the script, just not the node.

You might want to use Quick Edit mode so as not to accidentally change the structure of a script.

Quick Edit Only

Quick Edit mode allows you (the feature-control-set member) to work only with scripts in Quick Edit mode. Edit is disabled and Quick Edit is enabled on the Script Editor toolbar. When changing from Monitor or Browse mode to Quick Edit mode, the Script Editor workspace background goes from gray to yellow.

In Quick Edit mode:

- You cannot add or delete a node.

- You can adjust most of the properties of the script nodes selected 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 on the node and selecting Properties, or by double-clicking on 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).

- Script > Make Active Version is disabled.

- Script > Make Active Version is disabled.

- You cannot import scripts.

- You cannot use the File > Script Locks tool.

Access Quick Edit Mode

You can access Quick Edit by performing any of the following steps:

Procedure

- Choose **Script** > **Quick Edit**.
- Click **Quick Edit** on the Script Editor toolbar.
- Right-click in the Script Editor workspace and choose **Quick Edit**.
- From within the script in Monitor mode, double-click the node you want to edit. (This is only available if Options > Preferences > Allow for Quick Edit from Monitor Mode is checked.)

What to do next

When in Quick Edit mode, the disabled nodes are removed from the object palette.

Script Node Control

Script Editor Feature Control allows an administrator to create feature sets that can be assigned to users. Administrators use a Node Control Table in the feature set to control which script nodes are accessible to the user.

Node Control Table

The Node Control table (on the Unified ICM Configuration Manager Script Editor Feature Control dialog box) has two columns, the Node column and the Available column. This table allows an administrator to create feature control sets that can be assigned to users. The feature control set controls which script nodes are accessible to the user.

If a script is opened that contains a disabled node, you can browse or monitor the script but you cannot put the script into edit mode. If you attempt to put this script into edit mode a message indicating you are not authorized to enter edit mode appears. However, you can still Quick Edit the script, just not the node.

Node Column

A node is an executable element within a script. A script consists of nodes, connections, routing targets, and comments. Every script begins with a Start node. This column lists of all the nodes that you can use in a script.

✎

Note   The Line Connector node is always available.

Available Column

Each checked node in this column appears on the editing palette of the feature-control-set user, regardless of the edit mode (Full Edit or Quick Edit Only).

Disabled nodes are removed from the object palette.

Create a Feature Control Set

The system administrator can create a feature control set using the Unified ICM Configuration Manager on Administration & Data Server:

Procedure

Step 1	Ensure the users the feature set is to be assigned to are configured.
Step 2	Start the Unified ICM Configuration Manager.
Step 3	Choose **Tools > List Tools > Feature Control Set List**.
Step 4	In the Feature Control Set section (on the left), click Add.
Step 5	Enter the name of the feature control set. The name appears in the left section when Enter or Tab is pressed.
Step 6	Optionally, enter a description.

Assign Users to a Feature Control Set

In the Unified ICM Configuration Manager, use the User List tool to select a feature control set to associate with the user.

Select Script Nodes and Edit Options for a Feature Control Set

Procedure

Step 1	In the Feature Control Set List dialog box, select the name of the feature control set to be assigned.

Step 2   Select Advanced (under Script Editor).

Step 3   In the Script Editor Feature Control dialog box, select the nodes for this feature control set and an edit option (Full Edit or Quick Edit).

Step 4   Click **OK**.

Step 5   Click **Save**.

CHAPTER **10**

Internet Script Editor

ISE Application

You can use either or both of the Internet Script Editor and the Script Editor to work with routing and administration scripts.

The Internet Script Editor provides the same functionality as the Unified ICM Script Editor software, without the need for a full Administration & Data Server.

Note   When the Unified ICM runs on a partitioned system, you cannot edit security information for a script with Internet Script Editor. Use Script Editor instead.

ISE Functionality

This section describes how Internet Script Editor works on and communicates with Administration & Data Server.

Internet Script Editor works through the IIS Web server on the Unified ICM Distributor. It uses HTTP or HTTPS to communicate with Administration & Data Server.

The Internet Script Editor and Unified ICM Script Editor GUIs are essentially the same. The menus, toolbars, palette, and work space are utilized in the same manner in both applications. The differences between the two occur primarily in the method by which each application communicates with Unified ICM.

ISE Requirements

This section describes Internet Script Editor requirements.

Internet Script Editor is supported on the operating systems listed in the Contact Center Enterprise Compatibility Matrix.

Note   For ISE to work properly, select the **Enable HTTP Keepalive** box on the website tab of Internet Information Services Manager/Default Web Site Properties.

Note   If you use Unified Contact Center Management Portal (Unified CCMP) or Unified Contact Center Domain Manager (Unified CCDM), you cannot use Transport Layer Security (TLS) v1.0 and v1.1 to connect with the Internet Script Editor.

TLS Requirements for ISE

Internet Script Editor connections use Transport Layer Security (TLS). TLS connections encrypt client requests and server responses.

Unified CCE has a utility (SSLUtil.exe) that provides the ability for Unified ICM Setup to create and install a self-signed server certificate. The certificate is generated, imported to the Local Machine Store, and installed on the web server.

A digital certificate is an attachment to an electronic message used for security purposes. A digital certificate verifies that a sender is who they claim to be and provides the receiver with the means to encode a reply. Your browser does not automatically recognize a self-signed certificate. A self-signed certificate does not provide any guarantee concerning the identity of the organization that is providing the website. Most browsers have a list of trusted CAs (Certification Authorities) whose certificates they automatically accept. If a browser encounters a certificate whose authorizing CA is not in the list, the browser asks you whether to accept or decline the connection.

Install the standalone encryption utility on the AW Real-time Distributor (in the AW Program Group). This enables you to change the default encryption settings (implemented by Setup). This utility contains the functionality to regenerate the self-signed certificate and replace the IIS installed certificate as needed.
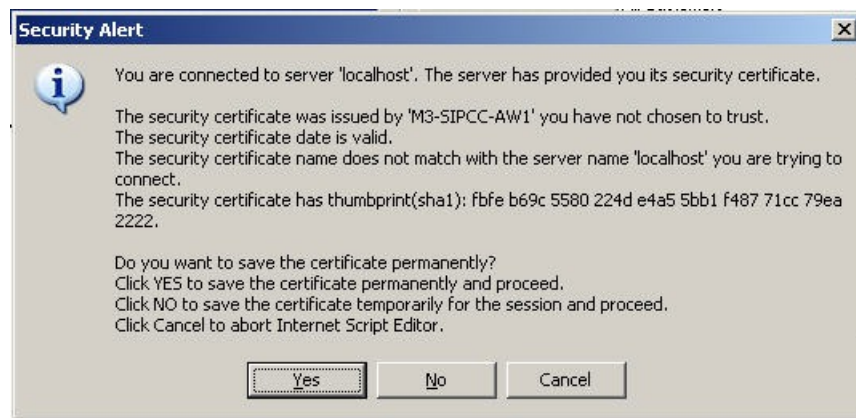
ISE Client

Unified ICM setup configures a secure connection (using port 443) by default and sets up the certificate on the ISE server (the Distributor Administration & Data Server). For new ISE client installations, configure the ISE client to use the secure connection (port 443) to connect to the server. If you configure the ISE client to use HTTP (port 80) to connect to the server and the connection fails, the ISE client tries the secure connection (port 443).

When the ISE client connects to the server. one of the following occurs.

- If the encrypted flag is not set:

- The client starts with setting up an HTTP connection.

  - If the client successfully connects to the server, it sends the user account and password in plain text. It then establishes the session with the server through HTTP .

  - If the client cannot connect to the server, it fails over and tries to connect to the server through HTTPS. If the HTTPS connection is set up successfully, the client sends the encrypted user account and password. It also sets the encrypted flag in the registry so that the next time it will use HTTPS.

- After the HTTPS connection is established, the server sends the certificate to the client. The client presents the certificate prompt, unless it has been previously saved locally.

Figure 116: Security Alert Dialog Box

- The client starts with setting up an HTTPS connection.

  - If the client successfully connects to the server, the client sends the encrypted user account and password.

  - After the HTTPS connection is established, the server sends the certificate to the client. The client presents the certificate prompt (see the Security Alert Dialog Box), unless it has been previously saved locally).

  - If the client cannot connect to the server, it prompts you to determine if it should failover to try to connect to the server via HTTP, or not. If you select **Yes**, the client sends in the user account and password in plain text once the HTTP connection is established. However, it does not set the encrypted flag in the registry so that the HTTPS connection will still be initiated the next time.

  - If you want to use the HTTP connection for all future connections, you must manually unset the flag in the login screen.

During runtime, the initialization is the same for every HTTPS request.

During upgrades or new installations on Windows Server, the secure connection is automatically configured to the default setting of 443 for SSL.

> ✎
>
> **Note**   The ISE client can revert back to unencrypted communication over port 80 if it fails to establish an HTTPS session.

Departmental Hosting

External Authorization Server for Internet Script Editor

The Internet Script Editor includes support for Departmental Hosting, which separates scripting authorization by user, group, or role. A script authorization server determines which configuration objects are valid for a user in the Internet Script Editor. This feature is supported when the deployment type is set (through CCE Administration) to one of the following:

- UCCE: 2000 Agents

- UCCE: 4000 Agents

- UCCE: 12000 Agents

- UCCE: 8000 Agents Router/Logger (for Non-Reference Designs only)

This feature is enabled by configuring a CCMP Authorization Server in Web Setup on the Admin Workstation.

The objects that can be authorized are:

- Call Type

- Dialed Number

- Label

- Precision Queue

- Network VRU Script

- Skill Group

An error appears if you open a script that contains data for which you are not authorized. An authorized user may need to change the user authorization configuration or the script to allow access.

Use the Feature Control Set to enforce Quick Edit and limit node access for the lower-level users. This limits which nodes a lower-level user can modify.

When you enable this feature by using Web Setup, the user cannot dynamically select targets on Precision Queue, Call Type, and Route Select nodes.

Access to Labels or Dialed Numbers by User

This scripting authorization restricts a label or a dialed number that is authorized for a particular department only to that department's users in the Internet Script Editor.

You only see list of labels for which you have authorization in the Label nodes and Dynamic label nodes.

ISE Installation and Upgrades

Install Internet Script Editor

You cannot install Internet Script Editor directly on a VM.

Procedure

Step 1	Point your browser to *server-name*`/install/iscripteditor.htm`, where *server-name* is the name of the computer on which you installed the distributor with the Internet Script Editor client package.
Step 2	Click **Download Internet Script Editor**.
	Note You can also open the `iscripteditor.exe` file directly from the web page.
Step 3	Navigate to the directory where you want to save `iscripteditor.exe`.
Step 4	Click **Save** to begin the download.
Step 5	After the download is complete, close the browser.
Step 6	On your desktop, navigate to `iscripteditor.exe` and run the file.
Step 7	When the InstallShield Wizard for Internet Script Editor starts, click **Next** to continue.
Step 8	Select the default Destination Folder by clicking **Next**; or click **Browse** to navigate to the desired Destination Folder, and then click **Next**.
Step 9	After the InstallShield Wizard indicates that the installation is complete, click **Finish**.

A shortcut for Internet Script Editor (IScriptEditor) appears on the desktop, and in the Start menu in the `Programs/Cisco Systems Inc.` program group.

Start Internet Script Editor

Procedure

Step 1	Double-click the desktop shortcut for Internet Script Editor (IScriptEditor).
Step 2	Click **Connection**.
Step 3	Enter the correct **Address**, **Port**, and **ICM Instance** information.
Step 4	Click **OK**.
Step 5	Enter your **User Name** and **Password**. Be sure to use a Security Account Manager (SAM) username, as the name must not exceed 20 characters in length.
Step 6	Enter the **Domain** of Unified ICM system.
Step 7	Click **OK**.
Step 8	Upgrade Internet Script Editor as necessary.

> **Note**　You require full access to `icm\<inst>\ra\dbagent.acl` on the Router to use Internet Script Editor. (By default, Setup creates the file and gives full read/write access to this file to every user signed into the system.) If the access attributes of this file are not full read/write access, you cannot start Internet Script Editor. In such cases, the following error appears in iseman log: "GetLock: lock denied/insufficient permission." The error message "Unable to access dbagent.acl during security check" appears in the dbagent log.

Upgrade Internet Script Editor

After you start Internet Script Editor, if there is a newer version, you receive a message informing you that you can upgrade Internet Script Editor.

> **Note**　In this release, the Internet Script Editor server only supports TLS 1.2 for communication with an Internet Script Editor client. Internet Script Editor client versions before Release 11.6(1) cannot properly establish a TLS 1.2 connection with the server. This prevents an automatic upgrade of the Internet Script Editor client to the current release.
>
> You can manually upgrade the ISE Client installer by entering the following URL in your browser:
>
> `https://<DistributorHost/addr>/install/upgradescripteditor.htm`
>
> This URL reaches the upgrade web page for the Internet Script Editor client. You can then upgrade the Internet Script Editor client.

> **Note**　Some upgrades are optional; these upgrades typically contain GUI enhancements. Other upgrades, typically involving protocol or database changes, are mandatory. You cannot use Internet Script Editor until you accept mandatory upgrades.

Procedure

Step 1　Accept a software upgrade.

A web page opens from which you can download the new Internet Script Editor.

Step 2　Click **Download Internet Script Editor**.

> **Note**　You cannot use Internet Script Editor during the upgrade.
>
> You can also open the `iscripteditor.exe` file directly from the web page.

Step 3　Navigate to the directory where you want to save `iscripteditor.exe.`

Step 4　Click **Save** to begin the download.

Step 5　After the download is complete, close the browser.

Step 6　On your desktop, navigate to `iscripteditor.exe` and run the file.

Step 7　When the InstallShield Wizard for Internet Script Editor starts, click **Next** to continue.

Step 8   Select the default Destination Folder by clicking **Next**; or click **Browse** to navigate to the desired Destination Folder, and then click **Next**.

Step 9   After the InstallShield Wizard indicates that the installation is complete, click **Finish**.

Troubleshooting Tools for Internet Script Editor

This section describes the tools that you can use to troubleshoot the Internet Script Editor

Client-Side Troubleshooting Tools for Internet Script Editor

The following table describes the client-side troubleshooting tools for Internet Script Editor:

Troubleshooting Method	Description
EMS trace files	Internet Script Editor writes to EMS logs and deletes old logs on startup, similar to the Unified ICM Script Editor.
Dr. Watson	Internet Script Editor is built without symbol tables to keep it small. This makes Dr. Watson output more difficult to debug.

Server-Side Troubleshooting Tools for Internet Script Editor

The following table describes the server-side troubleshooting tools for Internet Script Editor:

| Troubleshooting Method	Description
IIS Logs	IIS logs its activity to the system event log or to an ODBC data source.
EMS trace files	ISAPI DLL generates trace output on the distributor. System administrators can use the Dumplog utility to display the contents of the logs.

Scripting Specifics in CCE Environment

Unified CCE Gateway

The Unified CCE Gateway PG allows a site Unified CCE system to connect to a Unified ICM. The Unified ICM parent views the system as an ACD.

Figure 117: Unified CCE Gateway PG
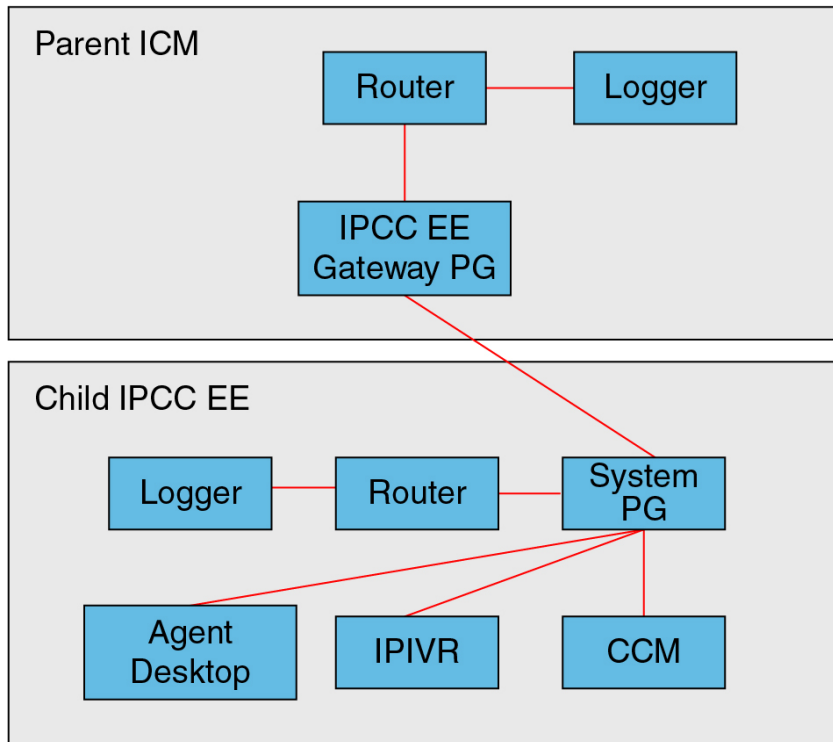
This view is important because it allows a "parent" Unified ICM to monitor and send calls to a "child" Unified CCE system. Previous to the availability of the Unified CCE Gateway feature, monitoring this using intelligent (PG-based) routing was not possible. This allows improved scaling, resiliency, and simpler, more standardized scripting across sites.

The Unified CCE Gateway PG looks to the Unified ICM like any other PG; it does not look like Unified CCE. The Gateway PG has a Peripheral type to allow connection to a "child" Unified CCE system. The child system is listed as "Unified CCE Gateway" in the setup screen for the PG. The Gateway PG does not support third-party call control (CTI) so that all agent desktops, etc., are connected to the child Unified CCE system. The link supports voice only. The child system may support multi-media but the Unified ICM only routes voice calls to the child.

Object tracking is accomplished the same as on all legacy PIMs, through matching Peripheral Numbers on the agent, skill group, and service table. Routing to the child is to peripheral targets (skill groups/services) as with all legacy TDMs, in contrast to the Unified CCE. The usual TDM functionality is supported: Pre-Routing, Pre-Routing with Translation routing, post routing, etc. Third-party call control is the only exception. Full variable passing is done between the parent and child (you can send/receive call variables 1-10 and ECC variables).

Scripting is consistent with traditional PGs, not Unified CCE. Scripts use LAA, MED, target services, and skill groups; but not agents.

Note NCT is not supported with the Enterprise Gateway because the CTI control is on the ACD (child) rather than on Unified ICM parent.

Deployment of Unified CCE Gateway for Unified CCE

Where to deploy Unified CCE Gateway for Unified CCE:

- The gateway PG is ideal for deployments that have several call centers spread geographically.

- The call centers are independent of the Central Controller so in a WAN outage, the call centers (System PG systems) can operate totally independently of Unified ICM.

- The gateway PG is also an ideal way to integrate new IP call centers into an existing Unified ICM environment with many TDM ACD sites.

Note   In a single script node (MED) peripherals that are legacy and gateway type can be used to select the best "site" to send the call to.

How does it work:

- The gateway PG works by monitoring all the events that happen on the child Unified CCE system. This is identical in function to a PG/PIM connected to any other ACD. By knowing the active call and agent states the Unified ICM router can have information available to decide if that "site" is the best to send a call to.

