Avaya Aura® Configuration—Orchestration Designer Application Development

Release 6.2
NN44400-510
03.04
30 July 2012

© 2012 Avaya Inc.

All Rights Reserved.

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

“Documentation” means information published by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya generally makes available to users of its products. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of documentation unless such modifications, additions, or deletions were performed by Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked Web sites referenced within this site or documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on its Hardware and Software (“Product(s)”). Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya’s standard warranty language, as well as information regarding support for this Product while under warranty is available to Avaya customers and other parties through the Avaya Support Web site: http://support.avaya.com. Please note that if you acquired the Product(s) from an authorized Avaya reseller outside of the United States and Canada, the warranty is provided to you by said Avaya reseller and not by Avaya.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, HTTP://SUPPORT.AVAYA.COM/LICENSEINFO/ ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AUTHORIZED AVAYA RESELLER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AUTHORIZED AVAYA RESELLER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA AUTHORIZED RESELLER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE (“AVAYA”).

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, or Hardware provided by Avaya. All content on this site, the documentation and the Product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

Third-party components

Certain software programs or portions thereof included in the Product may contain software distributed under third party agreements (“Third Party Components”), which may contain terms that expand or limit rights to use certain portions of the Product (“Third Party Terms”). Information regarding distributed Linux OS source code (for those Products that have distributed the Linux OS source code), and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply to them is available on the Avaya Support Web site: http://support.avaya.com/Copyright.

Trademarks

The trademarks, logos and service marks (“Marks”) displayed in this site, the Documentation and Product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the Documentation and Product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

Avaya is a registered trademark of Avaya Inc.

All non-Avaya trademarks are the property of their respective owners, and “Linux” is a registered trademark of Linus Torvalds.

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support Web site: http://support.avaya.com.

Contact Avaya Support

See the Avaya Support Web site: http://support.avaya.com for product notices and articles, or to report a problem with your Avaya product. For a list of support telephone numbers and contact addresses, go to the Avaya Support Web site: http://support.avaya.com, scroll to the bottom of the page, and select Contact Avaya Support.

Contents

Chapter 1: New in this release 13
- Features 13
  - Preference changes 13
  - Fixing application errors 13
- Other changes 14
  - Rebranded application 14

Chapter 2: Introduction 15
- Related resources 16
  - Support 16
  - Avaya Mentor videos 16

Chapter 3: Getting Started in Avaya Aura® Contact Center 19
- Initial configuration tasks 19
- Configuring contact center resources 21
- Downloading the most recent Orchestration Designer 21
- Opening Orchestration Designer 22
- Using a default application 25
- Configuring contact routes 29
- Adding prompts 30

Chapter 4: Application upgrade 33
- Prerequisites to application upgrade 33
- Application upgrade procedures 34
- Reviewing existing applications and variables in Orchestration Designer 35

Chapter 5: Application planning 37
- General application writing process 37
- Determine the types of contacts 38
- Match contact types with agent skillsets 38
- Define the application objective 39
- Identify available resources 39
- Identify and create variables 40
- Compose the application 40
- Correct validation errors 40
- Synchronize the applications in Orchestration Designer 40
- Activate the application 41

Chapter 6: Application creation 43
- Prerequisites to application creation 43
- Application creation procedures 43

Chapter 7: Application editing 47
- Prerequisites to application editing 47
- Application editing procedures 48

Chapter 8: Application demonstration 51
- Prerequisites for application demonstration 51
- Opening and configuring your Orchestration Designer 52
- Creating a variable in Orchestration Designer 55
- Creating a logic block for closed days 58

Connecting the start node to the first block ... 59
Creating a transition ... 60
Creating the if closed conditional expression for holidays ... 61
Creating recorded announcements ... 64
Connecting the transition to the holiday message ... 66
Creating a disconnect block ... 66
Creating block to identify closed hours ... 68
Creating the If closed conditional expression for the hours ... 70
Completing the closed hours branch of the flow application ... 71
Creating a logic block for out-of-service skillset ... 72
Creating a queue to skillset block ... 74
Creating a treatment block to play music ... 76
Creating a logic block to check if the call is queued ... 77
Rechecking skillset service ... 79
Playing a technical message for disconnecting contact ... 80
Copying a block ... 81
Playing a hold message ... 82
Creating a wait block ... 84
Exiting calls from the flow application ... 85
Synchronizing the views ... 86
Activating the application ... 90
Chapter 9: Integrated reporting ... 91
How integrated reporting works ... 91
Types of IVR and contact center integration ... 91
Front-End IVR processing ... 92
Front-End Contact Center Processing ... 94
Chapter 10: Orchestration Designer interface ... 95
Modes of operation ... 95
Session management ... 97
Views ... 98
Menus ... 101
Toolbar items ... 106
Chapter 11: Contact Center and Local view operations ... 107
Prerequisites for Contact Center or Local view operations ... 108
Showing or hiding the Contact Center view ... 108
Showing or hiding the Local view ... 108
Creating a Contact Center Manager Server node ... 109
Creating a new application ... 110
Configuring the contact router ... 113
Copying data from Contact Center view to the Local view ... 114
Opening an existing application ... 115
Changing the properties of an application ... 116
Creating a Contact Center Manager Administration node ... 116
Viewing Contact Center Manager Administration properties ... 117
Deleting a Contact Center Manager Administration node ... 117
Viewing Contact Center Manager Server properties ... 118
Deleting a Contact Center Manager Server node ... 118

Importing an application ... 119
Exporting an application ... 119
Activating an application ... 120
Deactivating an application ... 121
Renaming an application ... 121
Saving an application ... 122
Closing an application ... 123
Deleting an application ... 123
Converting the script application to a flow application ... 124
Checking applications for errors ... 125
Generating application documentation ... 126
Printing an application ... 127
Chapter 12: Problems view operations ... 129
Prerequisites for Problems view operations ... 129
Displaying the Problems view ... 129
Resolving errors in the application ... 130
Chapter 13: Synchronization view operations ... 131
Prerequisites for working in the Synchronization view ... 132
Displaying the Synchronization view ... 132
Synchronizing objects from the Local view ... 132
Adding an application to the Contact Center view ... 134
Updating an application in the Contact Center view ... 135
Removing an application from the Contact Center view ... 135
Adding an application, application manager data, or variable to the Local view ... 136
Updating an application, application manager data, or variable in the Local view ... 137
Removing an application, application manager data, or variable from the Local view ... 138
Updating a conflicted application ... 138
Updating all non conflicted applications in the Contact Center view ... 139
Updating all non conflicted applications, application manager data, or variables in the Local view ... 140
Updating all non conflicted objects to both the Local and Contact Center views ... 140
Updating an application in the Contact Center view with an invalid local application ... 141
Printing a Synchronization View report ... 142
Chapter 14: Flow Editor operations ... 143
Prerequisites ... 144
Adding blocks to the flow ... 144
Adding blocks using accelerator keys ... 146
Configuring general details for blocks ... 147
Renaming blocks and nodes ... 151
Defining block processing ... 152
Creating a default transition ... 156
Defining conditional transitions ... 156
Defining switched transitions ... 159
Copying and pasting blocks ... 160
Deleting blocks, nodes or transitions ... 161
Deleting a transition ... 162
Deleting multiple objects ... 162
Disconnecting blocks ... 163

Arranging blocks in the flow diagram ... 163
Inserting a new page ... 164
Deleting pages ... 164
Renaming a page ... 165
Adding a note to a block ... 165
Using Zoom to Fit ... 166
Zooming in ... 166
Zooming out ... 167
Resetting the size of the diagram ... 167
Moving the flow application ... 168
Viewing the block summary ... 169
Selecting application management data in block configuration ... 169
Chapter 15: Block creation ... 171
Prerequisites for block creation ... 171
Creating a logic block ... 172
Creating a queue to skillset block ... 174
Creating a queue to network skillset block ... 179
Creating a queue to agent block ... 181
Creating a queue to networked ACD block ... 183
Creating an input block ... 185
Creating a block to play a RAN ... 190
Creating a block to play a RAN in an AML-based contact center ... 192
Creating a block to play prompts ... 194
Creating a host data send info block ... 197
Creating a host block to request information ... 199
Creating a custom block ... 203
Creating a reference block ... 204
Creating a block to play music ... 206
Creating a ringback treatment ... 207
Creating a silent treatment ... 209
Creating a wait block ... 211
Creating an IVR block ... 212
Creating an IVR block to play prompts in the SIP configuration ... 214
Creating an IVR block to play prompts and collect digits in the SIP configuration ... 216
Creating an IVR block to send instant messages in the SIP configuration ... 219
Creating an IVR block for free form SIP configuration ... 221
Creating a menu block ... 223
Creating a disconnect block ... 229
Creating a busy block ... 230
Creating an overflow block ... 232
Creating a route contact block ... 233
Using an anchor node ... 235
Using an anonymous node ... 236
Configuring a source shortcut node ... 237
Configuring a destination shortcut node ... 238
Creating an exit node ... 238
Chapter 16: Script Editor operations ... 239

Prerequisites to script editor operations ... 239
Adding commands to script applications ... 239
Displaying or hiding line numbers ... 240
Going to a particular line number ... 240
Undoing or redoing the last change ... 241
Chapter 17: Condition and Assignment builder operations ... 243
Prerequisites for condition and assignment builder operations ... 243
Creating conditional and assignment expressions ... 243
Adding assignments and expressions ... 245
Deleting an expression ... 250
Chapter 18: Application manager data and variable configuration ... 251
Prerequisites to variable configuration ... 254
Creating application manager data ... 254
Renaming application manager data ... 256
Viewing application manager data properties ... 256
Deleting application manager data ... 257
Adding a variable to Orchestration Designer ... 257
Changing the properties of a variable ... 259
Removing a value from a set ... 260
Deleting a variable ... 261
Starting the Script Variables tool in Contact Center Manager Administration ... 261
Creating variables with one value ... 262
Creating application variables with multiple values ... 264
Checking variables for referencing applications ... 266
Changing a variable property ... 267
Copying variable properties ... 268
Deleting a variable from Contact Center Manager Server ... 269
Refreshing Orchestration Designer ... 269
Printing the variables to a file ... 270
Printing a variable to a printer ... 271
Chapter 19: Threshold configuration ... 273
Prerequisites to threshold configuration ... 273
Creating new threshold class ... 273
Selecting a threshold class for an application ... 275
Chapter 20: Web Service Integration ... 277
Activating the SOA Web service ... 277
Importing Web services to the Database Integration Wizard ... 277
Configuring variables in Orchestration Designer ... 282
Creating required blocks to handle data ... 282
Calling the Web service and collecting return values ... 284
Chapter 21: User preferences ... 287
Prerequisites for user preferences ... 287
Configuring network connection preferences ... 287
Configuring automatic saving ... 288
Configuring application preferences ... 289
Resetting the Contact Center perspective ... 291
Refreshing data ... 292

Chapter 22: Getting help ... 293
Viewing online Help in Orchestration Designer ... 293
Viewing context-sensitive Help in Orchestration Designer ... 293
Chapter 23: Scripting tips ... 295
Guidelines for providing feedback ... 295
Using variables to minimize script maintenance ... 297
Chapter 24: General programming commands reference ... 299
ASSIGN TO ... 299
EXECUTE ... 299
EXECUTE SCRIPT ... 300
IF-THEN-END IF ... 300
IF-THEN-ELSE-END IF ... 300
QUIT ... 301
SECTION ... 301
WAIT ... 302
Chapter 25: Basic contact processing commands reference ... 305
APPEND ... 305
CHANGE PRIORITY IN AGENT ... 305
CHANGE PRIORITY IN SKILLSET ... 306
CONTAINS ... 307
CONVERT ... 308
DISCONNECT ... 308
GIVE BUSY ... 309
GIVE MUSIC ... 309
GIVE OVERFLOW ... 310
GIVE RAN ... 311
GIVE RINGBACK ... 311
GIVE SILENCE ... 312
MATCHES ... 312
QUEUE TO AGENT ... 313
QUEUE TO SKILLSET ... 314
REMOVE FROM AGENT ... 315
REMOVE FROM SKILLSET ... 316
ROUTE CALL ... 316
Chapter 26: Advanced script command reference ... 319
EVENT HANDLER ... 319
LOG ... 320
READVAR and SAVEVAR ... 321
WHERE-EQUALS ... 322
Chapter 27: Network commands ... 325
Using network skillsets ... 325
CHANGE PRIORITY IN NETWORK SKILLSET ... 328
QUEUE TO NETWORK SKILLSET ... 329
REMOVE FROM NETWORK SKILLSET ... 330
CHANGE PRIORITY IN NACD ... 331
QUEUE TO NACD ... 333
REMOVE FROM NACD ... 334

Chapter 28: Voice processing for Avaya Communication Server 1000 337
Choosing the most efficient commands 337
GIVE IVR 340
COLLECT DIGITS 342
OPEN VOICE SESSION and END VOICE SESSION 343
PLAY PROMPT 346
GIVE CONTROLLED BROADCAST ANNOUNCEMENT 348
Chapter 29: Host Data Exchange commands 353
SEND INFO 354
SEND REQUEST 356
GET RESPONSE 357
Chapter 30: Contact processing for SIP 359
Configuration guidelines and general scripting considerations 359
GIVE_IVR 362
Instant message commands 368
ROUTE CALL URI 370
Chapter 31: Intrinsics overview 371
Types of intrinsics 371
Return value 371
Intrinsic usage 372
Chapter 32: Call intrinsics 375
AGE OF CALL 375
CALL DATA 375
CALL FORWARD 377
CALL FORWARD BUSY 378
CALL FORWARD DO NOT DISTURB 378
CALL FORWARD NO ANSWER 379
CDN 379
CLID 379
CONFERENCED 381
CONSULTED 381
DIALED DN 382
DIRECT CALL 382
DNIS 383
INTERNATIONAL CALL 383
LOC 384
NETWORK CALL 385
NPA 385
NXX 386
NPANXX 386
ON HOLD 386
QUEUED 387
ROUTE NUMBER 388
TRANSFERRED 388
Chapter 33: Multimedia intrinsics 391
%AGENT%AGENT 391
%DATE%DATE 392

%INTEGER%WAITTIME ... 393
%PRIORITY%PRIORITY ... 393
%SKILLSET%SKILLSET ... 394
%TIME%TIME ... 395
CONTACTSUBTYPE ... 396
CUSTOMERID ... 397
CUSTOMFIELD1 ... 397
FROMADDRESS ... 398
NEXTYEAR ... 399
QUEUETYPE ... 400
TOADDRESS ... 402
Chapter 34: Skillset intrinsics ... 403
ANSWERED COUNT ... 403
AVERAGE SPEED ANSWER ... 404
EXPECTED WAIT TIME ... 405
IDLE AGENT ... 407
IDLE AGENT COUNT ... 407
LOGGED AGENT COUNT ... 408
LOGGED OUT AGENT ... 409
LONGEST IDLE AGENT ... 410
MOST LOGGED AGENTS ... 411
OLDEST ... 413
OUT OF SERVICE ... 414
POSITION IN QUEUE ... 415
PRIORITY IN QUEUE ... 416
PRIORITY IN NETWORK QUEUE ... 418
QUEUED COUNT ... 419
Chapter 35: Time intrinsics ... 421
DAY OF WEEK ... 421
DATE ... 421
DAY OF MONTH ... 422
MONTH OF YEAR ... 423
TIME OF DAY ... 423
Chapter 36: Traffic intrinsics ... 425
CALL RATE ... 425
CONTACT RATE ... 425
TOTAL ACTIVE CALLS ... 426
TOTAL ACTIVE CONTACTS ... 426
Chapter 37: Open queue intrinsics ... 429
AGE OF CONTACT ... 429
CONTACT DATA ... 429
Contact Type ... 431
ROUTE POINT ... 431
URI ... 433
Chapter 38: SIP intrinsics ... 435
Contact Type ... 435
CmfContactID ... 435

Provider ... 435
ProviderContactID ... 436
SIP_CALL_ID ... 436
SIP_INITIAL_TEXT ... 436
SIP_REQUEST_URI ... 436
SIP_FROM_ADDRESS ... 436
SIP_TO_ADDRESS ... 437
SIP_LOCATION ... 437
SIP_PREFERRED_LANGUAGE ... 437
SIP_SUBJECT ... 438
SIP_USER_AGENT ... 438
SIP_CALL_TYPE ... 438
SIP_MAIN_CONTACT_ID ... 438
Skillset ... 439
SIP_RETURNED_TEXT_n or SIP_RETURNED_DIGITS_n ... 439
Chapter 39: P-Intrinsics SIP Header ... 441
Typical solution using P-Intrinsics ... 441
P-Intrinsics format ... 442
Sample Voice XML applications using P-Intrinsics ... 443
Chapter 40: Troubleshooting applications ... 445
Application execution problems ... 445
Voice processing problems ... 446
Networking problems ... 449
Phantom voice contacts ... 450
Validation errors ... 451
Appendix A: Reserved Keywords ... 461
Reserved keywords ... 461
Reserved keywords for Avaya Communication Server 1000 PBX ... 466
Appendix B: Icons Rebranded in Orchestration Designer Release 6.2 ... 467
View and Editor icons changes ... 467
General icons changes ... 467
Flow Operation icons changes ... 468
Flow Node icons changes ... 468
Appendix C: Commands and intrinsics ... 471
Commands–General ... 471
Commands–Call Procedures ... 472
Commands–Advanced ... 473
Commands–Networking ... 474
Commands–Voice Processing ... 475
Commands–HDX ... 476
Intrinsics–Skillset ... 476
Intrinsics–Time ... 477
Intrinsics–Traffic ... 478
Intrinsics–Call ... 479
Intrinsics–Multimedia ... 481
Intrinsics–SIP ... 481
Index ... 485

Chapter 1: New in this release

The following sections detail what is new in *Avaya Aura® Contact Center Configuration – Orchestration Designer Application Development* (NN44400-510).

- Features on page 13
- Other changes on page 14

Features

See the following sections for information about feature changes:

- Preference changes on page 13
- Fixing application errors on page 13

Preference changes

The Preference settings in Orchestration Designer integrate with other Avaya products. You can locate the preferred settings in Contact Center, Contact Center Manager Administration Connection, or Default Options.

The following procedures in this document describe the changes to the preferences:

- Displaying or hiding line numbers on page 240
- Configuring network connection preferences on page 287
- Configuring automatic saving on page 288
- Configuring application preferences on page 289

Fixing application errors

Orchestration Designer contains a different method to edit problems in the applications. In the Problems view, right-click the problem and choose to go to the block, or open the editor for the block from the Problems view.

For information about fixing errors, see Checking applications for errors on page 125 and Resolving errors in the application on page 130.

Other changes

See the following section for information about other changes:

- Rebranded application on page 14

Rebranded application

The Service Creation Environment has been renamed Orchestration Designer. There are icon changes, and updates to fields and labels throughout the application.

A list of the changed icons appears in the appendix of this document. See Icons Rebranded in Orchestration Designer Release 6.2 on page 467.

Chapter 2: Introduction

Use Orchestration Designer to create applications. Applications contain instructions that determine the sequence of steps that a contact follows after the contact arrives at Avaya Aura® Contact Center. These steps can include call treatments (such as music or ringback), call routing (such as skill-based routing), or interaction with the caller (entering account numbers).

Applications perform two major functions: they define the path a contact follows, and they provide treatments to a contact as the contact moves through Contact Center. You can also use the applications to track and record information about each step in the progress of a contact, and use this information to analyze how your contact center functions to improve service.

To create effective applications, you must understand the objectives of the contact center. Generally, a contact center has three major objectives:

- Maximize contact center efficiency—agents process contacts that they are best prepared to deal with based on the requirements (information, media) of the contact.
- Maximize user satisfaction—route contacts to a qualified agent initially to improve the wait time, or provide activities for caller to further route their contact.
- Analyze your contact center performance, and decide on how to improve service—store contact data in a database to report contact center functionality.

Specific applications run when certain types of contacts enter Contact Center. These applications must recognize the contact requirements and route the contacts to an agent who can deal effectively with these requirements. As an application designer, you must write the applications to ensure that contacts are routed to the qualified agents as quickly as possible. A contact is not always answered immediately by an agent. However, you can provide treatments to the voice contacts while they wait in a queue. These treatments can tell callers the estimated amount of time before their call is answered, or play music to callers while they wait in queue.

When a contact enters Contact Center, the contact starts running the master application. Typically, the master application routes contacts and provides treatments based on the incoming contact information. The master application directs the contact to primary applications based on this information. For example, if a Calling Line Identification (CLID) number for a voice contact is determined to be on the list of important callers, the voice contact can queue to an agent or skillset reserved for contacts requiring preferential treatment. Additionally, you can use time of day, day of week, or contact center activity to determine how the contact is handled.

A primary application is directly called from the Master Script. A secondary application is an application that is called from a primary application. The system can peg only primary applications in reporting. No reporting is completed for secondary applications.

Orchestration Designer is a graphical tool that you use to create applications. If you are familiar with previous releases of Avaya NES Contact Center, the applications can also be viewed as script files.

This guide is divided into the following sections:

- The first section provides the processes for working with applications in your contact center. You can upgrade from a previous release of the contact center software, create a new application, or make changes to an existing application.
- The second section contains general procedures for working with Orchestration Designer, including configuring the optional preferences and changing the appearance of your environment.
- The third section of this document contains programming reference for the commands that you can use when creating script applications.
- The appendices contain general information, including a quick reference to commands for the individual platforms and reserved keywords.

Prerequisites

- Install Orchestration Designer on your Contact Center Manager Administration server or your client desktop.
- Configure the Contact Center Manager Server in Contact Center Manager Administration. For more information, see the Contact Center Manager Administration online Help.

Related resources

Support

Visit the Avaya Support website at http://support.avaya.com for the most up-to-date documentation, product notices, and knowledge articles. On the Avaya Support website at http://support.avaya.com, search for notices, release notes, downloads, user guides, and resolutions to issues. Use the Web service request system to create a service request. Chat with live agents to help answer questions. If an issue requires additional expertise, agents can quickly connect you to a support team.

Avaya Mentor videos

Avaya Mentor is an Avaya-run channel on YouTube that includes technical content on how to install, configure, and troubleshoot Avaya products.

Visit http://www.youtube.com/AvayaMentor and do one of the following:

- Enter a key word or key words in the Search channel to search for a specific product or topic.
- Click the name of a playlist to scroll through the posted videos.

Chapter 3: Getting Started in Avaya Aura® Contact Center

Use this chapter to configure the resources required in your Contact Center. By following the procedures in this chapter, you can quickly route contacts to your agents.

You can add, configure, and delete resources using one of the following methods:

- upload and download bulk data using the Configuration Tool spreadsheets
- add or remove resources individually using the Web-based user interface

For information about uploading and downloading bulk data, see *Avaya Aura® Contact Center Manager Administration – Client Administration* (NN44400-611). This chapter describes adding resources individually.

Initial configuration tasks

About this task

The following task flow shows you the sequence of procedures you perform to configure the resources for the Contact Center system to route contacts.

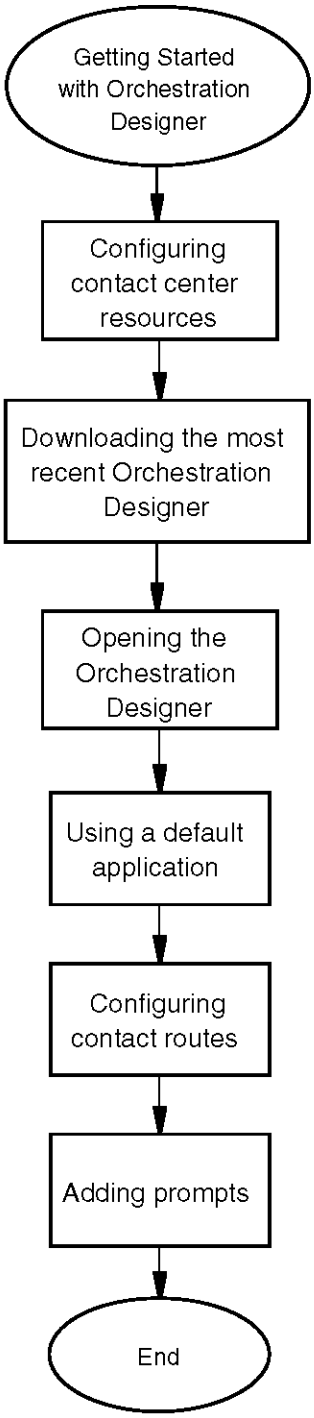


Figure 1: Contact center initial configuration tasks

Configuring contact center resources

Before you begin

- Install your contact center.
- Configure your contact center resources in Contact Center Manager Administration (CCMA). For more information see *Avaya Aura® Contact Center Manager Administration – Client Administration* (NN44400-611).

About this task

Configure the contact center resources that are required for your applications. Contact Center Resources include CDNs, media servers, skillsets, agents, and Dialed Number Identification Services (DNISs). The following procedure describes the minimal resources that are required for Orchestration Designer.

Procedure

1. Log on to CCMA.
2. Verify the Server Information.
3. Configure and acquire a SIP CDN (Route Point).
4. Add a media server.
5. Add a skillset.
6. Add an agent.

Downloading the most recent Orchestration Designer

Before you begin

- Download the most recent version of Orchestration Designer.exe file to your Contact Center Manager Administration server.
- Ensure that you reset your IceAdmin password on the Contact Center Manager Administration server. See *Avaya Aura® Contact Center Commissioning* (NN44400-312).

About this task

Download the most recent Orchestration Designer to your Contact Center Manager Administration client to ensure that you are working with the most recent version of the software.

If you attempt to open a version of Orchestration Designer that is not installed on the client, a prompt appears asking you to download the correct version of the software.

Procedure

1. Open the Contact Center Manager Administration browser.
2. Log on to the Contact Center Manager Administration application as a user with scripting access.
3. Click **Scripting**.
4. Choose **Orchestration Designer** > **Launch Orchestration Designer**.
5. If prompted to download Orchestration Designer, click **OK**.
6. In the File Download - Security Warning message dialog box, click **Run**.
7. In the Installation Welcome window, click **Next**.
8. In the Customer Information window, type a **User Name** and **Organization Name** in the appropriate boxes.
9. Under **Install this application for**, select the option for your installation.
10. Click **Next**.
11. In the Destination Folder window, select the installation folder for Orchestration Designer.
12. Click **Next**.
13. In the Ready to Install the Program window, click **Install**.
14. After the installation is complete, click **Finish**.

Procedure job aid

Alternatively, you can save the .exe file on your client machine, and use the steps from the procedure (Step 7 to Step 14) to install Orchestration Designer on your client.

Opening Orchestration Designer

About this task

Open Orchestration Designer to configure the routing in your contact center environment.

Procedure

1. Open the Contact Center Manager Administration browser.

2. Log on to the Contact Center Manager Administration application as a user with scripting access.
3. Click **Scripting**.
4. Click **Orchestration Designer** > **Launch Orchestration Designer**.

Procedure job aid

The following figure shows Orchestration Designer for Contact Center. Each part of the window contains a label that describes what appears in the panel.

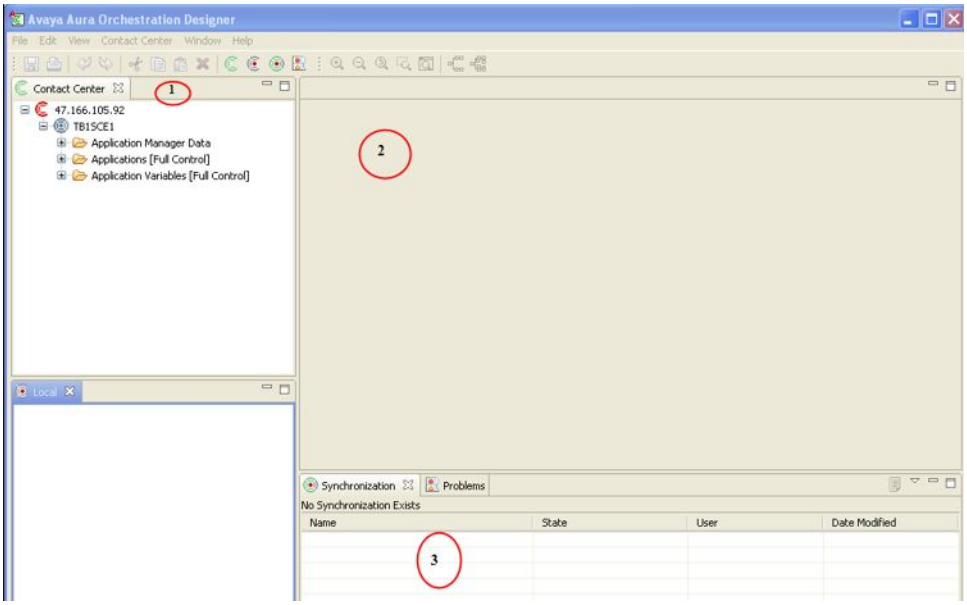


Figure 2: Orchestration Designer for Contact Center

1	Contact Center view: The Contact Center view of Orchestration Designer shows all applications, application variables, and application management data currently configured in your Contact Center.
2	Application editor view: The Application editor is the main tool to create or modify the default applications. It provides the canvas on which to place the blocks.
3	Synchronization view: The Synchronization view shows the difference between the objects in the Local view and the Contact Center view for the Contact Center Manager Server.

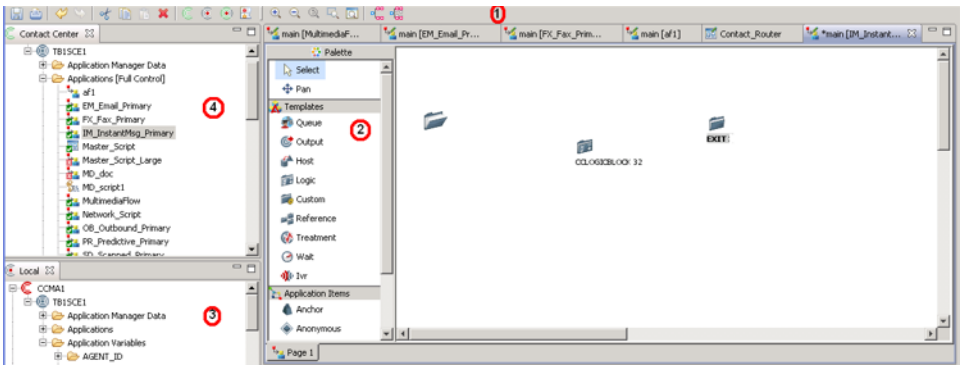


Figure 3: Orchestration Designer tabs

1	View tabs: The tabs located across the top of the Application editor represent main pages and block editors for the flow applications on which you work.
2	Palette bar: The icons represent blocks that you can use to build your Contact Center applications. The blocks you see depend on the switch you use in your Contact Center.
3	Local view: The Local view provides a user work space on a desktop to work with copies of the variables and applications.
4	Application Manager Data folder contains a list of all the agents, skillsets, CDNs, and DNISs. Applications folder contains a list of all the applications in the system. Applications are used to control how contacts are routed through the Contact Center and the treatment each contact receives. Applications Variables contains a list of all the variables in the system. Variables are used to change the nature of a flow at run time without changing the application.

You can also start Orchestration Designer from the Start menu. Click **Start** > **All Programs** > **Avaya** > **Contact Center** > **Orchestration Designer**.

If you start Orchestration Designer using the Start menu, you can create and work with applications and variable data in a local version of Orchestration Designer without affecting the working contact center.

The local version of Orchestration Designer allows you to perform the following tasks:

- Access all information without restrictions by access classes.
- Perform updates without affecting your Contact Center applications.
- Create applications using Orchestration Designer before the rest of the Contact Center software is installed.

By default, the Local and Problems views appear in your Orchestration Designer window. The top right corner is reserved for the script or flow application editor.

If you start Orchestration Designer from the Contact Center Manager Administration application, there are no partition restrictions. You log on to Contact Center Manager Administration as administrator and work with blocks and variables in Orchestration Designer as administrator.

Only one instance of Orchestration Designer can run at a time.

For more information about the menu options, see Orchestration Designer interface on page 95.

View name	Description
Contact Center view	The Contact Center view shows all of the applications, variables, and application management data that are currently inactive or active in your Contact Center. You can make minor changes to applications in the Contact Center view. However, Avaya recommends that you work on a copy of the application in the Local view to make significant changes. For more information about the Contact Center view, see Contact Center and Local view operations on page 107.
Local view	The Local view shows all of the applications, variables, application management data, and intrinsics saved on the local machine. You need not be connected to a Contact Center Manager Administration or to the network to work with this data. You can upload applications to the Contact Center view after you finish your modifications. For more information about the Local view, see Contact Center and Local view operations on page 107.
Synchronization view	The Synchronization view shows the differences between all objects stored on the Contact Center Manager Server (Contact Center view) and the objects stored on the Local client (Local view) after you use the Synchronization command. For more information about the Synchronization view, see Synchronization view operations on page 131.
Problems view	The Problems view shows the errors in the current application. You can use the problems view to determine where the problem is, and determine the reason for the problem. For more information about the Problems view, see Problems view operations on page 129.

Using a default application

Before you begin

- Start Orchestration Designer.
- Ensure that the High Priority Skillset and Low Priority Skillset skillsets are in service.

About this task

Use one of the four default applications to instantly access contact routing in your contact center:

- An application for routing any contact is called the Contact Router. No configuration is required to use this default application.
- A Simple Greeting application. This application requires minimal configuration to ensure it routes contacts correctly.
- An application for routing Multimedia contacts. You cannot customize the Multimedia default application.
- An application for routing contacts in a network of contact center servers.

Use this procedure if you want to modify the SimpleGreeting application for your contact center. Detailed information about the application is provided after the procedure.

Procedure

1. In the Contact Center pane, select your Contact Center Manager Server name.
2. Expand **Applications**.
3. Select the **SimpleGreeting** application.
4. Configure the default skillset as the skillset to which you want to send the incoming calls in your contact center.

   Important:

   You must change the skillset for the High Priority Queue block and the Low Priority Queue block. The skillsets you choose must be in service.