- The child system can also send up route requests to the parent Unified ICM to decide where to send a call (Post/Translation routing).

- The event monitoring is accomplished by the PIM connecting to the child systems CTI Server modified version of the CTI Protocol.

Object Mapping Overview (Enterprise)

- Agents on the child map to agents on the parent

- Skill groups on the child map to skill groups on the parent

- Call Types on the child map to services on the parent.

Note   This is due to the child's reporting being based on Call Types and the parents reporting being based on services.

Unified CCE

This topic contains information for writing routing scripts when Unified ICM is part of a Unified CCE environment.

Prioritize Agents

Follow these steps to prioritize agents within a skill group:

Procedure

Step 1   Use a Select node and group the agents that have a higher priority in a skill group in the first Select Longest Available Agent (LAA) node,

Step 2   Look for available agents in a subsequent Select node that have another lower priority grouping of agents.

What to do next

This is done in the case where there are idle agents when a call comes in. The order of the Select nodes within the script determines the agent prioritization.

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 (1-20 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 Unified 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.

Select Node

You can use the Select node to check for the Longest Available Agent (LAA). However, the Select node cannot check for Minimum Expected Delay (MED) of a service, because Unified CCE skill groups do not have a valid expected delay because the calls in the queue are on Unified IP IVR PG service.

By default, the Queue to Skill Group node uses LAA for agent selection. Do not put LAA Select nodes before or after a Queue to Skill Group node for the same skill group. You can only use this node if the call is not already queued to a skill group.

There is, however, a lengthy workaround to determine the MED for a call using an IF node.

Related Topics

Queue to Skill Group Node

If agents are assigned to base skill groups, use base skill groups in this node.

If the script has both Queue to Skill Group and Queue to Enterprise Skill Group nodes, then do not include skill groups defined within the Queue to Skill Group nodes that belong to the enterprise skill groups that are defined within the Queue to Enterprise Skill Group node.

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.

Busy Node

Use the Busy node for initial overflow conditions. Do not use this treatment if the call has already received ringback tone or given announcements or music. If a Busy CTI Route point and CTI port are defined on Unified IP IVR and the Cisco Unified Communications Manager, then Unified IP IVR tells Cisco Unified Communications Manager to return a busy signal. If a Busy CTI Route point and CTI port are not defined on Cisco Unified Communications Manager and Unified IP IVR and the call is already connected to an Unified IP IVR port, then Unified IP IVR port plays a busy tone from a .wav file for 30 seconds and then disconnects the call. If the VRU port plays a busy tone from a .wav file then, answer supervision returns to the far end. This causes charges to occur for the call. If the call is a toll free call (for example, 800#), then the Contact Center accrues the charges; otherwise the caller has to pay.

You must define a Busy CTI route point and one CTI port on the Cisco Unified Communications Manager and associated it with the Unified IP IVR user. The one CTI port can be used for multiple calls. In Unified IP IVR Administration, a new CTI Route point of Type Busy and a CTI port need to be defined whose Dialed Numbers match what was previously defined on the Cisco Communications Manager.

Define a label of type busy with a Label field that corresponds to the aforementioned CTI Route point dialed Number in the Cisco Unified Communications Manager and Unified IP IVR.

If the call is not at a VRU port and the Busy CTI Route point is not defined on the Cisco Unified Communications Manager and the Unified IP IVR, and the Unified ICM returns a busy label to the Cisco Unified Communications Manager, the Cisco Unified Communications Manager returns a fast busy to the caller for 30 seconds and then the call is disconnected.

Ring Node

You can use the Ring node for diverting blocked list callers. It is supported as a post route from the Cisco Unified Communications Manager. The call is given ringback tone until the caller ends the call. No answer supervision is returned for this call because it is not connected to a VRU port. This means that if this is the initial treatment given the caller, then no charges will be accrued for this call.

You need to define a Ring No Answer CTI port group in the Cisco Unified Communications Manager, but you do not have to define CTI ports. In Unified IP IVR Administration, you need to define a new CTI Route Point of Type Ring No Answer whose Dialed Number matches what was previously defined on the Cisco Unified Communications Manager.

Define a label of type Ring with a Label field corresponds to the CTI Route Point Dialed Number in Cisco Unified Communications Manager and Unified IP IVR.

Release Call Node

Use the Release Call node for initial overflow conditions. It is supported as a post route from the Cisco Unified Communications Manager. Do not use this node if the call has already received ringback tone or announcements or music. The call is given busy tone for 30 seconds before it is disconnected on the Cisco Unified Communications Manager. No answer supervision is returned for this call because it is not connected to a CTI port. This means that no charges will be accrued for this call.

No configuration needs to be done within the Cisco Unified Communications Manager or Unified IP IVR. No label needs to be defined within Unified ICM.

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.

Call Treatment Comparison

Attributes/Treatment	Busy Node	Ring Node	Release Node	End Node
Answer Supervision	No if Busy CTI Route Point defined Yes if Busy CTI Route Point not defined	No	No	No
Voice port required	Yes (only 1 CTI port needed)	No	No	No
Timing	Initial treatment only	Anytime	Initial treatment only	Initial treatment only
Applications	Overflow Conditions	Blocked list Callers	Overflow Conditions	Overflow Conditions
Treatment provided by Unified IP IVR	Unified IP IVR		Cisco Unified Communications Manager	Cisco Unified Communications Manager

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.

For a transfer to a specific target agent, the initiating agent enters the target agent ID. The DNP entry matching the dialed number (agent ID) must have a DNP type of PBX. For this DNP type, the PIM enters the dialed

number (agent ID) into the Caller-Entered Digits (CED) field while sending the route request to the Router. To have the Router handle the call properly, specify the CED field as the location of the agent ID in the Agent to Agent node.

Agent IDs must not match any of the extensions on the Unified Communications Manager cluster. If you make all agent IDs of the same length and begin with the same number, a generic wildcard string can match all agent IDs. With that wildcard string, you need only one entry in the DNP for agent-to-agent routing.

The agent-to-agent node requires that you specify the PIM. If your environment has multiple PIMs, use an agent ID number plan to determine which PIM contains an agent. Agent IDs are associated with a specific PIM and are not unique by themselves. You can set up a consistent agent ID assignment plan (such as, all agent IDs on PIM 1 begin with 1, and so on). That pattern enables you not to repeat agent IDs across the enterprise. Then, you can parse the CED field in the script editor to determine which PIM contains a specific agent.

Service and Enterprise Service Nodes

You do not need the Service node or Enterprise Service node to route calls to Unified CCE agents because services are not required for Unified CCE agents.

Scheduled Select and Divert Label Nodes

Cisco Communications Manager does not support the Scheduled Select node and Divert Label node. You cannot use these nodes to route calls to Unified CCE agents.

Unified IP IVR as a queue point

Unified CCE relies on the Unified IP IVR to queue the call while it is waiting for an available agent.

The Unified ICM sends the call to the VRU port for queuing:

- To provide the call with a termination point that allows the VoIP Gateway to return the correct signals or messages back to the PSTN.

- 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.

- To obtain further information from the caller that is not sent from the network

The Unified IP IVR lets Unified ICM know when the caller disconnects through the Event Report Message with an Event Type of either DISCONNECT or ABANDON. When an agent becomes available, Unified ICM automatically instructsUnified IP IVR 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 or Queue to Enterprise 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 Unified ICM only looks for available agents for that call when the VRU script is finished and the call runs 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 Configuration Manager, under the Add Network VRU script configuration. Neither the VRU or the Unified ICM script can overwrite this setting.

VRU (IVR) Types

Listed in the table below are the VRU types that are supported for the Unified CCE. These types are defined in the Unified ICM in the Network VRU Configuration. These VRU types are used by the Unified ICM to determine the routing client, routing type, script node used, and VRU port status. Type 3 and 7 can only be Network VRUs. However, Types 2 and 6 can be Network or On-premise VRUs.

VRU Type	Routing Client	Routing Type	Scripting Node Used to Send to VRU	Unified ICM Knows VRU Port Status
2, 8	Yes	Post-route from Cisco Communications Manager	Translation Route to VRU	Yes
3, 7	No	Pre-route	Send To VRU	No
5, 6	Yes	Post-route from VRU	N/A	Yes

Usually the VRU has been configured to support only one VRU type, although the Unified IP IVR can support both Type 2 and Type 6 based on whether the call is routed directly to the VRU or the call comes into a Translation Route point. In this case, the VRU type must be defined as Type 2.

Translation Route to VRU Node

The Translation Route to VRU node is used by Type 2 VRUs that are post routed from the Cisco Unified Communications Manager, when the call is not at the VRU, but at another routing client. The call can be a pre-route from the IXC or it can be a post route from a Cisco Unified Communications Manager CTI Route point.

The Translation Route to VRU node is used to send the call to the VRU port. Translation routing is needed for:

- Cradle to grave reporting (Termination Call Detail Reporting)

- Ability to send call context data like the calling line ID or Dialed Number to the VRU

- Ability to check if the VRU peripheral is online or if all VRU ports are busy before sending the call

The Translation Route to VRU node is typically followed by a Run External Script node. If some checking is done before the Run External Script node is initiated, the time it takes the Unified ICM to run the nodes in

the script must not exceed the VRU Request Time-out timer. Information obtained from the caller during the VRU session can be passed to the Unified ICM for further processing.

No Default Skill Groups in Routing Scripts

Ensure your Unified ICM routing scripts do not reference the default skill group. This ensures that the default skill group does not capture statistics for any Unified ICM-routed calls.

For more information, see *Cisco Unified Contact Center Enterprise Installation and Upgrade Guide* and *Administration Guide for Cisco Unified Contact Center Enterprise*.

Router Behavior (During PG Failure)

In a Unified CCE environment, if the router is disconnected from the two PGs (available on either side of the network/site between the routers) and the PG fails, the router cannot receive any information about the failure of the peripheral or about agents who are logged out. In such a scenario, the router uses low frequency heartbeats (60 seconds) to determine the failure of the peripheral and declares the peripheral to be offline at this point. The router discards the real time values for the peripheral (including service, skill group, enterprise agent, and agent state at their last known value) and continues to send calls to these agents who were in the "Ready" state before the connectivity to the PGs was lost. After the router suspends the peripheral and declares it offline, you need to use an "IF" node with "Peripheral.online=1" in the script to avoid sending or queuing calls to an offline peripheral. You should make this change manually.

Scripts in an Outbound Environment

Outbound Option only supports both Type 2 (Generic PG environment) and Type 9 (Unified CCE System PG environment). For Unified CVP, Outbound Option supports Type 10, Type 7, and Type 5 (Unified CVP Comprehensive Model).

If using the transfer to VRU feature, create a transfer to VRU campaign. Transfer to VRU campaigns might require translation routes if using a Type 2 VRU. For more information about setting up translation routes, see *Cisco Unified Contact Center Enterprise Installation and Upgrade Guide*. (Note that translation routes are not required when using the Unified CCE System PG.)

Note   The transfer to VRU feature is only supported for Outbound Option on Unified CCE. You cannot use this feature in the Direct Preview or the regular Preview modes.

The dynamic routing client feature is used when an outbound agent transfers a call using the Unified CVP to a Type 2 VRU. Make sure the routing client for the translation route labels is the Cisco Unified Communications Manager, which makes the outgoing call.

Control OutboundControl Variable and Skill Group Reservation Percentage with Script

Use the Unified ICM Script Editor application to create an administrative script for each skill group to control the OutboundControl variable and the skill group reservation percentage (you can use one script to control all Outbound Option skill groups or multiple scripts to control multiple Outbound Option skill groups). The Outbound Option Dialer looks at the value of the OutboundControl variable to determine which mode each skill group uses. If the OutboundControl variable is not set, the skill group defaults to inbound. If using the transfer to VRU feature, create a transfer to VRU campaign.

> **Note** The transfer to VRU feature is supported only for the Outbound Option on the Unified CCE. You can not use this feature in the Direct Preview or the regular Preview modes.

How to create an administrative script to control the OutboundControl variable and skill group percentage:

Procedure

Step 1 Open the Unified ICM Script Editor application.

Step 2 Create an administrative script using a Start node, a Set node, an End node, and an If node (all required).

Step 3 Set the OutboundControl variable. OutboundControl Variable Settings:

a) INBOUND: Indicates that this skill group is disabled for outbound use and only takes inbound calls.

b) PREDICTIVE_ONLY: Dials several customers per agent. After reaching a live contact, the Predictive Dialer transfers the customer to a live agent along with a screen pop to the agent's desk. The predictive algorithm is designed to calculate the number of lines to dial per available agent to keep agent wait time to a minimum.

c) PREDICTIVE_BLENDED: Agents receive inbound calls, but could be used for an outbound call when available.

d) PREDICTIVE_BLENDED: Agents receive inbound calls, but could be used for an outbound call when available.

e) PREVIEW_ONLY: Reserves an agent prior to initiating an outbound call and presents the agent with a screen pop. The agent might then **Accept** the call: Dials the customer and transfers the call to the agent. **Skip** the call: Agent receives another customer call. **Skip-Close** the call: Skips the current preview call and closes the record so it cannot be called again. **Reject** the call: Releases the agent. At this point, the system might deliver the agent another preview outbound call or a new inbound call. **Reject-Close** the call: Rejects the current preview call and closes the record so it cannot be called again.

f) PREVIEW_BLENDED: Agents receive inbound calls, but could be used for an outbound preview call when available.

g) PREVIEW_DIRECT_ONLY: Agents can only place outbound calls and hear ring tones, such as phone ringing or busy signal.

h) PREVIEW_DIRECT_BLENDED: Agents can receive inbound calls, place outbound calls, and hear ring tones, such as phone ringing or busy signal.

i) PROGRESSIVE_ONLY: Similar to PREDICTIVE_ONLY; however, lines to dial per agent is not calculated—users configure a fixed number of lines that will always be dialed per available agent.

j) PROGRESSIVE_BLENDED: Similar to PREDICTIVE_BLENDED, but a fixed number of lines will always be dialed per available agent.

Note	• Verify that the outbound control variable mode is spelled correctly.
	• Make sure that the skill group being controlled is the base skill group, and not the primary or secondary skill groups. Although agents may be logged into just the primary or secondary skill group, the outbound control variable must always be set on the base skill group
	• If the administrative script (where the OutboundControl variable or reservation percentage is set) is running, but the modes/percentages are not being updated at the Dialer, perform one of the following tasks:
	• Use the Set node to set skill group variables (OutboundControl and OutboundPercent).

Step 4   Set the OutboundPercent variable by entering the agent percentage in the Value field of the Set Properties window. This variable controls the percentage of agents, logged in to a particular skill group, to be used for outbound dialing. For example, if there are 100 agents logged in to skill group N, and the OutboundPercent variable is set to 50%, 50 agents would be allocated for outbound dialing.

Note	• This variable does not allocate specific agents for outbound dialing, just a total percentage.
	• Use the Set node to set skill group variables (OutboundControl and OutboundPercent).

What to do next

The following illustration displays a very simple administrative script where both the OutboundControl variable and the outbound percentage are set for a skill group. A script in a production call center is typically be more complex, perhaps changing these variables due to time of day or service level.

Note	• For more information about using the Script Editor, see Unified ICM Script Editor online help.
	• For more information about sample administrative and routing scripts for the Unified CCE System PG, see the section for sample administrative and routing scripts for the Unified CCE System PG.

Figure 118: Setting Skill Group Variables (OutboundControl and OutboundPercent)
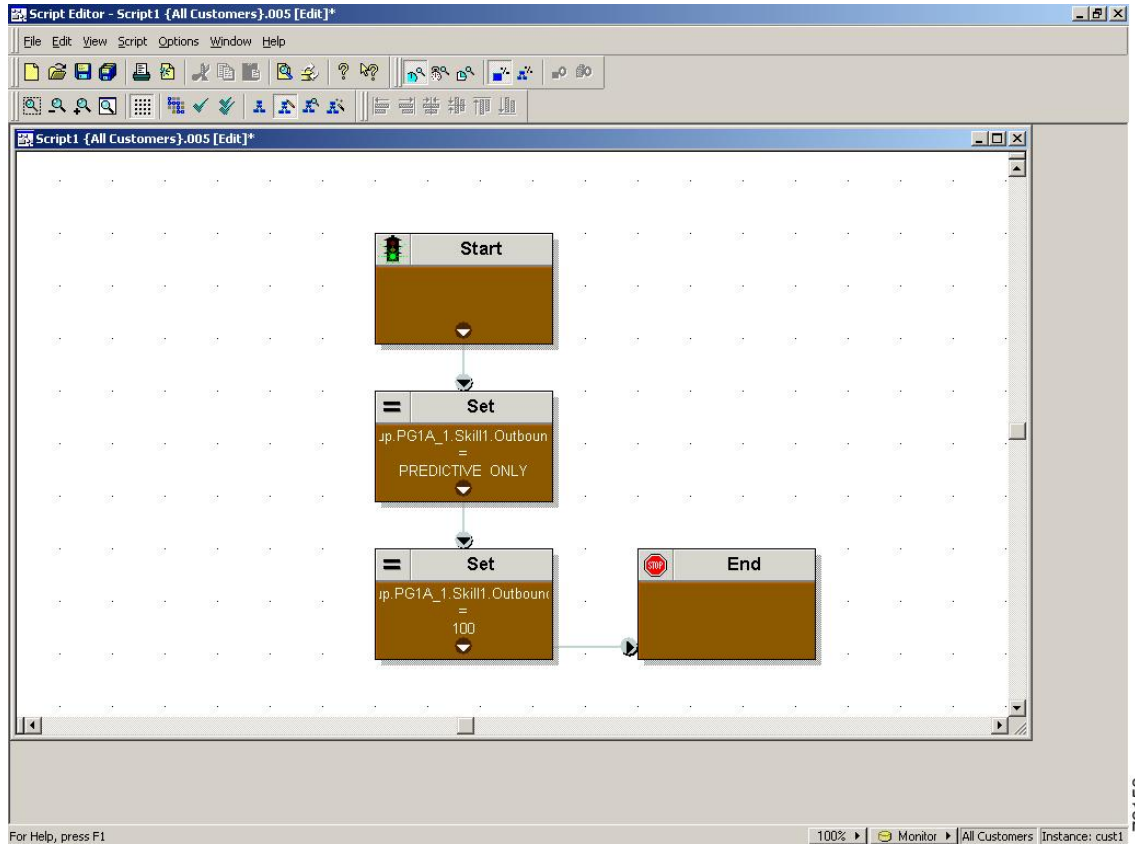
Transfer to VRU Using Outbound Option with Unified IP IVR

The following illustration displays a routing script for a transfer to VRU campaign using the Outbound Option with the Unified IP IVR. (For more information about configuring a Outbound Option transfer to VRU campaign, see *Outbound Option Guide for Unified Contact Center Enterprise*.)

Figure 119: Example Routing Script for a Transfer to VRU Campaign Using Outbound Option with Unified IP IVR
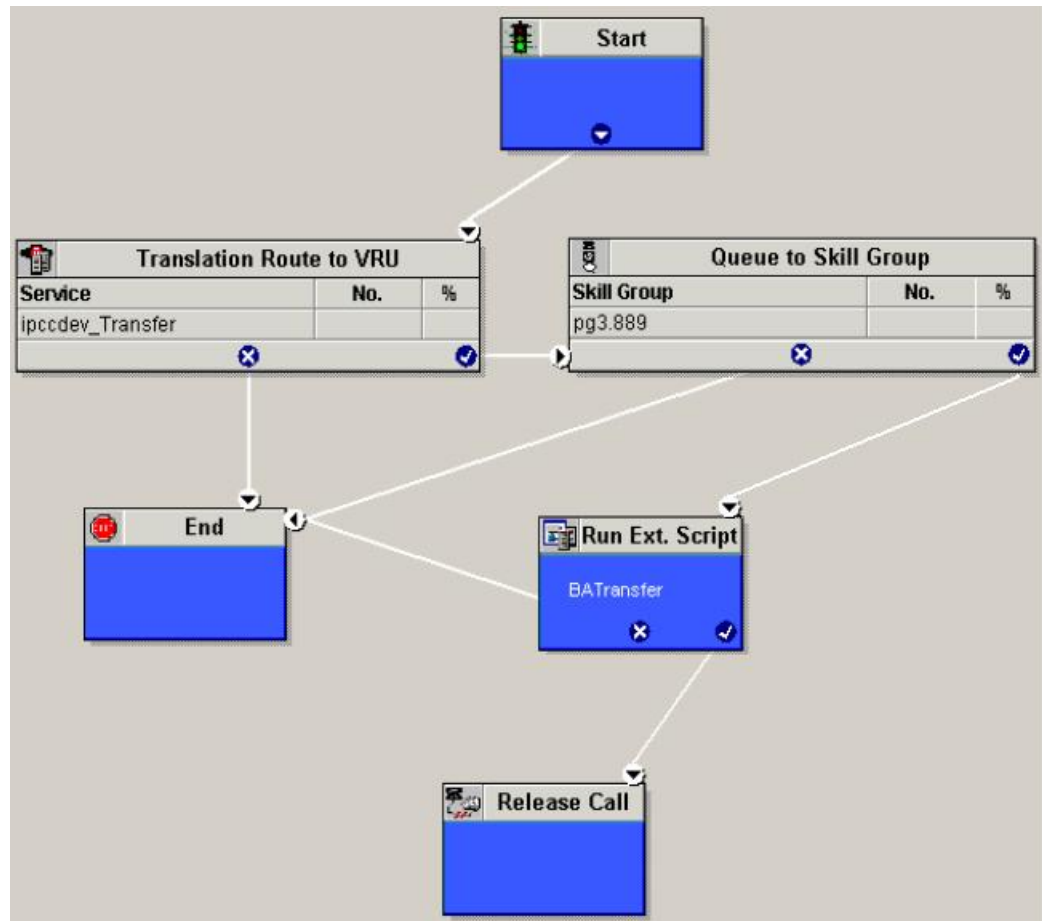
Transfer to VRU Campaign Using Outbound Option with Unified CVP

The following illustration displays a routing script for a transfer to VRU campaign using the Outbound Option with the Unified CVP. (For more information about configuring a Outbound Option transfer to VRU campaign, see *Outbound Option Guide for Unified Contact Center Enterprise*.)

Figure 120: Example Routing Script for a Transfer to VRU Campaign Using Outbound Option with CVP/ISN
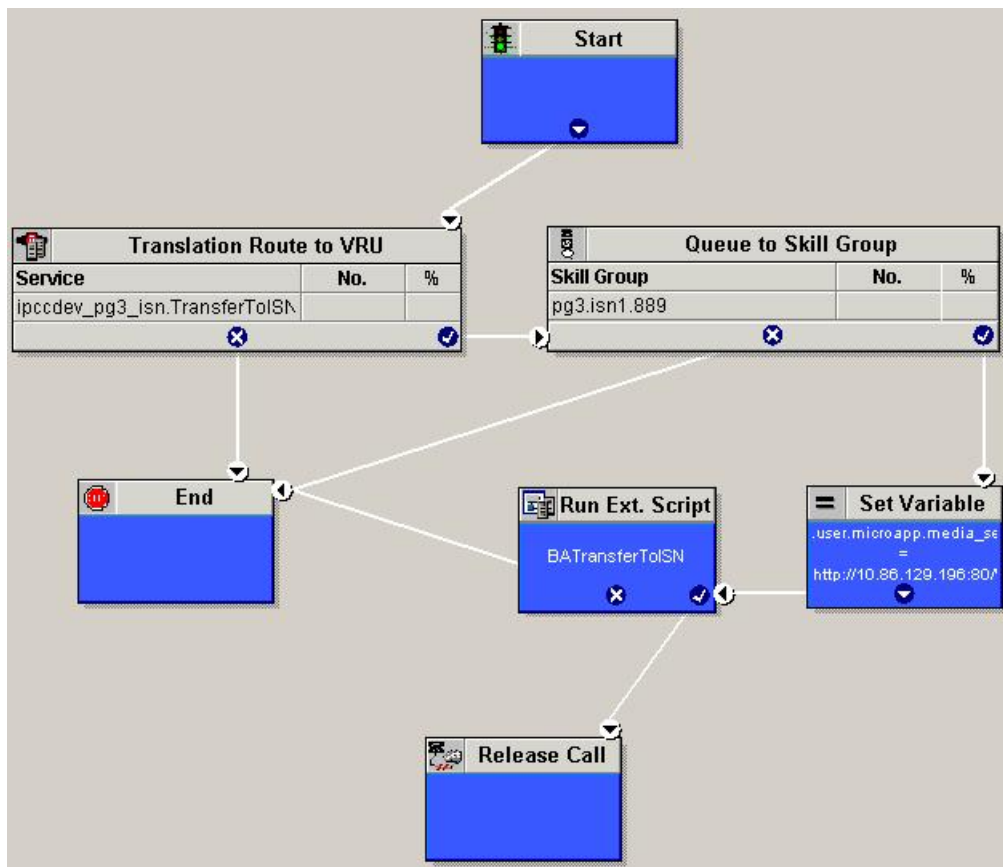
Use the Unified ICM Script Editor application to create a routing script that uses the Dialed Number for the MR routing client, and routes through a Select node to the previously configured skill group.

Use the Unified ICM Script Editor Call Type Manager to associate the MR (and Personal Callback, if used) Dialed Number(s) with the configured call type and newly created routing script.

Note   For more information about using Unified ICM Script Editor's features, see the *Unified ICM Script Editor online help*.

The following diagram displays an example routing script that uses the objects mentioned above.

Figure 121: Example Routing Script Using the Dialed Number for the MR Routing Client
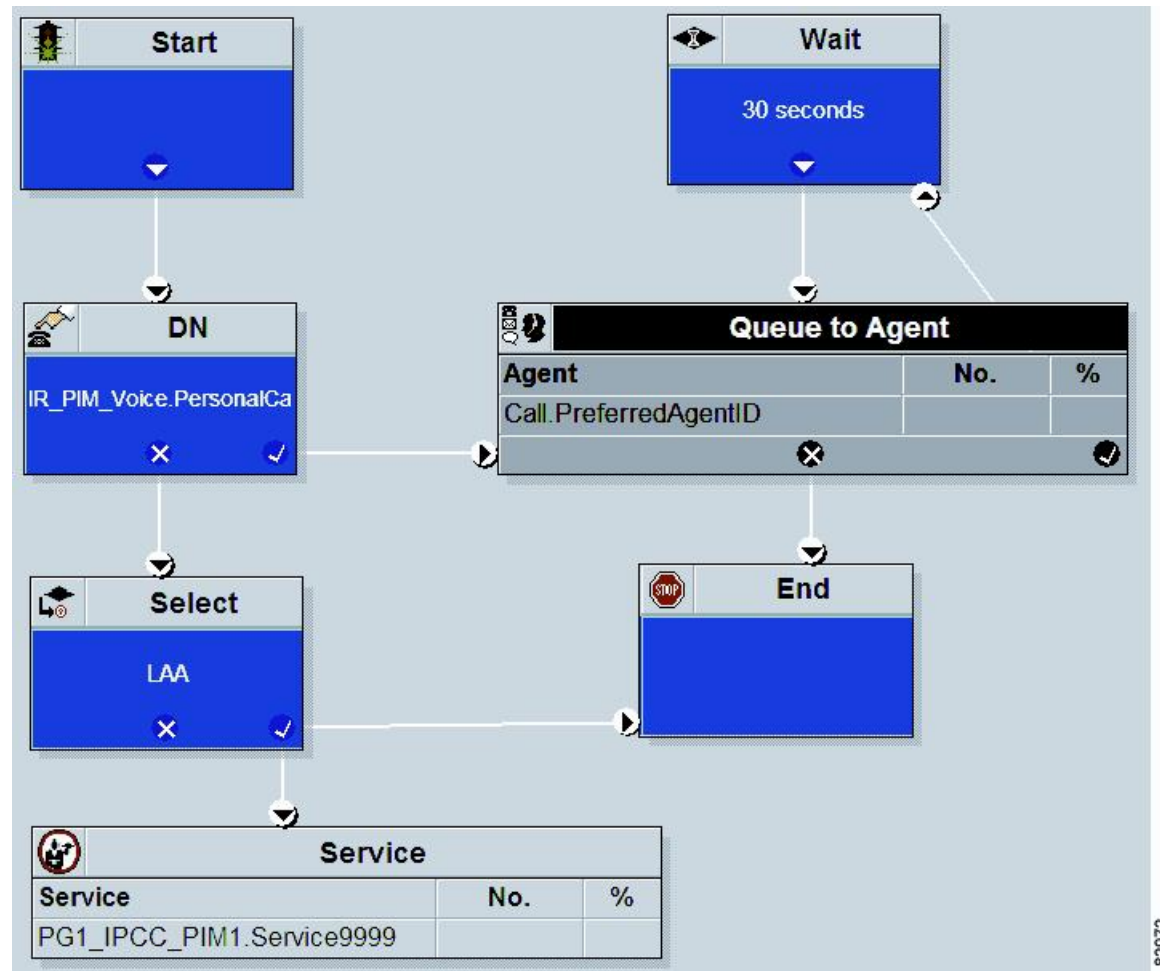
Note Lines connecting objects cannot appear on top of objects and therefore, partially display under the objects. For example, the line connecting the "X" (output terminal failure) on the Select object to the End object runs partially under the Select object.

Note Translation routes are not used in the Unified CCE System PG, so routing scripts using this PG do not need to use this object.

Queue to Agent Node Configuration

The following illustration displays the Queue to Agent tab of the Queue to Agent Properties dialog box.

Figure 122: Queue to Agent Properties
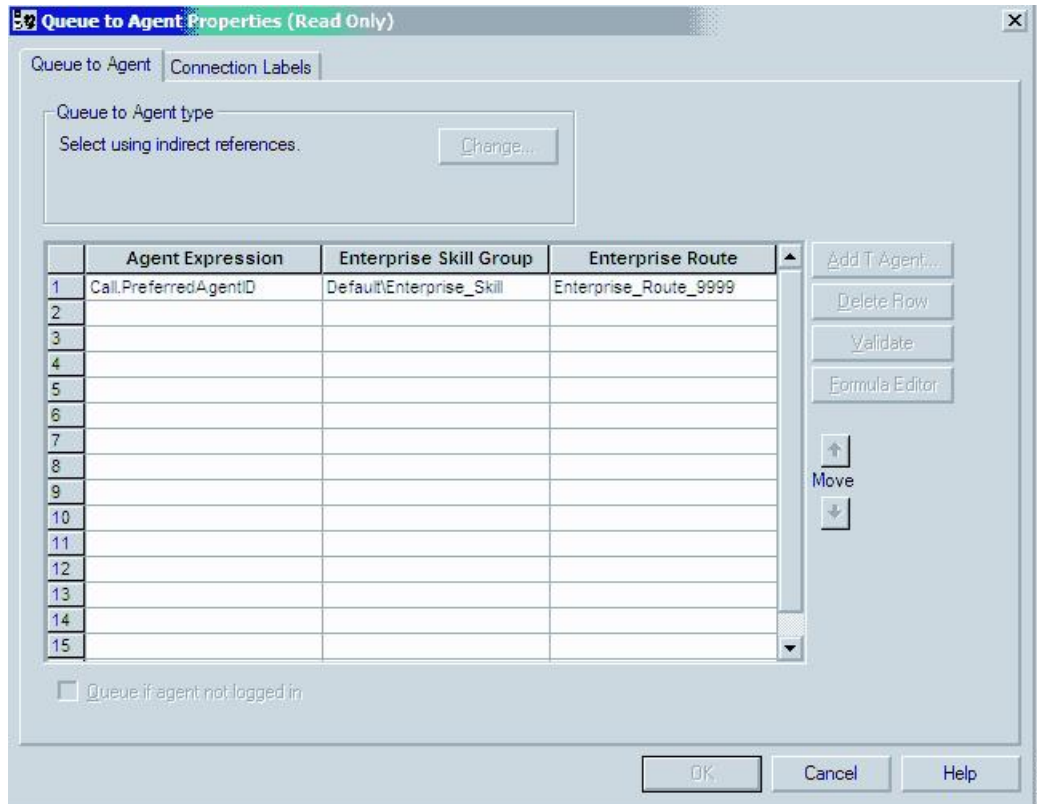
Unified CCE System PG for Outbound Option

The Unified CCE System PG combines two PGs (Cisco Communications Manager and VRU PGs) into one PG and is only supported with Unified IP IVR.

Setting up an Unified CCE System PG for Outbound Option campaigns consists of two tasks:

- Configuring the Unified CCE System PG.

- Adding the PG.

Use the Unified ICM Script Editor and create an administrative script. The following illustration is an example of an administrative script.

Figure 123: Example Unified CCE System PG for Outbound Option Campaign Administrative Script
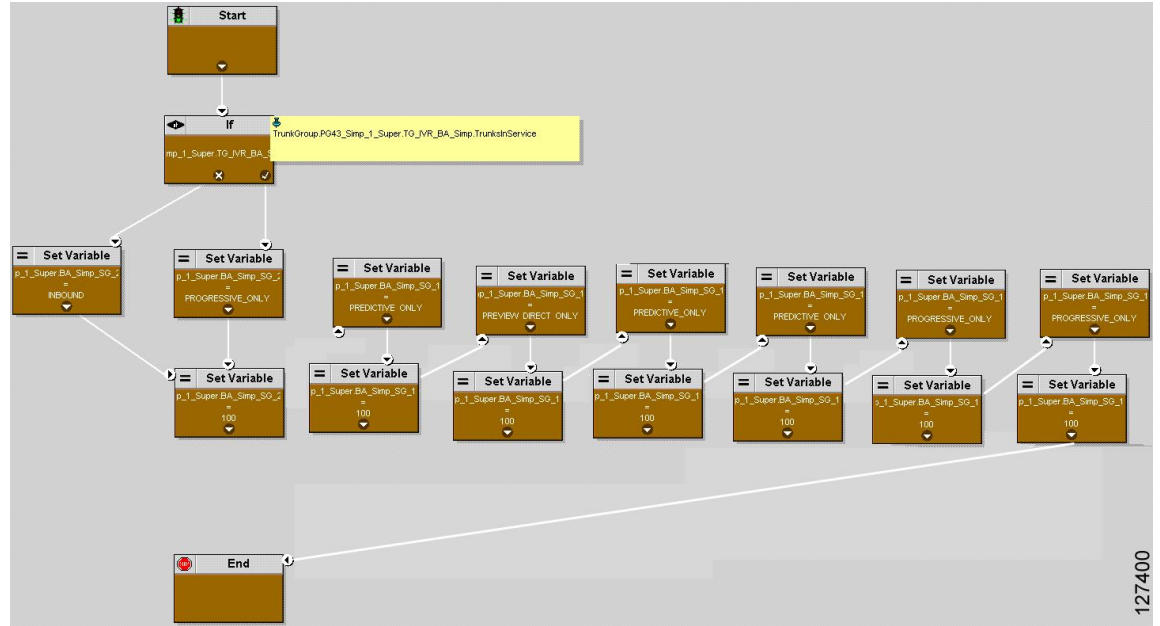
The following diagram displays an example routing script using the Queue to Skill Group node.

Figure 124: Example Routing Script Using the Queue to Skill Group Node
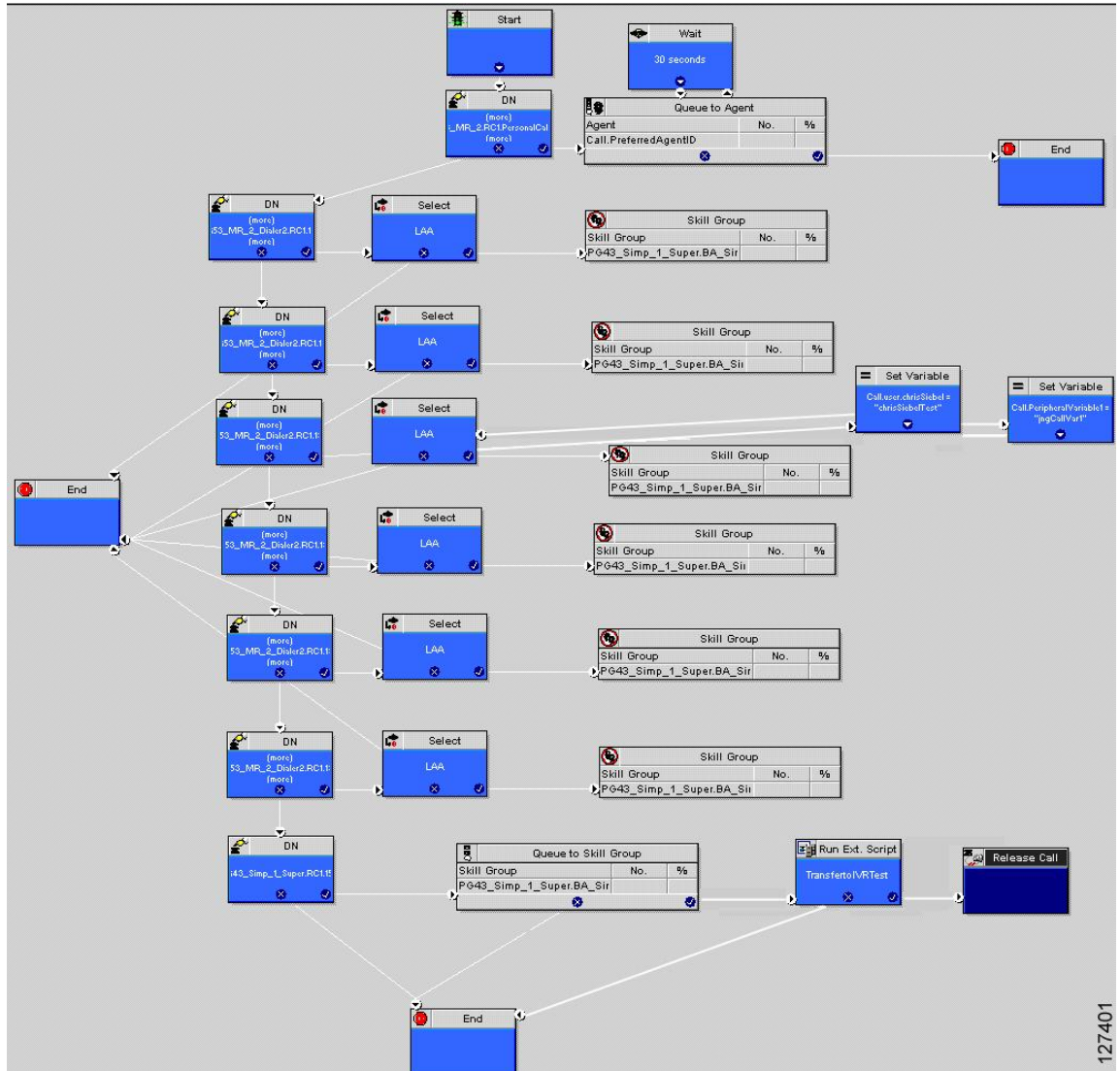
C H A P T E R 12

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 125: Start Properties
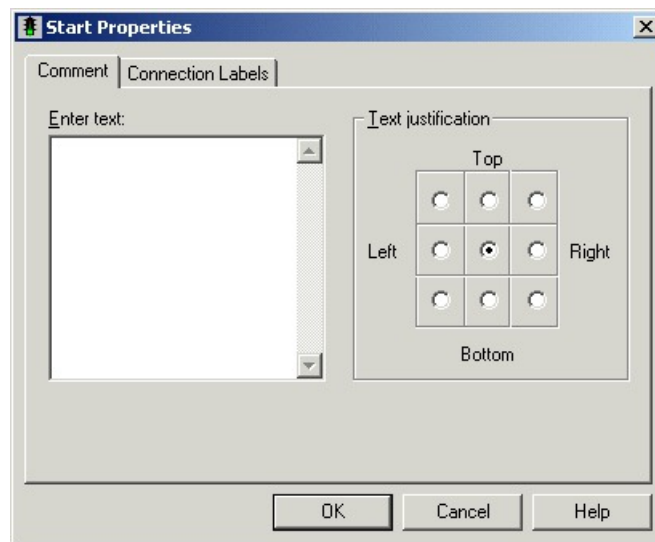
Comment Node

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 126: The Comment Icon

Figure 127: Comment Properties
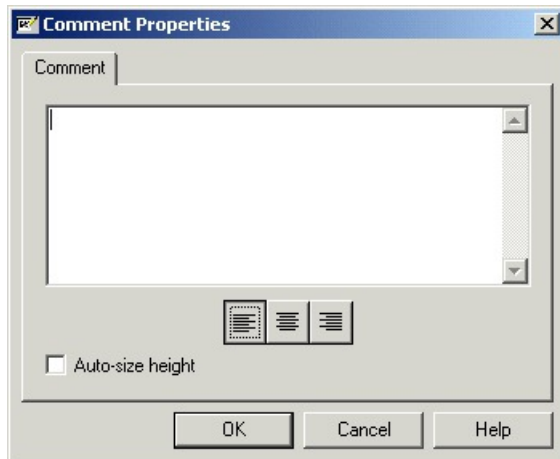
For example, you might add a comment describing the purpose of the script.