Procedure job aid

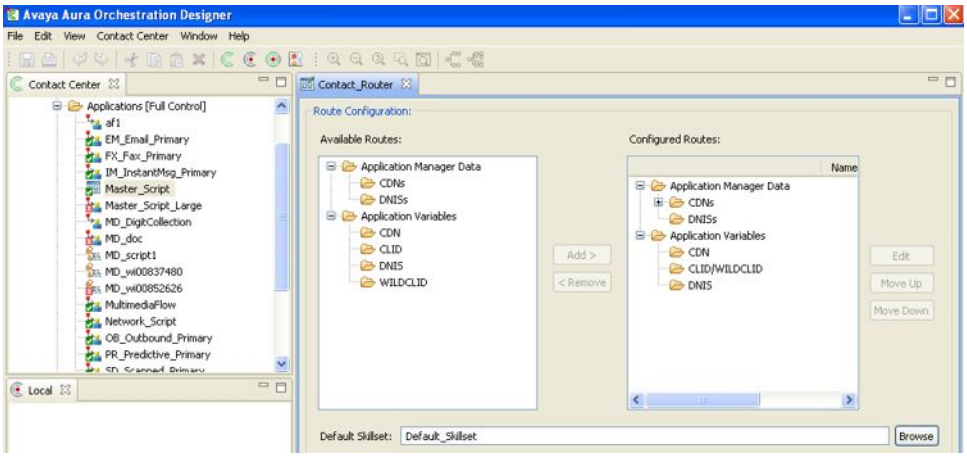


Figure 4: Contact Router application configuration

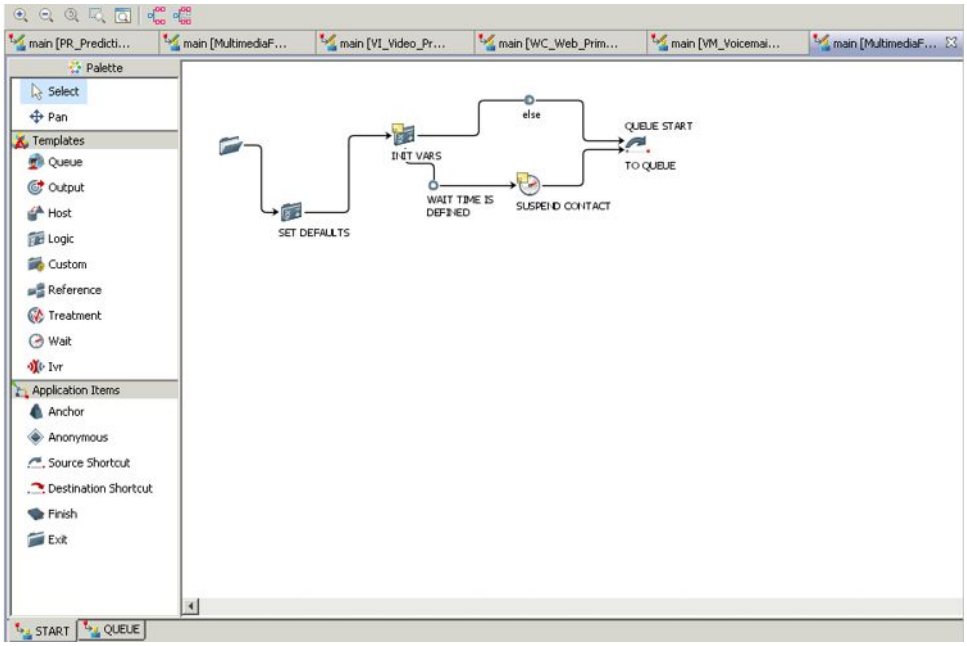


Figure 5: Multimedia application

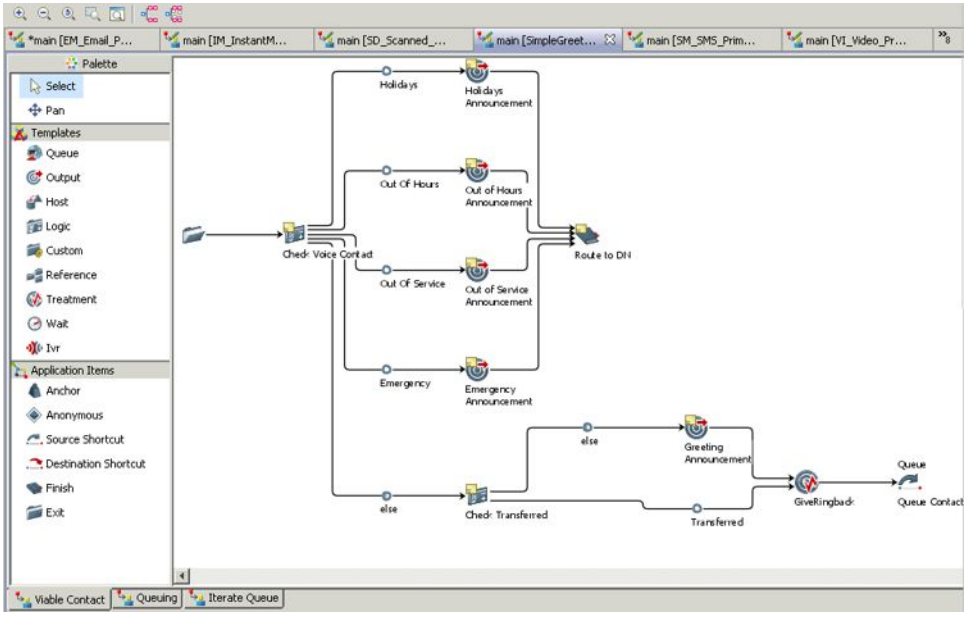


Figure 6: Simple Greeting application

The following table lists the blocks used in the Simple Greeting application.

Block	Default
Holidays Announcement	Thank you for contacting our Customer Support. The Contact Center is closed due to unforeseen circumstances. Please leave a message and we will get back to you as soon as possible.
OutOfHours	Thank you for contacting our Customer Support. The Contact Center is currently closed. It opens daily between 8 a.m. and 6 p.m. Please leave a message and we will get back to you as soon as possible.
Holidays	Thank you for contacting our Customer Support. The Contact Center is currently closed on January first for public holidays. The Contact Center will re-open on January second at 8 a.m. Please leave a message and we will get back to you as soon as possible.
OutOfService	Thank you for contacting our Customer Support. All our agents are busy at present. Please leave a message and we will get back to you as soon as possible.
Q Hi Priority Skillset	Default skillset.
Q Low Priority Skillset	Default skillset.
FirstRan	All of our agents are currently busy, please hold and you will be routed to the next available agent.
SecondRan	Your call is valuable to us. Please continue to hold.
Greeting	Welcome to Customer Support.

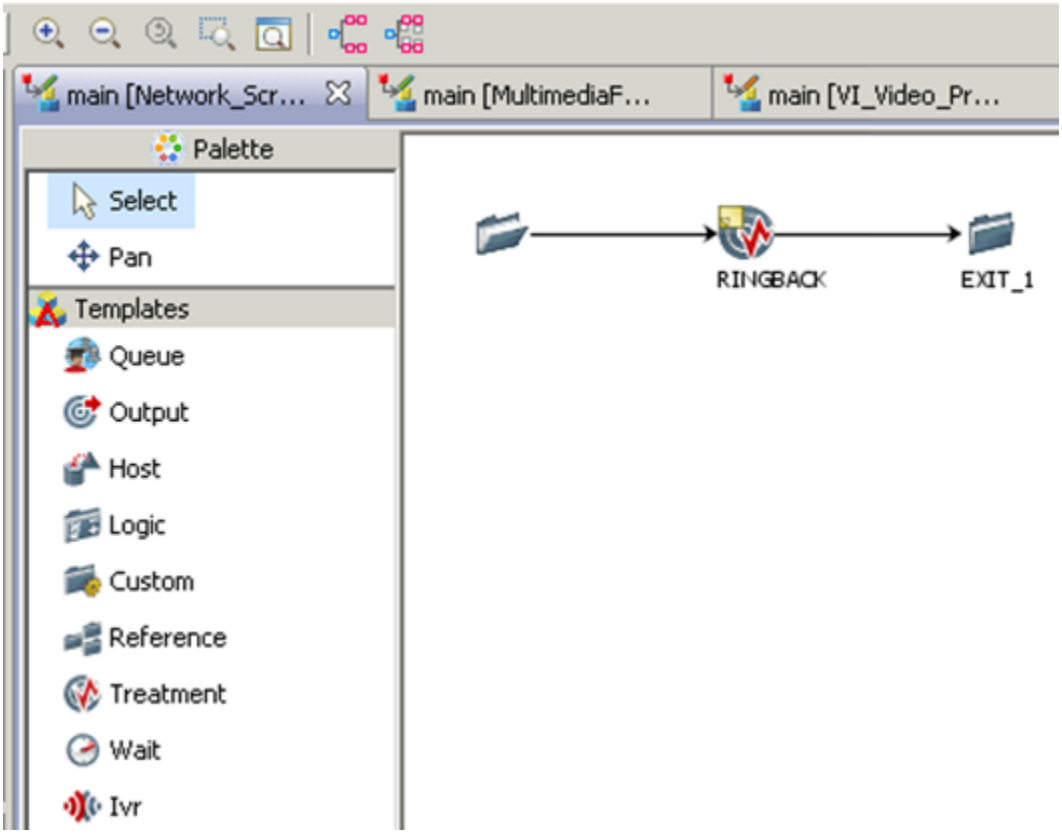


Figure 7: Network application

Configuring contact routes

Before you begin

- Create an application in Orchestration Designer.
- Create an agent in your contact center.
- Create at least one CDN (Route Point) in your Contact Center.

About this task

Configure the contact routes for incoming contacts. Contact routes are assigned based on CDNs (route points), DNISs, or CLID number of a contact. For each CDN (route point) or CLID number, you can designate a flow application that controls contact handling.

Procedure

1. Expand **Applications**.
2. Double-click **Master_Script**.

3. In the Contact Router editor, configure the general settings for your contact center.
4. If opening Master Script the first time, browse to a location on your computer.
5. Click **OK**.
6. Under **Available Routes**, select a specific **CDN** (Route Point) or **CLID**.
7. Click **Add**.
8. In the **Application Chooser**, select a valid application to associate with the route.

   You can choose an application you created, an application you customized, or a default application (SimpleGreeting, Multimedia).
9. Choose **File** > **Save**.
10. Repeat steps 6 to 9 for each route to configure.
11. Choose **File** > **Save**.

Variable definitions

Name	Description
Default skillset	If no matching route is found, caller is routed to this skillset.
Available routes	A list of routes that are currently configured in Contact Center Manager Administration for your server.
Configured routes	A list of routes with an assigned application for routing the contacts that enter the Contact Center on each route. The list provides the name of the configured application.

Adding prompts

Before you begin

- Install Contact Center on the server.
- Identify the prompt or music file you want to create or modify.
- Know the name of your content name space (the same name as the SIP domain).

About this task

Add the prompts you can use for your Contact Center applications.

Procedure

1. Obtain the default prompts from the recording studio.

   OR

   Create the .wav files that you want to use to play to callers.

   The recordings must be saved as Linear Mono 16 Bit PCM sampled at 8 KHz.
2. Create a folder on the server with the same name as the SIP domain.
3. Create a subfolder with the same name as the MS Locale. For example, use en_us.

   To get the local subscriber settings open **Start** > **Programs** > **Avaya** > **Contact Center** > **Manager Server** > **Server Configuration** > **Local Subscriber**.
4. Copy the newly recorded .wav files into that folder.
5. Zip the contents of the SIP domain folder.
6. Log on to the **Element Manager**.
7. Click **Tools** > **Media Management**.
8. Under **Media Management**, select the name of your content name space.
9. Select **Browse**.
10. Select the **Content Group** with the name of your MS Locale.
11. Click **More Actions**.
12. Select **Batch File Provision**.
13. Browse to the zip file of your recorded .wav files.
14. Select **Upload**.
15. To incorporate a voice prompt in your application, open Orchestration Designer and the application where you want to change the recording. Open the block and type the name of the new .wav file. Save the application.

Chapter 4: Application upgrade

If you upgrade Avaya NES Contact Center Release 6.0 or Avaya NES Contact Center Release 7.0 to Avaya Aura® Contact Center Release 6.2, you can use the same scripts in the new version of the software as you did in the previous release. Use the procedures in this chapter to perform steps required to review or update your scripts from the previous releases of Avaya NES Contact Center in the new Orchestration Designer.

You can use the same text-based script language to view your scripts, or convert them to graphical flows for customer presentations in Contact Center Release 6.2.

This chapter provides the steps you require to activate and validate your existing scripts on your new server.

By default, the master application and network application appear in the Contact Center view as active. You can choose to use the master application, or perform the steps in this chapter to use an existing application from the previous release.

You can also review the sample applications in Orchestration Designer.

Prerequisites to application upgrade

- Install and commission the Contact Center Manager Administration server. See *Avaya Aura® Contact Center Installation* (NN44400-311) and *Avaya Aura® Contact Center Commissioning* (NN44400-312).
- Ensure that your application management data, such as CDNs, DNISs, skillsets, and agents, are backed up and restored on the new Contact Center Manager Server.
- Ensure that your script variables are restored on the new Contact Center Manager Server, or you create the variables that you need for your application. See Application manager data and variable configuration on page 251.
- Know the name of the server on which your existing Contact Center applications are stored.
- Know which scripts are currently active and handling contacts in your contact center.

Application upgrade procedures

About this task

This task flow shows you the sequence of procedures you perform to upgrade your scripts from a previous release of Contact Center Manager Server to an application in Orchestration Designer.

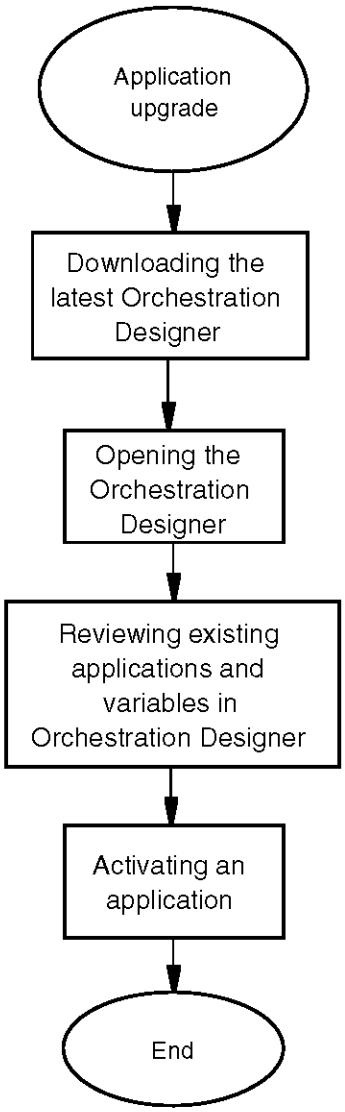


Figure 8: Application upgrades procedures

Reviewing existing applications and variables in Orchestration Designer

About this task

Review your existing application manager data, applications, application variables, and intrinsics in Orchestration Designer Contact Center view.

Note:

If you create new application variables that have the same name as an existing variable, you cannot save or run your applications.

Procedure

1. In Orchestration Designer, locate the **Contact Center** view.
2. In the **Contact Center** view, expand the Contact Center Manager Administration server name.
3. Select the name of the Contact Center Manager Server where your existing applications are stored.
4. Expand **Applications**.
5. Review the list of applications.
6. Ensure that you have at least one active application and all of the applications called by that application have a check mark on the icon beside the script name.

   The master application, and all applications called by the master are also active.

   If no check mark or a red X appears on the icon, the application is not active.

Procedure job aid

The following table shows inactive and active applications in the Contact Center view.

Icon	Description
	Valid script application
	Active script application
	Invalid script application
	Valid flow application

Icon	Description
	Active flow application
	Invalid flow application

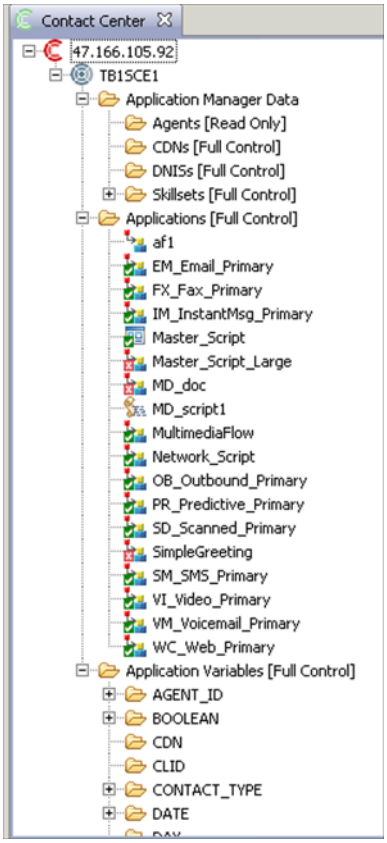


Figure 9: Contact Center view

Chapter 5: Application planning

An efficient contact center successfully matches contacts and their specific requirements with qualified agents. If you know which types of users contact your contact center, and the specific information or services they required, then you can write effective applications.

General application writing process

About this task

The following steps provide a guideline for creating new applications, or planning to change existing applications in your contact center.

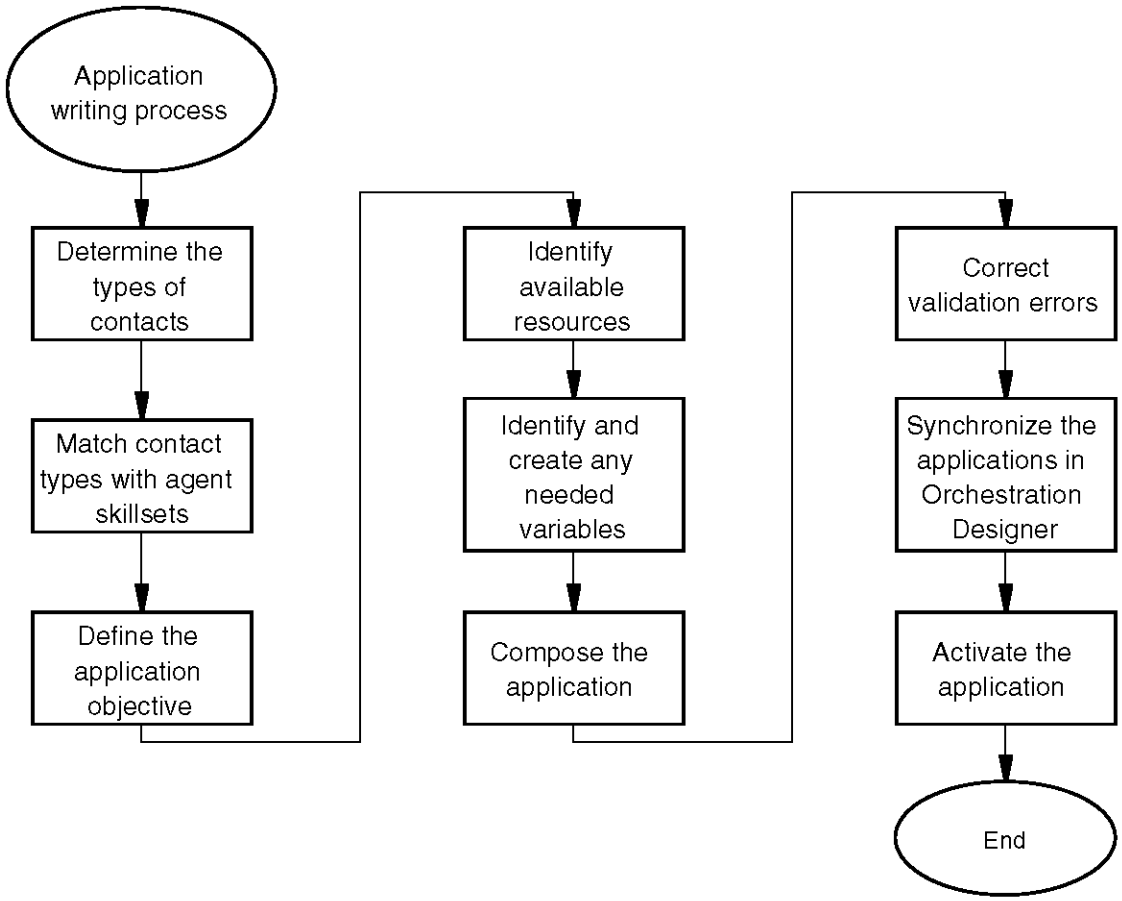


Figure 10: General application writing process

Determine the types of contacts

After you determine the types of contacts that enter your contact center, you can create skillsets that correspond to the contact types and assign agents to the skillsets.

Match contact types with agent skillsets

Skill-based routing is the ability to match qualified agents with related contact types based on subject of the contact and the type of the contact.

Define the application objective

Determine the purpose of the application. Each application can be referenced by, or can reference, other applications. Consider the following questions:

- Which kinds of applications do you need for your system?
- How many applications do you need to meet these requirements?
- How do the applications interact with one another?
- Does the application support Integrated Reporting?

In the Flow Editor in Orchestration Designer, you can create a flow chart to outline the logic of your applications, and develop the application at the same time. The flow chart helps you visualize the sequence of steps that a contact follows after it enters the contact center.

If the application is to support Integrated Reporting then it must be designed and implemented to do so. For more information on implementing applications to support Integrated Reporting see Integrated reporting on page 91.

Identify available resources

You can configure resources for your applications to control what happens to a contact when it enters the contact center. Most of the resources are available for voice contacts only and can include:

- CDNs
- recorded announcement (RAN) routes
- music routes
- skillsets
- number of agents in each skillset
- contact center working hours and holidays
- Interactive Voice Response (IVR) queues
- call treatments
- CLIDs and DNISs

For more information about configuring these resources, see *Avaya Aura® Contact Center Manager Administration – Client Administration* (NN44400-611) or the Contact Center Manager Administration online Help.

Identify and create variables

A variable is a placeholder that stores a value or group of values. For example, you can create a variable named holidays_gv to store information about the days when your contact center is closed. You can use variables to test for conditions that affect treatments given to the contact or the data collected from a contact as it moves through the contact center. You can create variables before, after, or during your application development.

For information about variables, see Application manager data and variable configuration on page 251.

You can add placeholders for the variables in the Local View to remember what variables you create. The placeholders can be uploaded to the Contact Center view.

Compose the application

Choose to write a script or flow application. The script application uses commands and the flow application uses the flow chart symbols from your planning.

For information about using Orchestration Designer, see Orchestration Designer interface on page 95.

Correct validation errors

Correct the validation errors using Orchestration Designer.

For information about correcting errors, see Problems view operations on page 129.

Synchronize the applications in Orchestration Designer

Resolve one or all differences between the Contact Center view and the Local view by updating the Contact Center Manager Server with the new version of your applications.

For information about synchronizing the applications, see Synchronization view operations on page 131.

Activate the application

When you activate the application, it becomes the application used on contacts that come into Contact Center Manager Server.

For instructions to activate a script application, see Activating an application on page 120.

Chapter 6: Application creation

When you create a script or flow application for the first time, Avaya recommends that you use the Local view in Orchestration Designer. The application is moved to the Contact Center view to activate it after it has been reviewed and approved.

In Orchestration Designer, you can create a visual flow application, that shows the nodes and paths that a contact can follow in a diagram. You can also choose to create a script application using the command syntax. After creating the command syntax, you can convert the script application to a flow to present to the customer.

By default, the master application and network application appear in the Contact Center view as active. You can choose to use the master application, or perform the steps in this chapter to create a new script or flow application for your contact center.

To follow a demonstration of the steps required to complete a new flow application, see Application demonstration on page 51.

Prerequisites to application creation

- Determine the purpose of your application, and understand the types of contacts you receive in your contact center.
- Create the application management data, such as CDNs, DNISs, skillsets, and agents, that you need for your new application. Configure CDNs, DNISs, and skillsets in Orchestration Designer. Configure Agents in the Management application.

  See the Contact Center Manager Administration online Help for information about creating these components.
- Create the variables that you need for your new application in Contact Center Manager Administration or Orchestration Designer. See Application manager data and variable configuration on page 251.

Application creation procedures

About this task

This task flow shows you the sequence of procedures you perform to create an application to route contacts using Orchestration Designer.

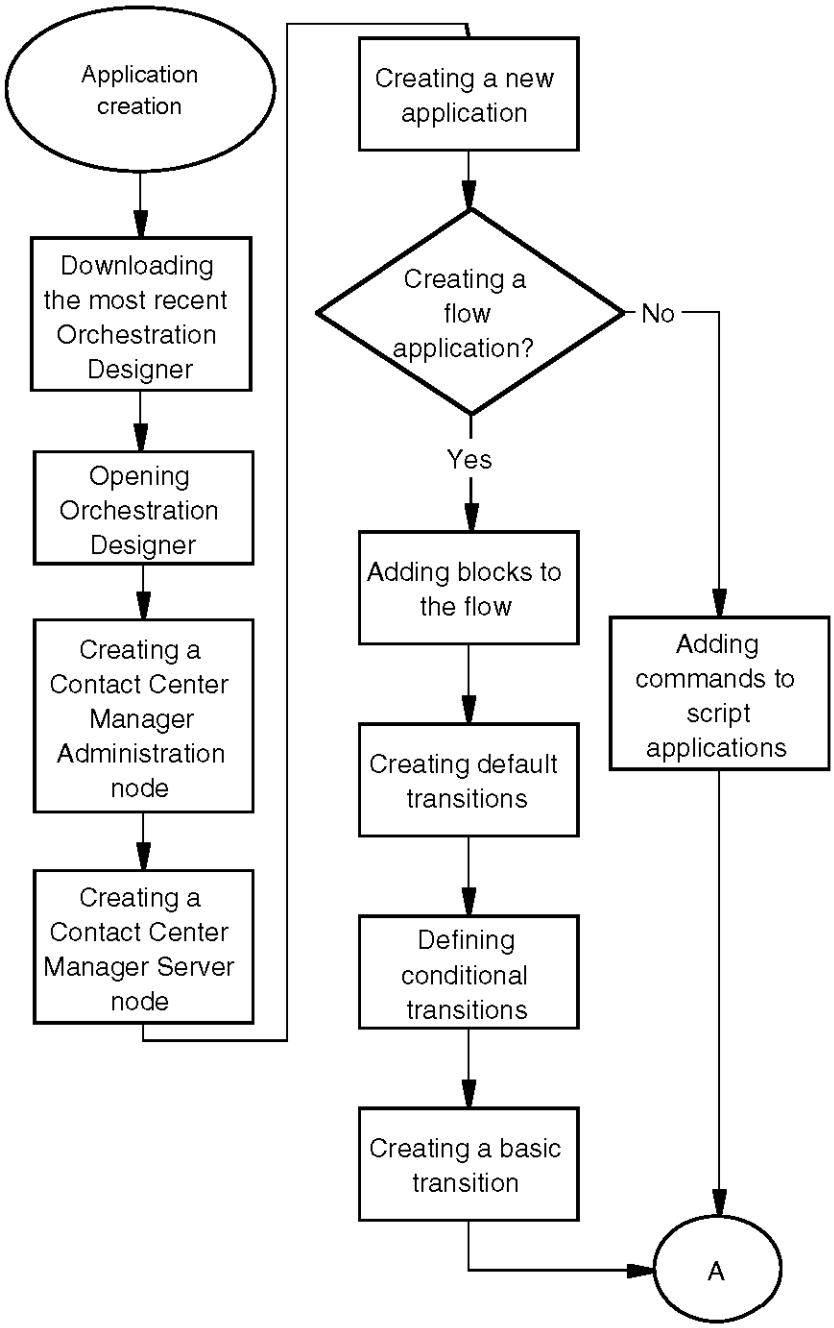


Figure 11: Application creation procedures

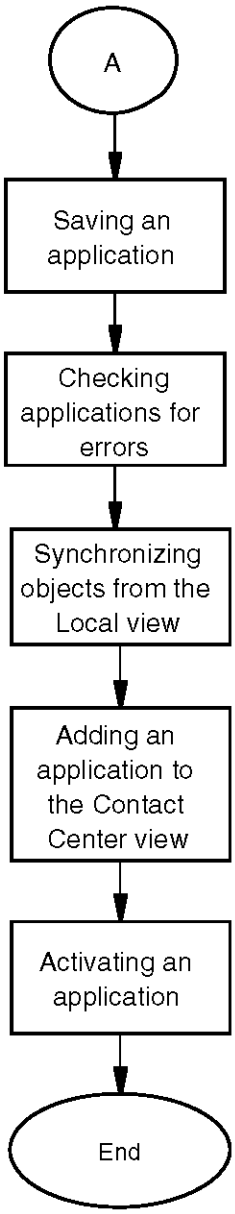


Figure 12: Application creation procedures (continued)

Chapter 7: Application editing

Whether you create an application for the first time in this release of Avaya Aura® Contact Center, or use an existing script application from a previous release of Avaya NES Contact Center, you must make modifications to accommodate new skillsets, new media types, or new information in your applications.

For minor changes, you can make your application changes in the Contact Center view. For more significant changes, Avaya recommends that you copy your application to the Local view to make the changes, and then synchronize your Local view and Contact Center view to activate your error-free applications in the Contact Center view. Working in the Local view significantly reduces the risk of affecting your contact routing. To access both the Contact Center view and the Local view, you must start your Orchestration Designer through the Contact Center Manager Administration application.

By default, the master application and network application appear in the Contact Center view as active. You can choose to use the master application as it is, or perform the steps in this chapter to change the application for your contact center. This chapter provides the steps you require to modify your application, and then activate it to route contacts.

Prerequisites to application editing

- Install the Contact Center Manager Administration server. See *Avaya Aura® Contact Center Installation* (NN44400-311).
- Create the application management data, such as CDNs, DNISs, skillsets, and agents, that you need for your new application. Configure CDNs, DNISs, and skillsets in Orchestration Designer. Configure Agents in the Management component.

  See the Contact Center Manager Administration online Help for information about creating these components.
- Create the variables that you need for your new application in Orchestration Designer, or in the Contact Center Manager Administration application. See Application manager data and variable configuration on page 251.
- Know the server on which your existing Contact Center scripts are stored.
- Know which scripts are currently active and handling contacts in your Contact Center.

Application editing procedures

About this task

This task flow shows you the sequence of procedures you perform to change an application that currently handles contacts in your contact center in Orchestration Designer in Contact Center.

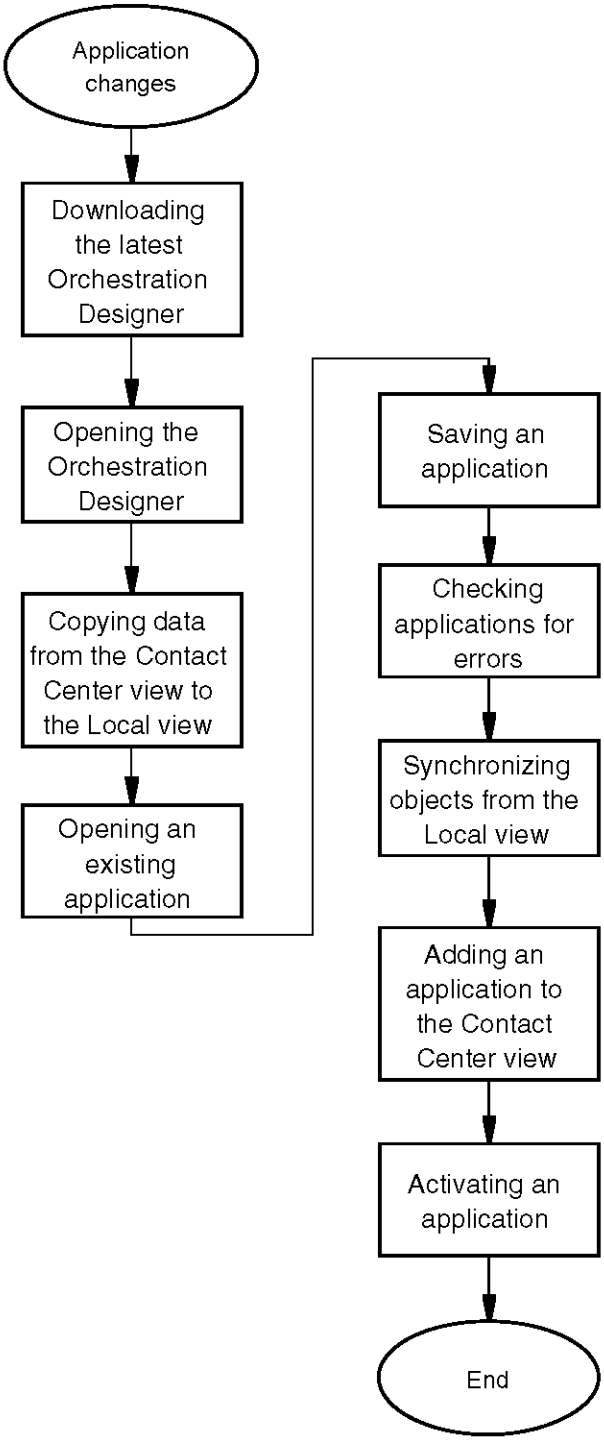


Figure 13: Application editing procedures

Chapter 8: Application demonstration

This chapter demonstrates how to configure your Orchestration Designer and create an application flow to perform a series of checks on a call and route it to the appropriate queue in the contact center.

The application for this demonstration performs the following steps:

- Check the time that the call is received.
  - If the contact center receives the call during the holiday season when the contact center is closed, play a holiday_message and disconnect the call.
  - If the contact center receives the call during a time of day when the contact center is closed, play a closed_message and disconnect the call.
- Check the status of the primary skillset.
  - If no agent with the primary skillset is logged on, play a technical message and disconnect the call.
  - Otherwise, queue the caller to the primary skillset.
- Play music to the caller.
- Check the status of the caller by determining whether the call is queued.
  - If the call is queued, play a continue to hold message.
  - If the call is not queued, attempt to requeue the call.
  - Otherwise, play a technical message and disconnect the call.
- Play a hold message every 60 seconds until the call is answered.

Avaya recommends that you create your new applications in the Local view of Orchestration Designer. The procedure that follows demonstrates how to work with the application in the Local view, and then what steps you must follow to make your application active in the Contact Center view.

Prerequisites for application demonstration

- Configure the Contact Center Manager Server in your Contact Center Manager Administration configuration tool.
- Install Orchestration Designer on your Contact Center Manager Administration server.

- Create the recorded announcements to play messages for holidays, closed times, hold, and a disconnect for technical reasons.
- Configure a music route for your voice contacts.

Opening and configuring your Orchestration Designer

About this task

Open Orchestration Designer and configure a Contact Center Manager Administration and a Contact Center Manager Server in your Local view to create a new flow application to route calls.

Procedure

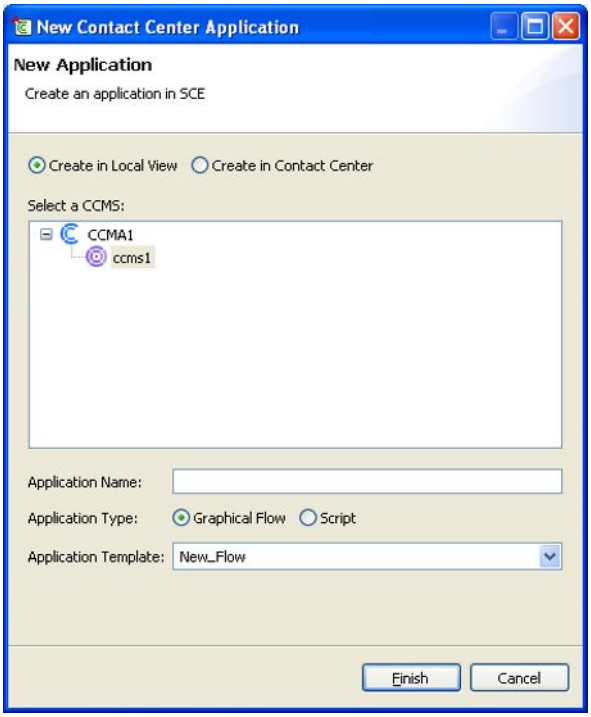
1. Log on to the Contact Center Manager Administration server.
2. Click **Start** > **All Programs** > **Avaya** > **Contact Center** > **Orchestration Designer**.