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 128: 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:

Figure 129: Line Connector Properties
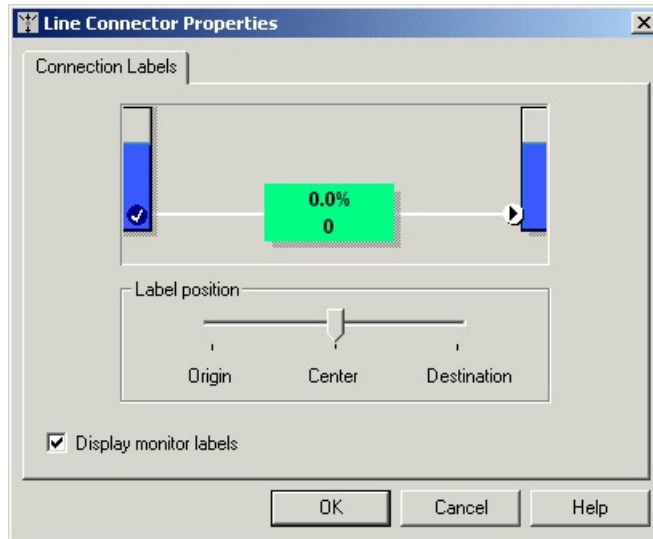
# Example Scripts

Example Collaboration Scripts

You can configure the Unified ICM and Enterprise Chat and Email so that the Unified ICM routes Web Collaboration requests that are processed by Enterprise Chat and Email.

Overview of ECE Web Request Routing Through Unified ICM

With the ECE integrated with the Unified ICM, the Unified ICM can route chat requests.

The process for routing Web requests through the Unified ICM can be divided into 5 parts, as shown in the following image:

Figure 130: Web Request Routing Process
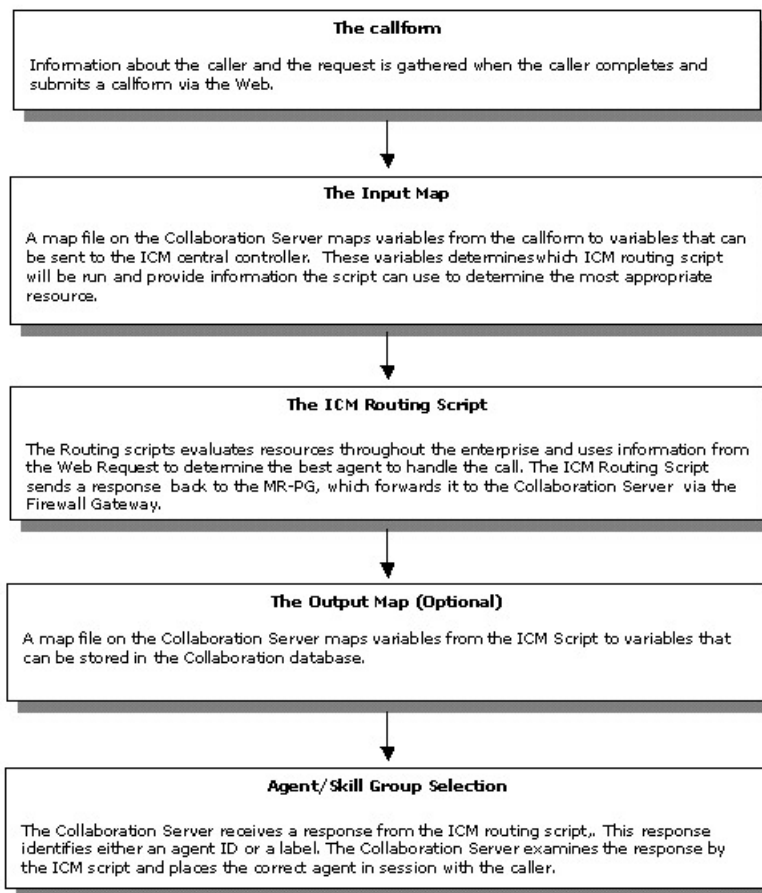
Web Requests and Media Routing Domains

Enterprise Chat and Email can take advantage of the following types of Media Routing Domains (MRDs):

- Non-voice MRDs
- Voice MRD

For more information about MRDs, see the *Administration Guide for Cisco Unified Contact Center Enterprise*.

Non-Voice MRDs

The Enterprise Chat and Email uses Non-voice MRDs for chat requests.

# The Voice MRD

The Voice MRD handles these types of requests:

- Blended chat with legacy ACDs (these are requests for chat, but with agent reservation on the ACD)

• Web callback and delayed callback with both ACDs and Unified CCE

Script That Queues a Web Request to a Skill Group

The following script example shows how a Web request can be queued to a skill group:

Figure 131: Example - Queuing a Web Request to a Skill Group
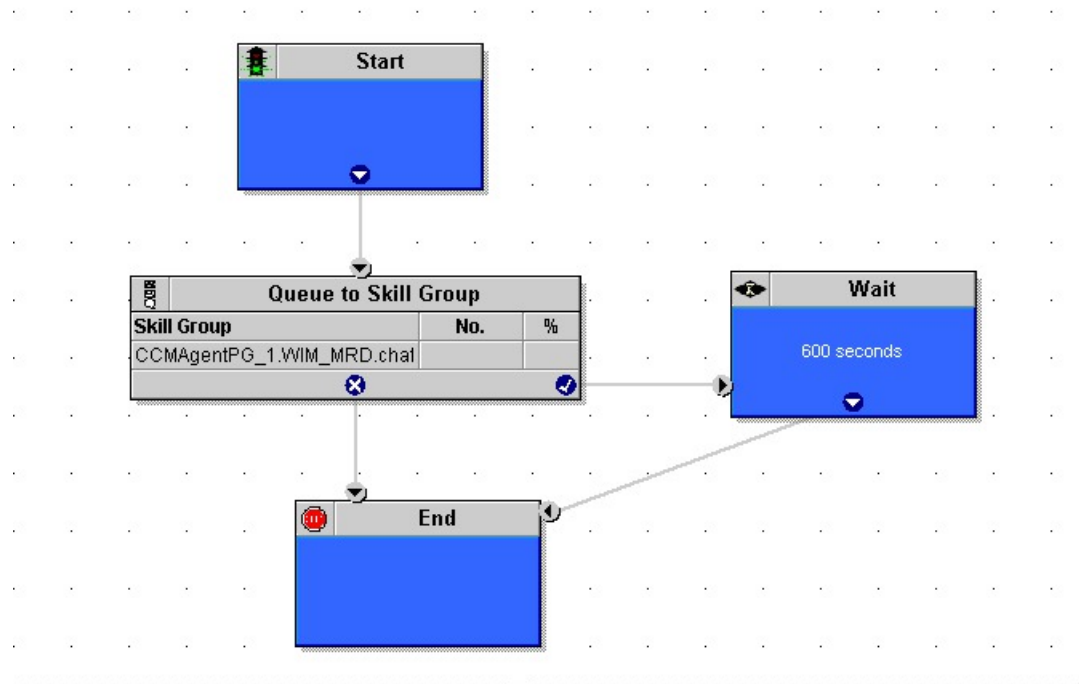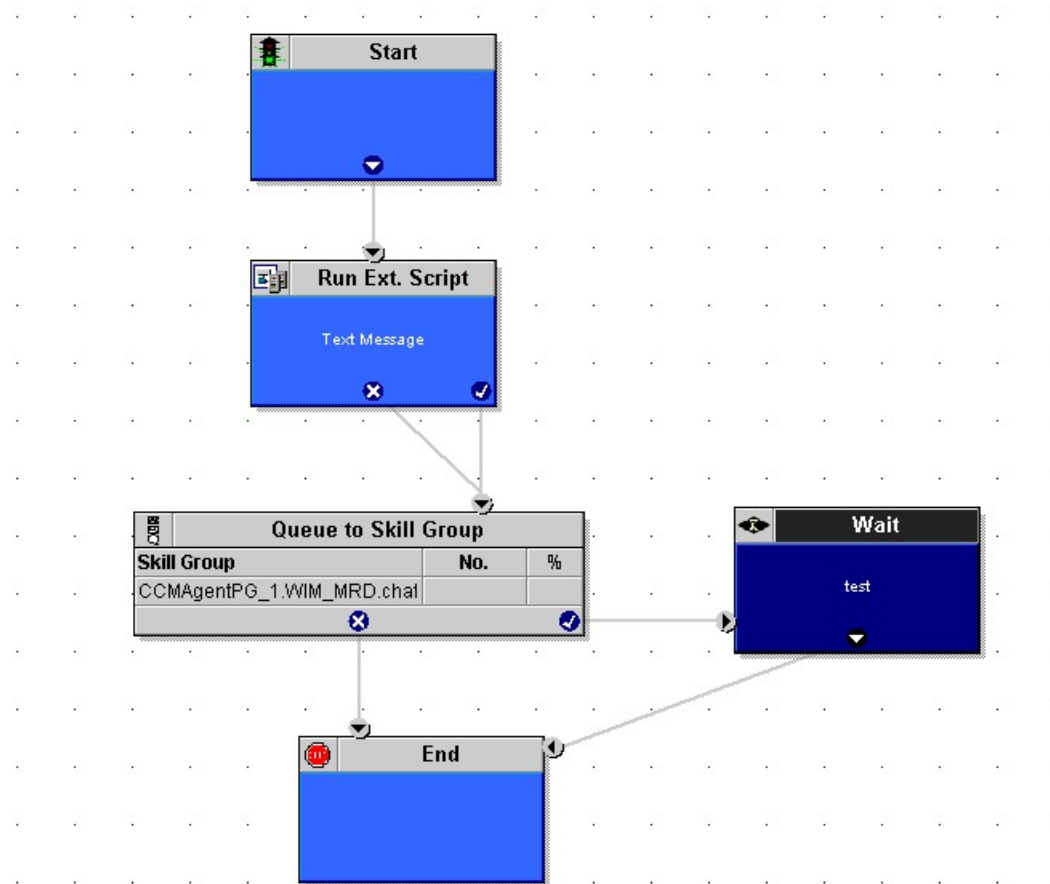

In this example:

1. This script is scheduled to run for web chat requests. For more information about call types and scheduling Web requests routing scripts, see Determination of Call Type for ECE Web Request.

2. The script queues the request to a chat skill group. At this time, Unified ICM attempts to find an available agent who is a member of that skill group. When the agent is found, the system software returns the agent ID to the Cisco Interaction Manager.

3. The script is run and the Enterprise Chat and Email processes a new chat request. This script is scheduled to run for chat requests.

4. The script queues the request to a chat skill group. At this time, the Unified ICM attempts to find an available agent who is a member of that skill group. When the agent is found, the Unified ICM returns the agent ID to the Enterprise Chat and Email.

Script That Pushes a URL to a Waiting Caller

The following script example shows how a script can push a URL to a waiting caller before the Web request is queued to a skill group:

Figure 132: Pushing URL to Waiting Caller

In this example, the Run External Script node pushes the selected URL to the caller's browser.

✎

Note For the Run External Script node to work, there must be an entry in the Network VRU list pointing to the URL map file. After the Run External Script node, the script functions just as the preceding example.

Script That Queues Directly to an Agent

The following script example shows how you can queue a Web request directly to an agent:
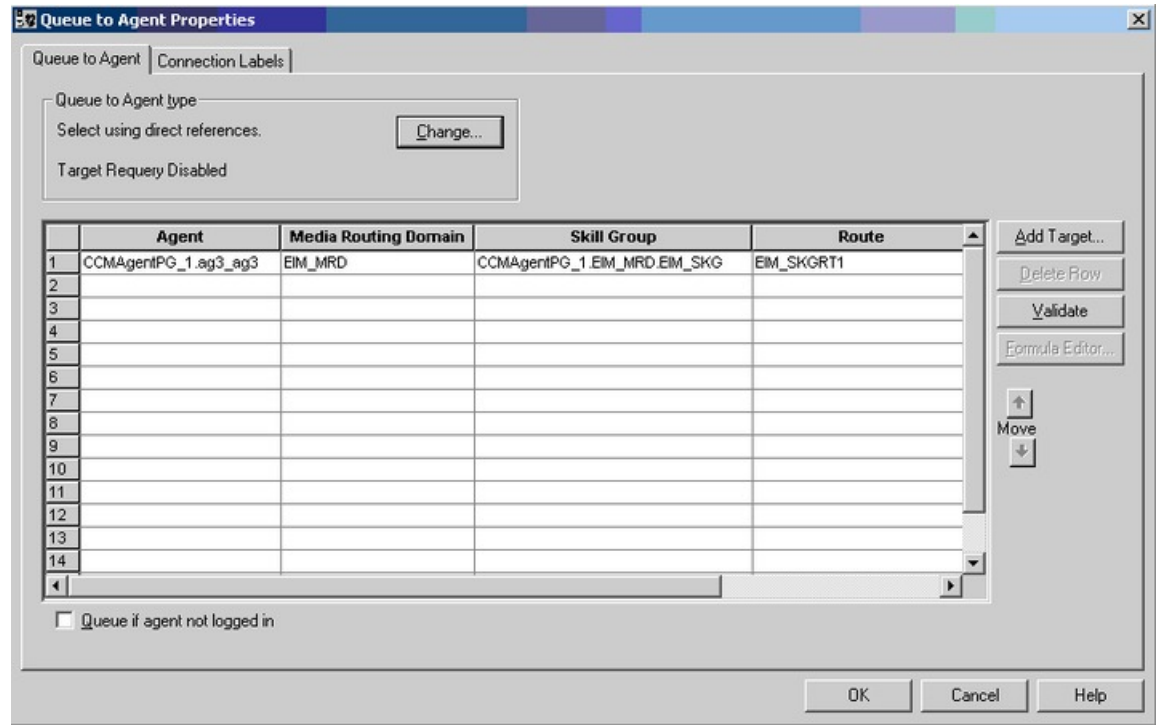
Figure 133: Queuing Directly to an Agent
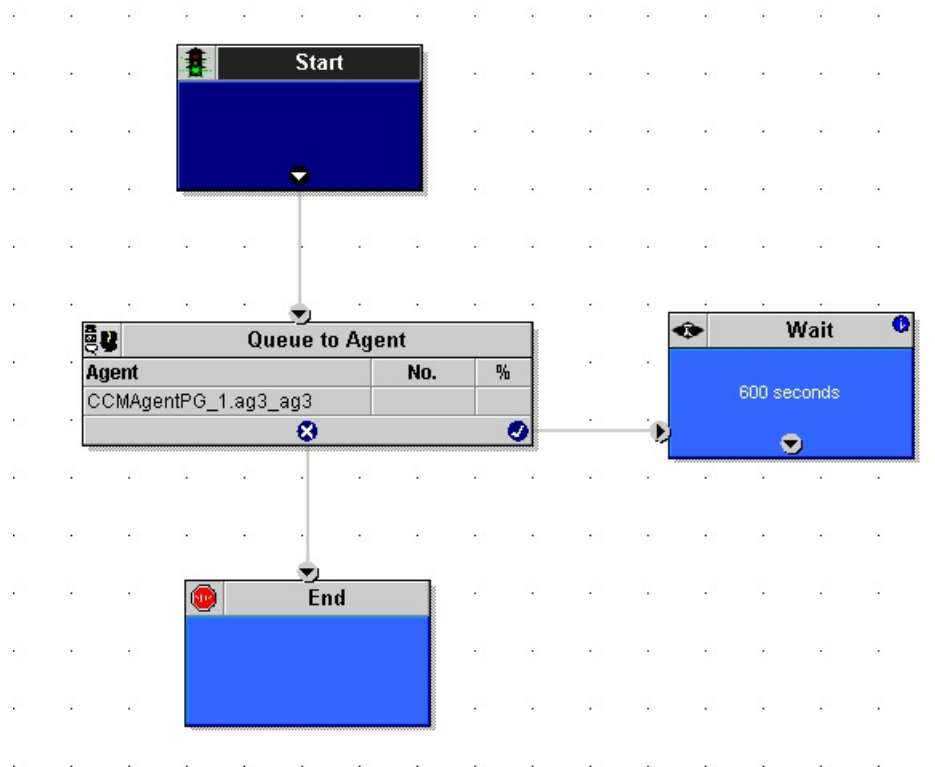
In this example:

1. You can schedule this script to run for a particular type of request.

2. The script attempts to queue the request directly to an agent using the Queue to Agent node. The Queue to Agent node uses a direct reference to the agent, as shown in its Properties dialog box:
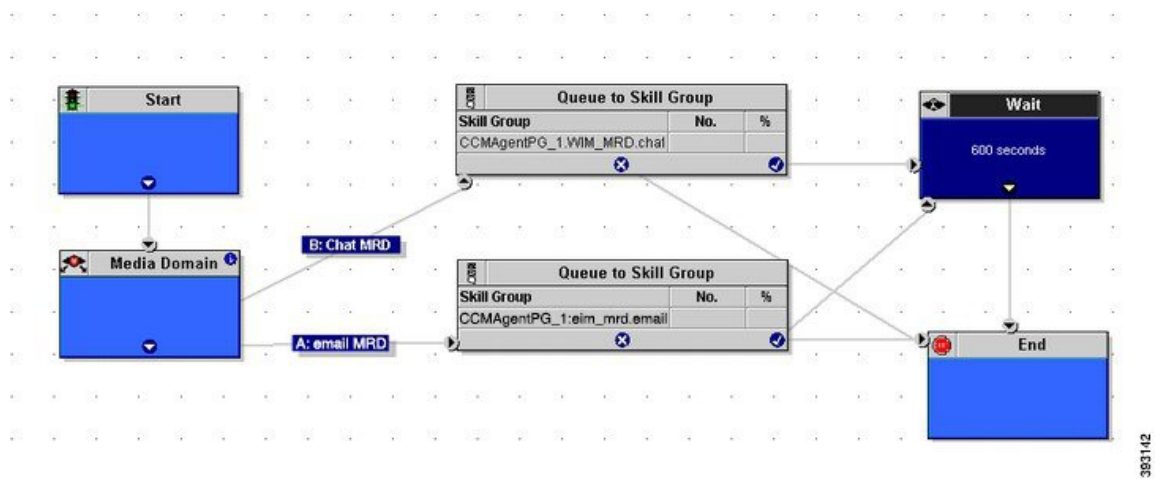
Figure 134: Direct Reference to Agent

3. The script tries to do this for 600 seconds before ending, as defined in the Wait node.

Script That Routes Based on the Media Routing Domain

The following script example shows how you can queue Web requests from different MRDs to different skill groups:

Figure 135: Routing Based on MRD

In this example:

1. The script is run to process a chat request. This script is scheduled to run for all these types of requests.

2. The script first detects the MRD of the request using the Media Routing Domain node. This node has two branches for the MRDs of the request: the email MRD (branch A) or chat MRD (branch B).