   OR

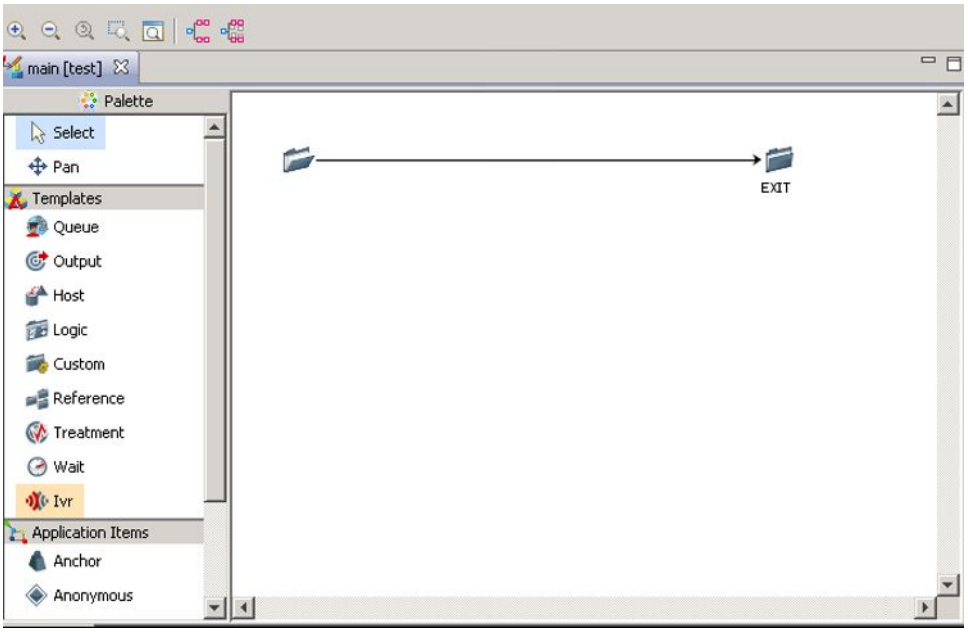
   Start Orchestration Designer from the Contact Center Manager Administration application.
3. Select the **Local** view, if it not already the current view.
4. On the **File** menu, click **New** > **CCMA**.
5. In the **CCMA Name** box, type `CCMA1`.
   You can choose any CCMA server name, such as your current CCMA server name. For this example, we are using CCMA1.
6. Click **Finish**.
7. Select **CCMA1**.
8. On the **File** menu, click **New** > **CCMS**.
9. In the **CCMS Name** box, type `CCMS1`.
   You can choose any Contact Center Manager Server (CCMS) server name, such as your current CCMS server name. For this example, we are using CCMS1.



10. Under **Server Type**, select **Standalone** or **Networking** to represent the CCMS server to which you are connecting.
11. In the **Server RLs** box, select the Contact Center Manager Server version number.
12. Under **Switch Type**, select the type of switch to which your Contact Center Manager Server is connected.
13. Click **Finish**.
14. In the **Local** pane, expand **CCMA1** > **CCMS1**.
15. Right-click **Applications**, and select **New** > **Application**.



16. In the **New Contact Center Application** dialog box, select **Create in Local View.**
17. Select a **CCMS** associated with a CCMA server.
18. In the **Application Name** box, type the name of your new flow application. For this example, the name is *ApplicationDemo1*.
19. In **Application Type**, select **Graphical Flow**.
    Graphical Flow is the default selection.
20. In **Application Template**, select **New_Flow**.
    New_Flow is the default selection. You can choose from a variety of existing templates. For this demonstration, start with a blank flow.
21. Click **Finish**.
    The Flow Editor opens with a Start node and an Exit node in the application flow.



Creating a variable in Orchestration Designer

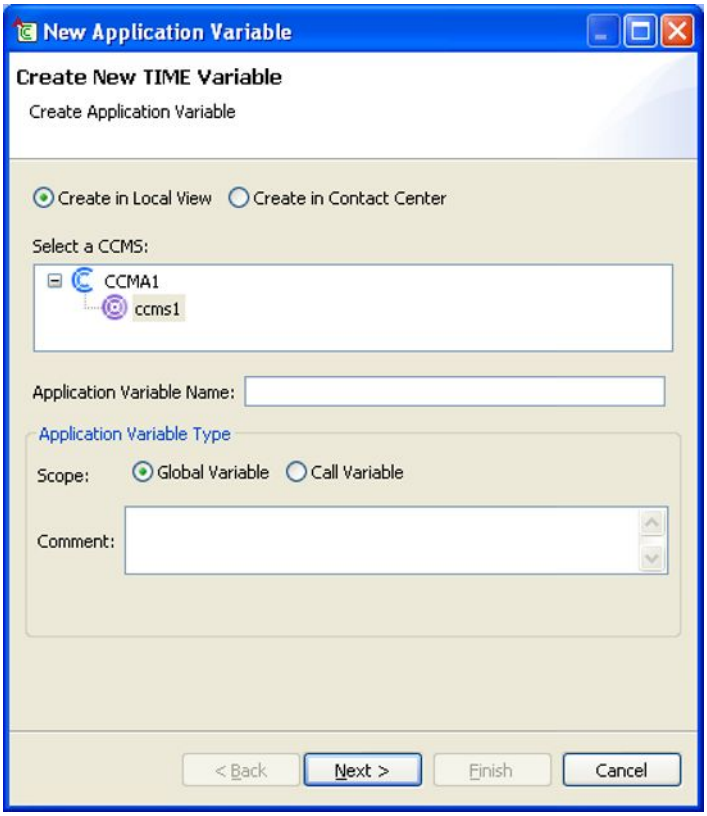
About this task

Create a call variable in Orchestration Designer that you can use in your Local view applications for testing. You must upload the call variable in the Local view to the Contact Center view later. For this example, a variable is required for holidays_gv which lists the days that the business is closed.

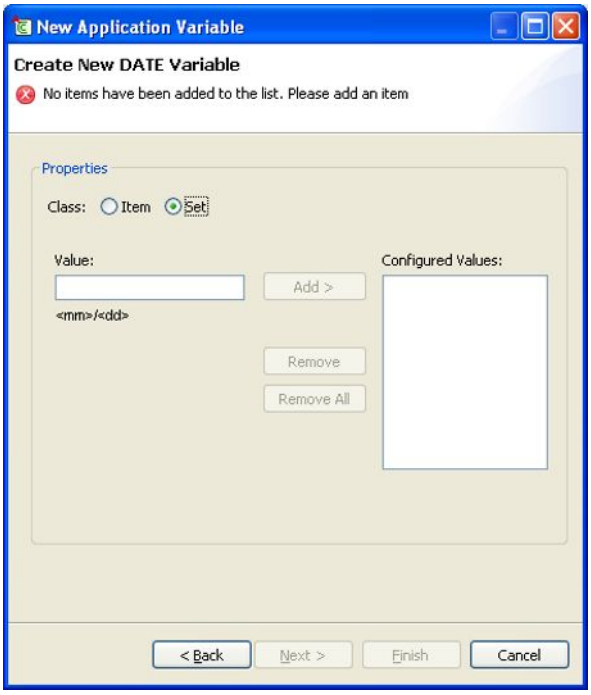
This procedure describes the steps to add the holidays_gv variable. Use the steps to create the other variables you need for this demonstration before you begin adding blocks to the flow.

Procedure

1. In the Local view, expand **Application Variables**.
2. Locate the **DATE** folder.
3. Right-click **DATE**, and click **New**.



4. In the **Application Name** box, type `holidays_gv`.
   The variable name to use is holidays_gv.
5. In the **Application Variable Type**, click **Global Variable**.
   A global variable contains the same value throughout the application.
6. In the **Comment** text box, enter a description for the variable.
7. Click **Next**.



8. Select **Set** as the **Class** type.
9. In the **Value** box, type `01/01`, and then click **Add**.
10. Repeat step 9 for each of the following dates: `04/04`, `12/25`, `12/26`.
    The values appear in the Configured Values box. For this example, the holidays are New Year's day (January 1), Easter (April 4), Christmas (December 25) and Boxing day (December 26). Use the notation mm/dd.
11. Click **Finish**.

Procedure job aid

The following table lists all the variables you need for the application demonstration. Use the preceding steps to create each variable. If you are using a SIP-enabled contact center, you must assign numbers to the RAN routes.

Variable type	Variable name	Values
DATE	holidays_gv	holidays of your contact center 01/01, 04/04, 12/25, 12/26
RAN	continue_to_hold_gv or assigned route number (SIP)	message to say continue to hold

Variable type	Variable name	Values
RAN	closed_hours or assigned route number (SIP)	message to say office is closed
RAN	holidays or assigned route number (SIP)	message to say office is closed for the holiday
RAN	technical_message or assigned route number (SIP)	message to say no agents are available
SKILLSET	Primary_Skillset_gv	name of skillset

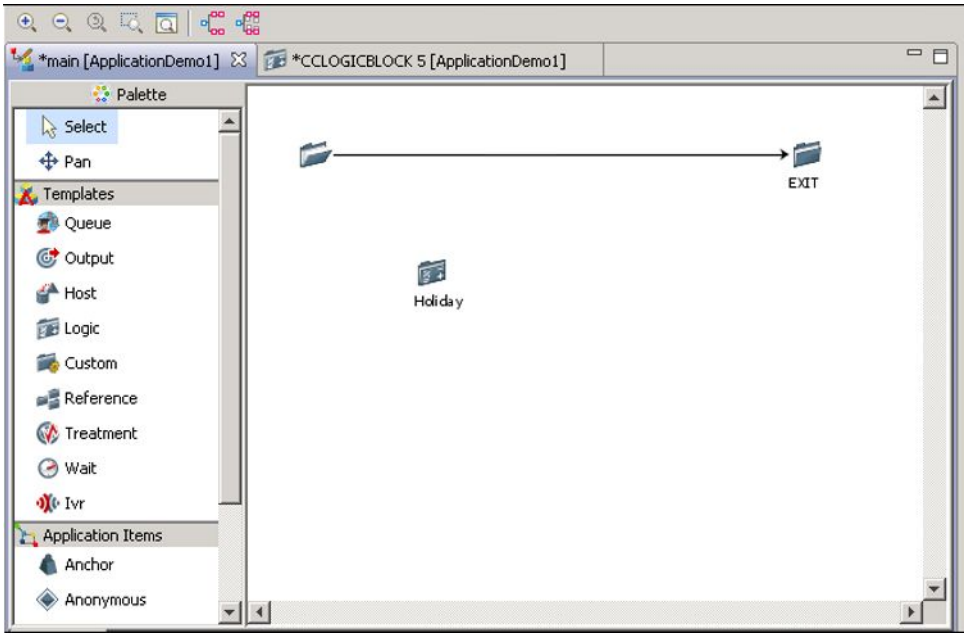
Creating a logic block for closed days

About this task

Create a logic block for this flow application. The first logic block checks the contact center for the days that the contact center is closed. If the contact is received on a holiday, this application plays a message and disconnects the contact.

Procedure

1. From the palette bar, select the logic block icon ().
2. Click the main panel.
   The CCLOGICBLOCK icon appears in the Main Flow Editor.
3. Double-click the **CCLOGICBLOCK** icon to open the details for the block.
4. Click the **Processing** tab.
5. In the **Block Name** box, type a descriptive title for the block. For this example, type the block name, `Holiday`.
6. In the **Description** box, type a description for the block for future reference. In this example, type `Determine holidays for the contact center`.
7. Click the **Main [ApplicationDemo1]** tab to return to the flow application.



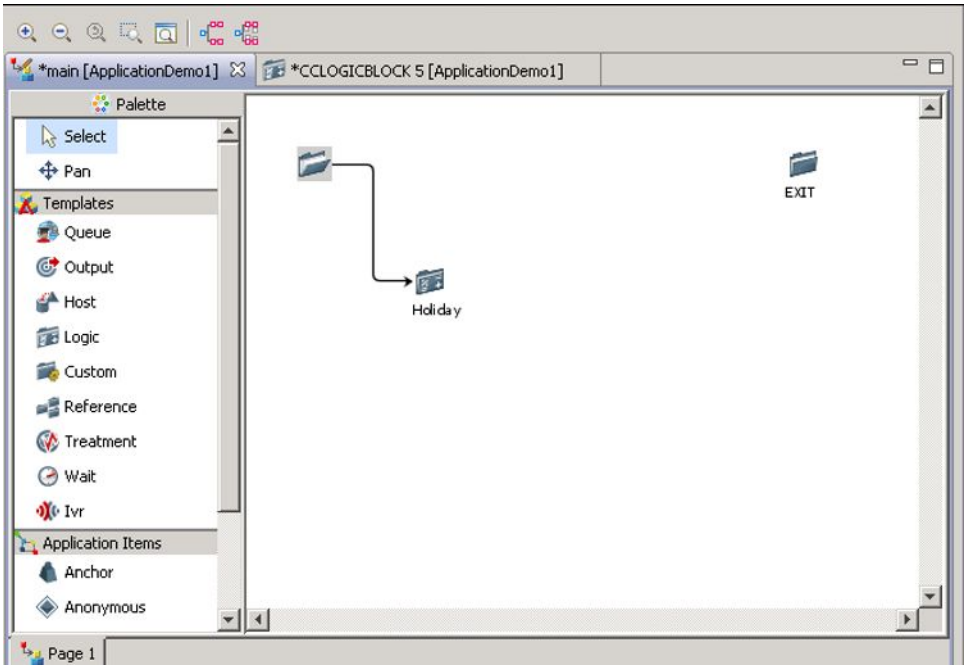
Connecting the start node to the first block

About this task

Connect the start node to the first logic block (Holiday) to configure the flow of calls in your contact center.

Procedure

1. In the **Main [ApplicationDemo1]** panel, select the **Start** node.
2. Right-click the **Holiday** block and select **(Dis)Connect**.
   A line shows the connection between the start folder and the Holiday logic block.



Creating a transition

About this task

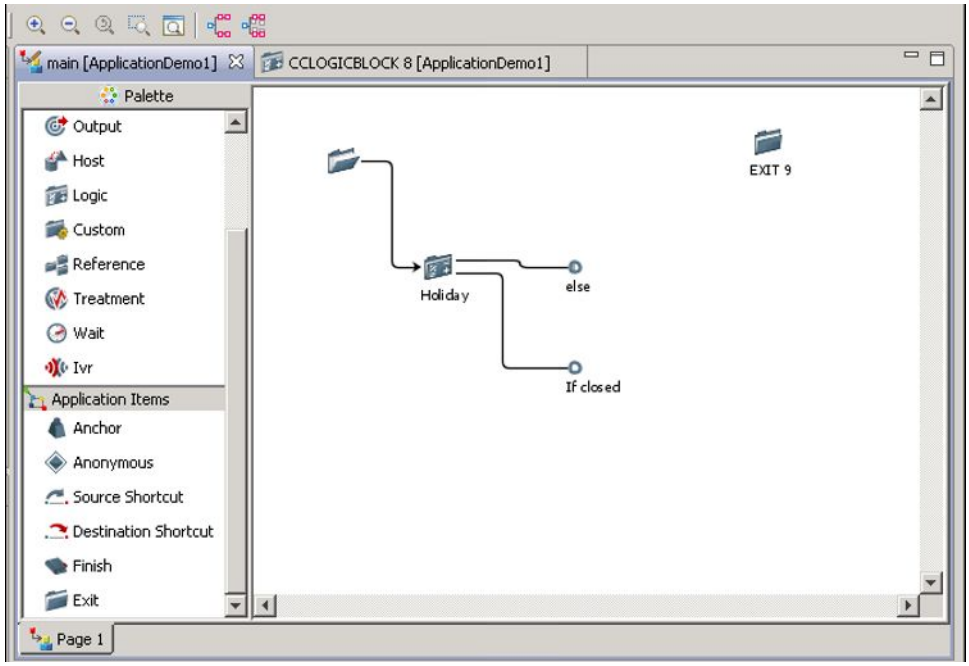
Create a transition for the logic block you created in the previous procedure to provide the process for this procedure. There are two conditions to consider:

- The call is received during a time of day when the contact center is closed.
- The call is received during a time of day when the contact center is open.

Transitions show the conditional statements so that you can view the path of calls for all conditions.

Procedure

1. Right-click the **Holiday** icon, and select **Add Transition**.



2. Right-click the new **tran** node, and select **Rename**.
   The second transition is automatically named else.
3. In the text box, type `If closed`.
4. Press **Enter**.
5. Click **File** > **Save**.
   Every time you save your flow application, the flow is validated. If there are errors in your flow application, you can resolve them after you complete your flow.
6. If you receive a confirmation box, click **OK**.

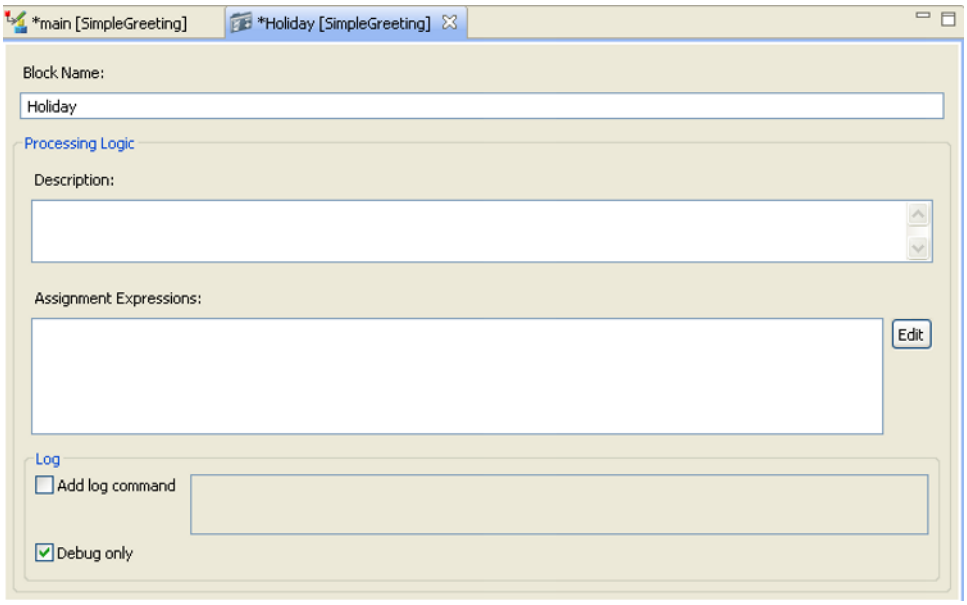
Creating the if closed conditional expression for holidays

About this task

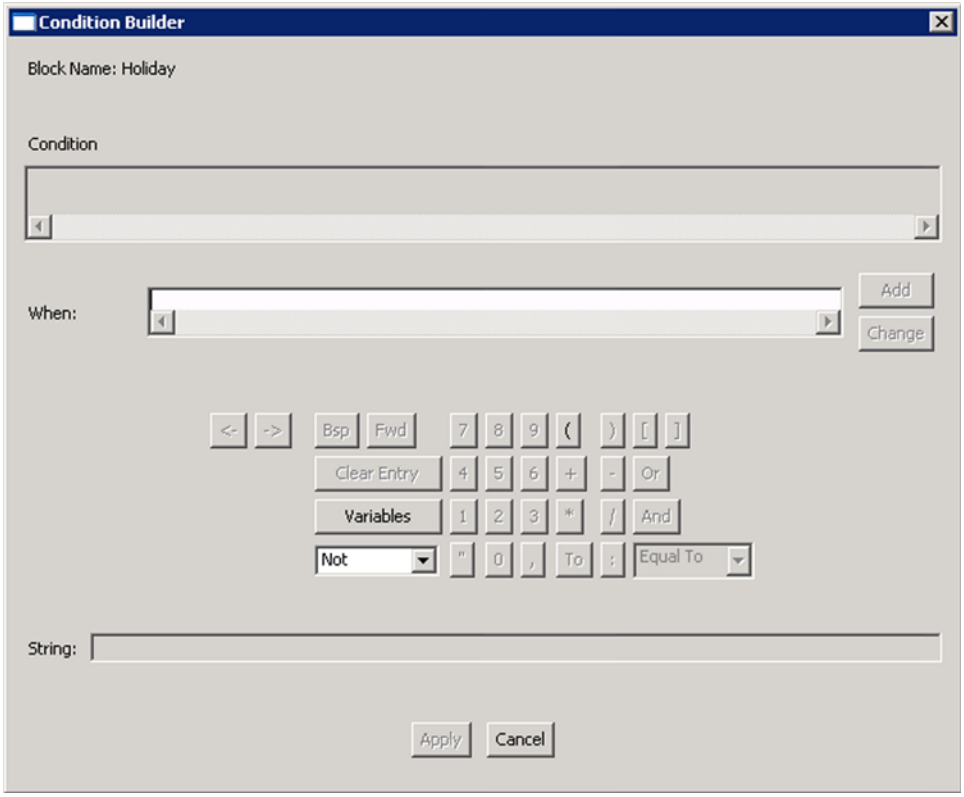
Create the if closed transition conditional expression that transitions calls during closed hours to a RAN message that indicates the business is closed.

Procedure

1. Double-click the **Holiday** logic block icon to open the details for the block.
2. Click the **Transition** tab.
3. Click the **If closed** tab.

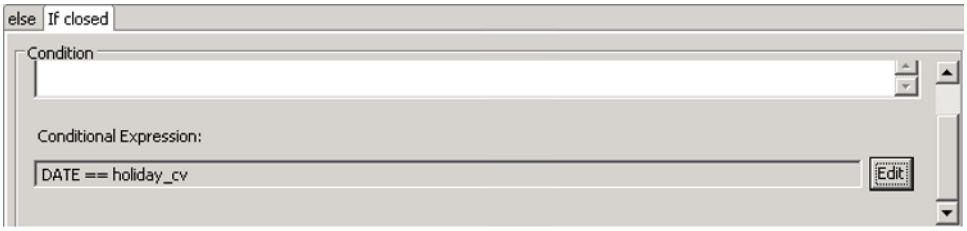


4. On the right of the **Conditional Expression** box, click **Edit**.



The following steps show how to create the conditional expression: DATE Is Equal to Holiday, where Holiday is the value of the variable holidays_gv. DATE is an intrinsic on the server.

5. Click **Variables**.
6. Expand **Intrinsics**, and **TIME**.
7. Select the intrinsic **DATE** from the list.
8. Click **OK**.
9. In the **Equal To** box, select **Equal To**.
10. Click **Variables**.
11. Expand **Application Variables** > **DATE**.
12. Select the **holidays_gv** variable you created earlier.
13. Click **OK**.
14. Click **Add** to add the expression to the condition list.
    The expression uses Orchestration Designer code to produce the expression DATE is equal to Holiday.
15. Click **Apply**.



16. Click **X** to close the **Logic block** tab.
    You do not have to configure conditions for the else statement.
17. Click **File** > **Save**.
    Every time you save your flow application, the flow is validated. The current errors in the flow application are dangling transitions. You resolve the dangling transitions while you perform the next procedure.
18. If you receive a confirmation box, click **OK**.

Creating recorded announcements

Before you begin

- Create a RAN Announcement to play to a caller when the contact center is closed for a public holiday.
- Assign a RAN variable or a RAN number (SIP) for the closed message. See Creating a variable in Orchestration Designer on page 55.

About this task

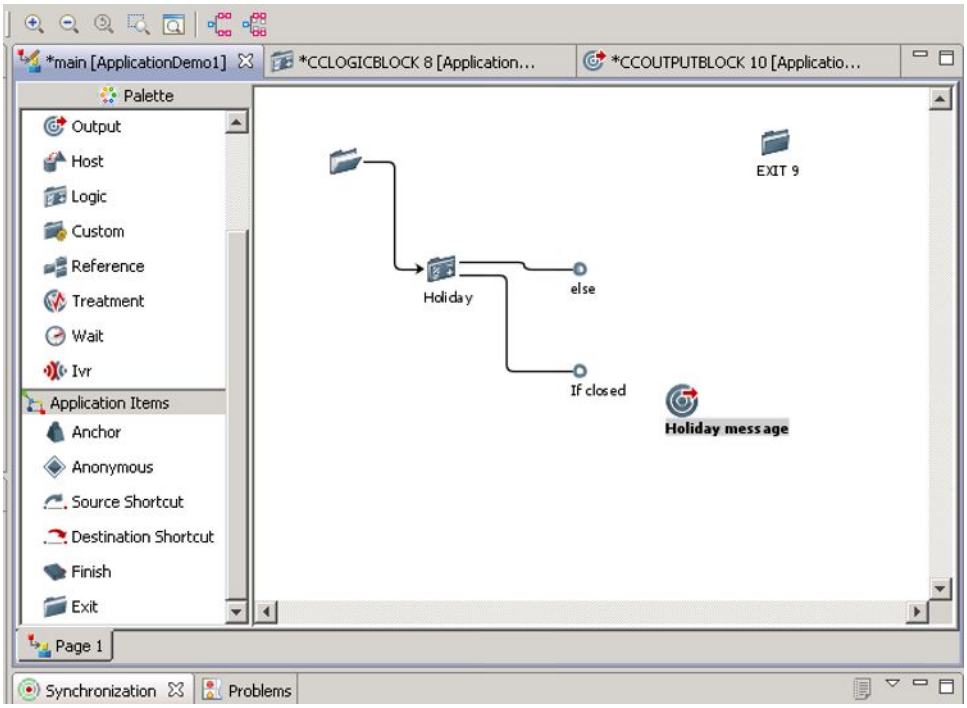
Create a step in the application to play a recorded announcement (RAN) to the customer. Use an Output block to play a recorded announcement (RAN) to the customer telling them that the contact center is closed for the holiday.

Procedure

1. From the palette bar, select the output block icon ( ).
2. Click in the main panel to the right of the **If closed** transition.
   The CCOUTPUTBLOCK icon appears in the Main Flow Editor. The block is given a default name starting with the block type and ending with a unique number.
3. Double-click the **CCOUTPUTBLOCK** icon to open the details for the block.
4. Click the **Setup** tab.
5. In the **Block Name** box, type a descriptive title for the block. For this example, type the block name `Holiday message`.
6. In the **Description** box, type a description for the block for future reference. In this example, type `Play holiday message`.
7. Click the **Prompts** tab.



8. If not already selected, under **Output Options**, click **RAN** to play a recorded announcement.
9. Click the RAN Route button ( ) to enable you to select the recorded announcement (RAN) route variable for holidays.
10. Click **Browse**.
11. Expand **Application Variables**, and **RAN**, then select your recorded announcement (RAN) variable.
12. Click **OK**.
13. Click the **Main [ApplicationDemo1]** tab to return to the flow.
    Notice the block name is updated.



14. Click the **X** to close the **Output block** tab.
15. Click **File** > **Save**.
    Every time you save your flow application, the flow is validated. If there are errors in your flow application, you can resolve them after you complete your flow.
16. If you receive a confirmation box, click **OK**.

Connecting the transition to the holiday message

About this task

Connect the If closed transition to the holiday message to create the sequence of events in the flow application.

Procedure

1. Select the **If closed** node.
2. Right-click the **Holiday message** block and select **(Dis)Connect**.
   A line shows the connection between the If holiday block and the Holiday message block.

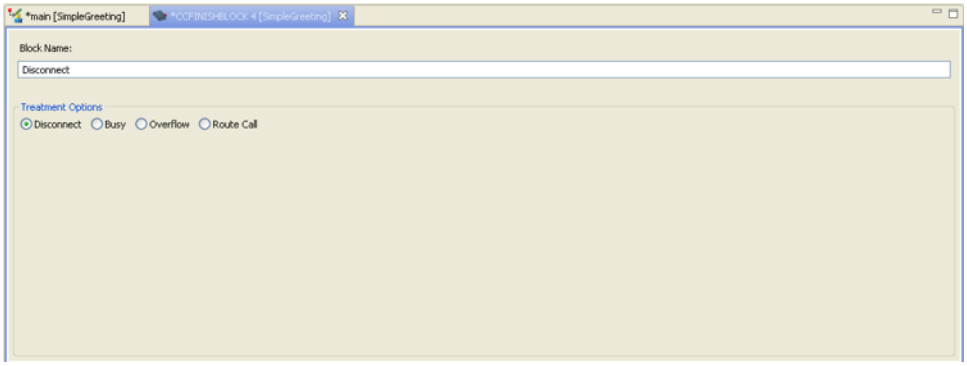
Creating a disconnect block

About this task

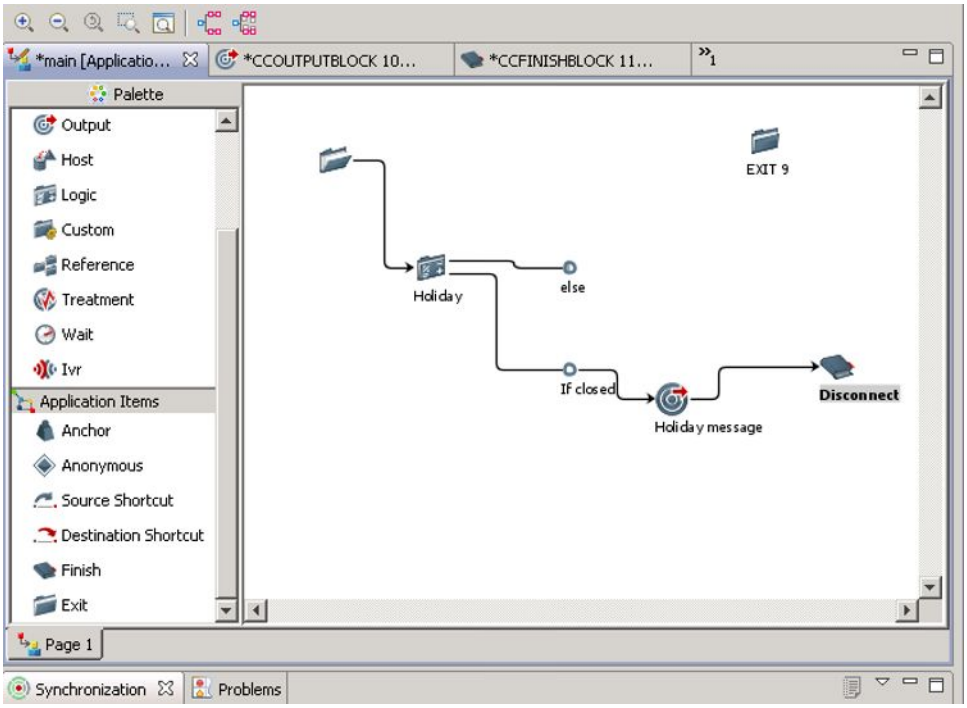
Create a disconnect block to disconnect the call. Disconnecting the call terminates the call and releases the line.

Procedure

1. From the palette bar, select the Finish block icon ( ) with your mouse.
   The Finish block icon is near the bottom of the palette bar.
2. Click in the main panel to the right of the **Holiday message** icon.
   The CCFINISHBLOCK icon appears in the Main Flow Editor. The block is given a default name starting with the block type and ending with a unique number.
3. Double-click the **CCFINISHBLOCK** icon.



4. In the **Block Name** box, type a descriptive title for the block. For this example, type the name `Disconnect`.
5. If not already selected, under **Treatment Options**, click **Disconnect**.
6. Click the **Main [ApplicationDemo1]** tab to return to the flow application.
7. Click the **Holiday message** block.
8. Right-click the **Disconnect** block and select **(Dis)Connect**.
   A line shows the connection between the Holiday message and the Disconnect block.



9. On the **Finish block** tab, click the **X** to close it.
10. Click **File** > **Save**.
    Every time you save your flow application, the flow is validated. Notice that the number of errors is different because one of the conditions for the Holiday logic block is complete.
11. If you receive a confirmation box, click **OK**.

Creating block to identify closed hours

Before you begin

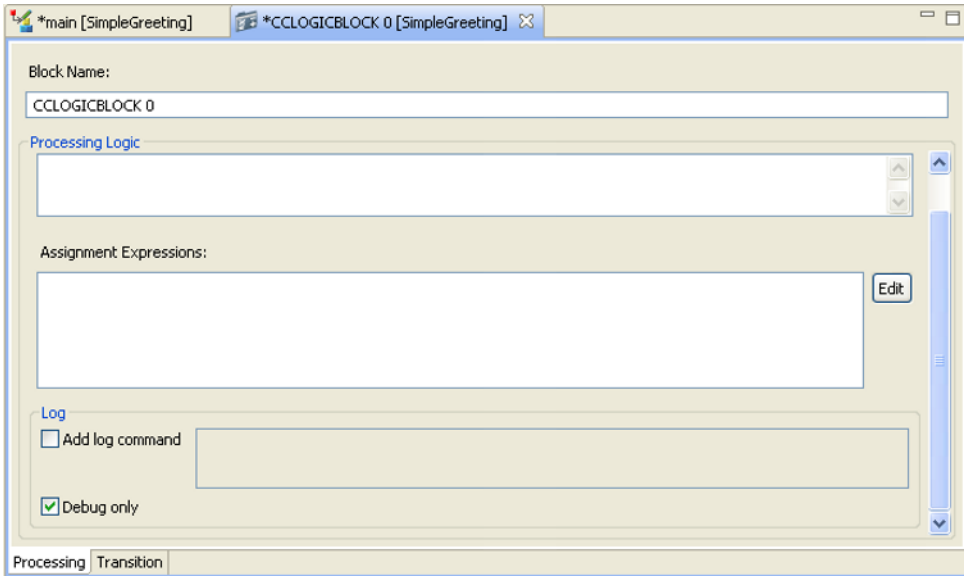
- Drag the EXIT node to the bottom right-hand side of the Flow editor so that it is out of the way for now.

About this task

Earlier in this demonstration, you created a logic block for configuring the holidays for the contact center. Now, create a similar logic block for the hours that the contact center is closed on the other days. This procedure combines creating the block and the transitions to other blocks.

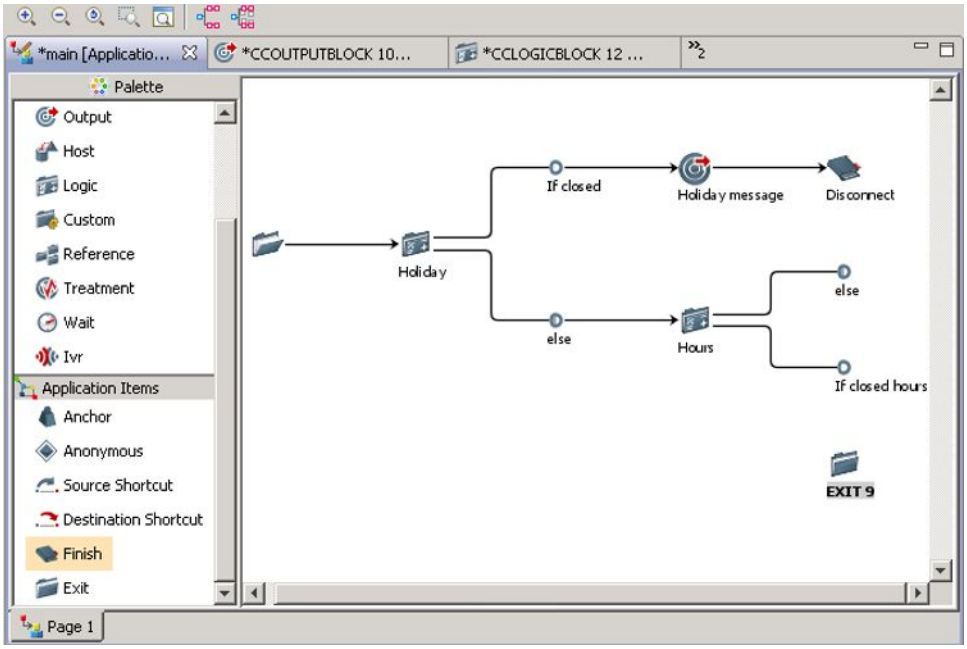
Procedure

1. From the palette bar, select the logic block icon ( ).
2. Click the main panel next to the else transition from the Holiday block.
3. Double-click the **CCLOGICBLOCK** icon to open the details for the block.
4. Click the **Processing** tab.



5. In the **Block Name** box, type a descriptive title for the block. For this example, type the block name, `Hours`.
6. In the **Description** box, type a description for the block for future reference. In this example, type `Determine hours for the contact center`.

7. Click the **Main [ApplicationDemo1]** tab to return to the flow application.
   Notice that the Logic block name now reflects the name from the Block Name box.
8. Right-click **Hours**, and select **Add Transition**.
   The objects in your flow application may overlap.
9. To prevent overlapping, click Incremental Layout ( ) in the Flow Editor toolbar to align the objects in the flow application.
   Full layout mode realigns all of the blocks in the first column. Since not all of the objects are connected, preserve the current layout by choosing Incremental Layout.
10. Right-click the new **tran** node and click **Rename**.
11. Change the name of the transition to **If closed hours**.
12. Click the **else** node associated with the Holiday block.
13. Right-click the **Hours** logic block and select **(Dis)Connect**.
14. On the **Logic Block** tab, click the **X** to close the tab.



15. Click **File** > **Save**.
    Every time you save your flow application, the flow is validated. If there are errors in your flow application, you can resolve them after you complete your flow.
16. If you receive a confirmation box, click **OK**.

Creating the If closed conditional expression for the hours

Before you begin

- Ensure that you have a RAN variable or value (SIP) for a message to play when the contact center is closed.

About this task

Create the If closed hours transition conditional expression that plays a RAN message to state the contact center is closed.

Procedure

1. Double-click the **Hours** block to open the details for the block.
2. Click the **Transition** tab.
3. Click the **If closed hours** tab.
4. On the right of the **Conditional Expression** box, click **Edit**.

   Use the Condition and Expression Builder to create the following expression: Time of day Is Not Equal to 09:00 To 17:00
5. Since Time of day is a time intrinsic, click **Variables**.
6. In the **Variables** box, expand **Intrinsics** > **Time**.
7. Click **TIME OF DAY**.
8. Click **OK**.
9. In the expression box on the lower right side of the controls, select **Not Equal To**.
10. Type `09:00`.
11. Select **To**.
12. Type `17:00`.
    This value represents 5:00 p.m. on the 24-hour clock.
13. Click **OK**.
14. Click **Add**.
15. Click **Apply**.
16. Click X to close the **Logic Block** tab.
17. Click **File** > **Save**.
18. If you receive a confirmation box, click **OK**.

Completing the closed hours branch of the flow application

Before you begin

- Create the closed hours message.
- Create the RAN variable or value (SIP) to play to the caller for the closed hours message.

About this task

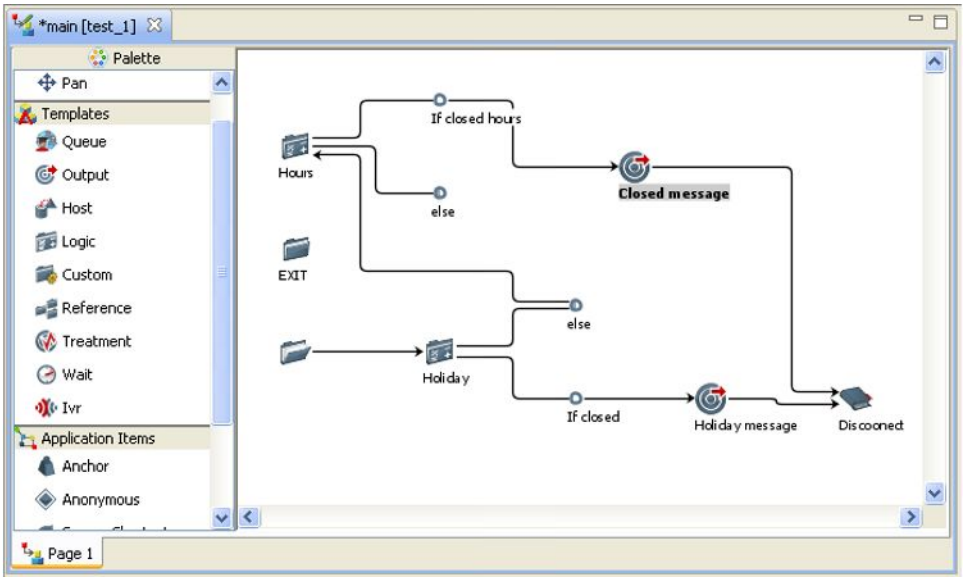
Complete the closed hours branch of the flow application by adding the Output block, the Disconnect block, and connecting the blocks in the correct order of the flow diagram.

The Output block plays a recorded announcement (RAN) to the customer about the closed hours.

Procedure

1. From the palette bar, select the Output block icon ( ).
2. Click the main panel.
3. Double-click the **CCOUTPUTBLOCK** icon to open the details for the block.
4. Click the **Setup** tab.
5. In the **Block Name** box, type a descriptive title for the block. For this example, type the block name `Closed message`.
6. In the **Description** box, type a description for the block for future reference.
7. Click the **Prompts** tab.
8. If necessary, under **Output Options**, click **RAN** to play the recorded announcement.
9. There is no previous treatment in the flow, so do not select the **Silence Previous Treatment** check box.
10. Click the **RAN Route** button ( ) to enable you to select the recorded announcement (RAN) variable.
11. Click **Browse**.
12. Expand the **Application Variables**, and **RAN** folder, then select your recorded announcement (RAN) variable.
13. On the **Output block** tab, click **X**.
14. Click the **Main [ApplicationDemo1]** tab to return to the flow application.

15. Select the **If closed hours** node.
16. Right-click the **Closed message** block and select **(Dis)Connect**.
17. Click the **Closed message** block.
18. Right-click the **Disconnect** block you created earlier and select **(Dis)Connect**. A line shows the connection between the Closed message and the Disconnect block.



19. Click **File** > **Save**.
20. If you receive a confirmation box, click **OK**.

Creating a logic block for out-of-service skillset

Before you begin

- Configure a skillset in your contact center.
- Create a SKILLSET variable named Primary_Skillset_gv. See Creating a variable in Orchestration Designer on page 55.

About this task

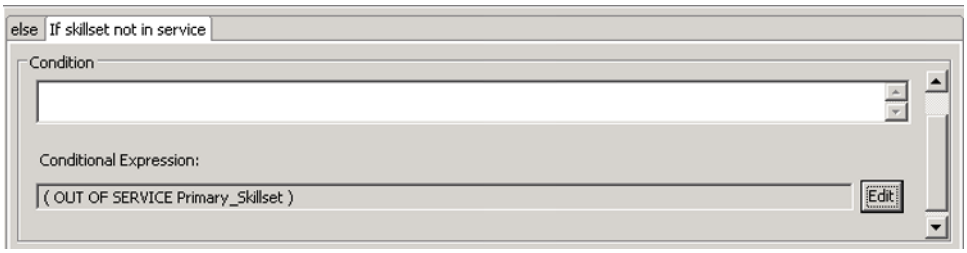
Create the logic block for this flow application that determines if the skillset is not in service.

The logic block checks to see if the primary skillset is out of service. If the skillset is in service, the contact is queued. If the skillset is not in service, another treatment is given to the contact.

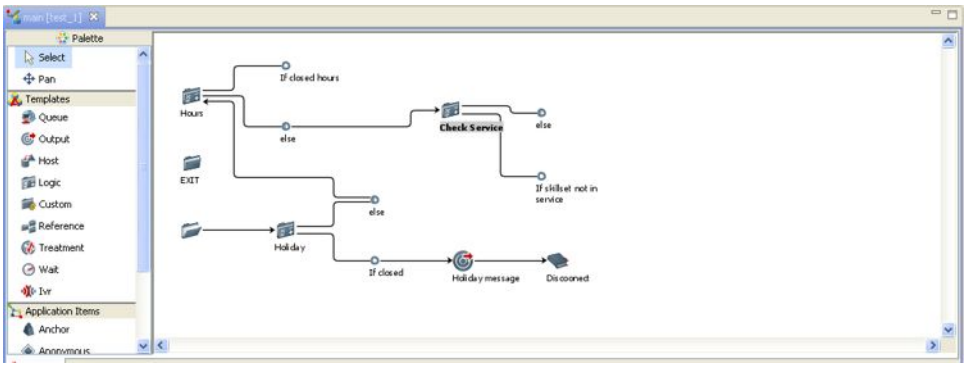
Procedure

1. On the palette bar, select the logic block icon ().
2. Click the main panel beside the unresolved **else**.
3. Click the **else** node.
4. Right-click the new logic block, and click **(Dis)Connect**.
5. Double-click the **CCLOCIGBLOCK** icon to open the details for the block.
6. Click the **Processing** tab.
7. Complete the **Block Name** and **Description** for the block. Name the block `Check service`.
8. Click the **Main [ApplicationDemo1]** tab to return to the flow application.
9. Right-click the **Check service** icon, and select **Add Transition**.
10. Right-click the new **tran** node and click **Rename**.
11. In the text box, type `If skillset not in service`.
12. Click **OK**.
13. Double-click the **Check service** block.
14. Click the **Transition** tab.
15. Click the **If skillset not in service** tab.
16. Beside **Conditional expression**, click **Edit**.
    Use the Condition builder to formulate the expression OUT OF SERVICE using the Skillset Intrinsic OUT OF SERVICE and the application manager data representing your primary skillset.
17. In the Condition Builder, click **(**.
18. Click **Variables**.
19. In the **Chooser** box, choose **Instrinsics** > **Skillset** > **OUT OF SERVICE**.
20. Click **OK**.
21. Click **Variables**.
22. In the **Chooser** box, choose **Application Variables** > **Skillset** > **Primary_Skillset_gv**.
23. Click **OK**.
24. In the Condition Builder, click **)**.
25. Click **Add**.

26. Click **Apply**.



27. Click **Main [ApplicationDemo1]** to return to the flow application.
28. Close the **Logic Block** tab.
29. Click **File** > **Save**.
30. If you receive a confirmation box, click **OK**.



31. To prevent the overlapping, click (Incremental Layout) in the Flow Editor toolbar.

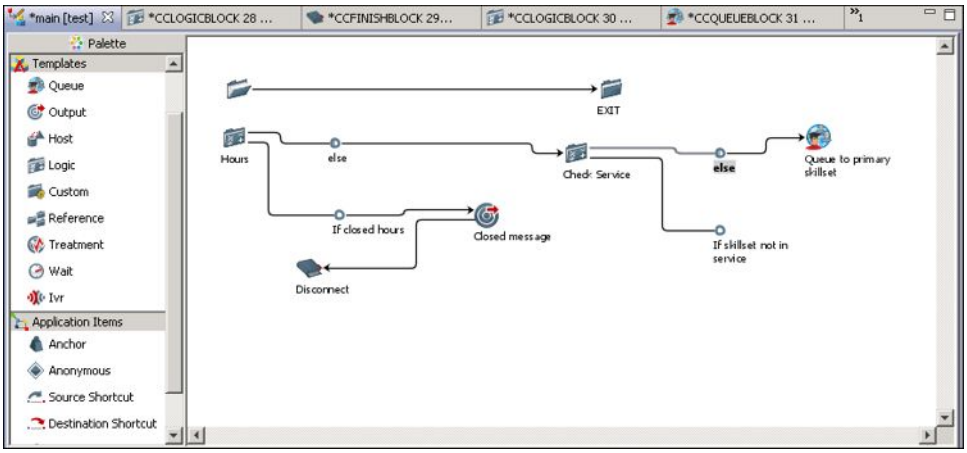
Creating a queue to skillset block

About this task

Create a queue to skillset block for the primary skillset if it is in service.

Procedure

1. On the palette bar, select the queue block icon ( ).
2. Click the main panel after the else condition.



3. Double-click the **CCQUEUEBLOCK** icon.
4. Click the **Setup** tab.
5. Complete the **Block Name** and **Description** for the block. Name the block `Queue to primary skillset`.
6. Click the **Queue** tab.
7. Under **Command**, select **Add**.
   You add the contact to the queue in the flow application.
8. Under **Queue To**, select **Skillset**.
9. Beside the **Skillsets** table, click **Add**.
10. Expand **Application Variables** > **SKILLSET**, and select **Primary_Skillset_gv**.
11. Click **OK**.
12. Beside **Priority**, use the scroll buttons to assign a priority of 3 to the contact.
    Assign a medium priority to the contact to ensure that the system handles the contact quickly.
13. In the **Main [ApplicationDemo1]** flow application, click **else**.
14. Right-click **Queue to primary skillset**, and click **(Dis)Connect**.
15. Click **X** on the Queue block to close it.
16. Click **File** > **Save**.
17. If you receive a confirmation box, click **OK**.

Creating a treatment block to play music

Before you begin

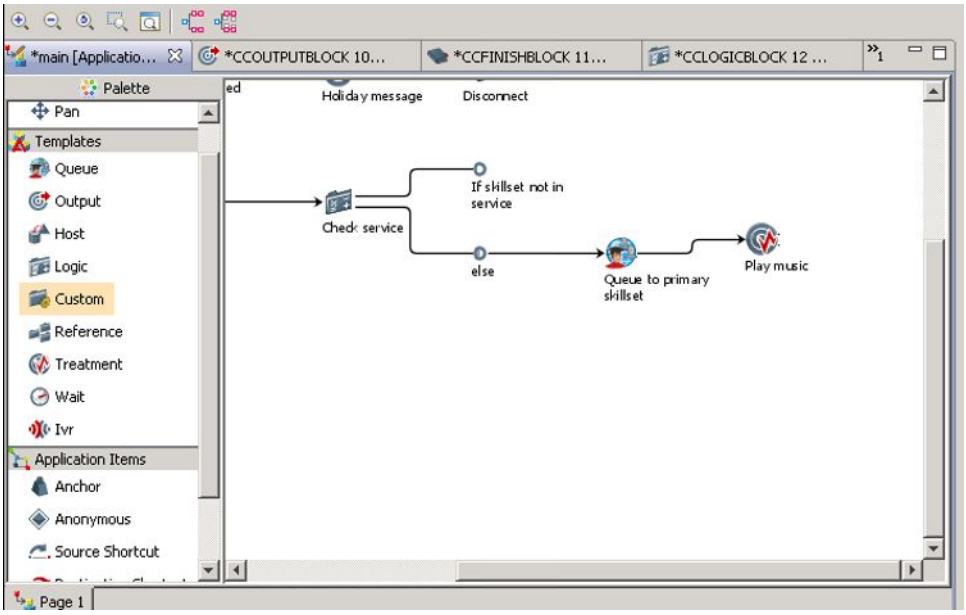
- Create the music routes to play the music to your callers.
- Know how to connect two blocks in a flow application. See Connecting the start node to the first block on page 59.

About this task

Create a treatment block to play music to the caller while they wait in queue. By default, the application variable CC_DefaultMusic is configured on your system.

Procedure

1. From the palette bar, select the treatment block icon ().
2. Click the main panel next to the **Queue to primary skillset** block.
3. Double-click the **CCTREATMENT** icon to open the details for the block.
4. Click the **Setup** tab.
5. In the **Block Name** box, type a descriptive title for the block. For this example, type the block name `Play music`.
6. Type an optional description for the treatment block.
7. Click the **Treatment** tab.
8. Under **Treatment Options**, click **Music**.
9. Click the music route button () to enable you to select the music route variable.
10. Click **Browse** to select the music variable.
11. Click **OK**.
12. Select the minimum duration of the played music.
13. Click the **Main [ApplicationDemo1]** tab to return to the flow application.
14. Close the new **Treatment** tab.
15. Connect the **Queue to primary skillset** block to the **Play music** block.



16. Click **File** > **Save**.
17. If you receive a confirmation box, click **OK**.

Creating a logic block to check if the call is queued

About this task

Create another logic block for this flow application that checks after music is played for a while to see if the call is queued.

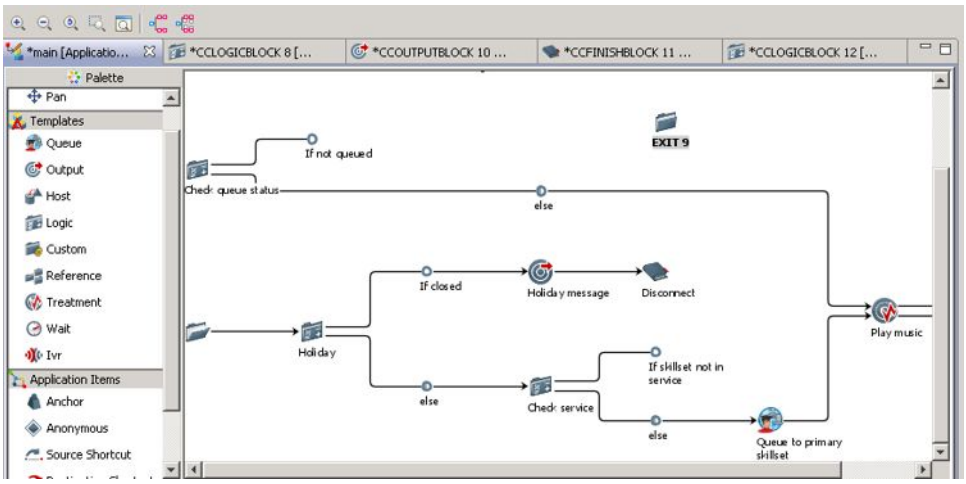
If the call is not queued, play a message. Otherwise, check the primary skillset again to see if it is out of service.

Procedure

1. On the palette bar, select the logic block icon ( ).
2. Click the main panel to the right of the **Play music** icon.
3. Connect the **Play music** block to the new logic block.
4. Double-click the **CCLOGICBLOCK** icon to open the details for the block.
5. Click the **Processing** tab.
6. Complete the **Block Name** and **Description** for the block. Name the block `Check queue status`.

7. Click the **Main [ApplicationDemo1]** tab to return to the flow application.
8. Right-click the **Check queue status** icon, and select **Add Transition**.
9. Right-click the new **tran** node and click **Rename**.
10. In the text box, type **If not queued**.
11. Double-click the **Check queue status** (logic) block.
12. Click the **Transition** tab.
13. If not open by default, click the **If not queued** tab.
14. Beside **Conditional Expression**, click **Edit**.

    Use the Condition builder to formulate the expression NOT QUEUED using intrinsics using the following steps.
15. Select **Not** from the list.
16. Click **Variables**.
17. Select **Intrinsics**, **Call**, and **Queued**.
18. Click **OK**.
19. Click **Add**.
20. Click **Apply**.
21. Close the **Logic Block** tab.
22. Click **File** > **Save**.
23. If you receive a confirmation box, click **OK**.



Rechecking skillset service

Before you begin

- Know how to create a logic block to check if the skillset is out of service. See Creating a logic block for out-of-service skillset on page 72.

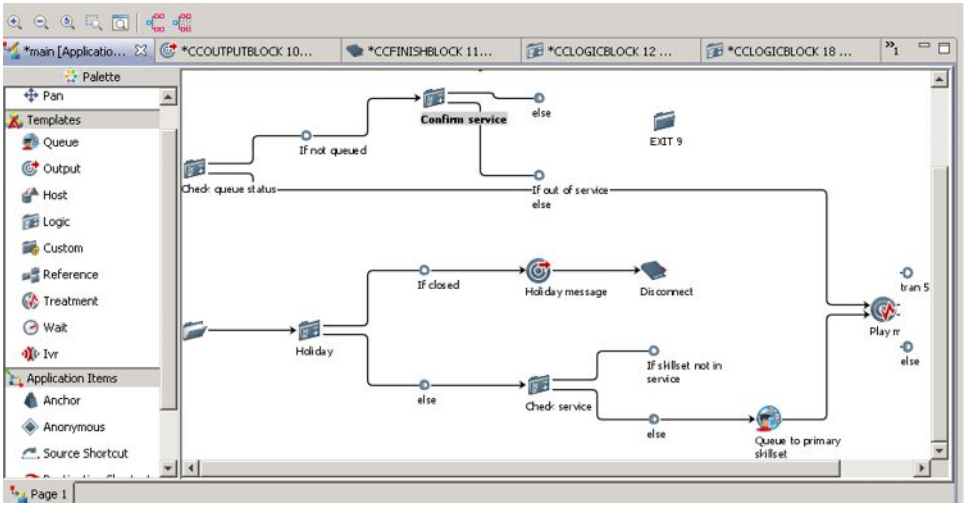
About this task

If the contact is not queued, best practices for application writing suggests that you check the skillset to ensure that it is not out of service now, so the caller is not left without feedback.

Procedure

1. On the palette bar, select the logic block icon ().
2. Click the main panel beside the **If not queued**.
3. Click the **If not queued** node.
4. Right-click the new logic block, and click **(Dis)Connect**.
   You need not follow a particular order to link icons or configure them.
5. Double-click the new **CCLOGICBLOCK** icon to open the details for the block.
6. Configure the Logic block with the title `Confirm service`.
7. Click the **Main [ApplicationDemo1]** tab to return to the flow application.
8. Right-click the **Confirm service** icon, and select **Add Transition**.
9. Right-click the new **tran** node and click **Rename**.
10. In the text box, type **If out of service**.
11. Click **OK** to complete the name of the transition condition.
12. Double-click the **Confirm service** block.
13. Click the **Transition** tab.
14. Click the **If out of service** tab.
15. Click **Edit**.
    Create the Conditional expression (OUT OF SERVICE Primary_Skillset_gv) in the Condition Builder using the following steps:
16. Click **Add**.
17. Click **Apply**.
18. Click **Main [ApplicationDemo1]** to return to the flow application.
19. Close the **Logic Block** tab.
20. Click **File** > **Save**.

21. If you receive a confirmation box, click **OK**.



Playing a technical message for disconnecting contact

Before you begin

- Create a RAN variable for informing users about the technical issues.
- Know how to add an output block for creating a message in your application. See Creating recorded announcements on page 64.
- Know how to add a finish block for disconnecting the contact. See Creating a disconnect block on page 66.

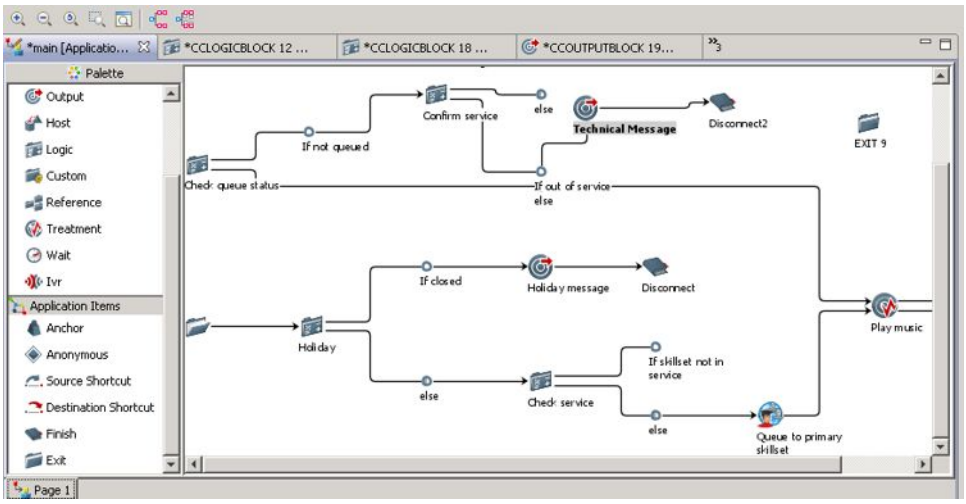
About this task

In this example, if the calls are queued, then the calls are queued to a skillset and are processed. If the contact cannot be queued, it is best practice to play a message to the customer, and disconnect the contact.

Procedure

1. Add an **Output** block in your application beside the else condition after the **If out of service** node.
2. Name the new Output block, Technical Message.
3. Configure the Technical Message block to play the technical message RAN.
4. Connect the **If out of service** node to the **Technical Message** block.
5. Add a **Finish** block in your application beside the **Technical Message** block.

6. Name the new Finish block, `Disconnect2`.
   You cannot name two blocks the same in your application.
7. Connect the **Technical Message** block to the **Disconnect2** block.



8. Click **File** > **Save**.
9. If you receive a confirmation box, click **OK**.

Copying a block

Before you begin

- Ensure the queue to skillset block is in your flow application.

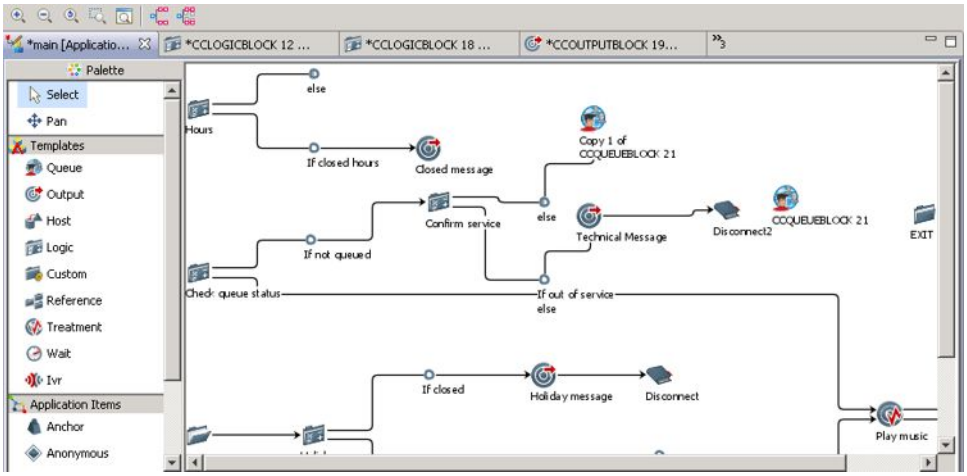
About this task

You can copy and paste blocks in your application to create a new block. After the service is confirmed, you want to queue to the same skillset used previously in the application.

Procedure

1. Select the **Queue to primary skillset** block in your flow application.
2. Right-click, and click **Copy**.
3. Select the Application Demo flow, right-click, and click **Paste**.
4. Locate the **Copy 1 of Queue to primary skillset** block.
5. Drag the **Copy 1 of Queue to primary skillset** block to the **else** node following the **Confirm service** block.

6. Connect the else node to the Copy 1 of Queue to primary skillset block.
7. Click **File** > **Save**.
8. If you receive a confirmation box, click **OK**.



Playing a hold message

Before you begin

- Record an announcement for on hold customers and configure the RAN in the Application variables named continue_to_hold_gv.

About this task

Add another output block to the flow application to play a continue to hold message while the caller waits for the call to be answered.

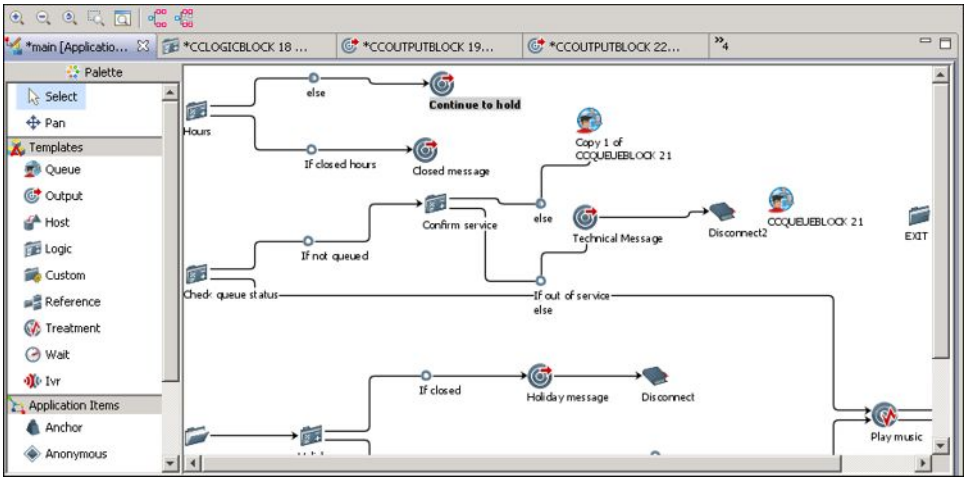
Play the hold message to all customers who are queued. Since there are two queue to skillset blocks, you can connect this block twice.

Since a music block appears before the hold message in the flow, when the caller is not listening to the message, the caller listens to music.

Procedure

1. From the palette bar, select the Output block icon ().
2. Click the main panel beside the Hours block **else** transition.
3. Double-click the **CCOUTPUTBLOCK** icon to open the details for the block.
4. Click the **Setup** tab.

5. In the **Block Name** box, type a descriptive title for the block. For this example, type the block name `Continue to hold`.
6. In the **Description** box, type a description for the block for future reference.
7. Click the **Prompts** tab.
8. If not already selected, under **Output Options**, click **RAN** to play the recorded announcement.
9. Click the RAN Route button ( ) to enable you to select the recorded announcement (RAN) file.
10. Click **Browse**.
11. Expand the **Application Variables**, and **RAN** folder, then select your recorded announcement (RAN) file.
12. Click **OK**.
13. Close the new tab.
14. Connect the **else** node of the **Hours** block to the **Continue to hold** block.
15. Connect the **Copy 1 of Queue to primary skillset** block to the **Continue to hold** block.
16. Click **File** > **Save**.
17. If you receive a confirmation box, click **OK**.



Creating a wait block

Before you begin

- Recall how to connect two blocks in your flow application. See Connecting the start node to the first block on page 59.

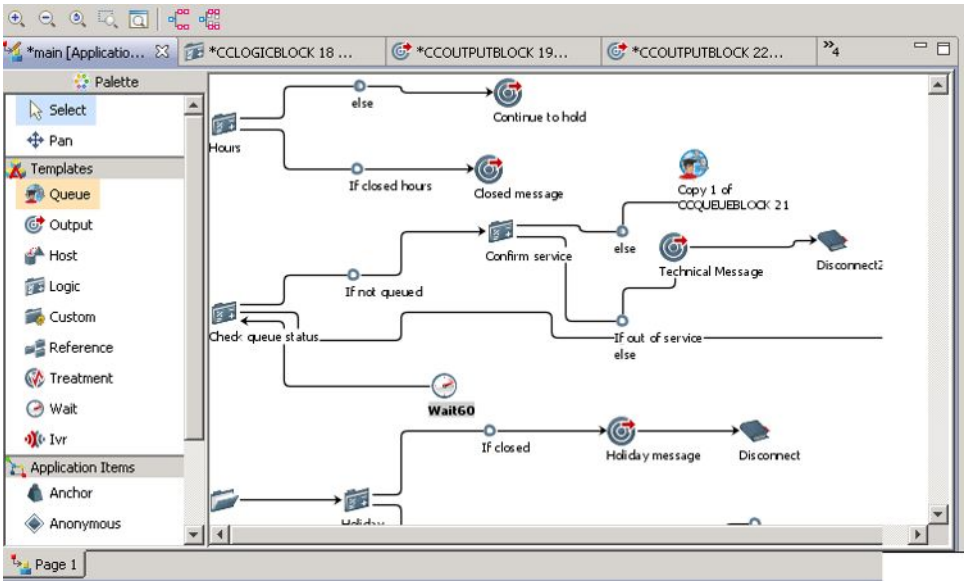
About this task

Create a wait block in your contact center to ensure that there is a time lapse between two steps in the call flow.

You can define a minimum wait time in the properties for your Orchestration Designer. See Configuring application preferences on page 289.

Procedure

1. From the palette bar, select the Wait block icon ().
2. Click the main panel.
3. Double-click the **CCWAITBLOCK** icon to open the editor.
4. Click the **Wait** tab.
5. In the **Block Name** box, type a name for the Wait block. Type `Wait60`.
6. In the **Duration** box, select the duration for the wait time. In this example, replace the default time with a wait time of 60 seconds.
   The wait time is automatically increased by 2 seconds, based on the default value for the flow application preferences.
7. On the **Wait block** tab, click **X**.
8. Connect the **Continue to hold** icon to the **Wait60** icon.
9. Connect the **Wait60** block to the **Check queue status** block to repeat the check to see if the contact is queued.



10. Click **File** > **Save**.
11. If you receive a confirmation box, click **OK**.

Exiting calls from the flow application

Before you begin

- Know how to copy the previous technical message block. See Copying a block on page 81.

About this task

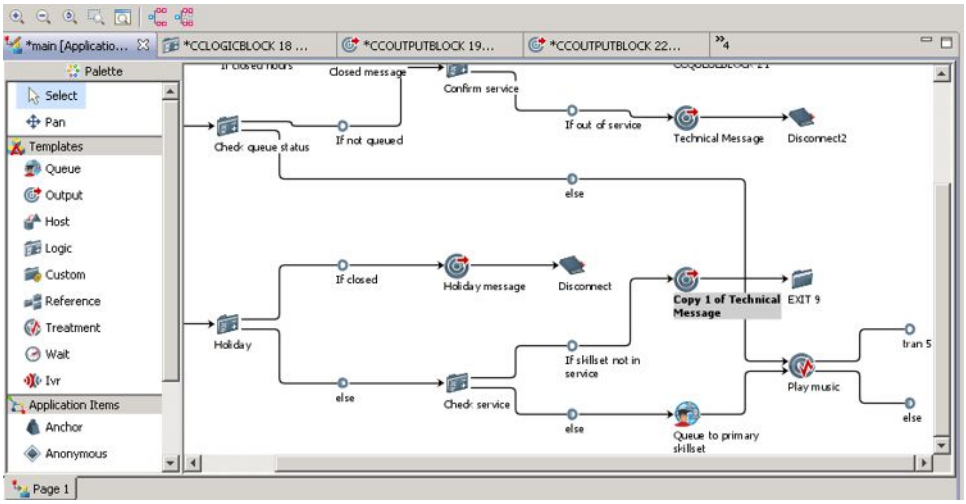
Exit calls from the flow application when they are complete. In this example, if the calls are queued, then the calls are queued to a skillset and are processed. If the contact cannot be queued, it is best practice to play a message to the customer, and disconnect the contact.