3. The script queues the request to the appropriate skill group for that type of request. At this time, the Unified ICM attempts to find an available agent who is a member of that skill group. When the agent is found, the system software returns the agent ID to the Cisco Interaction Manager.

Modify CCE Scripts for Experience Management Voice, SMS, and Email Surveys

In Script Editor, modify your CCE call routing scripts for incoming calls as follows:

Add nodes to invoke the call studio survey script, if needed. The following example explains when you might need to explicitly add nodes to call the survey script.

A script is called that asks callers if they want to participate in a survey. The script then sets the **user.microapp.isPostCallSurvey** variable according to the caller's response.

Note You can use POD.ID to send *cc_CustomerID*, and this *cc_CustomerID* can be used to filter the data in finesse gadget.

POD.ID is an optional field in Experience Management voice survey and the supported POD. ID format will be xxx, where xxx is the Customer_id.

Create Experience Management Routing Script for Voice

Create the following CCE call routing scripts for incoming calls.
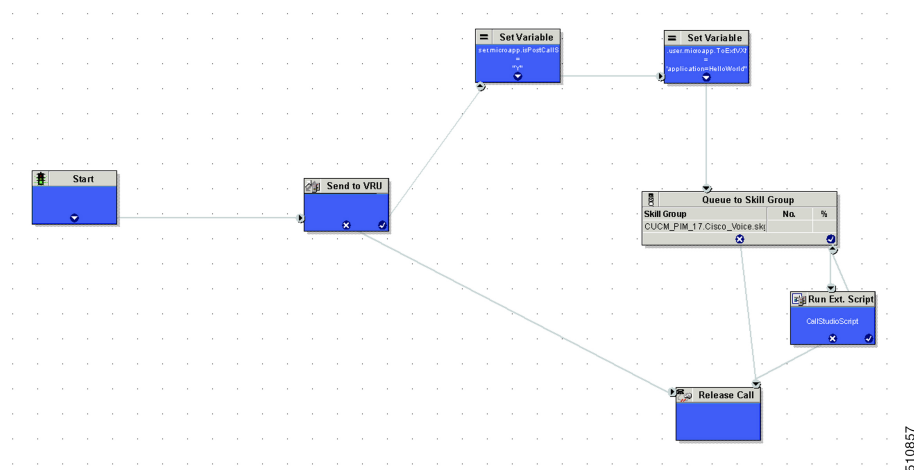
Create the following routing script for the **Experience Management** Call Type to play your survey script or application to the caller.
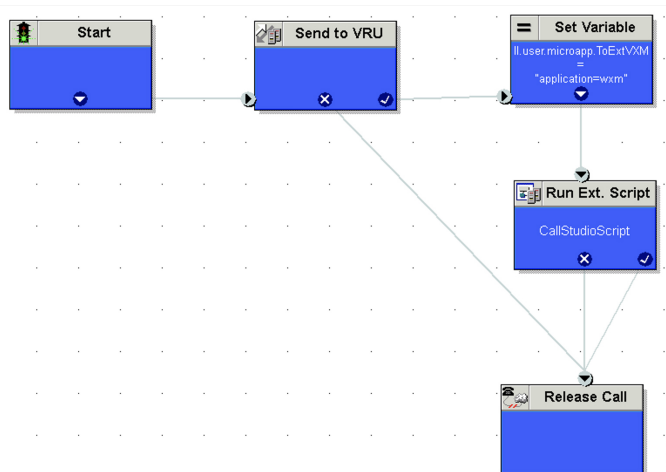
Procedure

Step 1 In Script Editor, create the routing script as shown in the example.

Step 2 Do the following to ensure the routing script is mapped to the call type you created for Experience Management.

   a) Navigate to **Script** > **Call Type Manager**.

   b) On the **Schedules** tab, from the **Call Type** drop-down list, select the call type that you created for Experience Management.

For more information refer to the topic *Call Type and Survey Association in Unified CCE Admin* in *Cisco Unified Contact Center Enterprise Features Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-feature-guides-list.html

   a) Click **Add**.

   b) Click **OK**.

For more information on optimizing the threshold values for SMS and Email batch processing, refer to the topic *Configure SMS/Email Thresholds* in *Configuration Guide for Cisco Unified Customer Voice Portal* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-installation-and-configuration-guides-list.html

Create Experience Management Routing Script for SMS and Email

Experience Management Deferred (SMS/Email) survey is used for getting feedback on the overall customer journey experience. For invoking this survey, the caller's mobile number, email address, and other details have to be collected from Call Studio and passed to ICM. To configure this, refer to the following sections:

Configure Call Studio App Data Format

Procedure

Create Call Studio application with the use of CVP `SubDialogReurn` element and fill the customer data as external VXML variables.
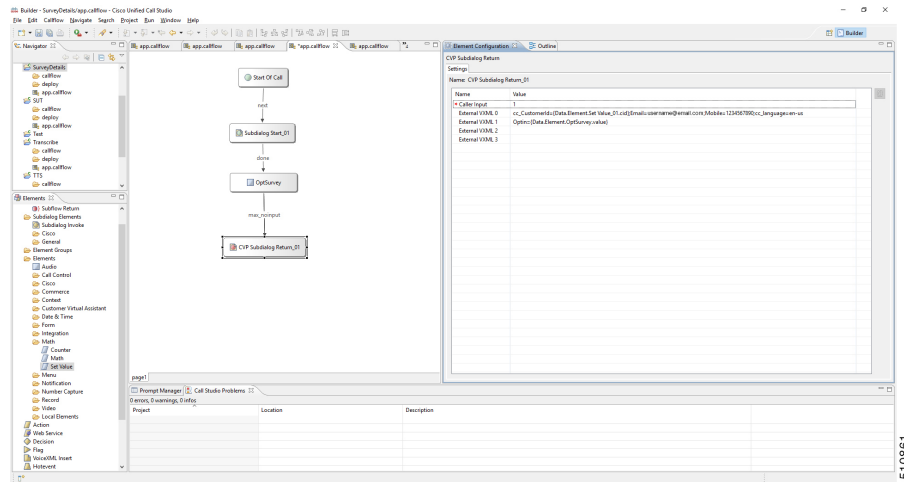
Example: `External VXML 0 = Email=abcd@cisco.com;cc_CustomerId=xyz;cc_language=en-US;Optin=yes;Mobile=911234567890;`

| **Note** | Each key-value pair is separated by a semi colon. |

| **Note** | The variable names are case sensitive. |

Figure 136: Example

Refer to the following table for the variables and their descriptions.

**Table 6: Variables and their descriptions**

Variable Name	Description	Required/Optional
cc_CustomerId	Unique ID for a customer across multiple channels.	Required
Email	Customer email ID	Either email or phone number is required
Mobile	Customer mobile number with country code. Example: 919911223344	Either email or phone number is required
cc_language	Survey language Example: en-US	Optional

| Variable Name	Description	Required/Optional
Optin	Option to check if customer wants to opt for the survey: Values: *yes/no* Default value is *yes* when the value is not captured from Call Studio.	Optional

Configure ICM Script

Procedure

Step 1　Create the ICM script as shown in the following screen shot:
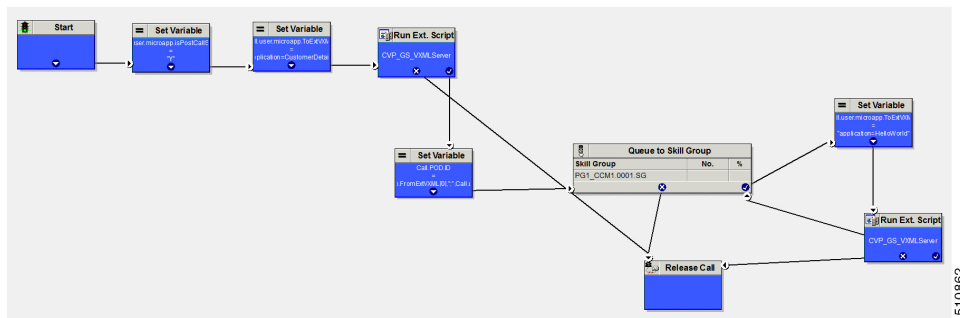
Step 2　Fill the ECC variable **POD.ID** values from `user.microapp.FromExtVXML` array and pass it to ICM.

Example:

```
POD.ID=cc_CustomerId=xyz;Email=abcd@cisco.com;cc_language=en-US;Optin=yes;Mobile=911234567890;
```
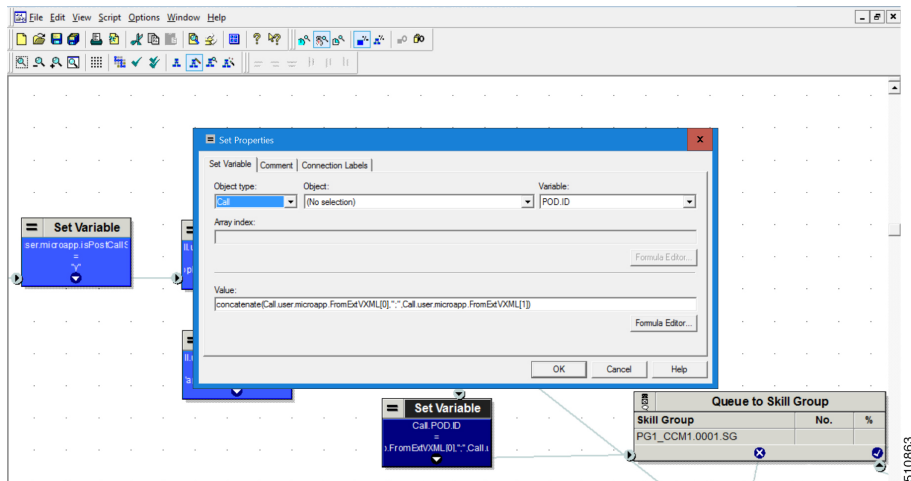

Step 3　Save the script.

Example Enterprise Chat and Email - Email Scripts

You can configure the Unified ICM and the Enterprise Chat and Email so that the system software routes email messages that are processed by Enterprise Chat and Email.

For more information about configuring Unified ICM and Enterprise Chat and Email in an integrated environment, see the following documents:

- *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*

- *Cisco Unified Web and E-Mail Interaction Manager Installation Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-email-interaction-manager/products-installation-guides-list.html

- *Cisco ICM Multichannel Software Implementation Map*

- *Cisco ICM Multichannel Software Overview*

Overview of ECE E-mail Routing Through Unified ICM

When theECE is in an integrated environment, the administrator can choose to defer message assignment to Unified ICM. The administrator determines if skill groups can initiate message routing through the system software when the skill groups are created. The administrator then defines rules to assign messages to those Unified ICM Routing skill groups.

In this situation:

1. A message enters the ECE.

2. A rule assigns the message to a Unified ICM Routing skill group.

3. ECE sends a request for assignment to Unified ICM.

Note   In a Unified ICM integrated with the ECE, the allowed responses for an invoked route request are: Return an Agent, Return a Label, or Failure (the script could not be found or something is wrong). The ECE deals with this by placing the e-mail message into the externalRoutingError system queue.

4. The Router runs a routing script to determine which ECE agent to assign the message to. The routing script can also determine that the message should be assigned to a local ECE routing skill group.

Note   To assign a message directly to an agent, the Unified ICM script must use the Queue to Agent node to route e-mail messages, not the Agent node.

Note   While the system software can assign a message directly to a local ECE routing skill group, you would not typically design a script to do this without first trying to queue the message to a skill group or directly to an agent. If the message, based on its content, is meant to be assigned to a local ECE routing skill group, you could make that assignment directly through ECE rules, and not use the software.

5. Based on the routing script's determination, the Unified ICM instructs the ECE to assign the message to a particular agent or ECE routing skill group.

6. The message is placed in the agent's or skill group's queue.

7. If the message is assigned to an agent by the Unified ICM, it is presented to the agent in push mode; if the message is assigned to an ECE routing skill group, it is placed in that skill group's queue.

Script That Queues E-mail to a Skill Group

The following script example shows how an e-mail message can be queued to a skill group:

Figure 137: Queuing E-mail to Skill Group
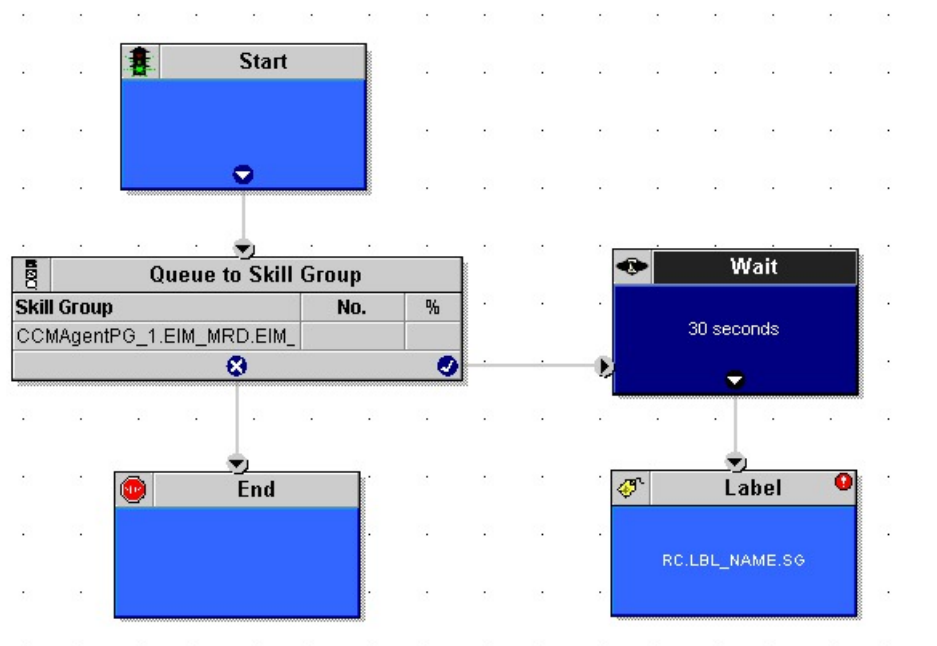
In this example:

1. The script is run when a e-mail message is assigned to a Unified ICM Routing skill group. This script is scheduled to run for messages assigned to that skill group. For more information about call types and scheduling e-mail routing scripts, see Determination of Call Type for ECE E-mail Contact.

2. The script queues the request to the skill group. At this time, the Unified ICM attempts to find an available agent who is a member of that skill group. When the agent is found, the system software instructs the Enterprise Chat and Email to push the e-mail message to that agent, so the agent can respond to the customer.

3. If no agent is found within 30 seconds, as defined in the Wait node, the script uses the Label node to return a label associated with the Enterprise Chat and Email local skill group. The Enterprise Chat and Email then places the message in that local skill group. Enterprise Chat and Email load balances the e-mail assignments to the available agent, based on load balancing.

Note	The ICM wait node can be used to keep the email in the queue when the agent is not available (for example, during a weekend) by setting the wait node to a high value such as 172800 seconds (48 hours). The maximum time that an activity can remain in the ICM queue is determined by the MRD setting "Max time in queue". By default, this setting is blank for all MRDs and the setting defined in the router registry is used.

Script That Routes a Message Based on the Priority

The following script example shows how you can queue an e-mail message directly to an agent or to a skill group, depending on the message's priority:

Figure 138: Routing Based on Priority
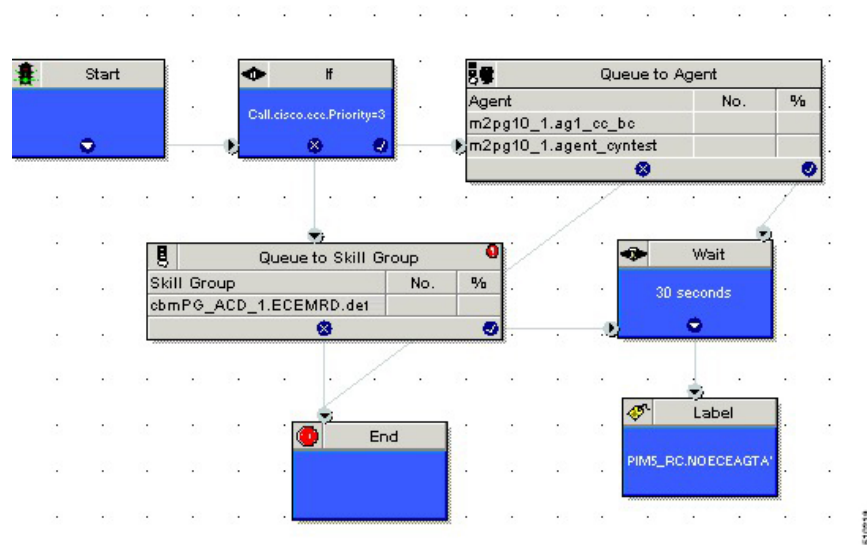
In this example:

1. The script is run when a e-mail message is assigned by the Enterprise Chat and Email to a Unified ICM Routing skill group. This script is scheduled to run for messages in that skill group. For more information on call types and scheduling e-mail routing scripts, see Determination of Call Type for ECE E-mail Contact.