If you review the Problems view, there is currently an error about a dangling transition. The Problems view is a good way to ensure that you have connections between all of your blocks.

Procedure

1. Copy the Technical Message block to your application beside the else condition after the **If skillset not in service** node.
2. Configure the Copy 1 of Technical Message block to play the technical message RAN.

3. Connect the **If skillset not in service** node to the **Copy 1 of Technical Message** block.
4. Locate the **EXIT** node in the flow application. Move it to the right side of the **Copy 1 of Technical Message** block.
5. Select the **Copy 1 of Technical Message** block.
6. Right-click the **Exit** block, and click **(Dis)Connect**.



7. Click **File** > **Save**.
8. If you receive a confirmation box, click **OK**.

Synchronizing the views

Before you begin

- Start Orchestration Designer from the Contact Center Manager Administration application. If you are working on a local copy of Orchestration Designer in your network, you must close it.

About this task

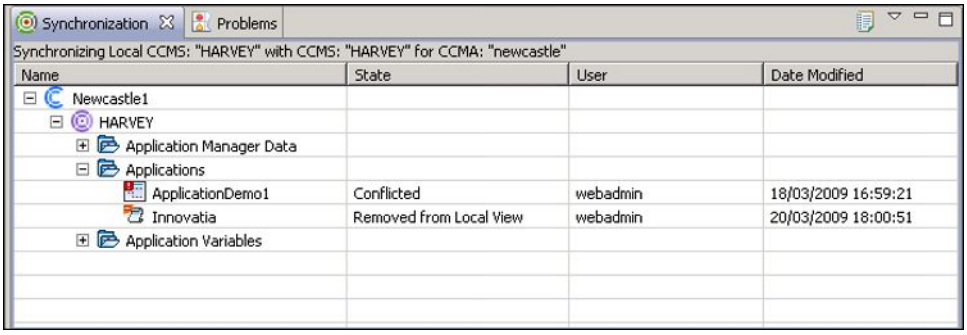
Synchronize the Local view with the Contact Center view to add your newly-created application and the variables to the Contact Center application.

Procedure

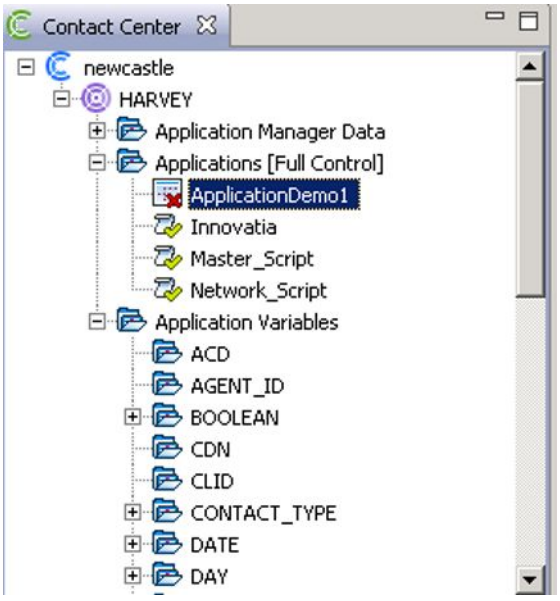
1. Select the Contact Center Manager Server in the Local view.

If the CCMS server in the Local view and the Contact Center view contain the same content, no objects appear in the Synchronization view.

2. Right-click, and click **Synchronize**.
3. In the Synchronization view, expand the Contact Center Manager Server.
4. Expand the **Applications** folder.
   Your Synchronization view may be different than the sample shown below.



5. Select **ApplicationDemo1**.
6. Right-click, and then click **Add to Contact Center**.
   The item is removed from the Synchronization view when you add the application to the Contact Center view.



7. Repeat the steps for the variables you created:
   - holidays_gv (DATE)

- closed_hours (RAN)
- continue_to_hold (RAN)
- holidays (RAN)
- technical_message (RAN)
- Primary_Skillset_gv (SKILLSET)

Procedure job aid

The following text shows the script text equivalent for the application presented in this chapter. Orchestration Designer generates this code.

```
SECTION HOLIDAY
/* Determine holidays for the contact center. */
IF (DATE = holiday_gv) THEN
     EXECUTE HOLIDAY_MESSAGE
END IF
/* ELSE Transition */
EXECUTE HOURS

SECTION HOLIDAY_MESSAGE
/* Play holiday message */
GIVE RAN holidays
EXECUTE DISCONNECT_0
SECTION DISCONNECT_0
DISCONNECT

SECTION HOURS
/* Determine hours for the contact center. */
IF (TIME OF DAY !== 08:00 To 17:00) THEN
     EXECUTE CLOSED_MESSAGE
END IF
/* ELSE Transition */
EXECUTE CHECK_SERVICE

SECTION CLOSED_MESSAGE
/* Play closed message */
GIVE RAN closed_hours
EXECUTE DISCONNECT_0

SECTION CHECK_SERVICE
IF ((OUT OF SERVICE Primary_Skillset_gv)) THEN
     EXECUTE COPY_1_OF_TECHNICAL_MESSAGE
END IF
/* ELSE Transition */
EXECUTE QUEUE_TO_PRIMARY_SKILLSET

SECTION QUEUE_TO_PRIMARY_SKILLSET
/* Queue to the Primary_Skillset_gv. */
QUEUE TO SKILLSET Primary_Skillset_gv WITH PRIORITY 3
WAIT 2
EXECUTE PLAY_MUSIC

SECTION PLAY_MUSIC
/* Play music to the caller. */
GIVE MUSIC CC_DefaultMusic
EXECUTE CHECK_QUEUE_STATUS
```

```
SECTION CHECK_QUEUE_STATUS
IF (NOT QUEUED) THEN
     EXECUTE CONFIRM_SERVICE
END IF
/* ELSE Transition */
EXECUTE CONTINUE_TO_HOLD

SECTION CONFIRM_SERVICE
IF ((OUT OF SERVICE Primary_Skillset_gv)) THEN
     EXECUTE TECHNICAL_MESSAGE
END IF
/* ELSE Transition */
EXECUTE COPY_1_OF_QUEUE_TO_PRIMARY_SKILLSET

SECTION TECHNICAL_MESSAGE
GIVE RAN technical_message
EXECUTE DISCONNECT2

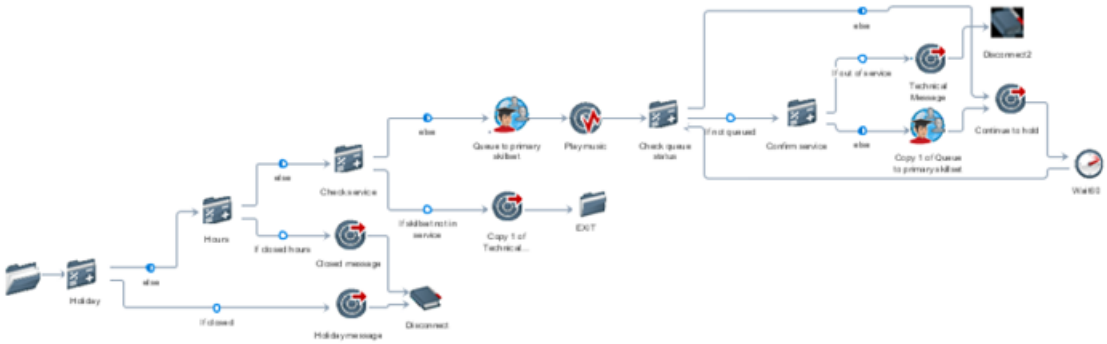
SECTION DISCONNECT2
DISCONNECTSECTION COPY_1_OF_QUEUE_TO_PRIMARY_SKILLSET
/* Queue to the Primary_Skillset_gv. */
QUEUE TO SKILLSET Primary_Skillset_gv WITH PRIORITY 3
WAIT 2
EXECUTE CONTINUE_TO_HOLD

SECTION CONTINUE_TO_HOLD
/* Play the continue to hold message */
GIVE RAN continue_to_hold
EXECUTE WAIT60

SECTION WAIT60
WAIT 60
EXECUTE CHECK_QUEUE_STATUS

SECTION COPY_1_OF_TECHNICAL_MESSAGE
GIVE RAN technical_message
EXECUTE EXIT
/*   EXIT */ SECTION EXIT
```

The flow application in the graphical format.



Activating the application

About this task

Activate the new application in the Contact Center view if you are ready to make the application live and use it to route contacts in your contact center.

Procedure

1. In the Contact Center view, select the script application that you want to activate.
2. On the **Contact Center** menu, click **Activate**.

   OR

   Right-click, then click **Activate**.

Chapter 9: Integrated reporting

This section describes how to implement an Interactive Voice Response (IVR) application to facilitate integrated reporting with a contact center.

The Contact Center Manager Server (CCMS) has an integrated reporting capability for Contact Center Multimedia (CCMM) and Avaya Media Server statistics.

How integrated reporting works

The Avaya Media Server IVR application assigns a unique number to each call. The IVR application passes the unique call number to the CCMS Open Networking Web service when the call is transferred to the CCMS. The CCMS uses the unique call number to match the Avaya Media Server information with the CCMS information and generate meaningful call flow statistics. The IVR application uses a Landing Pad, or a CDN controlled by the CCMS, to transfer the call to the contact center.

Types of IVR and contact center integration

Two basic Interactive Voice Response (IVR) and contact center integration scenarios are available:

- Front-End IVR processing on page 92
- Front-End Contact Center Processing on page 94

In Front-End IVR, a call is handled by an IVR application on the Avaya Media Server. The caller interacts with the application and the call is transferred to a Contact Center Agent.

In Front-End contact center integration, a call is handled by using a contact center script. The script sends the call to an Avaya Media Server application by using the Give-IVR treatment.

Contact center operates on an AML or IP network. Avaya Media Server operates on an IP network. AML uses two gateway types, AudioCodes and SIP Line Gateway (SLG). These gateways provide an interface between the AML and IP networks.

Use Orchestration Designer to write IVR applications.

Front-End IVR processing

In Front-End IVR, a call is presented to the Avaya Media Server IVR application. The IVR application answers the call and gathers information from the caller. The call can be transferred to an agent.

- The Avaya Media Server IVR application session is recorded in an SDR (Session Detail Record) on the Avaya Media Server database.
- The Contact Center Agent session is recorded in a CSR (Call Session Record) on the CC database.

The Avaya Media Server application must perform the following steps to facilitate integrated reporting:

- Store information for integrated reporting on page 92
- Landing Pad Reservation on page 93
- Performing Call Transfer on page 93

Store information for integrated reporting

The Avaya Media Server database contains five fields that the IVR application uses to store information. This information is copied to the CCMS and used to generate integrated reports.

Table 1: Avaya Media Server database fields used for integrated reporting information

Field Name	Values
OpaqueField1	String of length 128
OpaqueField2	String of length 128
OpaqueField3	Integer
OpaqueField4	Integer
OpaqueField5	Integer

The IVR application must store values in these specific Avaya Media Server database SDR fields using a Custom Block. The Custom Block is associated with a Java server page (JSP). The JSP page uses expressions to obtain values from Orchestration Designer environment and finalize the script code. A JSP expression starts with the characters **<%=** and ends with **%>**, as shown in the following example.

```
<script>sdr.OpaqueField3=<
%=((main)sceBean.getAppData()).ApplicationData.DataVariables.data_xferKind %>;</
script>
```

Landing Pad Reservation

The IVR application must call the Landing Pad Reservation Web service. The IVR application builds an XML message with specific information, sends this message to the Web service, reviews the response, and parses the response to retrieve the Landing Pad CDN to which it performs the transfer. The IVR application uses an Orchestration Designer Web Service Block to invoke the Landing Pad Web service. The Web service block information is prepared by reading the WSDL associated with the Web service. The WSDL file for the Landing Pad web service can be obtained directly from the Landing Pad Web server (running on the contact center) using a Web browser with the following URL:

http://<application-server>:9080/SOAOI/services/OpenNetworking?wsdl

<application-server> is the server name or IP address of the CC host that runs the Landing pad Web server.

The following is an example of a URL:

http://47.0.1.250:9080/SOAOI/services/OpenNetworking?wsdl

After the block is prepared, the values to be passed as parameters can be supplied using Java code. The values to be retrieved from the response are also extracted using Java code and assigned to a global variable for use later (for transferring the call).

Performing Call Transfer

The IVR Application can transfer the call to the Contact Center using the following methods.

AML Environment with SLG Gateway

The AML environment is the traditional call switching system. The call switch is equipped with a SLG gateway to send the calls to Avaya Media Server IVR application. The Avaya Media Server IVR application must use the Computer Telephony Integration (CTI) interface to perform the call transfer operation.

The Avaya Media Server CTI interface is used to transfer AML calls. The CTI interface is a collection of Web service operations used to interface to the Avaya Communication Server 1000 via a CTI link. The application URI and DN data are required to perform the transfer. You must also perform CTI Start Monitor and CTI register for an Established event.

AML Environment with AudioCodes Gateway

The AML environment is a traditional switching system on which the switch connects to an AudioCodes gateway to send calls to Avaya Media Server. This AML environment also requires the use of the CTI interface to perform the call transfer. The AudioCodes Gateway differs from

the SLG Gateway in the way the mydn value is obtained. All other CTI operations are the same.

CTI Transfer

Orchestration Designer uses the Custom Block to perform a CTI transfer.

Front-End Contact Center Processing

In Front-End CC the call is first presented to the Contact Center. The Contact Center script provides GIVE-IVR treatment to the call that results in the call being presented to an IVR application. The IVR application on Avaya Media Server interacts with the caller. The Contact Center GIVE-IVR treatment is recorded in a Call Session Record (CSR) on the contact center database. The Avaya Media Server session is recorded in an Session Detail Record (SDR) on the Avaya Media Server database.

AML Environment with SLG Gateway

When the call arrives from the contact center, the IVR application must perform the following steps to facilitate Integrated Reporting:

- Answer the call.
- Get the Call ID.
- Store the call details in a SDR record in the Avaya Media Server database.

AML Environment with AudioCodes Gateway

The AML environment is processed the same as for performing call transfers in front-end IVR processing.

Chapter 10: Orchestration Designer interface

Orchestration Designer consists of a visual layout of the perspectives, views, and blocks that make up the components to build, view, check errors, and activate applications to route contacts in Avaya Aura® Contact Center.

Modes of operation

You can use Orchestration Designer started within the Contact Center Manager Administration application or started on a local machine. The following table describes the differences between the two modes of operation.

Table 2: Differences between modes of operation in Orchestration Designer

	Contact Center Manager Administrator application	Local or client
Start application	Log on to CCMA client application. Open Scripting component. Click Orchestration Designer, Launch Orchestration Designer.	Choose Start, All Programs, Avaya, Contact Center, Orchestration Designer.
Views visible	Contact Center Local Application (the Script or Flow Editor) Synchronization Problems	Contact Center view (if connect to Orchestration Designer on CCMA) Local Application: the Script or Flow Editor when you edit an application Synchronization view (if connect to Orchestration Designer on CCMA) Problems
How change impacts contact center	Can affect contacts routed immediately.	Does not impact contacts. Application must be Synchronized and activated.

	Contact Center Manager Administrator application	Local or client
Accessibility to applications	Restrict users by access classes configured on Contact Center Manager Server. • If access level is None, users cannot access Orchestration Designer component. • If access level is Read Only, users can read applications, but not create new or modify existing applications. • If access level is Read & Update, users can read, create new, and modify existing applications. • If access level is Full Control, users can read, create new, modify existing and delete applications. **Important:** Orchestration Designer supports access classes, but it does not support partitioning in Contact Center Manager Administration.	No restrictions. User has full access to all applications.
• Accessibility to application manager data for Skillsets, CDNs and DNIS	Restrict users by access classes configured on Contact Center Manager Server. • If access level is None, users cannot access Orchestration Designer component. • If access level is Read Only, users can read applications, but not create new or modify existing applications. • If access level is Read & Update, users can read, create new, and modify existing applications. • If access level is Full Control, users can read, create new, modify existing and delete applications.	No restrictions. User has full access to all applications.

	Contact Center Manager Administrator application	Local or client
Accessibility to applications variables	Restrict users by access classes configured on Contact Center Manager Server. • If access level is None, users cannot access Orchestration Designer component. • If access level is Read Only, users can read applications, but not create new or modify existing applications. • If access level is Read & Update, users can read, create new, and modify existing applications. • If access level is Full Control, users can read, create new, modify existing and delete applications.	No restrictions. User has full access to all applications.
Accessibility to application data for Agents	Users restricted by access classes • If access level is None, users cannot access Orchestration Designer component. • If access level is Read Only, users can read applications, but not create new or modify existing applications.	No restrictions. User has full access to all applications.

Session management

Orchestration Designer is associated with the Contact Center Manager Administration where it was started shares a session management system. If the session expires or becomes invalid, Orchestration Designer starts running in local or client mode. In local or client mode, the user cannot use the Contact Center or Synchronization views. To work in the Contact Center Manager Administration mode again, you must close and restart Orchestration Designer from Contact Center Manager Administration.

The shared session contains two timeout values. The values are:

- Max Session time—The total time a session is active. After the Max Session time expires, Orchestration Designer enters the local or client mode.
- Max Idle time—The time that a session is idle before it becomes invalid. When the Max Idle time is reached without actions in Orchestration Designer, Orchestration Designer enters the local or client mode.

When Contact Center Security Framework is not enabled, Orchestration Designer and the CCMA browser use an internal CCMA session. The maximum session time is 24 hours, and the maximum idle time is 12 hours.

When Contact Center Security Framework is enabled, Orchestration Designer and the CCMA browser use a Security Framework session where the timeout values are configured on the security server.

Views

The following views are available in Orchestration Designer:

- Contact Center view on page 98
- Local view on page 99
- Synchronization view on page 99
- Flow Editor on page 100
- Script Editor on page 100
- Problems View on page 101

Contact Center view

The Contact Center view of Orchestration Designer shows all applications, application variables, and application manager data currently configured in your Contact Center.

You can access the Contact Center view if you are launching Orchestration Designer from the Contact Center Manager Administration application, or if you connect to the CCMA server from a client.

Variables, and application manager data (except agents) appear in the Contact Center view:

- Configure variables in Orchestration Designer.
- Configure CDNs, DNISs, and skillsets in Orchestration Designer.
- Configure Agents in the Management component.

See the online help for instructions to configure the application manager data or application variables for your contact center.

The Contact Center view refreshes automatically when you start Orchestration Designer, or you can choose to manually refresh the Contact Center Manager Server.

Before you can begin to manage applications, Orchestration Designer must verify that your user account has the correct access rights. Orchestration Designer receives information about the access classes apply to all data in the Contact Center from Contact Center Manager Server.

You can make minor changes to script or flow applications in the Contact Center view. However, for major changes, Avaya recommends that you copy the script or flow application from the Contact Center view to the Local view for modifications.

Local view

The Local view of Orchestration Designer shows data that is on the local machine. You need not be connected to Contact Center Manager Administration or to the network in order to work with this data. You can upload applications to the Contact Center View through the synchronization process. You can upload application manager data and application variables from the Local view to the Contact Center view.

The Local view is used if you start Orchestration Designer on a stand-alone client or from Contact Center Manager Administration.

Use the Local view to create and modify new applications without affecting the Contact Center Manager Server and the applications that are currently processing contacts.

Synchronization view

The Synchronization view shows the differences between all applications, application variables and application manager data stored on the Local machine and the applications, application variables and application manager data on the Contact Center Manager Server. Use the Synchronization view to compare the new and changed content in the Contact Center view and the Local view.

The Contact Center Manager Administration access classes are not applicable to the Local view, and you have full access to all data when working locally. However when you try to upload data from Local view to Contact Center view using the Synchronization view, your access rights determine what data you can upload or change.

For more information about the Synchronization view, see Synchronization view operations on page 131.

Flow Editor

The Flow Editor is the main tool for creating and modifying flow applications. The Flow Editor appears in the top-right hand corner of Orchestration Designer window when you create or open an existing flow application. It provides the canvas on which to place the blocks and link them to each other.

Each time you open a flow application, the main page of an application appears in a tab at the top of the Flow Editor.

Three Flow Editor components provide the controls to edit the blocks and transitions:

- View tabs: Located horizontally across the top of the Flow Editor, the tabs represent the block editors and main pages for your current flow application. An asterisk (*) on the tab indicates that data is modified on the tab and must be saved.
- Edit tool bar: Located horizontally above the Flow Editor, you have quick access to cut, copy, paste, delete and uniformly layout your flow.
- Block palette: Located vertically to the left of the Flow Editor, the icons represent blocks to build your Contact Center flow applications.
- Flow pages: Located at the bottom of the Flow Editor, if your flow application spans more than one page, view each main page of the flow application by clicking on the tabs.

When you save a flow application, you save the positional information of all dialog blocks on the diagram. When you reopen an application, the last saved positions of the blocks and transitions are restored.

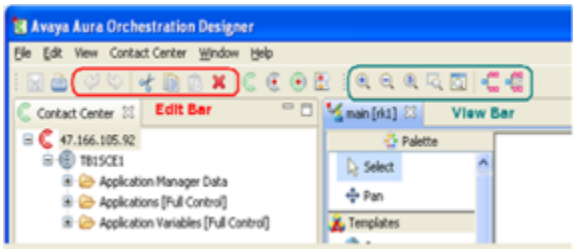


Figure 14: Flow Editor

Script Editor

The Script Editor is the main tool to create a script application. The Script Editor appears in the top-right hand corner of Orchestration Designer window when you create or open an existing script application. It provides the text-based tool to create the command line syntax for your script applications.

When creating applications, the script editor automatically indents text and color-codes the components of your application to make the application easier to read and review.

The following table shows the color coding that appears in the Script Editor.

Text	Color
Comments	Green text started with /* and ending with */
Logics	Blue text that is automatically indented
Commands	Green text in uppercase letters
Intrinsics	Red text in uppercase letters
Conditions	Blue text that is automatically indented
Loop	Blue text
Optional parameter	Green text
Contact Center Data	Black text
Operators, numeric data	Black text
Content Assistant	Press Ctrl + Spacebar to open a window with valid syntax

Problems view

The Problems view shows the errors in the current script or flow application. You can use the problems view to determine the location and cause of the problem.

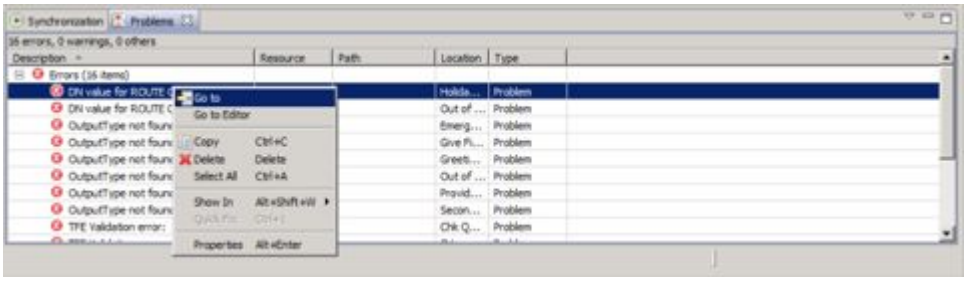


Figure 15: Problems View

For more information about the Problems view, see Problems view operations on page 129.

Menus

The following menus are available on Orchestration Designer menu bar.

- File menu on page 102
- Edit menu on page 102

- View menu on page 103
- Contact Center menu on page 103
- Window menu on page 104
- Help menu on page 105

If a menu item is unavailable, you cannot perform that option in the active view.

Table 3: File menu

Item	Description
New	Open a submenu of items you can add to Orchestration Designer: • CCMA: Add a Contact Center Manager Administration server name to the environment. (Local view only) • CCMS: Add a Contact Center Manager Server name to the environment. (Local view only) • Application: Create a new application.
Open	Open application: Open the application as a script in the Script Editor. Open Application in Flow Editor.
Close	Close the current editor.
Close All	Close all open editors.
Save	Save the contents of the current editor.
Save As	Save the contents of the current editor with a different file name.
Save All	Save the contents of all open editors.
Rename	Change the name of the selected resource.
Refresh	Refresh the Contact Center view with the information (variables) from Contact Center Manager Administration.
Print	Print the contents of the current script application.
Import	Start the import wizard. You can import applications from another source.
Export	Start the export wizard to export one application to your computer.
Properties	Select Properties to display the properties of the selected object.
Exit	Close Orchestration Designer.

Table 4: Edit menu

Item	Description
Undo	Reverse the most recent edit. The action that was last made is shown in the menu.
Redo	Reapply the edit that was most recently reversed by the Undo action.

Item	Description
Cut	Remove the selection and places it on the clipboard.
Copy	Place a copy of the selection on the clipboard.
Paste	Place the text or object on the clipboard at the current cursor location in the current view or editor.
Modify	Modify the application data item.
Delete	Remove the current selection.
Select All	Select all text or objects in the current view or editor.
Find and Replace	Search for an expression in the editor and, optionally, replaces the expression with a new expression.
Find Next	Search for the next item in the document that matches the item in the Find box.
Find Previous	Search for the previous item in the document that matches the item in the Find box.

Table 5: View menu

Item	Description
Zoom In	Zoom In to isolate portions of large and complex dialog flows.
Zoom Out	Zoom out on the Flow Editor to view larger portions of the dialog flow.
Reset Zoom	Reset Zoom button to return the contents of the Flow Editor to the previous scale.
Select Zoom Area	Select Zoom Area to draw a zoom box on the flow editor to select the area to zoom in.
Zoom to Editor	Zoom to Editor to make the flow fit in the screen area available in the flow editor.
Layout All Nodes	Layout All Nodes rearranges the position of the nodes and blocks to minimize line crossing.The blocks and nodes are completely rearranged.
Layout Incremental Nodes	Incremental Layout Nodes realigns the position of the nodes and blocks into rows and columns with respect to the current position. Incremental layout attempts to maintain the current layout of nodes and blocks.

Table 6: Contact Center menu

Item	Description
Connect to CCMA	Connect to the Contact Center Manager Administrator application for your contact center.

Item	Description
Activate Application	Activate an application that is not currently activated.
Deactivate Application	Disable an application that is currently activated.
Convert Application to Flow	Convert the Script application to a visual Flow application.
Copy CCMS to Local View	Copy the contents from the current CCMS in the Contact Center view to the Local view for updates.
Generate Documentation	Generate a document file that represents the content of the application. You can send the document to a printer. For more information, see Printing an application on page 127.
Synchronization	Synchronize the applications in the Local view with those on the Contact Center view. • Synchronize • Add To Local View • Update In Local View • Remove From Local View • Add To Contact Center • Update In Contact Center • Remove From Contact Center • Accept Updates To Local View • Accept Updates To Contact Center • Accept All Non-conflicted Updates • Show Validation Errors • Generate Report For more information about the Synchronization submenu, see Synchronizing objects from the Local view on page 132.

Table 7: Window menu

Item	Description
Show View	Display the views in the current perspective which depend on the version of Orchestration Designer you use: • Orchestration Designer Contact Center • Orchestration Designer Local

Item	Description
	• Orchestration Designer Synchronization • Orchestration Designer Problems
Reset Perspective	Change the layout of the current perspective to its original configuration which depends on the version of Orchestration Designer you use (through Contact Center Manager Administration or on your local client).
Navigation	• Show • Maximize Active View or Editor • Minimize Active View or Editor • Activate Editor • Next Editor • Previous Editor • Switch to Editor • Next View • Previous View
Preferences	Open the preferences dialog box to configure the General Settings and Contact Center preferences for Orchestration Designer.

Table 8: Help menu

Item	Description
Help Contents	Display the Help contents in a Help window or external browser. The Help contents contain Help books, topics, and information related to the workbench and installed features.
Download Sample Prompts	Download sample materials for applications.
View Contact Center Screen Cams	View demonstrations for creating applications.
About Avaya Contact Center Orchestration Designer	Provide information about Orchestration Designer, including the version number.

Toolbar items

Orchestration Designer toolbar, below the menu bar, provides shortcuts to commonly used functions. The contents of the toolbar change based on the editor you use.

Table 9: Toolbar icons

Icon	Description
	Save the project. If the project is closed, the project reopens in the last saved configuration. Saving the project also runs a validation check. Errors from the validation check are displayed in the Problems view.
	Print the application.
	Remove the selection and places it on the clipboard.
	Place a copy of the selection on the clipboard.
	Paste the contents of the clipboard.
	Delete the item in the script or flow application.
	Open the Contact Center view in Orchestration Designer. The Contact Center view shows all of the application manager data, applications, and application variables that are currently active in your Contact Center.
	Open the Local view in Orchestration Designer. The Local view shows all of the application manager data, applications, and application variables on the local machine. Use the local view to create and modify new applications without affecting the Contact Center Manager Server and the applications that are currently processing contacts.
	Open the Synchronization view in Orchestration Designer. The Synchronization view displays the differences between all objects on the Local machine and the objects on the Contact Center Manager Server. Use the Synchronization view to review the differences between the Local view and the Contact Center view, and update the information.
	Open the Problem view in Orchestration Designer. The problems view shows the errors in the current flow or script application.

Chapter 11: Contact Center and Local view operations

In Orchestration Designer, there are two views: the Contact Center view and the Local view.

The Contact Center view contains a real-time view of the Contact Center Manager Administration server and all of the Contact Center Manager Servers to which Orchestration Designer can connect.

Under each Contact Center Manager Server, the Contact Center view provides the user with the current configuration of the contact center applications, variables, and application manager data. You can configure the applications, variables, and application manager data (except agents) in the Contact Center view.

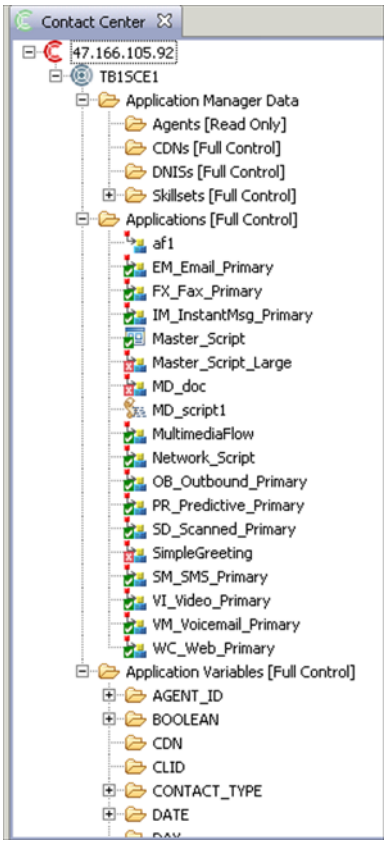


Figure 16: Contact Center view

The Local view provides a user work space on their own desktop to work with copies of the variables and applications. The variables and applications remain local until you synchronize the data to the Contact

Center view. You can configure the applications, variables, and application manager data with placeholders for agents in the Local view.

Prerequisites for Contact Center or Local view operations

- Start Orchestration Designer.

Showing or hiding the Contact Center view

About this task

By default, the Contact Center view appears when you launch Orchestration Designer from the CCMA client. You can show or hide the Contact Center view to see a list of all contact center flow applications, application manager data (AMD), and application variables.

Procedure

On the **Window** menu, click **Show View** > **Orchestration Designer Contact Center**.

Showing or hiding the Local view

About this task

By default, the Local view appears whenever you start Orchestration Designer. You can show or hide the Local view to view more or less of the other panels of the Orchestration Designer window. The Local View shows the same content as the Contact Center view, but you can customize it to appear as a copy of the Contact Center view, or use different labels.

Procedure

On the **Window** menu, click **Show View** > **Orchestration Designer Local**.

Creating a Contact Center Manager Server node

Before you begin

- Create the Contact Center Manager Server Administration node. See Creating a Contact Center Manager Administration node on page 116.

About this task

Create a Contact Center Manager Server node in the Contact Center or the Local view. The node is a collection of application manager data, applications and application variables associated with a Contact Center Manager Server in your contact center.

In the Contact Center view, you can see the Contact Center Manager Servers in the corresponding Contact Center Manager Administration node.

In the Local view, you must include Contact Center Manager Server nodes that match the Contact Center view to synchronize applications and variables with the Contact Center view. You can also create test server configurations to review and test applications.

Procedure

1. In the Contact Center or Local view, click **File** > **New** > **CCMS**.
2. Select the Contact Center Manager Administration server to which you want to add the Contact Center Manager Server.
3. In the **CCMS Name** box, type the name of your CCMS server.
4. Under **Server Type**, select **Standalone** or **Networking** to represent the CCMS server to which you are connecting.
5. In the **Server RLs** box, select the Contact Center Manager Server version number.
6. Under **Switch Type**, select the type of switch to which your Contact Center Manager Server is connected.
7. Click **Finish**.

Procedure job aid

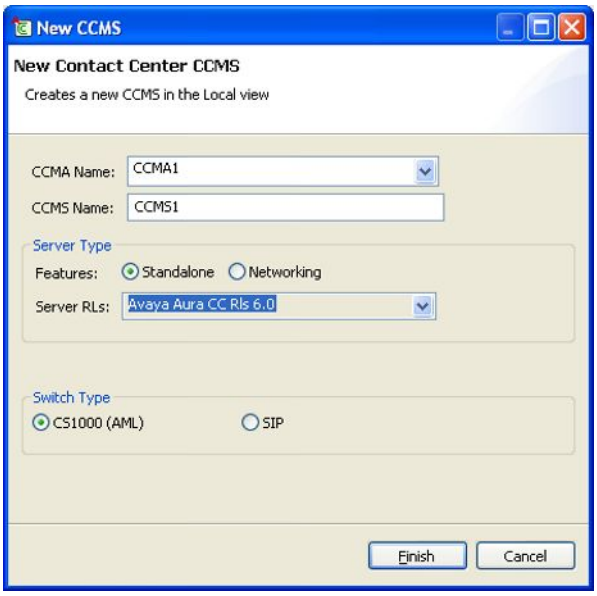


Figure 17: New Contact Center Manager Server window

Creating a new application

About this task

Create a new application in the Contact Center or Local view to begin placing your blocks (actions for the call to take place) and transitions (links between blocks) in your application.

The application can become active only if it is in the Contact Center view, and you activate it.

Procedure

1. In the Contact Center or Local view, select a Contact Center Manager Server on which to create the application.
2. On the **File** menu, click **New** > **Application**.
3. In the New Contact Center Application dialog box, select **Create in Local View** or **Create in Contact Center**.
4. Select a **CCMS** associated with a CCMA server.
5. In the **Application Name** box, type a name for the application.
6. In **Application Type**, select **Graphical Flow** or **Script**.

7. If you select Graphical Flow, in **Application Template**, select a template (sample) application on which to base your application.

   OR

   Choose **New_Flow** to create a blank application.

8. Click **Finish**.
   The new application appears in the Editor.

Variable definitions

Name	Description
Flow Name	The name of the application. The application has an extension .app. • Application names must start with a letter. • Application names are not case-sensitive. • The application name must be unique. • Do not use spaces or punctuation marks in flow application names. • Underscores (_) are allowed. • Do not use a keyword as a flow application name. For a list of keywords, see Reserved Keywords on page 461. • The maximum length of the application name is 30 characters. Avaya recommends that you use the following naming prefixes for multimedia applications: • EM_ for e-mail • FX_ for FAX message • IM_ for instant message • SD_ for scanned document • SM_ for SMS text message • OB_ for outbound • VM_ for voice mail • WC_ for Web communication

Name	Description
Application Templates	The user can select a template (sample) application on which to base their flow or they can choose to start with a blank flow. The templates provide a structure and varying formats to perform the following: • Uses IVR block to play prompts and collect digits. • Uses broadcast announcement in an output block to play prompts and collect digits. • Uses RAN output to play prompts and announcements. The flows have default data and default prompts which make them fully functional. The content of the flow depends on the switch type. These are the available templates: • New_Flow, DigitCollectionIVR, DigitCollectionIVRBroadcast, DigitCollectionRAN, MultiLevelMenuIVR, MultiLevelMenuIVRBroadcast, MultiLevelMenuRAN, SimpleGreetingIVR, SimpleGreetingIVRBroadcast, SimpleGreetingRAN, SimpleQueuing

Procedure job aid

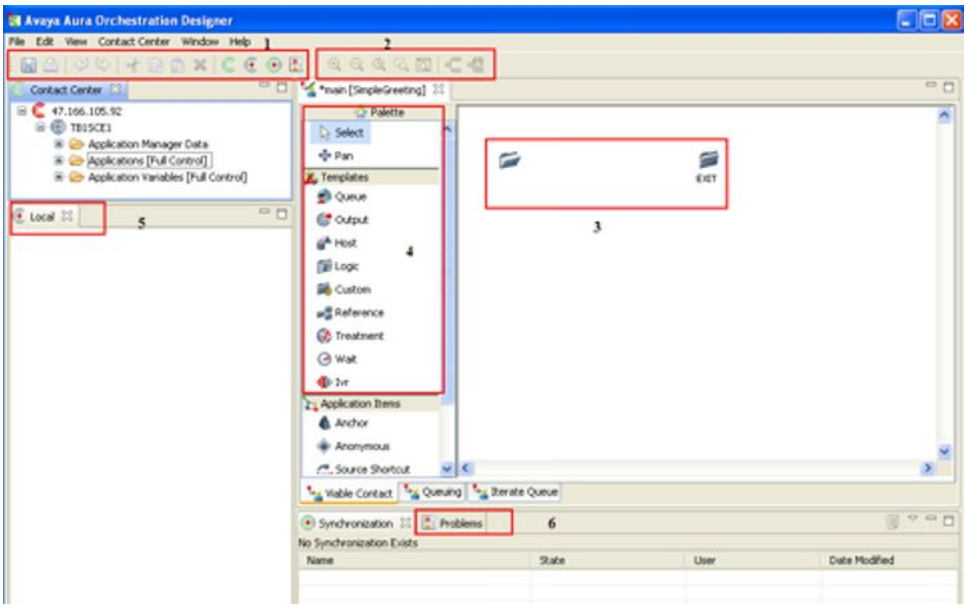


Figure 18: New application in Local view

1	Orchestration Designer toolbar
2	Editor tabs
3	New application flow
4	Block palette
5	Local view
6	Problems view

For more information about the functions of the Flow Editor, see Flow Editor operations on page 143.

Configuring the contact router

Before you begin

- Create an agent in your contact center.
- Create at least one CDN (route point) in your Contact Center.

About this task

Configure the contact router for incoming contacts. Contact routes are assigned based on CDN (route point) or the DNIS number of a contact. For each CDN (route point) or DNIS number, you can designate an application that controls contact handling.

Procedure

1. Open the **Applications** folder.
2. Right-click on the **Master_Script**.
3. Select **Open**.
   The Contact_Router pane appears.
4. In the **Route Configuration** editor, under **Available Routes**, select the required CDNs from the **CDNs** list.
5. Click **Add >**.
6. On the Application Chooser dialog box, select the application to process this CDN, and click **OK**.
7. To position the application, click **Move Up** or **Move Down**.
8. To change the application, click **Edit**.
9. Under **Available Routes**, select the required DNISs from the **DNISs** list.
10. Click **Add >**.
11. From the **Default Skillset** list, click **Browse** and select the skillset.
12. Click **File** > **Save**.

Copying data from Contact Center view to the Local view

Before you begin

- Create a Contact Center Manager Administration node in the Local view. See Creating a Contact Center Manager Administration node on page 116.

About this task

Copy your Contact Center Manager Server and other data to the Local view to perform tasks such as convert a script application to a flow application.

If the Local view already has a Contact Center Manager Server name with the same name as your current Contact Center Manager Server name in the Contact Center view, rather than copying the active server, you want to synchronize your Contact Center view and Local view. For more information about synchronizing views, see Synchronization view operations on page 131.

Procedure

1. In the Contact Center view, select an active Contact Center Manager Server node.
2. Right-click, and click **Copy to Local view**.
3. In the **CCMA Name** box, select the name of the Contact Center Manager Administration server in the Local view where you want to copy the Contact Center Manager Server.
4. In the **CCMS Name** box, enter the name of the Contact Center Manager Server name.
   To ensure consistency, ensure that the name of the Contact Center Manager Server in the Local view is the same as the copy in the Contact Center view.
5. Click **Finish**.
6. Expand the Contact Center Manager Administration server name in the Local view to see the newly copied content.

Opening an existing application

Before you begin

- Ensure that an application is in the Local view or Contact Center view.

About this task

Open an existing application in the Local view or Contact Center view to make changes or review the application.

Procedure

1. In the **Local** or **Contact Center** view, expand the Contact Center Manager Server where your application is located.
2. Expand **Applications**.
3. In the list of application, double-click the application name.

Procedure job aid

For information about making changes to your flow application, see Flow Editor operations on page 143.

For information about making changes to your script application, see Script Editor operations on page 239.

Changing the properties of an application

Before you begin

- Ensure that an application is in the Local view or Contact Center view.

About this task

View the properties of an existing application in the Local view or Contact Center view.

Procedure

1. In the **Local** or **Contact Center** view, expand the Contact Center Manager Server where your application is located.
2. Expand **Applications**.
3. Right-click the application to view, and click **Modify**.
4. Review and change, if required the properties for the application.

Creating a Contact Center Manager Administration node

Before you begin

- Start Orchestration Designer on the local machine (see Opening Orchestration Designer on page 22).

About this task

Create a Contact Center Manager Administration node in the tree in the Local view to identify a collection of Contact Center Manager Servers in your contact center.

In the Contact Center view, you can see the Contact Center Manager Administration server that has associated Contact Center Manager Servers.

In the Local view, you must add Contact Center Manager Administration nodes that match the Contact Center view to synchronize applications and variables.

Procedure

1. In the **Contact Center** or **Local** view, click **File** > **New** > **CCMA**.

2. In the **New CCMA** dialog box, type the name of the Contact Center Manager Administration server.
3. Click **Finish**.

Viewing Contact Center Manager Administration properties

About this task

View the Contact Center Manager Administration properties such as the server IP address, and the host name to confirm the connection to the correct server.

Procedure

1. In the **Contact Center** or **Local** view, select the Contact Center Manager Administration server name.
2. Right-click and click **Properties**.
3. Click **OK** to close the **Properties** dialog box.

Deleting a Contact Center Manager Administration node

About this task

Delete a Contact Center Manager Administration node in the Local view to reduce the displayed contents in your Local view. You cannot delete Contact Center Manager Administration nodes in the Contact Center view.

Deleting the Contact Center Manager Administration node also deletes all subcomponents of the Contact Center Manager Administration such as associated Contact Center Manager Servers, applications, application manager data and application variables.

Procedure

1. In the **Local** view, select the Contact Center Manager Administration server you want to remove.
2. Right-click and click **Delete**.
3. In the **Delete** dialog box that appears, click **OK** to confirm your choice.

Viewing Contact Center Manager Server properties

About this task

View the Contact Center Manager Server properties such as the server IP address, and the host name to confirm the connection to the correct server.

You can review the Contact Center Manager Server in both the Contact Center view. The Contact Center view and Local view contain the same data if you synchronize the views.

Procedure

1. In the **Contact Center** or **Local** view, select the Contact Center Manager Server server name.
2. Right-click and click **Properties**.
3. Click **OK** to close the **Properties** dialog box.

Deleting a Contact Center Manager Server node

Before you begin

- Ensure that a Contact Center Manager Server node exists.

About this task

Delete a Contact Center Manager Server node in the Local view to reduce the contents in your Local view.

Deleting the Contact Center Manager Server node also deletes all subcomponents of the Contact Center Manager Server such as associated applications, application manager data and application variables.

Procedure

1. In the **Local** view, select the Contact Center Manager Server you want to remove.
2. Right-click and click **Delete**.
3. In the **Delete** dialog box that appears, click **OK** to confirm your choice.

Importing an application

Before you begin

- Map the network drive to the location of the application you want to import, if the application is on a network drive.

About this task

Import an application from the file system on a local machine or a server on your network into your Local view. The application can be a sample application you want to modify.

Script applications use the file extension .s. Flow applications have the file extension .app.

You can import the sample scripts from the installation directory for Orchestration Designer. See c:\Program Files (x86)\Avaya\Contact Center\Orchestration Designer\<Orchestration Designer version number>\Samples or c:\Program Files\Avaya\Contact Center\Orchestration Designer\<Orchestration Designer version number>\Samples (32-bit operating systems).

Procedure

1. On the **File** menu, click **Import**.

   OR

   Right-click and select **Import**.
2. Click **Add** to navigate to the folder where the application is stored.
3. Ensure that the **File Type** box shows the correct file type for your application.
4. Select the application.
5. Click **Open**.
6. Select a CCMS to which to add the imported applications.
7. Click **Finish**.

Exporting an application

Before you begin

- Map the network drive to the location to which you want to export your application.
- Ensure there is an application to export.

About this task

Export an application from the Contact Center view or the Local view to a file system on the local PC or to your network. After you export the application, you can share it with colleagues or clients.

Procedure

1. On the **File** menu, click **Export**.

   OR

   Right-click and select **Export**.
2. Select the location option for the origin of the application to export.
3. Select the application to export.
4. Beside **To directory**, click **Browse** to open the folder to where you plan to export the application.
5. Under Options, choose to **Convert the graphical flow to a script**, or to **Overwrite existing files without warning**.
6. Click **Finish**.
   The exported application appears in the folder.

Activating an application

Before you begin

- Ensure that your application is free of errors.

About this task

Activate the application in the Contact Center view if you are ready to make the application live and use it to route contacts in your contact center.

If the application you want to activate is in the Local view, you must synchronize your views before activating the application in the Contact Center view.

If the application references another application, the referenced applications are also activated. If the referenced applications cannot be activated because they are not valid, the primary application cannot be activated.

Procedure

1. In the Contact Center view, select the script application that you want to activate.
2. On the **Contact Center** menu, click **Activate**.

   OR

Right-click, and then click **Activate**.

Deactivating an application

Before you begin

- Ensure that the application is active (Contact Center view only)

About this task

Deactivate an application in the Contact Center view to remove the application from the status of processing contacts. You must ensure that the application you plan to deactivate is not referenced by another application. If the application is referenced by another application, you cannot deactivate the application until the main application is deactivated or you remove the reference to the referenced application. After deactivation, the application remains valid, but not active.

The master application and other applications called by the master application are always active in the contact center and cannot be deactivated. The master application and references to other application can process contacts in the contact center to ensure that at least one application is always active.

Procedure

1. Select the application.
2. On the **Contact Center** menu, choose **Deactivate**.

   OR

   Right-click and click **Deactivate**.

Renaming an application

Before you begin

- Deactivate the application.

About this task

Rename an application in the Contact Center or Local view to make the name more appropriate for the application.

You can rename the master application or network application only in the Local view. You cannot rename an active application in the Contact Center view.

Procedure

1. In the **Local** view, select the application that you want to rename.
2. On the **File** menu, choose **Rename**.

   OR

   Right-click and select **Rename**.
3. In the Rename dialog box, type a new name for the application.
4. Click **OK**.

Saving an application

Before you begin

- Add all of the required commands or blocks to your application. For more information, see the following sections:
  - Adding blocks to The flow on page 144
  - Creating a default transition on page 156 or Defining conditional transitions on page 156
  - Adding commands to script applications on page 239

About this task

Save your complete application in the Contact Center or Local view to protect the changes you made to your application.

When you save an application, it is automatically checked for errors. Errors in the application appear in the Problems view.

Procedure

1. To save the application with the current name, choose **File** > **Save**.
2. If there is a problem with the application, a message dialog box appears.
3. In the message box, click **OK**.
4. Review the errors and correct them.

Closing an application

Before you begin

- Ensure the application is open.
- Save changes to the application. See Saving an application on page 122.

About this task

Close an application in either the Contact Center or Local view when you finish working with it.

Procedure

1. Select the application to close.
2. Choose **File** > **Close**.
3. If there are unsaved changes on the application, a dialog box appears prompting you to save your changes.
4. Click **OK**.
5. In the message box, click **Yes** to save the changes, **No** to disregard the changes, or **Cancel** to cancel the close.

Deleting an application

Before you begin

- Deactivate the application. See Deactivating an application on page 121.

About this task

Delete an application from the Contact Center or Local view when it is not required in your contact center to free file space on your server.

Procedure

1. Select the **Applications** folder that contains the script you want to delete.
2. Select the script you want to delete.
3. Choose **Edit** > **Delete**.

   OR

   Right-click and select **Delete**.

4. On the confirmation dialog box, click **OK**.

Converting the script application to a flow application

Before you begin

- Save a script application. See Saving an application on page 122.
- Ensure the script application is free of errors. See Checking applications for errors on page 125.

About this task

Convert your script application to a flow application if you want to present a user-friendly application to your client. The flow application presents information in a visual display of decisions, facts, and paths, rather than programming language code which is often easier to understand and follow.

When you convert a script application to a flow application, the flow application replaces the existing script application.

You must be working in the Local view of your Orchestration Designer to save the script application as a flow.

The script application does not exactly correspond to individual blocks in a flow application because the blocks in a flow application represent structured command patterns, not individual commands.

If your script application has a comment statement, it may cause the script to flow conversion to fail. Comments in your scripts must be placed at the end of the commands in order to translate to the flow application correctly. Some sets of commands must not be split by a comment, or the script to flow conversion fails. For example, the HDX SEND REQUEST and GET RESPONSE commands must not be divided by a comment. Another example of a set of commands that cannot be split by a comments is OPEN VOICE SESSION and CLOSE VOICE SESSION.

Procedure

1. In the **Local** view, select the script application you want to convert to a flow application.
2. Right-click, and then click **Convert to Flow**.
   The script application is replaced with a flow application with the same name.

Procedure job aid

When you convert script applications to flow applications, each block in a flow application represents a group of structured command patterns, not an individual command. For example,

a GIVE RINGBACK command followed by a WAIT command in a script. In a flow application, a treatment block can contain both the GIVE RINGBACK and WAIT command together.

If your script contains a QUEUE command that precedes a WAIT command, the WAIT command can appear to have a value in the flow application different from that in the script application. The WAIT block in the flow application uses the default wait value configured for the QUEUE commands in the Default Options preferences configured in Orchestration Designer application. The preference default value ensures that the QUEUE command is always followed by a short delay time according to the best practices for call flows. If your TFE script contains the following code:

```
QUEUE TO SKILLSET sales_skillset
WAIT 4
```

and the default wait value is 2, your flow application shows a wait block with a wait value of 2. The WAIT block makes up the difference to apply the originally intended wait value of 4 as in the original script application sequence.

Checking applications for errors

About this task

Check applications for errors when you save the application. Use the following procedure to help identify the errors.

When you save an application in the Contact Center view, the validator checks to see that the size of the scripts is within the appropriate limits and that the syntax is correct. If you save an application in the Local view, the validator only checks the syntax of the application; it does not check the size.

Procedure

1. In the **Problems** view, select an application.
2. Select a row in the Problems view to get a description of the problem.
3. Right-click the row in the Problems view.
4. Choose one of the following items from the menu:
    - Click **Go to** to view the block in the flow application.
    - Click **Go to Editor** to view the edit window for the block in the flow diagram.
5. Fix all issues.
6. Choose **File** > **Save**.

Procedure job aid

See the Troubleshooting chapter for more information about problems with the applications.
See Troubleshooting applications on page 445.

Generating application documentation

Before you begin

- Ensure the application is open, visualized and has focus.

Important:

You can only open Orchestration Designer-generated RTF documents in Microsoft Word.

About this task

Generate documents in Portable Document Format (PDF), Rich Text Format (RTF), and Hypertext Markup Language (HTML) that contain detailed information about Applications, Application Manager Data, and prompts configured using Orchestration Designer. These documents are versioned and can be used for tracking or customer sign-off purposes.

Note:

If you click **Cancel**, the documentation is generated for the current task, but no new documents are generated.

Procedure

1. From the **Contact Center** menu, choose **Generate Documentation**.
2. To document local applications, choose **Generate from Local View**.
3. To document Contact Center applications, choose **Generate from Contact Center**.
4. To select the applications to document, expand CCMA.
5. Expand the CCMS node.
6. Expand the **Applications** folder.
7. Select the Applications to document.
8. Click **Browse**, to locate a storage location for the documentation.
9. Select the documentation storage directory.
10. Click **OK**.

11. From the **Format** list, select the document format; HTML, PDF, RTF, or all formats.
12. Under **Revision History Details**, in the **Document Version** box, type the document version.
13. In the **Author** box, type the name of the author.
14. Under **Comments**, type the comments you want to appear in the documentation.
15. Under **Options**, select the **Open in Explorer upon completion** check box to open completed HTML documents.
16. Select the **Overwrite existing files without warning** check box to overwrite existing files without warning.
17. Click **Finish** to generate the application documentation.

Printing an application

Before you begin

- Configure a printer on your network with access to Orchestration Designer.

About this task

Print an application to obtain a hard copy of your application.

Procedure

1. Open the application that you want to print in the editor.
2. On the **File** menu, choose **Print**.
3. Select the printer.
4. Click **Print**.

Chapter 12: Problems view operations

When you save a script or flow application, Orchestration Designer saves and validates the application.

The validator ensures that the application is complete with no compile time errors. Some examples of items the validator catches are queue blocks where the queue field is empty, or queue blocks where the skillset references is not present on the Contact Center Manager Server. The validation problems, including code errors appear in the Problems view. If the current application is a script, the Problems view describes the error and provides the line number and position of the error. If the current application is a flow, the Problems view lists the block and the error type.

Your script or flow application must be valid before you can activate it, thus you must correct all errors in the Contact Center view.

Clicking on another flow editor refreshes the Problems view. The Problems view filters errors depending on the flow editor that is selected.

If the script is a Master Script, Network Script, or an active script, Orchestration Designer attempts to validate and activate the script or flow application before saving it. If the validation or activation of one of the active script fails, then the save does not occur. The script current in the Contact Center Manager Server memory continues to run.

Prerequisites for Problems view operations

- Save your application. See Saving an application on page 122.

Displaying the Problems view

Before you begin

- Start Orchestration Designer.

About this task

Display the Problems view to review the location and error in your applications.

The Problems view shows the errors when you save the current application.

Procedure

On the **Window** menu, click **Show View** > **Problems**.

Resolving errors in the application

About this task

Resolve errors in the application that you want to make active in your contact center. Orchestration Designer marks applications with a red X.

To avoid affecting your live contact routing, copy your application to the Local view to change the application.

Procedure

1. Double-click the application that has an error.
2. In Orchestration Designer, open the Problems view.
3. Right-click the row in the Problems view.
4. Choose one of the following items from the menu:
   - Click **Go to** to view the block in the flow application.
   - Click **Go to Editor** to view the edit window for the block in the flow diagram.
5. Click **File** > **Save**.
   Saving the application automatically checks the validation of the application. If there are no errors, the application is complete. If there are errors, you must correct them.

Procedure job aid

Use the following items to help you correct errors in your application.

- General text editing procedures. See Script Editor operations on page 239.
- Review script commands and syntax. See Checking applications for errors on page 125.
- Review the other error messages. See Validation errors on page 451.

Chapter 13: Synchronization view operations

The Synchronization view is initially empty. When you synchronize the Contact Center view and the Local view, the Synchronization view shows the differences between the objects in the Local view and the Contact Center view for the Contact Center Manager Server. Differences between applications, applications manager data, and variables appear in the Synchronization view. If no differences exist, the Synchronization view appears empty.

You can use the Synchronization view to perform the following operations:

- view the differences between the objects on a local machine and objects on a Contact Center Manager Server
- resolve one or all differences by updating either the local machine or the Contact Center Manager Server with the latest version of each application, applications manager data, or variable

The Contact Center Manager Administration access classes are not applicable to the Local view, and you have full access to all data when working locally. However when you try to upload data from Local view to Contact Center view, your access rights determine what data you can upload or change.

If you copy activate applications to the Local view, make changes, and synchronize the applications to the Contact Center view, only error-free scripts are automatically reactivated.

For example, you can copy Master, and the referenced applications Application1, Application2, and Application3 to the Local view and make changes to the four applications. When your changes are complete, synchronize the Local view with the Contact Center view and select Accept all updates to the Contact Center view. If there is an error in Application2, during the synchronization process, Orchestration Designer lists all of the other applications in the Contact Center view in their current active state, and continues to use the existing version of Application2, since Orchestration Designer cannot activate the new version of Application2.

Prerequisites for working in the Synchronization view

- Ensure there are changes in the Contact Center view or Local view to compare.

Displaying the Synchronization view

Before you begin

- Start Orchestration Designer.

About this task

Display the Synchronization view to show the differences between the applications, application manager data, and variables between the Contact Center view and the Local view.

You must run the Synchronization action to view the differences between the Contact Center view and the Local view.

Procedure

On the **Window** menu, click **Show View** > **Synchronization**.

Synchronizing objects from the Local view

About this task

Synchronize applications, application manager data, or variables in the Contact Center view with the applications, application manager data, or variables in the Local view. Synchronizing objects in the Local view and Contact Center view performs a comparison to see which items are different.

Objects only appear in the Synchronization view when there are differences between the objects in the Local view and the Contact Center view.

During the first synchronization, you must select the Contact Center Manager Server with which to synchronize.

Procedure

In the **Local** view, right-click the Contact Center Manager Server and select **Synchronize** from the menu.

If the CCMS server in the Local view and the Contact Center view contain the same content, no objects appear in the Synchronization view.

Procedure job aid

If the Contact Center Manager Server in the Local view and Contact Center view are different, the changes appear in the Synchronization view, along with a simple statement reflecting the change. For changes to application variables and applications, the Synchronization view shows the users who made the change and the date of the change.

The Synchronization state of each object appears in the State column of the Synchronization view. The context menu relates to the state of the object that you select in the tree.

Important:

It is possible to update application manager data (excluding agents) and application variables in Contact Center using Orchestration Designer.

State	Description
Added to Local View	A new object is in the Local view and an object of the same name and type does not exist in the Contact Center Manager Server.
Updated in Local View	An object with the same name appears in both the Local view and the Contact Center Manager Server but, the local version is different.
Removed from Local View	An object with the same name appear in both the Local view and the Contact Center Manager Server, but the local version is missing.
Added to Contact Center	A new object is in the Contact Center Manager Server and an object of the same name and type does not exist in the Local view.
Updated in Contact Center	An object with the same name appears in both the Local view and the Contact Center Manager Server, but the Contact Center Manager Server version is different.
Removed from Contact Center	An object with the same name appears in both the Local view and the Contact Center Manager Server, but the Contact Center Manager Server version is missing.
Conflicted	One of the following occurred:

State	Description
	• a new object appears in the Local view while an object of the same name and type exists in the Contact Center Manager Server. • a new object appears in the Contact Center Manager Server while an object of the same name and type exists in the Local view. • an object is initially Synchronized between the Local view and Contact Center Manager Server, and then both versions are modified. • an object is initially Synchronized between the Local view and the Contact Center Manager Server, and then one version is modified and one version is deleted.

Adding an application to the Contact Center view

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Add an application to the Local view.
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Add an application you modified in the Local view to the Contact Center view to prepare the application for activation in your contact center.

You can add a single application, or update all non conflicted applications to the Contact Center view. For more information, see Updating all non conflicted applications in the Contact Center view on page 139.

Procedure

1. In the **Synchronization** view, expand the Contact Center Manager Server.
2. Expand the appropriate folder to locate the new application.
3. In the **State** column, verify the application has the state **Added to Local View**.
4. Right-click, and then click **Add to Contact Center**.
   The item is removed from the Synchronization view when you add the application to the Contact Center view.

Updating an application in the Contact Center view

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Add or modify an application to the Local view.
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Update an application you updated or added in the Local view to the Contact Center view to prepare the application for activation in your contact center.

You can update a single application, or update all non conflicted applications in the Contact Center view. For more information, see Updating all non conflicted applications in the Contact Center view on page 139.

Procedure

1. In the **Synchronization** view, expand the Contact Center Manager Server.
2. Expand the appropriate folder to locate the updated application.
3. In the State column, verify the application has the state **Updated in Local View**.
4. Right-click, and then click **Update in Contact Center**.
   The item is removed from the Synchronization view when you update the Contact Center view.

Removing an application from the Contact Center view

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Delete an application from the Local view.
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Remove an application from the Contact Center view if it is removed from the Local view to ensure the list of applications in the Contact Center Manager Server in both views are the same.

You can remove a single application, or remove all non conflicted applications in the Contact center view. For more information, see Updating all non conflicted applications in the Contact Center view on page 139.

Procedure

1. In the **Synchronization** view, expand the Contact Center Manager Server.
2. Expand the appropriate folder to locate the removed application.
3. In the **State** column, verify the application that has the state **Removed from Local View**.
4. Right-click, and then click **Remove from Contact Center**.
   The item is removed from the Synchronization view when you delete the application from the Contact Center view.

Adding an application, application manager data, or variable to the Local view

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Add an application to the Contact Center view (or application manager data or variables using the Contact Center Manager Administration application).
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Add an application, application manager data, or variable to the Local view from the Contact Center view to ensure that the copy in the local view is the same as the copy in the Contact Center view.

You can add a single application, application manager data or variable, or add all non conflicted applications, application manager data or variables in the Local view. For more information, see Updating all non conflicted applications, application manager data, or variables in the Local view on page 140.

Procedure

1. In the **Synchronization** view, expand the Contact Center Manager Server.
2. Expand the appropriate folder to locate the added application, application manager data, or variable.
3. In the **State** column, verify the object has the state **Added to Contact Center**.

4. Right-click, and then click **Add to Local view**.
   The item is removed from the Synchronization view when you add the item to the Local view.

Updating an application, application manager data, or variable in the Local view

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Update an application to the Contact Center view (or application manager data or variables using the Contact Center Manager Administration application).
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Update an application, application manager data, or variable to the Local view from the Contact Center view to ensure that the copy in the local view is the same as the copy in the Contact Center view.

You can update a single application, application manager data or variable, or update all non conflicted applications, application manager data or variables in the Local view. For more information, see Updating all non conflicted applications, application manager data, or variables in the Local view on page 140.

Procedure

1. In the **Synchronization** view, expand the Contact Center Manager Server.
2. Expand the appropriate folder to locate the added application, application manager data, or variable.
3. In the **State** column, verify the object has the state **Updated in Contact Center**.
4. Right-click, and then click **Update in Local view**.
   The item is removed from the Synchronization view when you update the item in the Local view.

Removing an application, application manager data, or variable from the Local view

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Remove an application from the Contact Center view (or application manager data or variables using the Contact Center Manager Administration application).
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Remove an application, application manager data, or variable from the Local view from the Contact Center view to ensure that the list of resources in both the Local and Contact Center views are the same.

You can remove a single application, application manager data or variable, or remove all non conflicted applications, application manager data or variables from the Local view. For more information, see Updating all non conflicted applications, application manager data, or variables in the Local view on page 140.

Procedure

1. In the **Synchronization** view, expand the Contact Center Manager Server.
2. Expand the appropriate folder to locate the added application, application manager data, or variable.
3. In the **State** column, verify the object has the state **Removed from Contact Center View**.
4. Right-click, and then click **Remove from Local view**.
   The item is removed from the Synchronization view when you remove the item to the Local view.

Updating a conflicted application

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Update an application in the Contact Center view.

- Update the same application in the Local view.
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Update an application in the Contact Center view or the Local view to ensure that the copy in the Contact Center view is the same as the copy in the Local view.

Procedure

1. In the **Synchronization** view, expand the Contact Center Manager Server.
2. Expand the appropriate folder to locate the conflicted application.
3. In the **State** column, verify the object has the state **Conflicted**.
4. Right-click, and then click either **Update in Local View** or **Update in Contact Center**.
   The item is removed from the Synchronization view when you update the item in the Local view or the Contact Center view.

Updating all non conflicted applications in the Contact Center view

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Add, modify, or delete multiple applications in the Local view.
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Update all non conflicted applications in the Contact Center view at once to save time.

You can update a single application to the Contact Center view. See Adding an application to the Contact Center view on page 134 or Updating an application in the Contact Center view on page 135.

Procedure

1. In the **Synchronization** view, select the Contact Center Manager Server.
2. Right-click, and then click **Accept Updates to Contact Center View**.

The items are removed from the Synchronization view when you accept the updates.

Updating all non conflicted applications, application manager data, or variables in the Local view

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Add, modify, or delete multiple applications in the Contact Center view (or application manager data or application variables in Contact Center Manager Administration).
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Update all non conflicted applications, application manager data, and variables in the Local view at once to save time.

You can update a single application, application manager data, or variable in the Local view. For more information, see Adding an application, application manager data, or variable to the Local view on page 136 or Updating an application, application manager data, or variable in the Local view on page 137.

Procedure

1. In the **Synchronization** view, select the Contact Center Manager Server.
2. Right-click, and then click **Accept Updates to Local View**.
   The items are removed from the Synchronization view when you accept the updates.

Updating all non conflicted objects to both the Local and Contact Center views

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Add, change, or remove multiple applications in the Contact Center view (or application manager data or variables using Contact Center Manager Administration).

- Add, change, or remove multiple applications in the Local view.
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Update all non conflicted objects to both the Local and Contact Center views simultaneously to ensure that both views contain the same data.

Procedure

1. In the **Synchronization** view, select the Contact Center Manager Server.
2. Right-click, and then click **Accept All Non-Conflicted Updates**.
   The items are removed from the Synchronization view when you accept the updates.

Updating an application in the Contact Center view with an invalid local application

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Update an application in the Local view.
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.
- Update the application in the Contact Center view (which fails if the application with the same name in the Contact Center view is activated, and the application in the Local view is not valid).

About this task

Update an application in the Contact Center view with an invalid local application. If you are updating an active Contact Center view application, all errors must be resolved in the Local view application before you can update the application in the Contact Center view. If you cannot synchronize the application, the Synchronization view shows the errors.

You can update a single application, or update all non conflicted applications to the Contact Center view. See Updating an application in the Contact Center view on page 135 or Updating all non conflicted applications in the Contact Center view on page 139.

Procedure

1. In the **Synchronization** view, expand the Contact Center Manager Server.
2. Expand the applications folder to locate the updated application with errors.

In the Synchronization view, the invalid application has an error icon and the line is red

3. Right-click, and then click **Show Validation Errors**.
4. In the dialog box that appears, select an error.
   The Local application opens and the line or block with the error is selected.
5. Fix all of the validation errors in the local version of the application.
6. Right-click and then select **Update in Contact Center**.
   The item is removed from the Synchronization view when you accept the updates.

Printing a Synchronization View report

Before you begin

- Copy the server from the Contact Center view to the Local view.
- Update objects in the Local view or the Contact Center view.
- Synchronize the objects in the Local view. See Synchronizing objects from the Local view on page 132.

About this task

Print a synchronization view report to create a record of the comparison between the applications, application manager data, and variables in the Contact Center and Local views. The printed report is in HTML.

Procedure

1. In the **Synchronization** view, click the Synchronization Report icon ( ).
2. In the dialog box, select a name and location for the report.
3. Click **OK**.
   An HTML report generates in the designated location. You can use an application to send the HTML file to a printer.

Chapter 14: Flow Editor operations

Use the Flow Editor in Orchestration Designer to view a graphical representation of the nodes, decisions, and paths in your script applications. This chapter contains the general procedures for manipulating the elements in your application using the Flow Editor.

The Flow Editor components provide the controls to edit the blocks and transitions:

- View tabs: Located horizontally across the top of the Flow Editor, the tabs represent main pages and block editors for all flow applications on which you are working.
- Edit tool bar: Located horizontally above the Flow Editor, you have quick access to cut, copy, paste, delete and layout your flow.
- Block palette: Located vertically to the left of the Flow Editor canvas, the icons represent blocks that you can use to build your Contact Center flow applications. The blocks you see depend on the switch you use in your Contact Center.
- Flow pages: Located at the bottom of the Flow Editor, if your flow application spans more than one page, you can view each main page of the flow application by clicking on the tabs.

When you save a flow application, you save the positional information of all dialog blocks on the diagram. When you reopen an application, the last saved positions of the blocks and transitions appear.

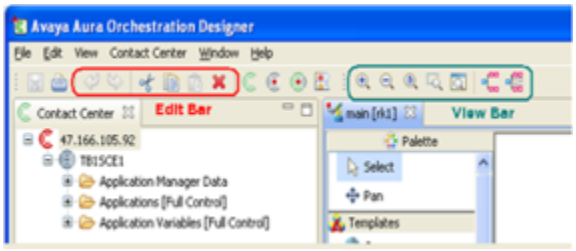


Figure 19: Flow Editor

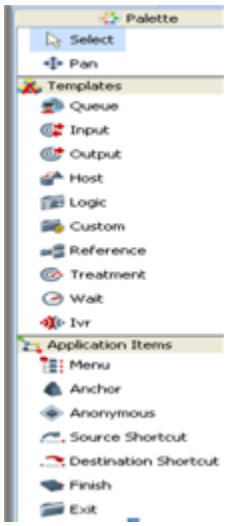


Figure 20: Palette bar on SIP OCS

Prerequisites

- Create a new flow application or open an existing script.
- Review the flow block information to see how you can use them in your flow application.

Adding blocks to the flow

About this task

Add blocks to the flow in the Flow Editor to graphically present an application and demonstrate the paths in a clear, graphical manner. Each block represents an action for the contact.

Procedure

1. Select a dialog block by clicking the dialog block icon on the Palette bar.
2. Click the area of the **Flow Editor** in which to place the dialog block.

Procedure job aid

The following table shows all of the block icons in the Block palette and the function for each block. For information about how to configure each type of block, see Block creation on page 171.