2. The script tests the message's priority, the value of the cisco.ece.Priority variable. For more information, see Data for Enterprise Chat and Email E-mail Requests.

3. If the message is marked "Urgent" (cisco.ece.Priority value of 3), control passes to the Queue to Agent node. That node lists two agents who are e-mail supervisors and who handle urgent messages. The Enterprise Chat and Email then pushes the message to the first of these agents who is logged in.

4. If the message is not urgent, control passes to the Queue to Skill Group node. The Unified ICM attempts to find an available agent who is a member of that skill group. When the agent is found, the Unified ICM instructs the Enterprise Chat and Email to push the e-mail message to that agent.

5. If no agent is found within 30 seconds, in either the Queue to Agent node or Queue to Skill Group node, as defined in the Wait node, the script uses the Label node to return a label associated with a Enterprise Chat and Email local skill group. The Enterprise Chat and Email then places the message in that local

skill group. The message is not pushed to an agent; rather, it waits in the local skill group queue for an agent to pick it.

> **Note**
> These Expanded Call Context (ECC) variables are optional. Before you use these ECC variables, configure them in Unified CCE and ECE.
>
> To configure these ECC variables in Configuration Manager's Expanded Call Variable List Tool, see the *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html.
>
> To configure these ECC variables in ECE, see at https://www.cisco.com/c/en/us/support/customer-collaboration/cisco-enterprise-chat-email/products-maintenance-guides-list.html.

Script That Routes Requests Based on Pick/Pull/Push

The following script example shows how you can use a single script to route a push request or a pick or pull request using an IF node:

Figure 139: Routing Based on Push/Pull/Pick
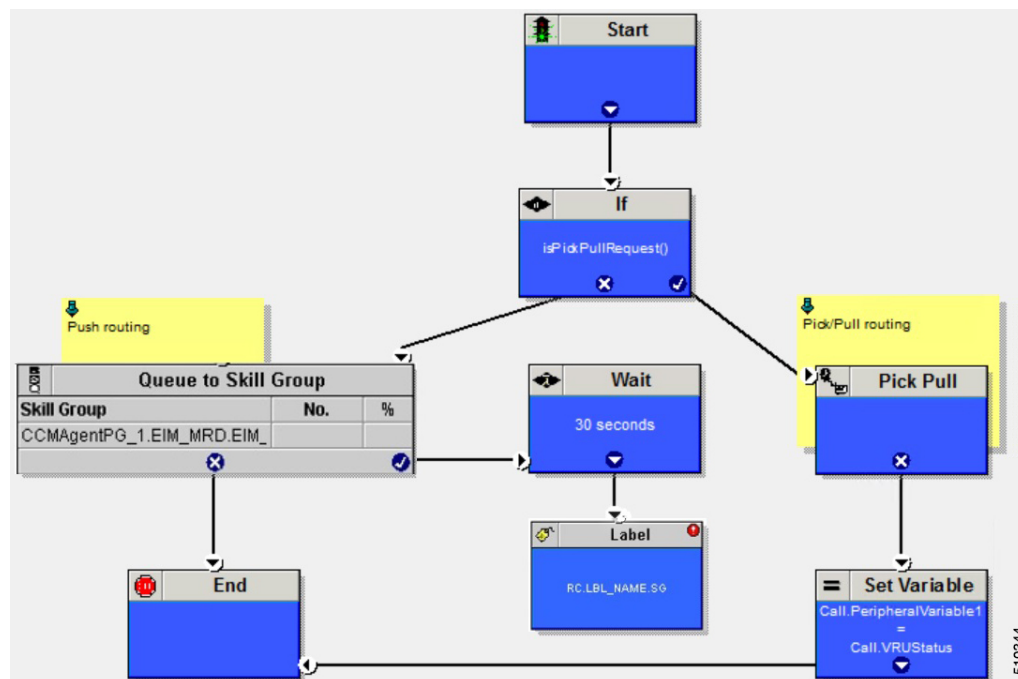
In this example:

1. The script is run when the Enterprise Chat and Email assigns an email message to a Unified ICM Routing Skill Group.

2. The script checks whether the service requested is one of the Pick or Pull service type using the builtin function – isPickPullRequest(). The success branch of **IF** node connects to Pick/Pull node and the failure

branch connects to a queue node, which can be either a Queue to Skill Group node or Precision Queue node.

3. If the **IF** node evaluates to False, control passes to path of Push routing. The script queues the request to the skill group. At this time, the Unified ICM attempts to find an available agent who is a member of that skill group. When an agent is found, the system software instructs the Enterprise Chat and Email to push the e-mail message to that agent, so the agent can respond to the customer.

4. If no agent found within 30 seconds, as defined in the **Wait** node, the script uses the Label node to return a label associated with the Enterprise Chat and Email local skill group. The Enterprise Chat and Email then places the message in that local skill group. Enterprise Chat and Email load balances the e-mail assignments to the available agent, based on load balancing.

5. If the **IF** node evaluates to True, control passes to Pick / Pull routing. Based on the pick or pull service requested, the request will be handled in the Pick/Pull node.

6. Pick / Pull node can be used directly without an **IF** node which is shown in the example. In such case, the failure path can be used for Push routing. In addition, the reason for the failure through the Pick / Pull node is captured in the VRUStatus variable which can be set in one of the peripheral variables to be captured in the Route Call Detail record.

Example Digital Channel Interactions Using Webex Connect

You can configure Contact Center Enterprise (CCE) and Webex Connect so that the system software routes the digital channel tasks that are processed by Webex Connect.

For more information about configuring CCE and Webex Connect in an integrated environment, see the *Digital Channels Integration Using Webex Connect* chapter in the following documents:

- *Cisco Unified Contact Center Enterprise Features Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-feature-guides-list.html

- *Cisco Packaged Contact Center Enterprise Features Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/series.html#%7Etab-documents

Overview of Digital Channel Routing using Webex Connect

In an integrated environment where Webex Connect is used to route digital channel contacts through the Digital Routing service, an administrator can set up a routing script that can either assign tasks directly to specific agent or to Skill Groups / Precision Queues across different media channels or media routing domains.

In this scenario:

1. A digital channel task enters Webex Connect and is routed to Unified CCE through Digital Routing service.

2. The Webex Connect platform uses channel-specific flows to pass task context to the Digital Routing service through call variables and ECC variables. Optionally, a Flow designer can set the **Preferred Owner** field in the **CCE Create Task** node in Webex Connect with the Agent Skill Target ID or Peripheral Number of an agent, who should be queued or routed to, if available. The value passed in the flow is populated as the call variable, Call.PreferredAgentID, which can then be used to queue the task using a QueueToAgent node in the routing script.

3. The Digital Routing service sends a request for assignment to Unified CCE.

4. The Router runs a routing script to determine which non-voice or digital channel agent to assign the task to. The routing script can also determine that the task should be assigned to a Digital Routing skill group or precision queue.

5. Based on the routing script's determination, the Unified CCE instructs the Digital Routing service to assign the task to a particular agent.

6. If the desired agent is not available, or all agents are busy in the Skill Group or Precision Queue, the task gets queued in the CCE Router. An administrator can choose to send an estimated wait time and additional task context that are carried by ECC variables through "Run External Script" node. The Digital Routing service sends the "QUEUED" webhook notification event, which invokes a "QUEUED" flow in Webex Connect to relay the estimated wait time and additional context to the customer who is active on a live chat session or on asynchronous social channels such as SMS.

Script that Queues a Digital Channel Task to a Skill Group

The following script example shows how a digital channel task can be queued to a skill group. In this example there are two scenarios:
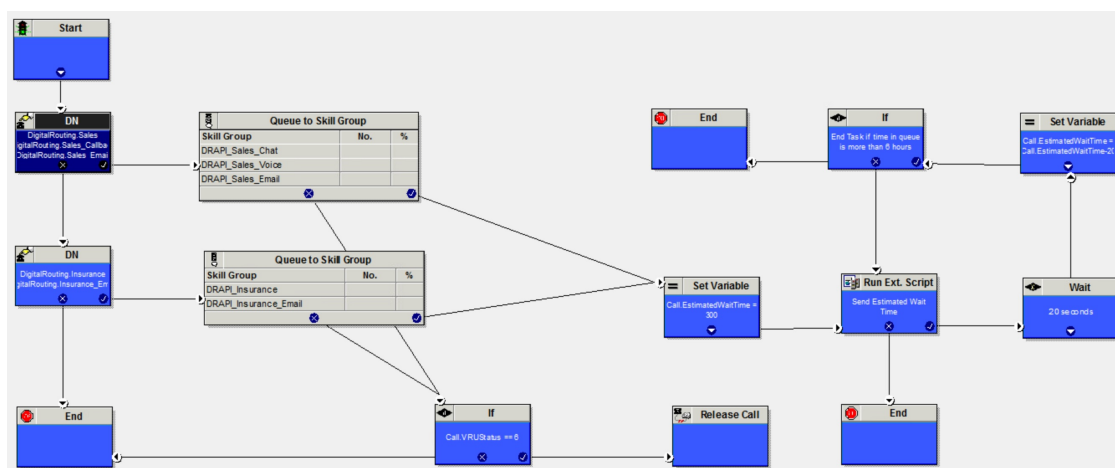
Scenario 1

The task is queued and routed to an agent based on the wait time defined.

1. The routing script decides the skill group from where an agent must be picked. The same script can be used to route tasks belonging to different media channels such as Email, Chat, SMS or Web Callback (which maps to Cisco_Voice MRD) and unique Script Selectors / Dialed Numbers. In this example, the script queues the task to "Sales" or "Insurance" skill groups corresponding to Chat and Email MRDs based on the Script Selectors / Dialed Numbers received in the route request from Webex Connect. For more information about how Unified CCE determines the Call Type for a task routed through digital channels, see Determination of call type for a Digital Routing task.

2. The script queues the task request to the skill group. At this time, the Unified ICM attempts to find an available agent who is a member of that skill group. The Unified ICM runs an external script to notify the customer an estimated wait time. When the agent is found, the system software instructs the Digital Routing API service to push the task to that agent, so the agent can respond to the customer.

3. In the example script, a loop is run to send periodic updates to the Digital Routing service about the Estimated Wait Time (EWT). An EWT is simulated by assigning it a 5 minutes (300 seconds) static value and decrementing the count in a loop, however, a custom formula can be used to relay the same based on your requirements. If the task is queued for more than 6 hours, the script bails out and ends the queueing.

The CCE wait node can be used to keep the task in the queue when the agent is not available (for example, during a weekend) by setting the wait node to a high value such as 172800 seconds (48 hours). The maximum time that an activity can remain in the ICM queue is determined by the MRD setting "Max time in queue". By default, this setting is blank for all MRDs and the setting defined in the router registry is used. The default value in the router registry is 3600 seconds.

Scenario 2

The router rejects the queuing request as there are no queue slots available.

1. When queueing the task to the skill group, the router checks the following:

   - The global limit of maximum queued calls and maximum queued calls per call type as defined in the router registry under the following registry hive: `HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\ICM\InstanceName\RouterA\Router\CurrentVersion\Configuration\Queuing`

     The "MaxCalls" denotes the maximum queued calls, which is set as 15,000 calls or tasks and the "MaxCallsPerCallType" denotes the maximum queued calls per call type, which is set as 10,000.

   - Whether the number of tasks queued in the skill group has exceeded the maximum number of tasks defined in the MRD for the skill group.

2. If it has exceeded the maximum number of tasks defined, then the script checks the VRUStatus call variable in the failure path of the Queue node.

3. If the value of the VRUStatus call variable is 6, that is STATUS_MAX_QUEUE_LIMIT_EXCEEDED, the script sends the call to the release node.

4. The router populates a special CauseCode as part of the Release node processing that instructs the Digital Routing service to queue the task at the head of the queue, such that the task routing does not fail. The Digital Routing service tries to queue the task again in a periodic manner, as queue slots become available.

Example Universal Queue Scripts

You can design scripts to route contacts from different media in a Universal Queue environment.

These scripts are only examples; your company's needs may differ.

For more information about configuring Task Routing with third-party multichannel applications see the *Cisco Unified Contact Center Enterprise Features Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-feature-guides-list.html.

For more information about configuring Enterprise Chat and Email, see the following documents, at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/tsd-products-support-series-home.html:

- *Configuration Guide for Cisco Unified ICM/Contact Center Enterprise*

- *Cisco Unified Contact Center Enterprise Installation and Upgrade Guide*

- *Administration Guide for Cisco Unified Contact Center Enterprise*

Selection of Agents from Skill Groups

The following script example shows how contacts from different channels can be routed to the Longest Available Agents in skill groups that are specific to the different channels:

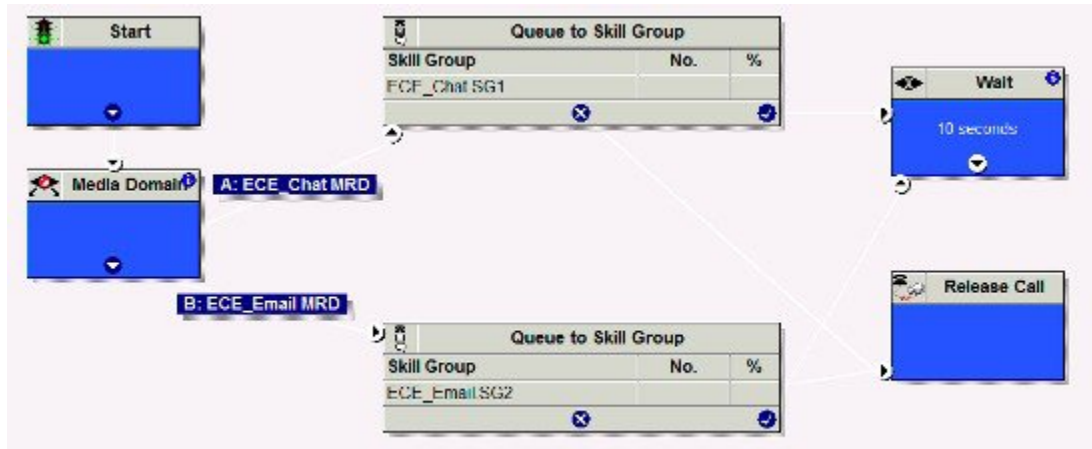*Figure 140: Selecting Agents from Skill Groups*
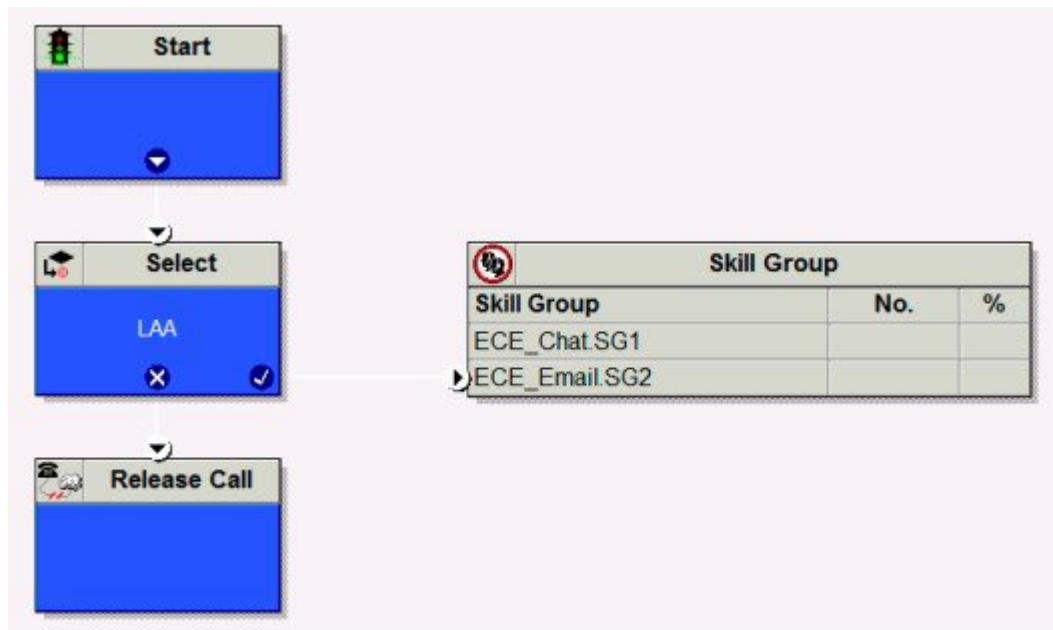
You schedule this script to run for Call Types associated with contacts from the different channels. The script then selects the Longest Available Agent from the skill group in the Media Routing Domain for that channel. The agents may be logged in to different Media Routing Domains and working with contacts from different channels; the Router determines an agent's availability across channels.

Categorization by Media Routing Domain with Skill Groups

The following script example shows how contacts can be categorized by Media Routing Domain, then queued to skill groups specific to that Media Routing Domain:

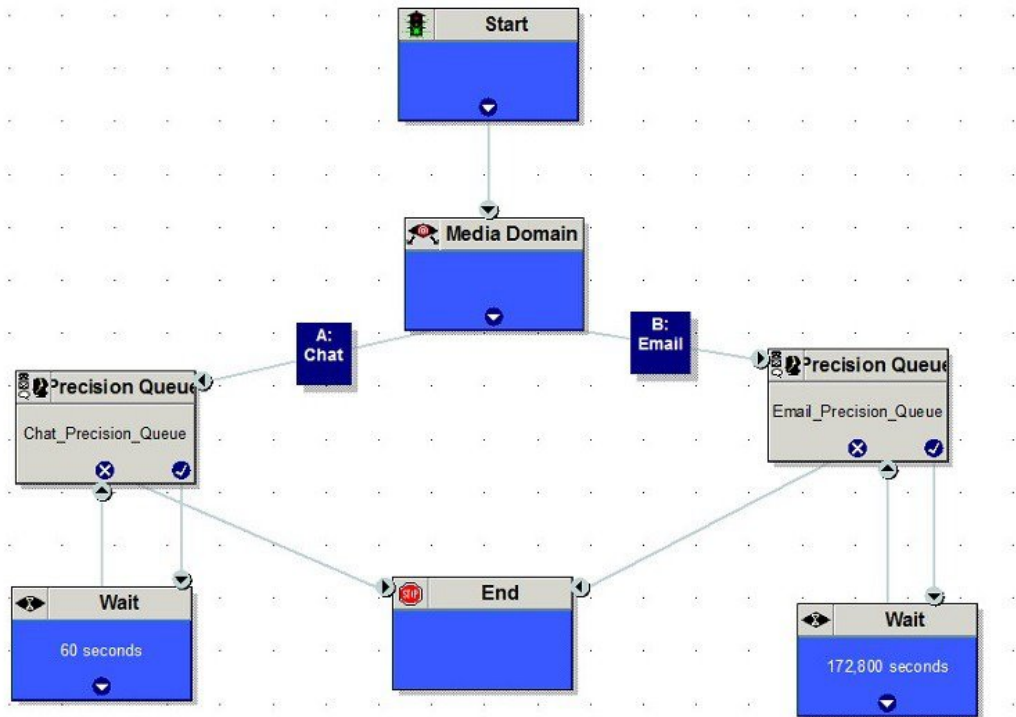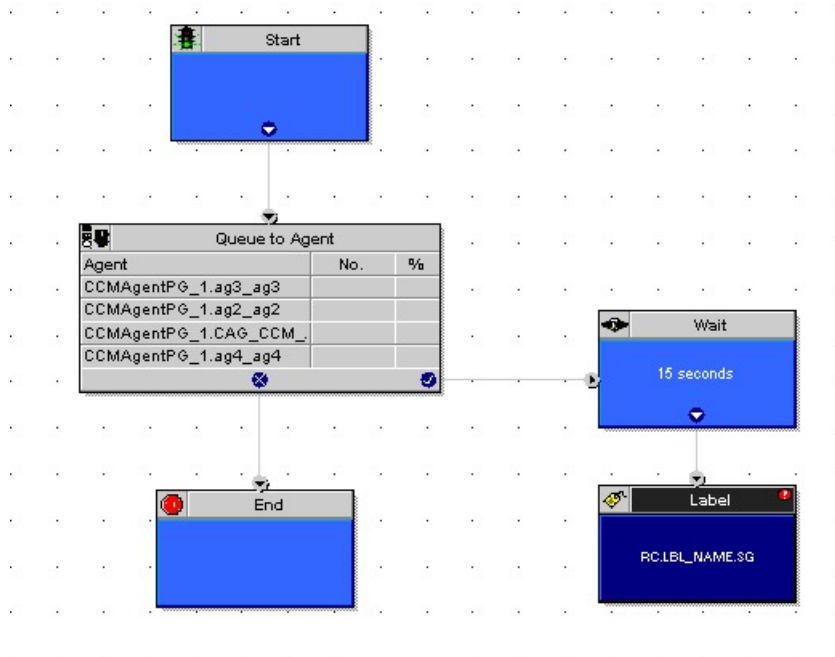*Figure 141: Categorizing by MRD with Skill Groups*

You would schedule this script to run for Call Types associated with contacts from the different channels. The script then uses the Media Routing Domain node to detect the MRD of the contact and branches to a Queue to Skill Group node that specifies skill groups specific to that MRD.