Icon	Description
	Queue block: A block that assigns the contact to a queue. The queue can be a skillset, a particular agent, or a Network Automatic Call Distribution (NACD) DN.
	Input block: A block that can collect input from callers. You can ask a caller a question and collect the responses. (Avaya Communication Server 1000 only)
	Output block: A block that can play prerecorded prompts and variable values to a caller.
	Host Data Exchange block: A block that you use to configure Host Data Exchange commands.
	Logic block: A block that defines conditional situations in a flow. The conditions are decisions that determine where to go next in the flow application.
	Custom block: A block that provides custom script commands that is not available in other block types.
	Reference block: A block that allows the flow application to call another flow. You can create multiple flow applications to reduce the size and complexity of the graphical images of the flow applications.
	Treatment block: A block that provides treatment to the contact. For calls, sample treatments include music, ringback or silence.
	Wait block: A block that of time to pause between treatments for a contact.
	IVR block: A block that sends the call to an IVR system. You can configure the IVR parameters.
	Menu block: A block that defines menu options for callers who enter Dual Tone Multi Frequency (DTMF) codes to choose the appropriate menu option to meet their requirements.

Icon	Description
	Anchor node: A node that allows dialog blocks to exist outside the main path of the application.
	Anonymous node: A node used to rearrange links in a different position to reduce the number of line crossings in a flow application and provide you with more control to display the links between blocks.
	Finish block: A block representing the termination point of an application, such as disconnect, give busy, or route call.
	Exit block: A block signifying the end of the flow application. There is only one Exit node in a flow application.
	Source block: A block pointing to another location in the flow application to allow repeats in the applications or navigation between pages. The Source Shortcut points to a Destination shortcut.
	Destination block: The block where a Source Shortcut points.

Adding blocks using accelerator keys

About this task

Add blocks to the Flow Editor using Accelerator keys to quickly add multiple blocks in sequence on the Flow Editor diagram or to create alternative paths for your application.

Procedure

1. On the **Flow Editor**, select the block that is the source of the connection.
2. Press **Alt** or **Alt+Shift** and click the block you want to add on the Block palette.

   If you press **Alt**, the new block connects to the block you selected. The original block remains active so that you can create alternative paths when you use appropriate blocks with the decision capabilities.

   If you press **Alt+Shift**, the new block connects to the block you select. The new block is selected so that you can create links in a single path.

Configuring general details for blocks

Before you begin

- Understand how to use the Condition and Assignment Builder. See Condition and Assignment builder operations on page 243.

About this task

Configure the general details for a block that appears in the Flow Editor to provide meaningful details such as a name and description for each block. The name appears under the block in the Flow Editor.

You can also specify variable assignments for the block when the contact enters the block in your flow application. The conditional or unconditional assignments are made when the contact reaches the block in the application. You must use call variables for general assignments in a block.

You can configure comments and debug functionality for your script using the general setup tab for your blocks in the flow applications.

Procedure

1. Select a block icon in the **Flow Editor**, right-click and click **Open**.

   OR

   Double-click the block icon in the Flow Editor.
2. Click the **Setup** tab.
3. In the **Name** box, type a name for the queue block.
4. In the **Description** box, type a description for the queue block.
5. To add assignments or conditional assignments, on the right of the **Assignment Expression** list, click **Edit** to open the Condition and Assignment Builder and define your expression.
6. If required, select **Add Log command**, and type a log message.

Variable definitions

Name	Description
Add log command	Add a log command to create an entry in the event log on Contact Center Manager Server. Avaya recommends this option only

Name	Description
	if you have concerns with the application. Otherwise, the Log command option slows the function of the application. For example, you can capture a message for each block in the event log. You can use the log command together with the Debug only option for debugging your applications. You can remove all log commands by clearing the Include Debug Logging check box in the preferences of Orchestration Designer.
Assignment Expressions	An optional list of assignment expressions for variables.
Debug only	Select the option to include the log commands only when in debug mode. Orchestration Designer assigns the debug flag to true for the application. Avaya recommends this option only if you have concerns with the application. Otherwise, the debug option slows the function of the application. IF debug THEN LOG ‘message’ END IF If the Debug only box is cleared, then the Log message appears in the event log on Contact Center Manager Server. You can remove all log commands and debug information by clearing the Include Debug Logging check box in the preferences of Orchestration Designer.
Description	A description of the block.
Name	The name of the block. By default, the name includes the type of block with a unique identifier.

Example: Using Setup tab for assignments

A locked assignment checks the value of a survey counter call variable. The assignment is locked to ensure that the data in the counter call variable is preserved for the entire flow application. If the value is less than or equal to 10, use an assignment expression to increase the survey counter by 1. If the counter is 10, use an assignment expression to assign the survey counter to 0. Use a logic block to make a decision that depends on the survey counter variable.

This conditional assignment must occur before the transitions from the logic block because the decisions about the next step in the application depends on the value of the variable.

To configure the assignment, create the call variable cv_survey_counter with an initial value of 0 in the Script Variables page (see Creating variables with one value on page 262), or as a placeholder for the call variable in the Local view (see Adding a variable to Orchestration Designer on page 257).

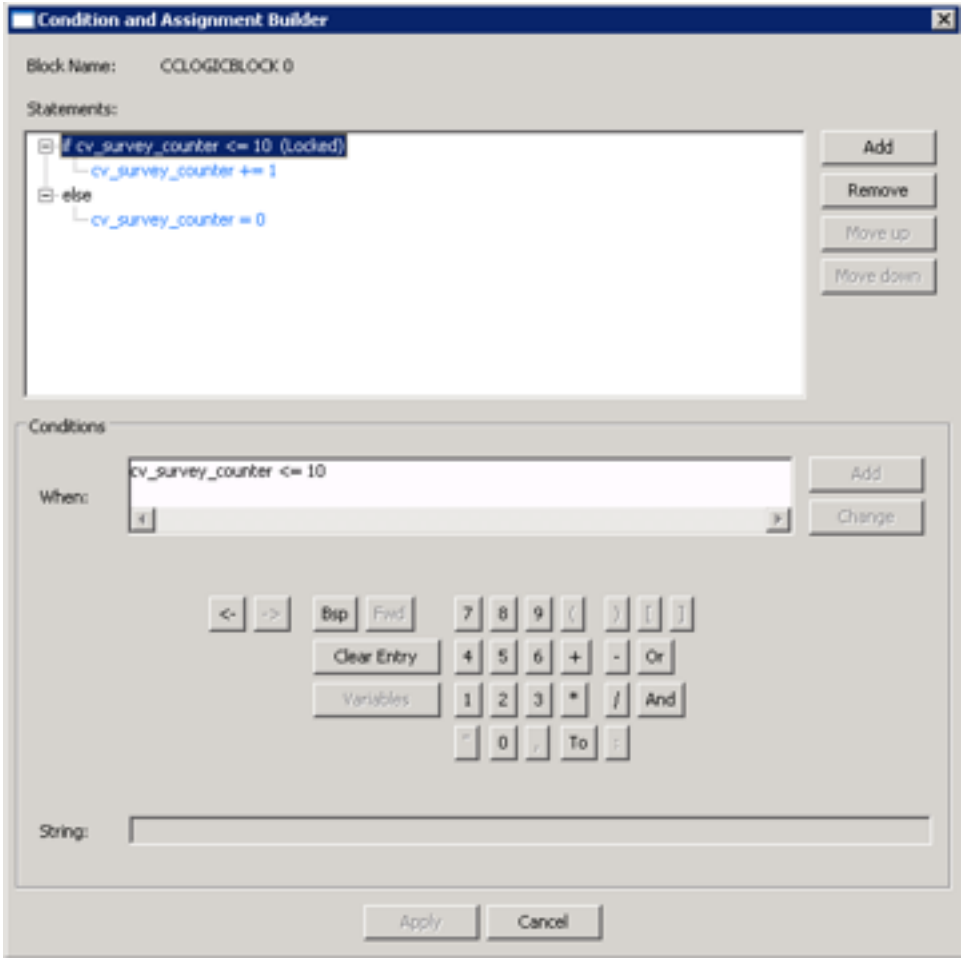


Figure 21: Condition and Expression builder



Figure 22: Processing tab for Logic block

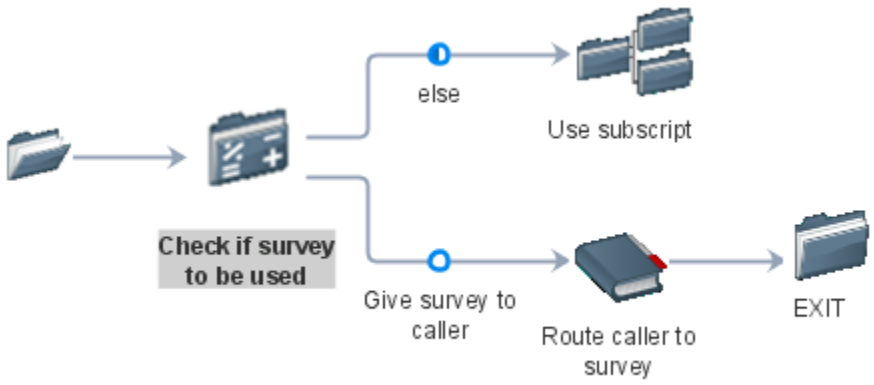


Figure 23: Flow using Logic block

Renaming blocks and nodes

About this task

Rename blocks or nodes in the Flow Editor to provide a meaningful name.

Important:

Meaningful names for blocks and nodes in your applications describe the action that is performed within the block or describes why the transition occurs.

Procedure

1. In the **Flow Editor** diagram, right-click the block or node to rename.
2. Select **Rename**.
3. Type a new name in the box below the icon.
4. Press **Enter** to accept the new name.

Defining block processing

Before you begin

- Understand how to use the Condition Builder. See Condition and Assignment builder operations on page 243.

About this task

Define block processing to create optional assignments or conditional assignments assigned after the block but before the transitional conditions, and optionally configure a log message for the flow application.

Block processing can occur in various types of blocks, such as queue, input, and output.

Procedure

1. Select the block icon in the **Flow Editor**, right-click and click **Open**.

   OR

   Double-click the block icon in the **Flow Editor**.
2. Click the **Processing** tab.
3. In the **Description** box, type a word description for the condition.
4. Beside **Assignment Expressions**, click **Edit** to open the Condition and Assignment Builder.
5. Use the Condition and Assignment Builder to create your assignment expression or conditional assignments.

Variable definitions

Name	Description
Add log command	Add a log command to create an entry in the event log on Contact Center Manager Server. Avaya recommends this option only if you have concerns with the application. Otherwise, the Log command option slows the function of the application. For example, you can capture a message for each block in the event log.

Name	Description
	You can use the log command together with the Debug only option for debugging your applications. You can remove all log commands by clearing the Include Debug Logging check box in the preferences of Orchestration Designer.
Assignment Expressions	An optional list of assignment expressions for variables.
Debug only	Select the option to include the log commands only when in debug mode. Orchestration Designer assigns the debug flag to true for the application. Avaya recommends this option only if you have concerns with the application. Otherwise, the debug option slows the function of the application. IF debug THEN LOG ‘message’ END IF If the Debug only box is cleared, then the Log message appears in the event log on Contact Center Manager Server. You can remove all log commands and debug information by clearing the Include Debug Logging check box in the preferences of Orchestration Designer.
Description	A description for the processing.

Example: Using the Setup and Processing tab together

Use the Host Data Exchange DIW capability to send the contact DNIS value to a host and, based on the DNIS, retrieve the customer type for the contact. Some callers are categorized as Gold and others as Platinum customers. The type of caller determines the priority to use when the contact is queued.

Use the Setup tab to explicitly assign the call variable to initialize it to 0. After the host data exchange processes are complete, if your call variable still contains the null value, you know that the host did not respond to the request.

Create the cv_response_string as a call variable to assign it a value in the variable definitions.

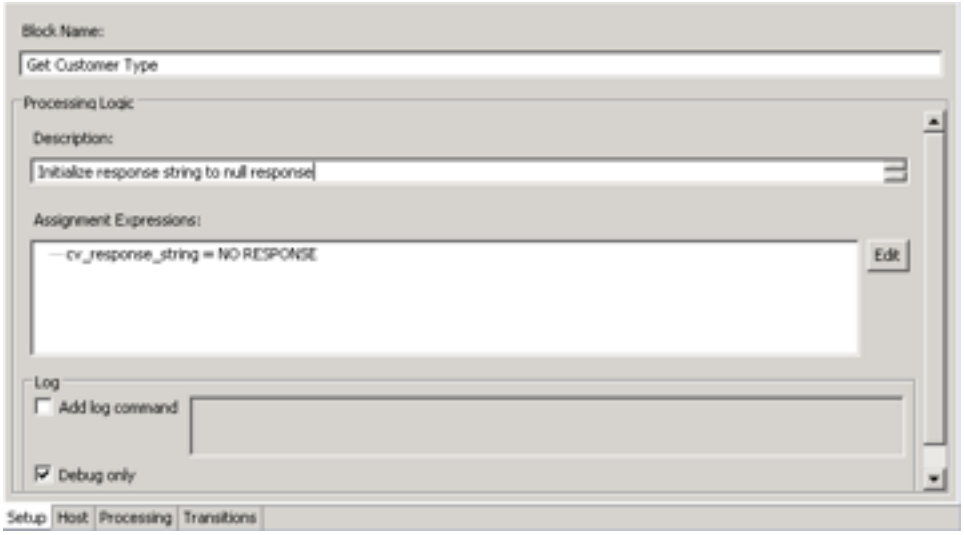


Figure 24: Setup tab showing initialization assignment string

In the Request Parameters field, include the store procedure that the host provider application uses for the DNIS lookup. The host provider also needs the DNIS value unique to the customer. The store procedure is an integer call variable. In the Response Parameters field, include the correct stored procedure for the retrieval of the customer type, and the string used to store the customer type (GOLD or PLATINUM). The store procedure is an integer call variable. The cv_response_string is currently assigned NO RESPONSE (from the Setup tab).

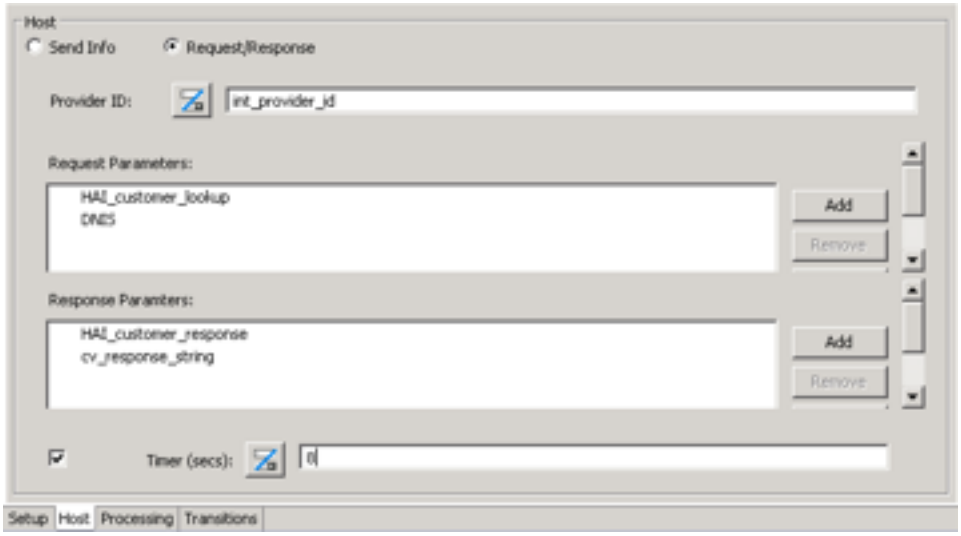


Figure 25: Host tab

Now use the processing tab to check the value in the customer type string to determine how to assign a priority. Gold customers are assigned a lower priority than the Platinum customers. If there is no response from the host application, assign the caller a lower priority than the other caller types.

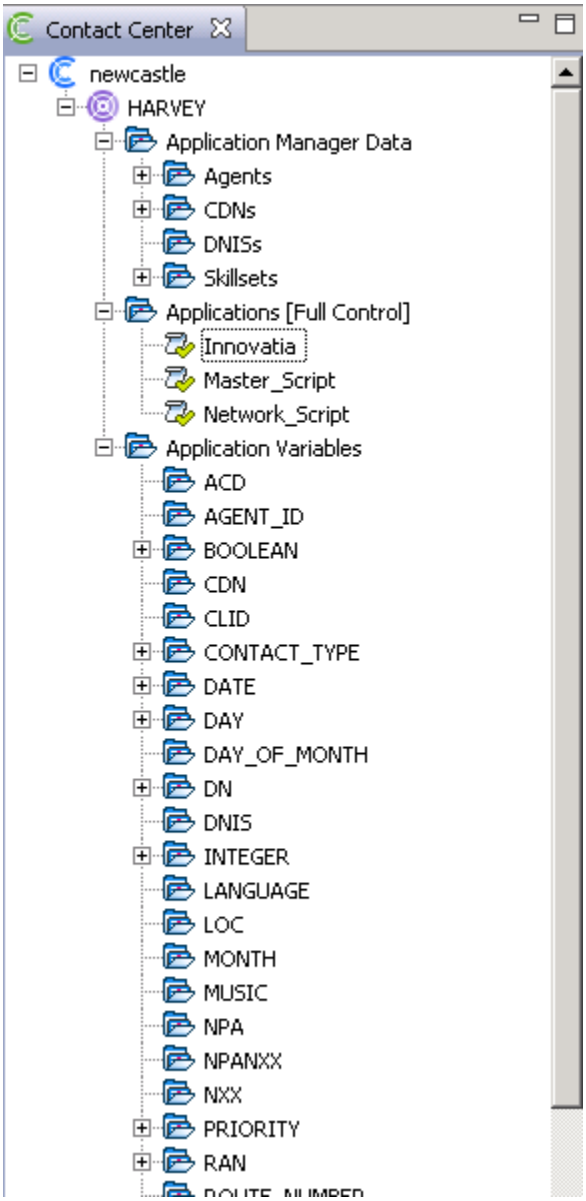


Figure 26: Processing tab

Creating a default transition

Before you begin

- Ensure that you have two blocks or nodes in the flow application to connect. See Adding blocks to the flow on page 144 or Adding blocks using accelerator keys on page 146.

About this task

Create a default transition between dialog blocks and nodes to create the pathway for an application flow. A default transition provides no conditions or expressions.

The source block is the block or node you want the transition to start, and the designation block is the block or node where you want the transition to terminate.

Procedure

1. Select the source block or node.
2. Right-click the destination block or node and select **(Dis)Connect**.

Defining conditional transitions

Before you begin

- Understand how to use the Condition Builder. See Condition and Assignment builder operations on page 243.

About this task

Define conditional transitions when a block has multiple transitions that lead from it. The multiple transitions, or conditional transitions follow if the condition statement associated with the transition evaluates to true. If the transition evaluates to false, the system processes the remaining transitions out of the block or proceeds along the path of the Else transition.

Each local transition that leads from the block is represented by its own tab on the Transitions tab of various blocks, such as Input, Queue, Output, and Host.

If only two conditions lead from a block, one transition is a Conditional transition and the other is an Else transition.

Procedure

1. Select the block icon in the **Flow Editor**, right-click and click **Open**.

   OR

Double-click the block icon in the **Flow Editor**.

2. Click the **Transition** tab.
3. Click **Add Transition**.
4. Select the transition tab you want to configure.
5. Beside **Conditions**, click **Edit** to open the Condition and Assignment Builder to create a conditional expression representing the condition that must be met for the transition to occur.
6. If you want to have assignment, conditional assignments or logs that are processed during the transition to the connected block, configure these items.
7. Repeat step 3 on page 157 to step 6 on page 157 for each transition, except for the Else transition.

Variable definitions

Name	Description
Add log command	Add a log command to create an entry in the event log on Contact Center Manager Server. Avaya recommends this option only if you have concerns with the application. Otherwise, the Log command option slows the function of the application. For example, you can capture a message for each block in the event log. You can use the log command together with the Debug only option for debugging your applications. You can remove all log commands by clearing the Include Debug Logging check box in the preferences of Orchestration Designer.
Assignment Expressions	An optional list of assignment expressions for variables.
Debug only	Select the option to include the log commands only when in debug mode. Orchestration Designer assigns the debug flag to true for the application. Avaya recommends this option only if you have concerns with the application. Otherwise, the debug option slows the function of the application. IF debug THEN

Name	Description
	LOG ‘message’ END IF If the Debug only box is cleared, then the Log message appears in the event log on Contact Center Manager Server. You can remove all log commands and debug information by clearing the Include Debug Logging check box in the preferences of Orchestration Designer.
Description	A description for the processing.

Procedure job aid

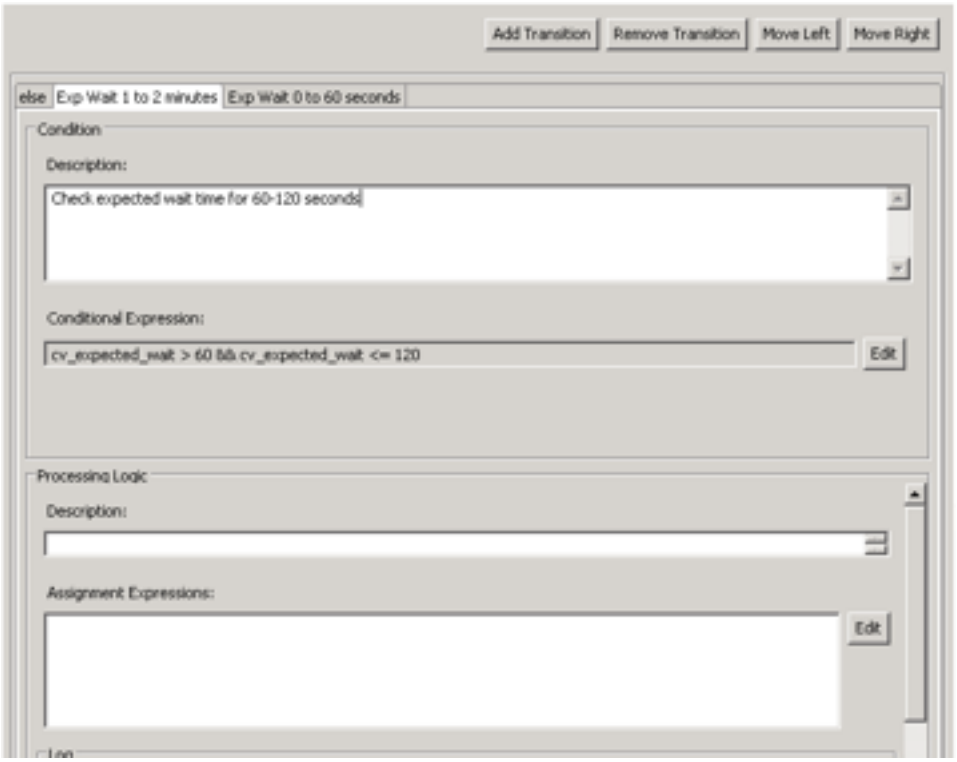


Figure 27: Transition tab in block editor

Defining switched transitions

Before you begin

- Understand how to use the Condition and Expression Builder. See Condition and Assignment builder operations on page 243.

About this task

Define switched transitions when a block has multiple transitions that lead from it. Each transition is based on the value of a variable or application manager data.

For example, in your contact center, you have DNs, you can have one action if the DN is 123, and another action if the DN is 234, and a third action if the DN is 345 and so on.

Depending on the value of the variable, one of the transitions is followed. If the value is not listed, an else transition handles the routing for the remaining values.

In Orchestration Designer, managing the multiple transitions can be difficult to view because the monitor is not wide enough. If you use a switch statement, the transitions are displayed in a list of cases.

Important:

When you create a conditional transition, label the transition so that you know the precise path is for the contact.

Procedure

1. In the **Flow Editor**, right-click the block from which you want to transition, and select **Specify Transition Type** > **Switch Statements**.
2. Open the block editor by double-clicking on a block in your flow application, click the **Transition** tab.
3. Click **Variables**.
4. Select the variable and the operation you want to configure.
5. Under **Cases**, select the value you want to configure.

   Important:

   Switch Transitions are ordered. You can alter the order of the **Cases** by using the **Move Up** and **Move Down** buttons. The order is directly reflected in the Task Flow Executor (TFE) script generated by Orchestration Designer.
6. Beside **Conditions**, click **Edit** to open the Condition and Assignment Builder to create assignments you may want for the condition you have selected.

7. Repeat step 4 on page 159 to step 6 on page 159 for each transition, except for the Else.

Procedure job aid

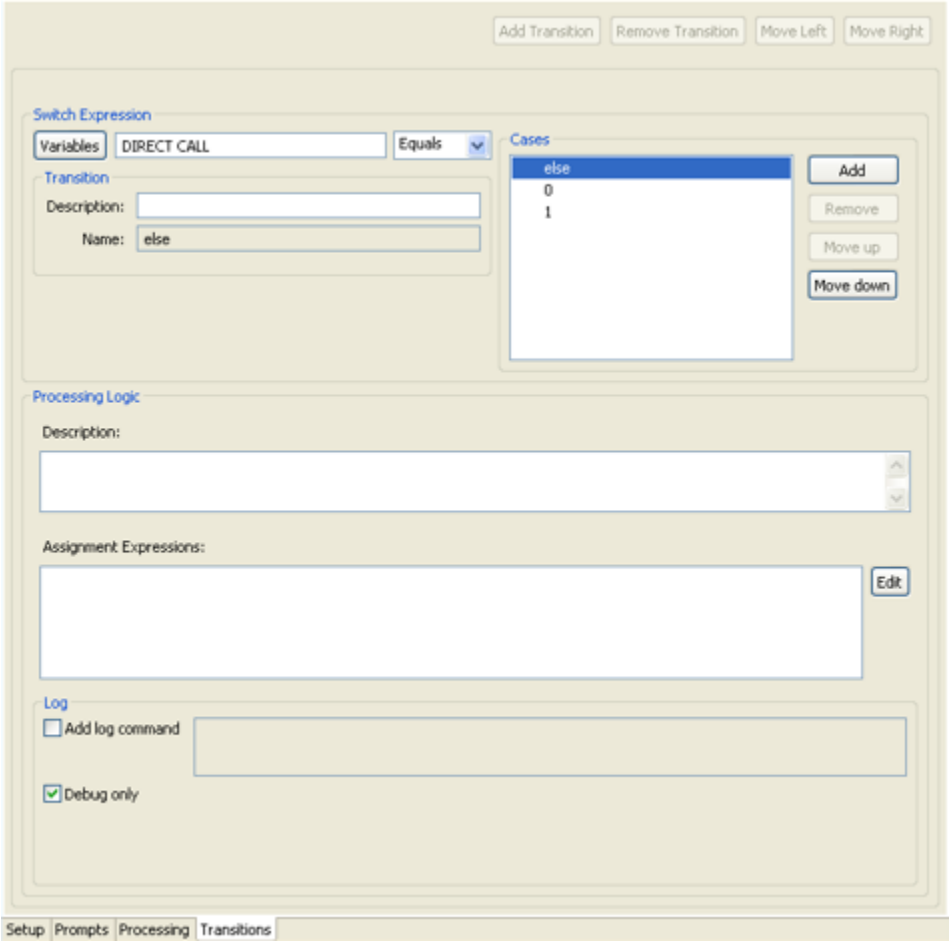


Figure 28: Switch transition tab in block editor

Copying and pasting blocks

About this task

Copy and paste blocks to reproduce the block and configuration information in another part of the application.

Copy of text indicates the copy of the block until you assign a new name to the new block.

You can copy and paste blocks within the same application, or paste the new block in a different application.

Procedure

1. Select a single object or a group of objects.

   You can select multiple objects by pressing **Ctrl** and clicking multiple objects.

   OR

   Click and hold the left mouse button and drag the selection tool around multiple objects.
2. Right-click the selected items, and click **Copy**.
3. Right-click the part of the **Flow Editor** where you want to add the object and select **Paste**.

Procedure job aid

Table 10: Shortcuts in Flow Editor

	Cut selected block.
	Copy selected block.
	Paste selected block into flow diagram.

Deleting blocks, nodes or transitions

About this task

Delete blocks, nodes, or transitions from the Flow Editor to remove them from the flow application.

Procedure

Right-click the block, node or transition you want to delete and select **Delete**.

OR

Select the block, node or transition you want to delete and, on the Edit bar, click **Delete**.

Procedure job aid

Table 11: Shortcut in Flow Editor

	Delete selected item.

Deleting a transition

About this task

Delete a transition in the Flow Editor to modify the connections between nodes in your flow application.

Procedure

1. In the **Flow Editor** diagram, right-click the transition.
2. Click **Delete**.

Procedure job aid

Table 12: Shortcut in Flow Editor

	Delete selected item.

Deleting multiple objects

About this task

Delete multiple objects to remove them from the Flow Editor.

Procedure

1. Select the objects in the **Flow Editor** that you want to delete.
   You can select multiple objects by pressing **Ctrl** and clicking multiple objects.
   OR

Click and hold the left mouse button and drag the selection tool around multiple objects.

2. Right-click the objects and select **Delete**.

Disconnecting blocks

Before you begin

- Ensure that you have two connected dialog blocks on the Flow Editor diagram.

About this task

Disconnect blocks to delete the pathway in a flow application.

When you disconnect blocks, click the same block you first click to create the connection, and then right-click the second block in the connection. Otherwise, Orchestration Designer creates another flow in the opposite direction.

Procedure

1. Select a source block.
2. Right-click the destination block and select **(Dis)Connect**.

Arranging blocks in the flow diagram

About this task

Arrange blocks and transitions using an automatic layout to configure the spacing between the blocks.

You can arrange the blocks in incremental mode or full layout mode.

Procedure

On the **Flow Editor** tool bar, click the appropriate layout button.

Procedure job aid

Table 13: Layout modes in Flow Editor

	Display blocks of flow diagram in incremental mode. Incremental Layout realigns the position of the nodes and blocks into rows and columns with respect to the current position. Incremental layout attempts to maintain the current layout of nodes and blocks.
	Display blocks of flow diagram in full layout mode. Full Layout rearranges the position of the nodes and blocks to minimize line crossing. The blocks and nodes are completely rearranged.

Inserting a new page

About this task

Insert a new page into the Flow Editor so that you can view large applications without scrolling. Tabs on the bottom of the Flow Editor pane represent pages in the flow.

Procedure

1. At the bottom of the **Flow Editor** diagram, right-click a tab.
2. Select **Insert Page**.

Deleting pages

About this task

Delete pages to remove them from the Flow Editor.

Procedure

Right-click the tab for the page to delete and select **Delete**.

Renaming a page

About this task

Rename a page in the Flow Editor to provide a meaningful name for the page of your flow application.

Procedure

1. Below the **Flow Editor** diagram, right-click the tab that you want to rename.
2. Select **Rename**.
3. In the **Input** window, type a new name for the page.
4. Click **OK**.

Adding a note to a block

About this task

Add a note to a block to add reminders or descriptions about the dialog block for future reference.

When you save a note for a block or node, the block or node icon appears with a yellow note symbol in your Flow Editor.

If you configure your Orchestration Designer preferences to include comments, your comments appear in the script applications if you export your flow application to a script.

Procedure

1. On the **Flow Editor** diagram, right-click the block or node to which you want to add a note.
2. Select **Notes**.
3. Type your explanation or reminder.
4. Click **OK**.

Using Zoom to Fit

About this task

Use the Zoom to Fit button to scale the dialog flow to view the entire application in the Flow Editor window.

Procedure

On the **Flow Editor** Edit bar, click **Zoom to Fit**.

Procedure job aid

Table 14: Shortcut in Flow Editor

	Zoom application to fit window.

Zooming in

About this task

Zoom in on the Flow Editor diagram to enhance portions of the dialog flow for viewing. Use this feature to isolate portions of large and complex dialog flows.

Procedure

1. On the **Flow Editor** bar, click **Zoom In**.
2. Click the area of the dialog flow that you want to enhance.
3. Repeat until you achieve the enhancement level.

Procedure job aid

Table 15: Shortcut in Flow Editor

	Zoom in on flow application blocks.
	Zoom to selected box area.

Zooming out

About this task

Zoom out on the Flow Editor to view larger portions of the dialog flow.

Procedure

1. On the **Flow Editor** bar, click **Zoom Out**.
2. Click the area of the diagram that you want to reduce.
3. Repeat the steps until you achieve the view you want.

Procedure job aid

Table 16: Shortcut in Flow Editor

	Zoom out on flow application blocks.

Resetting the size of the diagram

About this task

Use the Reset Zoom button to return the contents of the Flow Editor to the previous scale.

Procedure

On the **Flow Editor** bar, click **Reset Zoom**.

Procedure job aid

Table 17: Shortcut in Flow Editor

	Reset zoom to original level.

Moving the flow application

About this task

You can move the flow application blocks around within the Flow Editor window either by moving one block at a time, or by simultaneously moving all the blocks to see a different part of the entire flow application.

Procedure

On the **Flow Editor** bar, click **Move flow**.

Procedure job aid

Table 18: Shortcut in Flow Editor

	Move flow to desired location.

Viewing the block summary

Before you begin

- Ensure the Flow Editor is active by clicking it.

About this task

View the block summary to see the dialog block configuration information without opening the block editor.

Procedure

Position the pointer over the dialog block or node.
A pop-up window that contains the block summary appears.

Selecting application management data in block configuration

Before you begin

- Configure the application management data (DNs, variables). For more information, see Application manager data and variable configuration on page 251.
- Open a block editor. See Adding blocks to the flow on page 144.

About this task

Select application management data when you configure blocks. In many block element configurations, you can select a DN, skillset, variable, or other application management data to represent timer duration, RAN routes, priorities and all other types of data.

You can also type literal strings or integers in the fields.

Procedure

1. In the block editor, locate a configuration item.
   The configuration items where you can apply application manager data are identified by an icon.



2. Type the literal text or numbers you want to use in the space provided.

   OR

Click .

If you click the button to choose the Application Management Data, the definition field appears.



3. Click **Browse**.
4. In the **Chooser** window, select the **Application Management Data** you want to use in your configuration.
5. Click **OK**.

Procedure job aid

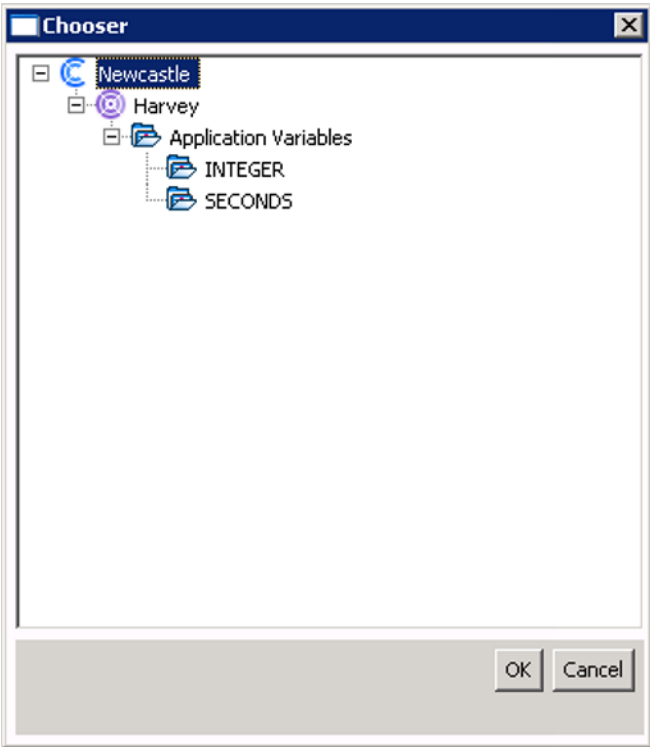


Figure 29: Chooser window for Application Management Data

Chapter 15: Block creation

In Orchestration Designer, you create blocks that contain information about what happens to a contact as it arrives, is processed by, and leaves the block. In the Flow Editor, each block appears as an icon in a diagram. Each block connects to the previous block and the next block by a transition, or connector.