Categorization by Media Routing Domain with Precision Queues

The following script example shows how contacts can be categorized by Media Routing Domain, then queued to precision queues specific to that Media Routing Domain:

Figure 142: Categorizing by MRD with Precision Queues

You would schedule this script to run for Call Types associated with contacts from the different channels. The script then uses the Media Routing Domain node to detect the MRD of the contact and branches to a Queue to Precision Queue node that a precision queue specific to that MRD.

Script That Queues to Agents

The following script example shows how contacts from different channels can be queued to agents:

Figure 143: Queuing to Agents

You would schedule this script to run for Call Types associated with contacts from the different channels. In the Queue to Agent node, each row defined for an agent also contains a Media Routing Domain selection. The script queues the contact to the agent with the selected MRD that matches the MRD of the contact.

RONA and Transfer Script

This example only applies to tasks submitted by third-party multichannel applications that use the Task Routing APIs. This example script shows the call priority increase if the service requested is 2 (TRANSFER).

Figure 144: Example RONA and Transfer Script
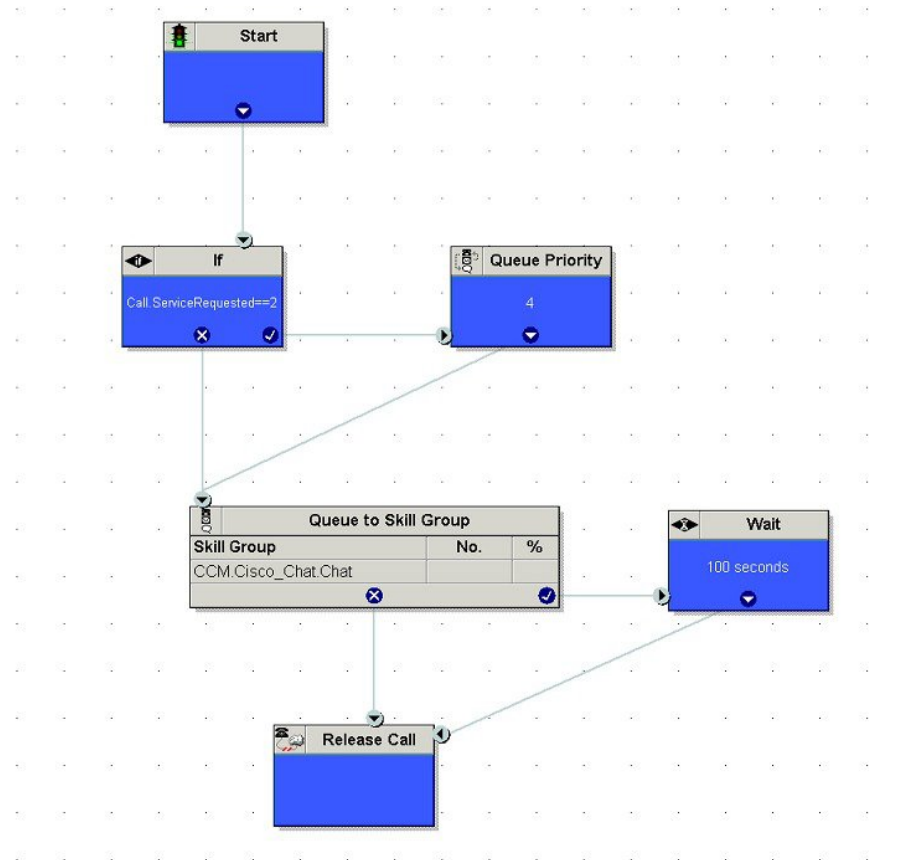
If the Call ServiceRequested call variable is set to 2 (TRANSFER) the call enters the Queue Priority node. The Queue Priority of the call is increased so that it is handled before any other calls in the queue. The Queue Priority node sends the call to the Queue to Skill Group node. If the Call ServiceRequested call variable is not set to 2 (TRANSFER), the call enters the Queue to Skill Group node.
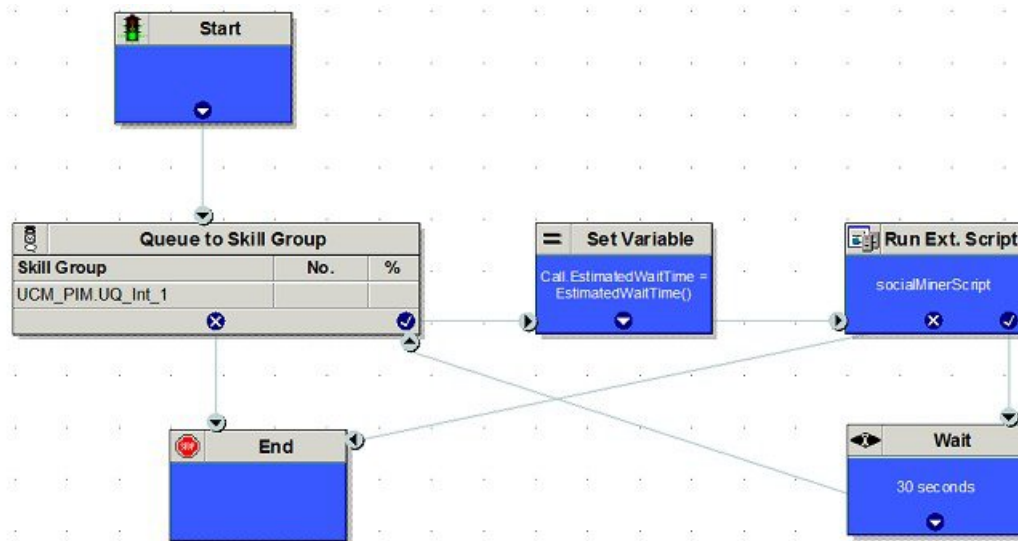
Estimated Wait Time Script

This example only applies to tasks submitted by third-party multichannel applications that use the Task Routing APIs.

Scripts for estimated wait time include:

  • A Set Variable Node to set the estimated wait time.

  • A Run External Script node to apply a Network VRU script that returns the estimated wait time to the customer.

Figure 145: Example Estimated Wait Time Script

Set Variable (Call.Estimated Wait Time) node: Set the estimated wait time as follows:

1. From the Set Variable node, select **Call** from the Object type drop-down menu.

2. From the Variable drop-down menu, choose **Estimated Wait Time()**.

   You can then work with the Formula Editor to use the default estimated wait value or create a formula and use your own value.

3. Click **Formula Editor**, and do either of the following:

   • To use the default estimated wait value, click the Built-In Functions tab and choose EstimatedWaitTime()

   • To create a formula and use your own value, click the Variables tab and choose an entry in the Object type list and an entry in the Object list. Then double-click a variable in the Variable list.

Run Ext Script node: Apply the Network VRU script as follows:

1. Click the Queue tab.

2. Click **Run External Script**.

3. Click inside the script. A Run External Script node appears.

4. Double-click the node and choose the Network VRU script from the list; then click **OK**.

   The call variable Estimated Wait Time now contains a value in the EstimatedWaitTime field and can be passed to peripherals.

   Note that a Run External Script node is required to send the EstimatedWaitTime to Customer Collaboration Platform.

Example Unified CCE Scripts

Following are example scripts for use when the Unified ICM is part of a Unified CCE environment.

- Redirection on Ring No Answer

- Agent Transfer

These scripts are only examples; your company's needs may differ.

For more information about configuring Unified ICM as part of a Unified CCE environment, see the following documents:

- *Cisco Unified Contact Center Enterprise Installation and Upgrade Guide*

- *Administration Guide for Cisco Unified Contact Center Enterprise*

✎

Note   Additional Unified CCE example scripts are available in the *Cisco Unified Contact Center Enterprise Reporting User Guide* at
https://www.cisco.com/en/US/products/sw/custcosw/ps1844/tsd_products_support_series_home.html.

Redirection on Ring No Answer

When configuring the Unified ICM in a Unified CCE environment, you configure Agent Desk Settings. When creating agents, you then associate each agent with one of the Agent Desk Settings you created.

One attribute of Agent Desk Settings is the **Ring no answer dialed number:**

Figure 146: Reroute on Ring No Answer

For calls to be routed when an agent does not answer, you must create and schedule a script for the Call Type mapped to the Dialed Number selected for the agent's Desktop Settings.

For example, you may schedule a simple script to run when agents do not answer that tries to select the longest available agent from a set of skill groups, and if that fails, requalifies the call to a new Call Type to have it rerouted:

Figure 147: Reroute on Ring No Answer Script
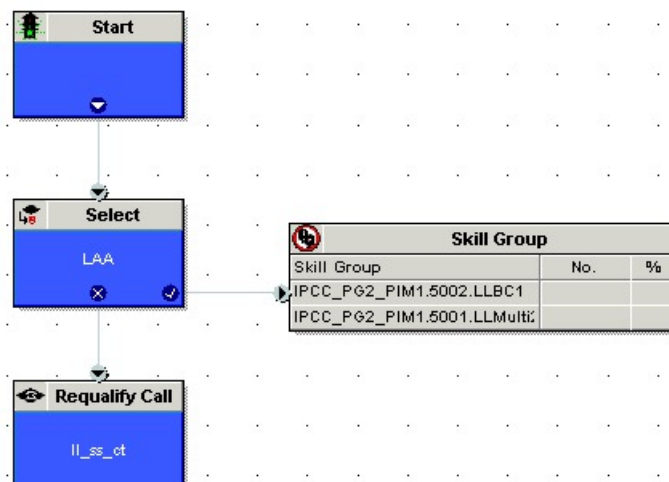
Agent Transfer

When configuring Call Types and dialed numbers in a Unified CCE environment, you typically may have a Dialed Number of *Routing_client*.9999 for internal calls from agent to agent. You would create a call type associated with the Dialed Number, and schedule a script for calls of this call type. The script routes internal calls, which also allows you to track and report on such calls.

For example, you may schedule a simple script to run for internal calls that tries to route directly to the agent using the Agent to Agent node, which selects the agent by peripheral and the expression Call.CallerEnteredDigits:

If the node fails, then the script tries to select the longest available agent from the set of supervisor skill groups:

Figure 148: Agent to Agent Node Script
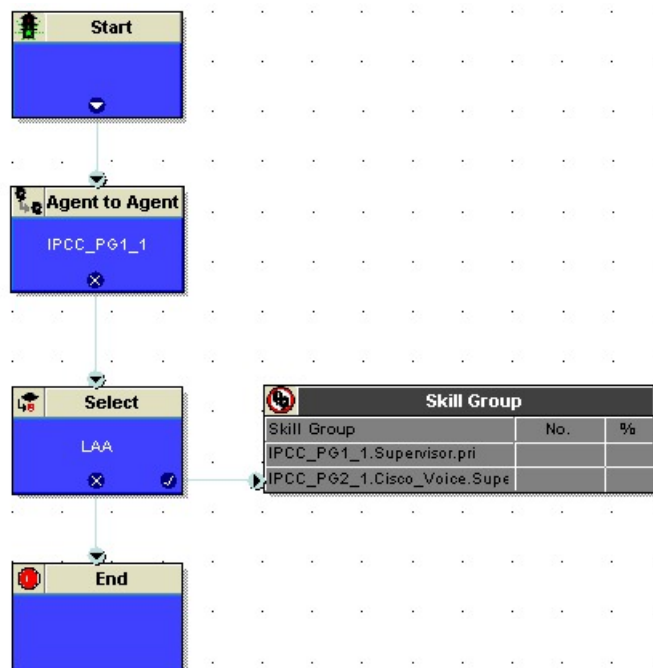
Additional Example Outbound Option Scripts

Following are example scripts for use when the Unified ICM is part of an Outbound environment:

- Setting the OutboundControl Variable and Skill Group Reservation Percentage
- Using the Dialed Number for the MR Routing Client and Routing through a Select Node to a Skill Group
- Transfer to IVR Using Outbound Option with Unified IP IVR
- Transfer to IVR Using Outbound Option with Unified CVP

These scripts are only examples, your needs may differ.

Control of OutboundControl Variable and Skill Group Reservation Percentage

You can use an administrative script to control the setting the OutboundControl variable and the skill group reservation percentage. The Outbound Option Dialer uses these values to determine which mode each skill group uses. You can use one script to control all Outbound Option skill groups, or multiple scripts can control multiple Outbound Option skill groups.

This administrative script comprises of Start, Set, If , and End nodes. Use the Set node to set skill group variables (OutboundControl and OutboundPercent).

Set the OutboundControl variable by entering it in the Value field of the Set Properties window:

- INBOUND: Indicates that this skill group is disabled for outbound use and only takes inbound calls.

- PREDICTIVE_ONLY: Dials several customers per agent. After reaching a live contact, the Predictive Dialer transfers the customer to a live agent along with a screen pop to the agent's desk. The predictive algorithm is designed to calculate the number of lines to dial per available agent to keep agent wait time to a minimum.

- PREDICTIVE_BLENDED: Agents receive inbound calls, but could be used for an outbound call when available.

- PREVIEW_ONLY: Reserves an agent prior to initiating an outbound call and presents the agent with a screen pop. The agent might then:

  Accept the call:

    - Dials the customer and transfers the call to the agent.

  Skip the call:

    - Agent receives another customer call.

  Skip-Close the call:

    - Skips the current preview call and closes the record so it will not be called again.

  Reject the call:

    - Releases the agent. At this point, the system might deliver the agent another preview outbound call or a new inbound call.

  Reject-Close the call:

    - Rejects the current preview call and closes the record so it will not be called again.

- PREVIEW_BLENDED: Agents receive inbound calls, but could be used for an outbound preview call when available.

- PREVIEW_DIRECT_ONLY: Agents can only place outbound calls and hear ring tones, such as phone ringing or busy signal.

- PREVIEW_DIRECT_BLENDED: Agents can receive inbound calls, place outbound calls, and hear ring tones, such as phone ringing or busy signal.

- PROGRESSIVE_ONLY: Similar to PREDICTIVE_ONLY; however, lines to dial per agent are not calculated—users configure a fixed number of lines that will always be dialed per available agent.

• PROGRESSIVE_BLENDED: Similar to PREDICTIVE_BLENDED, but a fixed number of lines will always be dialed per available agent.

Note   If the administrative script (where the OutboundControl variable or reservation percentage is set) is running, but the modes/percentages are not being updated at the Dialer, do the following:

- Make sure that the skill group being controlled is the base skill group, and not the primary or secondary skill groups. Although agents may be logged in to just the primary or secondary skill group, the outbound control variable must always be set on the base skill group

- Verify that the outbound control variable mode is spelled correctly.

Set the OutboundPercent variable in the same administrative script. Select the OutboundPercent variable in the Set Properties window and enter the agent percentage in the Value field. This variable controls the percentage of agents that are logged in to a particular skill group, which should be used for outbound dialing. For example, if there are 100 agents logged into skill group N, and the OutboundPercent variable is set to 50%, 50 agents would be allocated for outbound dialing.

Note   This variable does not allocate specific agents for outbound dialing, just a total percentage.

The following diagram displays a very simple administrative script where both the OutboundControl variable and the outbound percentage are set for a skill group. A script in a production call center would typically be more complex, perhaps changing these variables due to time of day or service level.

Figure 149: Setting the OutboundControl Variable and Skill Group Reservation Percentage
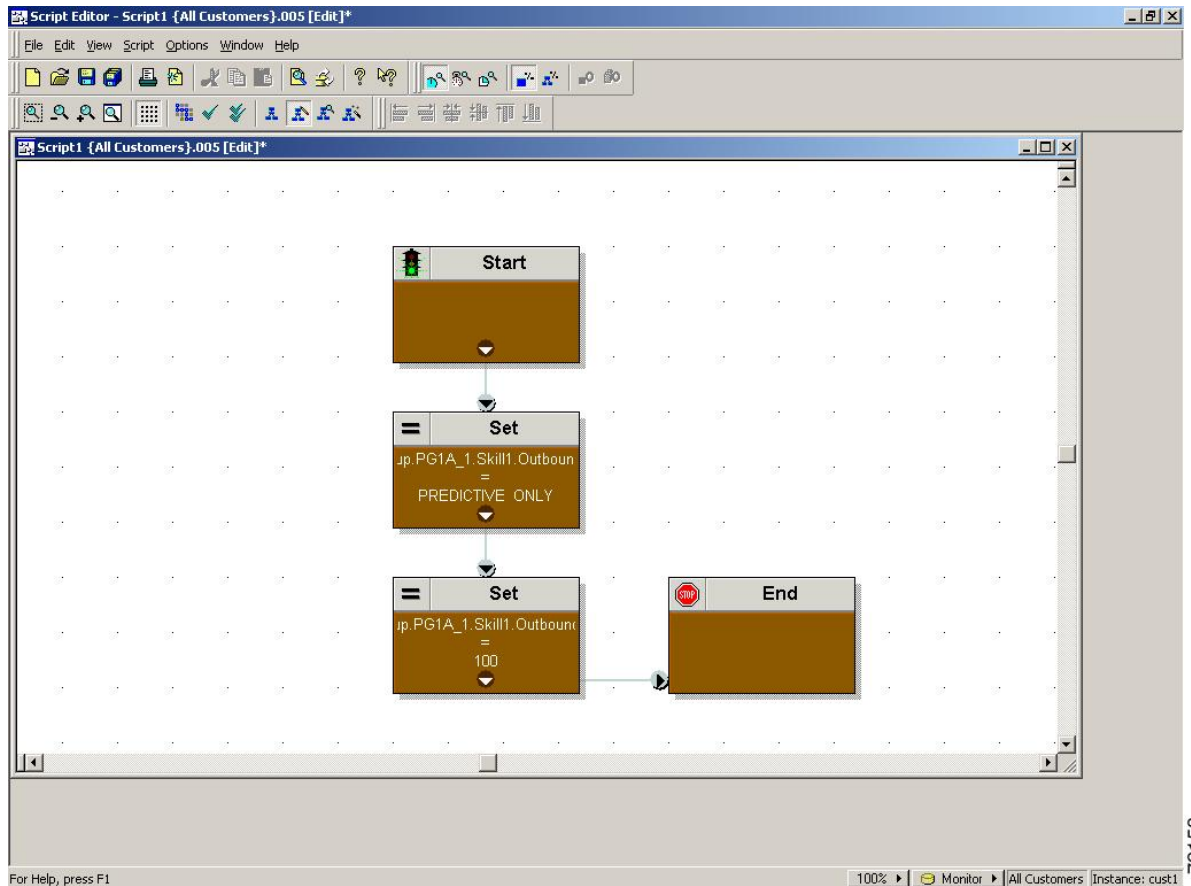
Use Dialed Number for MR Routing Client

The following diagram displays a sample routing script that uses the Dialed Number for the MR routing client and routes through a Select node to a previously configured skill group. Add additional DN nodes to route to agents in additional skill groups as you must maintain a 1:1 ratio of dialed numbers to skill groups.

Warning   You must set the outbound percentage in the Set node to a value other than zero (0) or it appears to the node that there are no agents assigned to the Outbound Option and no outbound calls will be made.

Warning   Do not use Select Route nodes, multiple skill groups, or services.

Note   Translation routes are not used in the Unified CCE System PG, so routing scripts using this PG do not need to use this object.

Configuring Queue to Agent Node

Procedure

|	Command or Action	Purpose
Step 1	Right-click the Queue to Agent node and select **Properties**.	
Step 2	Click **Change** in the "Queue to agent type" section.	
Step 3	Click **Lookup agent reference by expression**, then click **OK**.	
Step 4	Enter the agent expression Call.PreferredAgentID.	
Step 5	Make sure the Peripheral column is left blank.	
Step 6	Select the enterprise skill group.	
Step 7	Select the enterprise route.	
Step 8	Click **OK** to save the Queue to Agent node.	

What to do next

Figure 150: Queue to Agent Node Properties
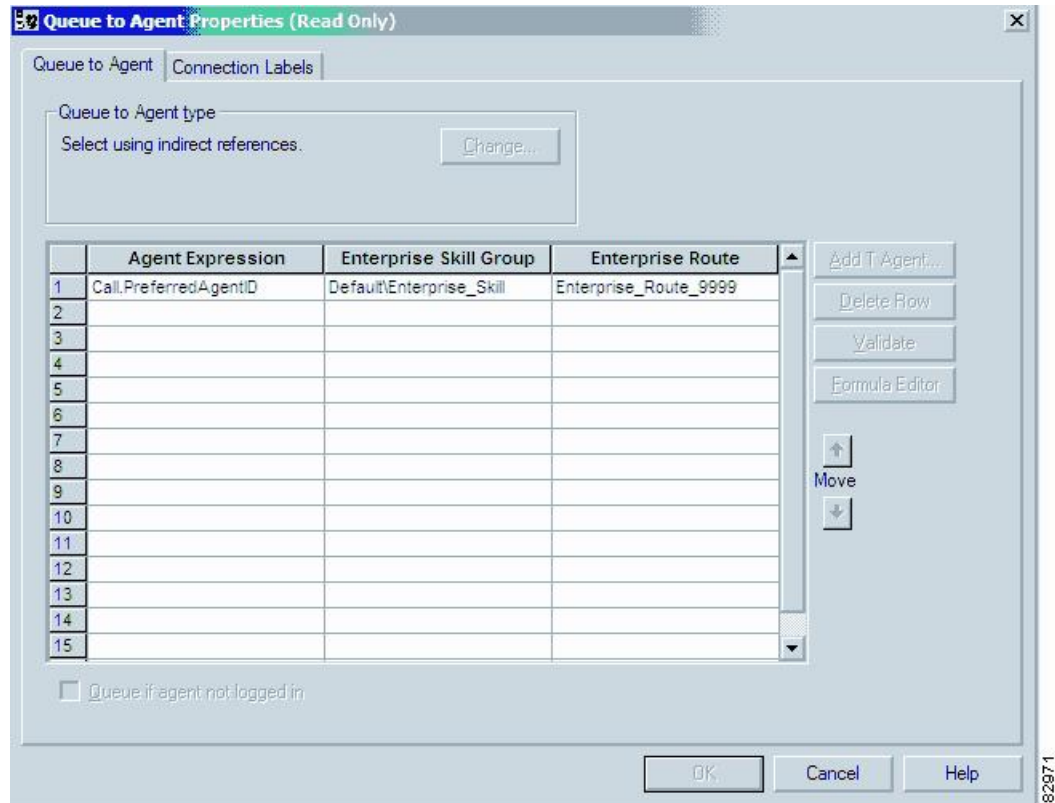
Note Lines connecting objects cannot appear on top of objects and therefore, partially display under the objects. For example, the line connecting the "X" (output terminal failure) on the Select object to the End object runs partially under the Select object.

Figure 151: Using the Dialed Number for the MR Routing Client and Routing Through a Select Node to a Skill Group
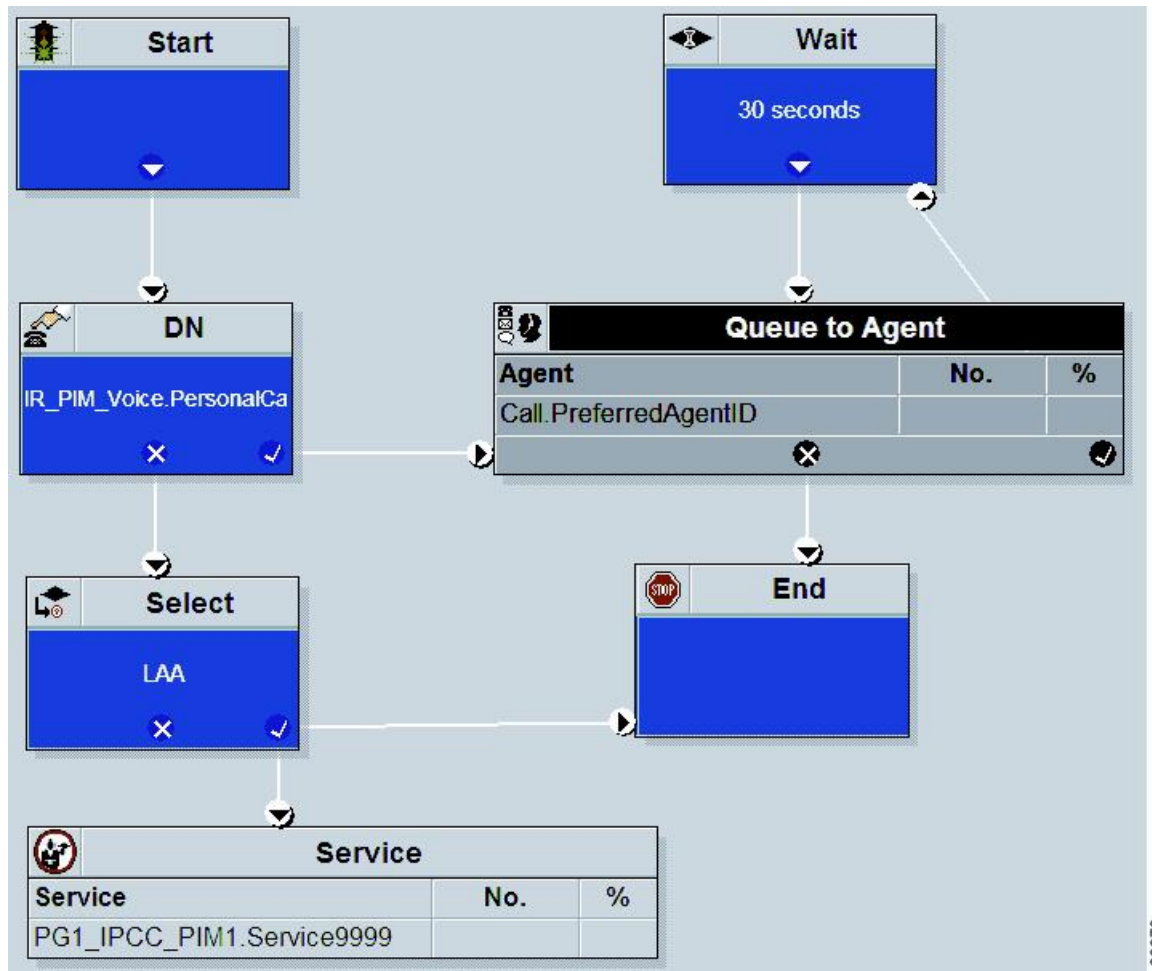
# Transfer to VRU Using Outbound Option with Unified IP IVR

The following diagram displays a routing script for a transfer to VRU campaign using the Outbound Option with the Unified IP IVR.

Figure 152: Transfer to VRU Using Outbound Option with Unified IP IVR
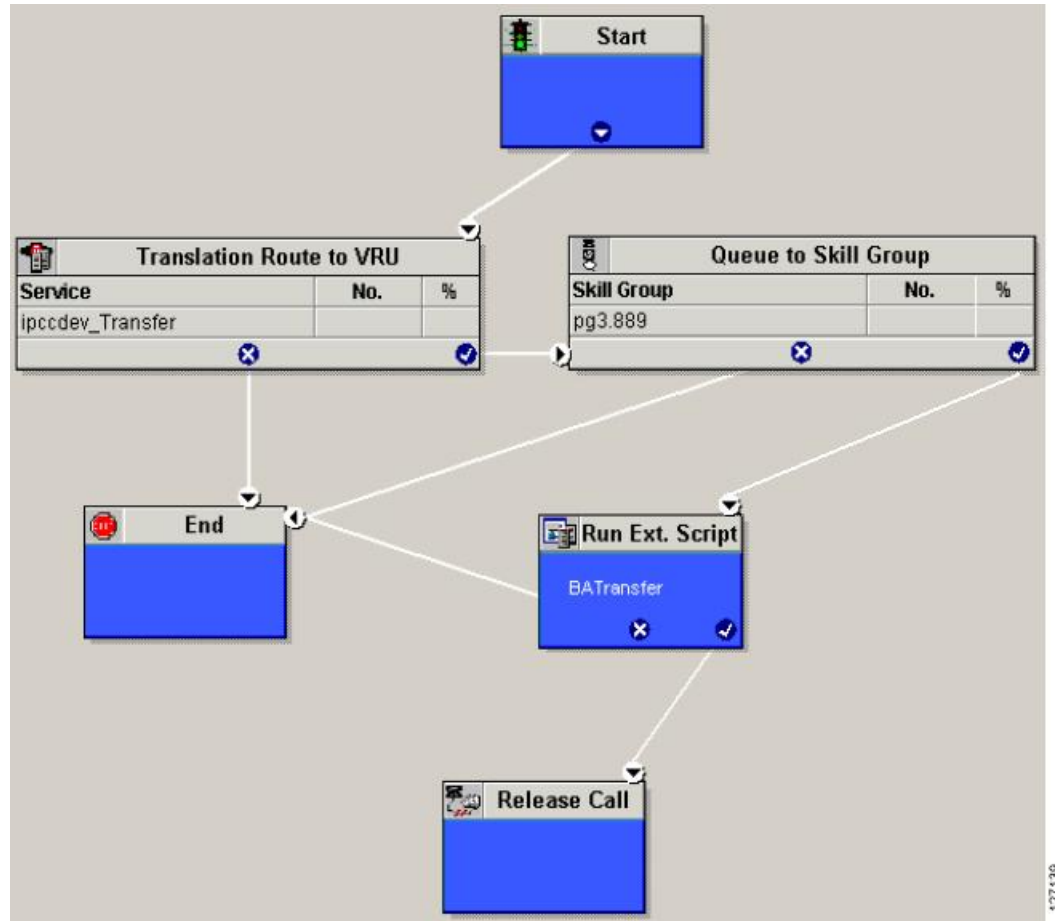
Transfer to VRU Using Outbound Option with Unified CVP

The following diagram displays a routing script for a transfer to VRU campaign using the Outbound Option with the Unified CVP.
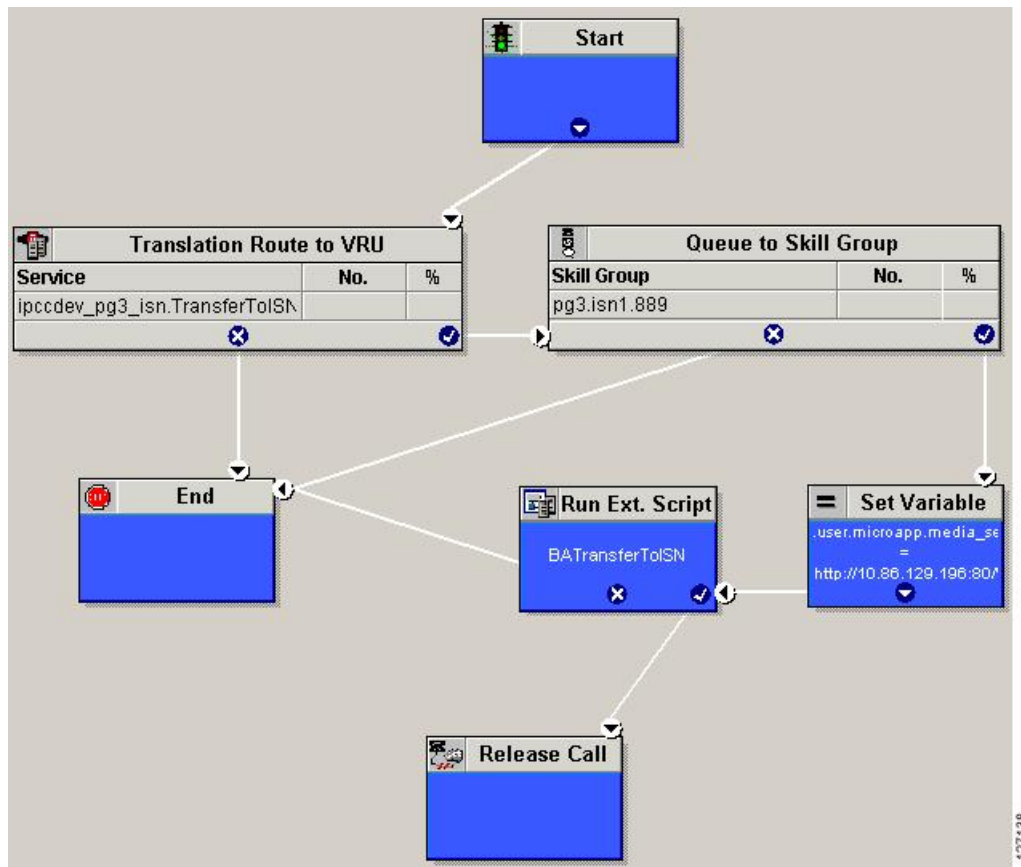
Estimated Wait Time (EWT) Queues

The Script Editor's built-in Minimum Expected Delay (MED) calculation does not apply to the Unified CCE. Instead, you can use the built-in function, EstimatedWaitTime, or write a formula on an IF node to determine the MED between two skill groups.

The default EstimatedWaitTime algorithm is based on a running five minute window of the rate of calls leaving the queue. Any calls which are routed or abandoned during the previous 5 minutes are taken into account as part of the rate leaving queue. For Precision Queues, the rate leaving queue represents the rate at which calls are delivered or abandoned from the entire precision queue, not any individual Precision Queue steps. The algorithm computes the wait time for each of the queues against which the call is queued (Skill Group(s) or Precision Queue(s)) and then returns the minimum estimated wait time. Queue to Agent(s) is not supported. The estimated wait time is calculated once, when the call enters the queue. The time is not updated as the position in the queue changes.

When a queue is just being built, the small number of calls in the queue makes the estimated wait time less accurate and the value fluctuates rapidly. As the queue operates with more calls over time, the estimated wait time is more accurate and consistent.

If the built-in EstimatedWaitTime function does not meet your needs, you can apply your own custom formula.

When to Use EWT Queues

Look for a secondary skillgroup choice in the following situations:

- No available agents in the primary or first choice skill group

- The call has already been sent to a VRU queue point

- The call has been queued to the primary or first choice skill group

- An adequate amount of time has elapsed (customer and call dependent)

After the above conditions have been met, if there are no agents available in the subsequent or secondary choice skillgroup, then Minimum Expected Delay (MED) of the subsequent skillgroup should be calculated.

Example EWT/MED Script Formula

You can use the example formula below in an IF node to determine the EWT/MED for a call that you can route to either a primary or secondary skill group. The call is routed to the skill group with the lowest MED:

```
EntSkill.Default\EnterpriseSkillgroupSec.Avail ||

(EntSkill.Default\EnterpriseSkillgroupPri.LoggedOn>0)

&&

(EntSkill.Default\EnterpriseSkillgroupSec.LoggedOn>0)

&&

((SkillGroup.CCM_PG_1.CCM_SG.sec.RouterCallsQNow+1)*

(ValidValue(EntSkill.Default\EnterpriseSkillgroupSec.AvgHandledCallsTimeTo5,<constant>))

/EntSkill.Default\EnterpriseSkillgroupSec.LoggedOn)

<

((SkillGroup.CCM_PG_1.CCM_SG.pri.RouterCallsQNow+1)*

(ValidValue(EntSkill.Default\EnterpriseSkillgroupPri.AvgHandledCallsTimeTo5,<constant>))

/EntSkill.Default\EnterpriseSkillgroupPri.LoggedOn)
```

EWT/MED Script Explained

If there are agents available, then no there is no need to calculate the MED and the IF node should return a TRUE value. If there are no agents available, then the MED of the secondary skillgroup should be calculated.

The formula first checks to make sure agents are logged in to both the primary (already queued to) and secondary (not queued to) skill groups. If there are agents logged in to both skill groups, then the minimum expected delay of the secondary skillgroup is compared to the minimum expected delay of the primary skillgroup to see which is less.

If the secondary skillgroup MED is less, then the call should be queued at the new secondary skillgroup. (The call can either be kept in the initial skillgroup or taken out of the initial skillgroup when it is queued to the secondary skillgroup.)

However, if the secondary skillgroup's MED is not less than the primary skillgroup's MED, then the call should remain queued at the initial skillgroup only. +1 is used to add this new call into the algorithm.

Because the `RouterCallsQNow` variable is only applicable at a Skillgroup and not an Enterprise Skillgroup level, then the aggregate of all the skill groups within the Enterprise Skillgroup `RouterCallsQNow` field must be multiplied by the Enterprise SkillGroup `AvgHandledCallsTime`.

The `<constant>` value is site dependent. It should be a value that represents the average Handle time of a call. This constant is only used if it is at the start of the day and the proper `AvgHandledCallsTime` has not been calculated.