This chapter provides a step-by-step look at each type of block and how to configure it.

Most blocks have three common configuration tabs:

- Setup—Configure the name, an optional description, create assignments or conditional assignments which you can determine prior to the block, and optionally configure a log message for the flow application. To configure the setup tab, see Configuring general details for blocks on page 147.
- Processing—Configure optional assignments or conditional assignments which process after the block configuration but before the transitional conditions, and optionally configure a log message for the flow application. To configure the processing tab, see Defining block processing on page 152.
- Transition—Configure the transitional condition for each connection from the block to other blocks in the application flow after the block, configure optional assignments or conditional assignments which are assessed to determine which transition to use, and optionally configure a log message for each transition in the flow application. To configure a transition, see Defining conditional transitions on page 156.

The Flow Editor palette contains the blocks that you can add to your flow application.

Prerequisites for block creation

- Open the Flow Editor. See Creating a new application on page 110 or Opening an existing application on page 115.
- Understand how to define the setup details for your block. See Configuring general details for blocks on page 147.
- Understand how to define processing details for your block. See Defining block processing on page 152.
- Understand how to define conditional transitions, if required, for your block. See Defining conditional transitions on page 156.
- Understand the condition editor for creating expressions. See Condition and Assignment builder operations on page 243.

Creating a logic block

Before you begin

- Add the Logic block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a logic block to configure conditional situations in the flow. It provides the conditional processing for a contact in the flow.

Many configuration items, such as name, description, assignments, and log messages are optional. If you do not complete a required field, an error message appears when you close the Logic block editor.

Procedure

1. Select the Logic block icon ( ), right-click and click **Open**.

   OR

   Double-click the Logic block icon.
2. Click the **Processing** tab to configure the general details for the block.
3. Click the **Transition** tab to configure transitions from this block.
4. Save the changes to the flow application.

Example: Using the logic block

Use a logic block to make a decision that depends on the survey counter variable. A locked assignment checks the value of a survey counter call variable. The assignment is locked to ensure that the data in the counter call variable is preserved for the entire flow application. If the value is less than or equal to 10, use an assignment expression to increase the survey counter by 1. If the counter is 10, use an assignment expression to assign the survey counter to 0.

This conditional assignment must occur before the transitions from the logic block since the decisions about the next step in the application depends on the value of the variable.

To configure the assignment, create the call variable cv_survey_counter with an initial value of 0 in the Application Variables page (see Creating variables with one value on page 262), for the call variable in the Local view (see Adding a variable to Orchestration Designer on page 257).



Figure 30: Processing tab for Logic block

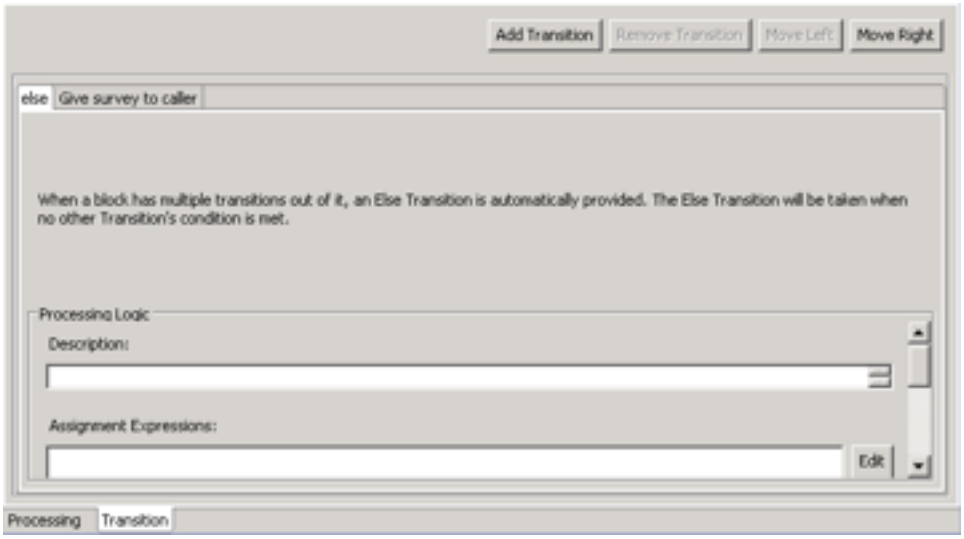


Figure 31: Transition tab for Logic block

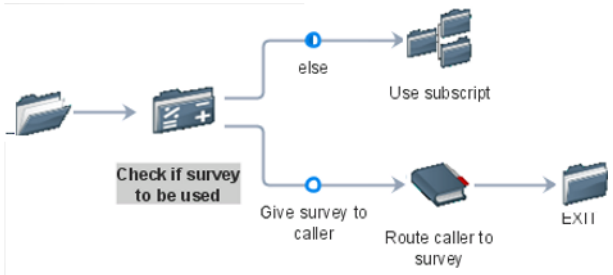


Figure 32: Flow using Logic block

Creating a queue to skillset block

Before you begin

- Add the queue block icon () to the flow application. See Adding blocks to the flow on page 144.
- Configure the skillsets in the Contact Center Manager Administration application.

About this task

Create a Queue block to assign the contact to a skillset queue so that an agent assigned to the skillset can handle the contact. You can also configure the skillsets for the Queue block by adding, changing, or removing the skillsets.

In addition to the main configuration tabs, the queue block also has the queue tab. Use the queue tab to configure the information for the queue, including assigning the properties for the queue, or a specific agent based on the queue time, and the priority of the contact.

Many configuration items, such as name, description, assignments, and log messages are optional. If you do not complete a required field, an error message appears when you close the Queue block editor.

Procedure

1. Select the queue block icon (), right-click and click **Open**.

   OR

   Double-click the queue block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Queue** tab.
4. Select **Add**, **Modify** or **Remove** to designate how to apply the contact to the queue.
5. Under **Queue To**, select **Skillset**.
6. Select the optional properties for agents in the skillset.

   OR

   Under **Skillsets**, click **Add** to select the skillset to queue the contact.
7. In the **Priority** box, select the priority for the contact in the skillset.

   OR

   Click  and then click **Browse** to specify an priority variable from the application manager data.

8. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
9. Click the **Transition** tab to configure transitions from this block.
10. Save the changes to the flow application.

Variable definitions

Name	Description
Command	Indicate how to handle the contact: • Add the skillset to the queue list. • Change the priority of the call in the skillset. • Remove the skillset from the queued call.
Most Idle	Automatically select the skillset that is most idle (received the least amount of work handling contacts). You need not assign a specific skillset for the queue.
Most Available Agent	Select the agent that has the least number of contacts automatically.
Most Logged in Agents	Automatically select the skillset with the most logged-in agents. You need not assign a specific skillset for the queue.
Priority	The rank of the contact where 1 is highest priority, meaning the contact is handled first and 10 is the lowest priority. A default value for the queue priority is set in the preferences.

Example: Queue to skillset block

Queue a call to a local skillset. Use the queue block to make decisions, based on the value of an Expected Wait Time variable to determine the message type plays using a RAN.

Use the Setup tab to name the queue block.

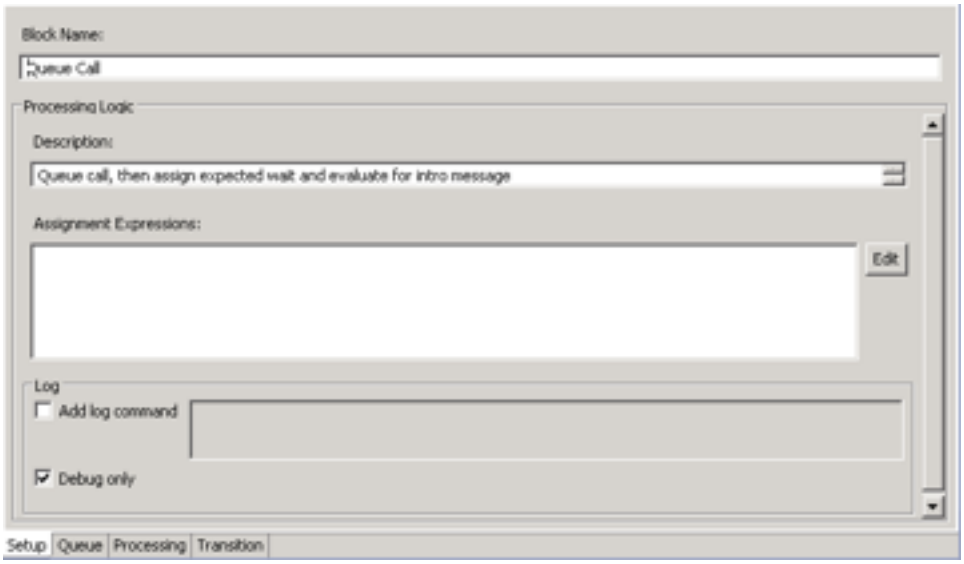


Figure 33: Queue to skillset Setup tab

Select the skillset and priority for the contact on the Queue tab.

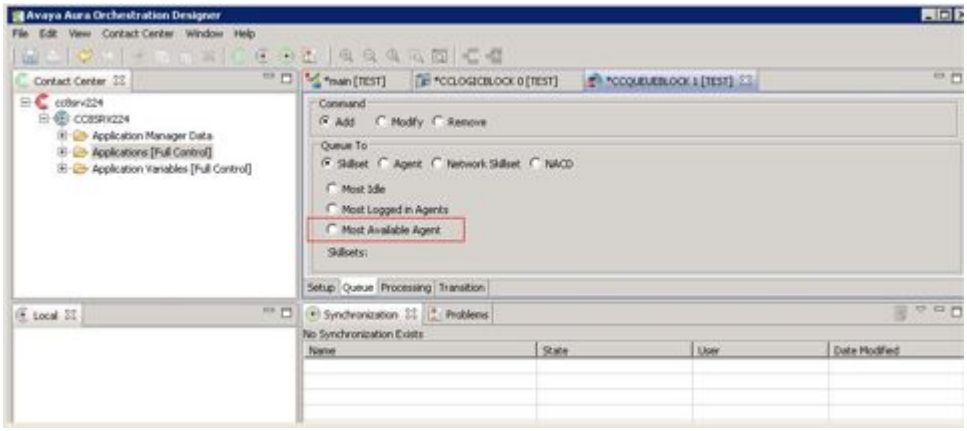


Figure 34: Queue to skillset Queue tab

Assign the Expected Wait Time for the skillset you select on the Queue tab.

The processing is ideal for this assignment because you decide the skillset to which to assign the contact.

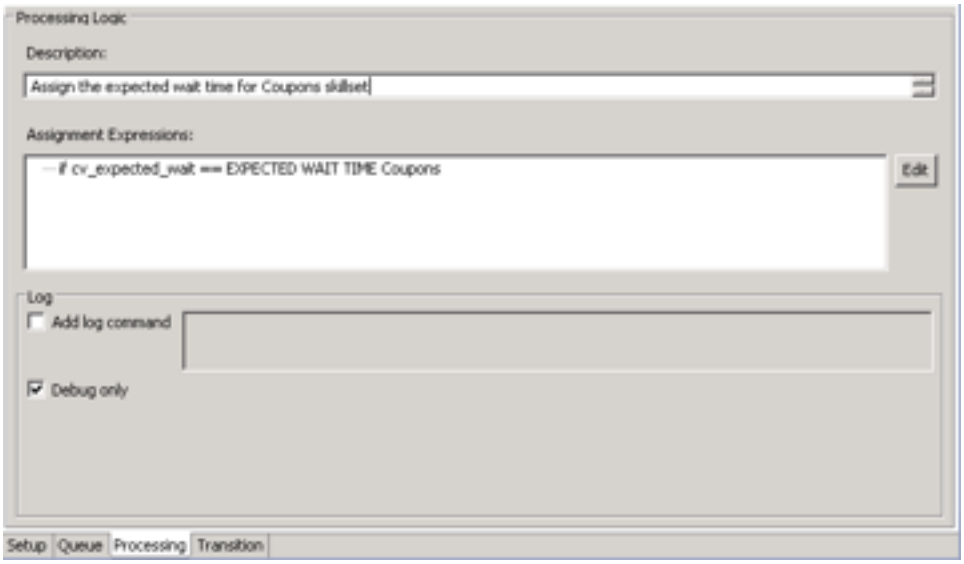


Figure 35: Queue to skillset Processing tab

The transitions describe the conditions to make the flow application easy to follow.

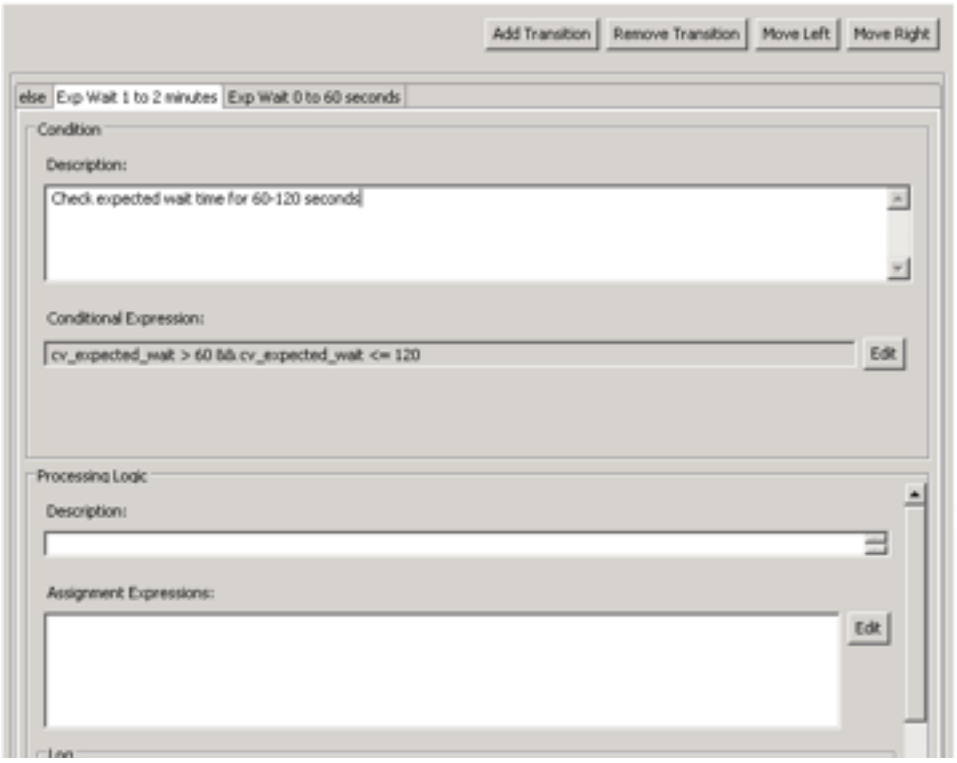


Figure 36: Queue to skillset Transition tab

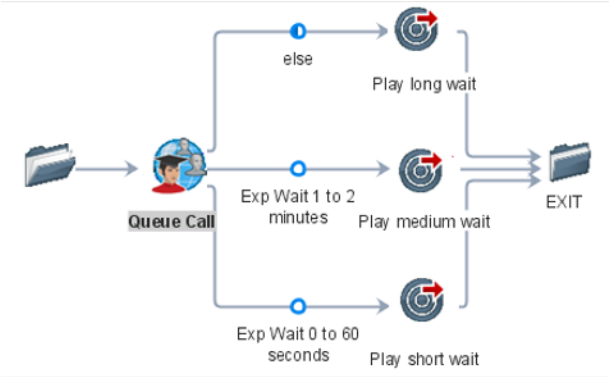


Figure 37: Queue to skillset flow sample

Creating a queue to network skillset block

Before you begin

- Add the queue block icon () to the flow application. See Adding blocks to the flow on page 144.
- Configure the network skillsets in the Contact Center Manager Administration application.

About this task

Create a queue network skillset block to assign the contact to a queue represented by a network skillset. The Network Control Center propagates a network skillset to all sites and the Network Routing tables determine the locations. You must be licensed for networking to use the networked skillsets.

In addition to the main configuration tabs, the queue block also has the queue tab. Use the queue tab to configure the information for the queue, including assigning the properties for the network queue, and the priority of the contact.

Many configuration items, such as name, description, assignments and log messages are optional. If you do not complete a required field, an error message appears when you close the Queue block editor.

Procedure

1. Select the queue block icon (), right-click and click **Open**.

   OR

   Double-click the queue block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Queue** tab.
4. Select **Add**, **Modify** or **Remove** to designate how to apply the contact to the queue.
5. Under **Queue To**, select **Network Skillset**.
6. Under **Skillsets**, click **Add** to select the network skillset to queue the contact.
7. In the **Priority** box, select the priority for the contact in the skillset.

   OR

   Click  and then click **Browse** to specify an priority variable from the application manager data.

8. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
9. Click the **Transition** tab to configure transitions from this block.
10. Save the changes to the flow application.

Variable definitions

Name	Description
Command	Indicate how to the handle the contact: • Add the skillset to the queue list. • Change the priority of the call in the network skillset. • Remove the skillset from the queued call.
Priority	The rank of the contact where 1 is highest priority, meaning the contact is handled first and 10 is the lowest priority. A default value for the queue priority is set in the preferences.

Procedure job aid

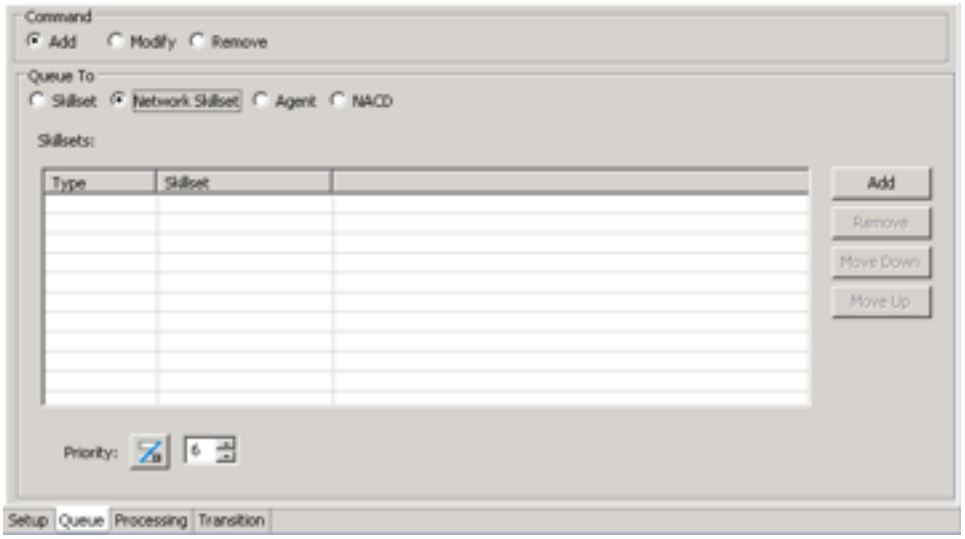


Figure 38: Queue to network skillset block configuration

Creating a queue to agent block

Before you begin

- Add the queue block icon () to the flow application. See Adding blocks to the flow on page 144.
- Configure the skillsets or agents in the Contact Center Manager Administration application.

About this task

Create a Queue block to assign the contact to an agent to handle the contact. You can assign a contact to a particular agent if you know that one agent meets the criteria for handling a particular contact.

In addition to the main configuration tabs, the queue block also has the queue tab. Use the queue tab to configure the agent or, if the longest idle agent or most available agent is selected, the skillset information for the queue, and the priority of the contact.

Many configuration items, such as name, description, assignments and log messages are optional. If you do not complete a required field, an error message appears when you close the Queue block editor.

Procedure

1. Select the queue block icon (), right-click and click **Open**.

   OR

   Double-click the queue block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Queue** tab.
4. Select **Add**, **Modify** or **Remove** to designate how to apply the contact to the queue.
5. Under **Queue To**, select **Agent**.
6. To select the longest idle agents for all skillsets, select **Longest Idle Agents**.

   To select the most available agents for all skillsets, select **Most Available Agent**.
7. If you select the longest idle agent or most available agent, in the **Agents** table, click **Add** to select the agents to which to queue the contact.

   OR

   If you did not select the longest idle agent or most available agent for all skillsets, in the **Skillsets** table, click **Add** to select the skillset to queue the contact.
8. In the **Priority** box, select the priority for the contact.

OR

Click  and then click **Browse** to specify an priority variable from the application manager data.

9. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
10. Click the **Transition** tab to configure transitions from this block.
11. Save the changes to the flow application.

Variable definitions

Name	Description
Agents	The agents that can handle the contact. You can choose a maximum of 10 agents. The agent who appears at the top of the list receives the contact.
Command	Indicate how to handle the contact: • Add the agents to the queue list. • Change the priority of the call for the agent. • Remove the agent from the queued call.
Longest Idle Agents	Choose an agent based on the longest agent idle over all skillsets. You can choose from which skillsets to review the idle time for the agents.
Most Available Agent	Select the agent that has the least number of contacts automatically.
Priority	The rank of the contact where 1 is highest priority, meaning the contact is handled first and 10 is the lowest priority.
Skillset	Select a group of skillsets for which to choose the Longest Idle Agent.

Procedure job aid

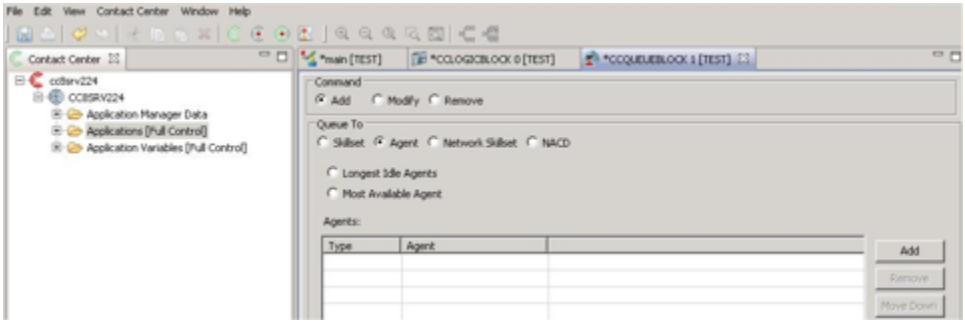


Figure 39: Queue to agent block configuration

Creating a queue to networked ACD block

Before you begin

- Add the queue block icon ( ) to the flow application. See Adding blocks to the flow on page 144.
- Configure the network ACD groups in the Contact Center Manager Administration application.

About this task

Create a Queue block to assign the contact to a networked ACD so that an agent can handle the contact. You must choose the acd_dn for the queue.

In addition to the main configuration tabs, the queue block also has the queue tab. Use the queue tab to configure the information for the queue, including assigning the properties for the queue, or a specific agent based on the queue time and the priority of the contact.

Many configuration items, such as name, description, assignments, and log messages are optional. If you do not complete a required field, an error message appears when you close the Queue block editor.

Procedure

1. Select the queue block icon ( ), right-click and click **Open**.

   OR

   Double-click the queue block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Queue** tab.

4. Select **Add**, **Modify** or **Remove** to designate how to apply the contact to the queue.
5. Under **Queue To**, select **NACD**.
6. In the **acd_dn** box, type the acd_dn for the contact.

   OR

   Click  and then click **Browse** to specify an acd_dn from the application manager data.
7. In the **Priority** box, select the priority for the contact.

   OR

   Click  and then click **Browse** to specify an priority variable from the application manager data.
8. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
9. Click the **Transition** tab to configure transitions from this block.
10. Save the changes to the flow application.

Variable definitions

Name	Description
Command	Indicate how to handle the contact: • Add the networked ACD DN to the queue list. • Change the priority for the networked ACD DN. • Remove the networked ACD DN from the queued call.
Priority	The rank of the contact where 1 is highest priority, meaning the contact is handled first and 4 is the lowest priority. A default value for the queue priority is set in the preferences.

Procedure job aid

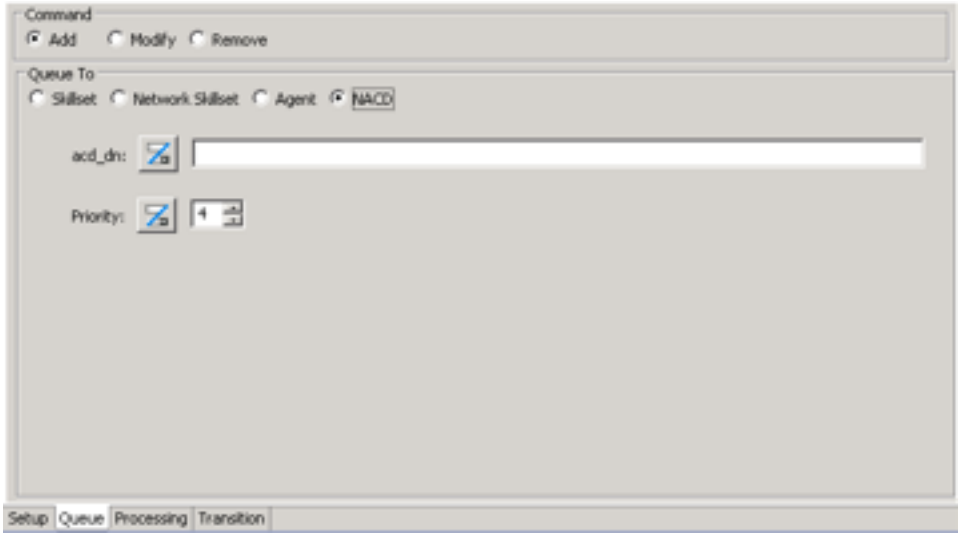


Figure 40: Queue to NACD block configuration

Creating an input block

Before you begin

- Add the input block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create an input block to prompt or ask the caller a question, process the caller's response, and determine what to do next. You can configure initial, retry, timeout, and conditional prompts that play during the data collection process.

In addition to the main configuration tabs, the input block also has two additional tabs:

- Prompts—Configure the maximum number of retries allowed for timeout and no match conditions. Also configure the prompt types and conditions for the prompt execution.
- Digit Collection—Configure the number of digits to collect and what digits are expected. Retry processing if unexpected digits are entered.

Many configuration items, such as name, description, assignments, and log messages are optional. If you do not complete a required field, an error message appears when you close the Input block editor.

Procedure

1. Select the input block icon (), right-click and click **Open**.

   OR

   Double-click the input block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Prompts** tab.
4. Beside **Max Retries**, select the maximum number of retries for the user input. If the Max Retries is greater than 0, the Retry and Timeout prompts display in the Prompt List.
5. To configure the Preamble prompt, right-click **Preamble**, then choose **Add Variable Prompt**.
6. Scroll down the **Prompts** tab to the **Selected Row Details** and complete the voice segment or digits, barge-in, and language details for your prompt.
7. To configure the Initial prompt, right-click **Initial**, then choose **Add Variable Prompt**.
8. Scroll down the **Prompts** tab to the **Selected Row Details** and complete the voice segment or digits, barge-in, and language details for your prompt.
9. To configure the Retry prompt, right-click **Retry**, then choose **Add Variable Prompt**.
10. Scroll down the **Prompts** tab to the **Selected Row Details** and complete the voice segment or digits, barge-in, and language details for your prompt.
11. To configure the Timeout prompt, right-click **Timeout**, then choose **Add Variable Prompt**.
12. Scroll down the **Prompts** tab to the **Selected Row Details** and complete the voice segment or digits, barge-in, and language details for your prompt.
13. Click the **Digit Collection** tab.
14. Beside **Number of Digits**, select the number of digits you expect the customer to type in response to the prompts. You can type a literal value for the number of digits, or press the button and click Browse to choose an application variable.
15. Beside **Timeout (secs)**, select the maximum number of seconds to wait for the customer before you play the Timeout message.
16. Beside **Terminating Character**, select the Terminating character the customer enters when they finish.
17. To flush the previously collected digits, select the **Flush previously collected digits** check box.
18. Configure the **Valid Input** check.

Include the conditional expressions for required retry/timeout prompting.

19. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
20. Click the **Transition** tab to configure transitions from this block.
21. Save the changes to the flow application.

Variable definitions

Name	Description
Barge-in	Select this check box to allow the caller to interrupt the prompt. Clear the check box to prevent the caller from interrupting the prompt or data collection.
Condition	The specific time when the prompt is played: • Always—The prompt plays for every contact. • If <condition>—The prompt plays for every contact that meets the condition you configure. Type a name and a description for the condition and use the Condition builder to create the conditional expression.
Flush previously collected digits	When this check box is selected, the system deletes all digits entered by the customer while the prompt plays. This forces the caller to start entering digits from the beginning.
Max Retries	The maximum number of retries for accepting user input.
Number of Digits	The number of digits you expect the caller to enter. For example if your caller is entering in an account number, and your account numbers only have eight digits, your number of digits is 8.
Prompt Type	There are two types of prompts: • The preamble prompt plays once to the customer when the contact enters this block in the flow application. Typically, the

Name	Description
	preamble prompt is used for introductory comments. • The initial prompt is the main collection prompt that plays. The initial prompt plays each time the block is entered. The prompt can be a voice segment or data gathered during the current input block.
Terminating character	The character the caller enters to identify that the last character of the number entered on their telephone keypad.
Timeout (secs)	The maximum delay the timer waits before playing the timeout prompt to the caller.
Voice Segment/Speak As	The name of the VOICE_SEGMENT variable file on the Contact Center Manager Server used for the prompt.
Wording/Value	Voice segment can be one of the following: • Voice Segment—A prerecorded statement stored on the Contact Center Manager Server. • Individual Digits—Echo the digits previously collected from the customer or digits that were entered into the text box as a literal value as individual digits. For example, 400 is spoken as four, zero, zero. • Whole Numbers—Echo the digits previously collected from the customer or digits that are entered into the text box as a literal value as a whole. For example, 400 is spoken as four hundred. **Note:** Variables of the DN type do not support leading zeros. DN variables containing leading zeros are spoken as individual digits. Use the Voice Segment box to navigate to the VOICE_SEGMENT variable file on the Contact Center Manager Server. Use Prompt Wording when you specify a Voice Segment. Use the text box to type a description of the voice segment file.

Procedure job aid

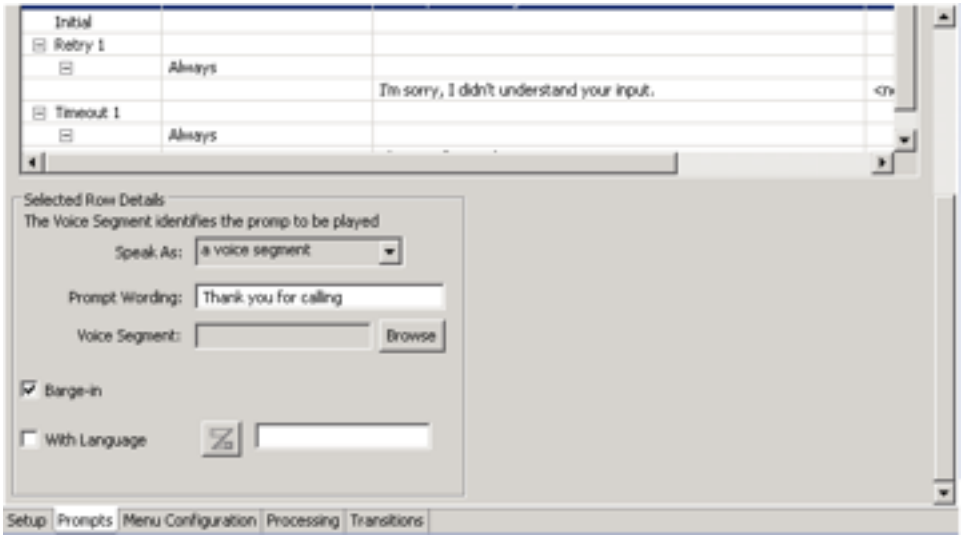


Figure 41: Input block configuration for prompts

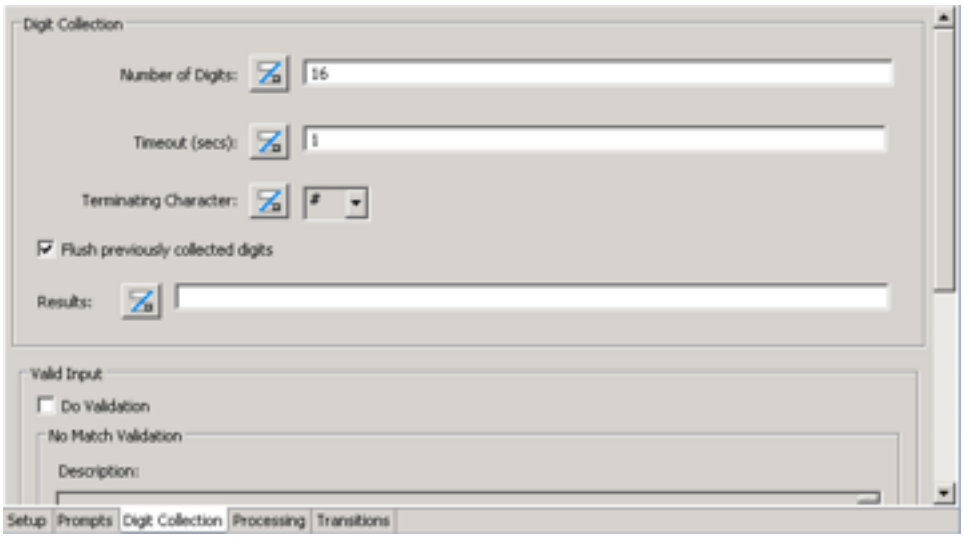


Figure 42: Input block configuration for digit collection

Creating a block to play a RAN

Before you begin

- Create a RAN file on the Avaya Media Server.
- Create and acquire a route in the Contact Center Manager Administration server. For more information on configuring and acquiring routes, see *Avaya Aura® Contact Center Manager Administration – Client Administration* (NN44400-611).
- Add the Output block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create an Output block to play a recorded announcement (RAN) to the caller. In a SIP-enabled contact center, you create and store a recorded announcement file on the Avaya Media Server. You create and acquire a route number for the Avaya Media Server RAN file on the Contact Center Manager Administration server. The route number allows Orchestration Designer to access the RAN file.

Many configuration items, such as name, description, assignments and log messages are optional. If you do not complete a required field, an error message appears when you close the Output block editor.

Note:

For RAN blocks in SIP-enabled contact centers, you must use a Music or Ringback block after the RAN block to provide a default treatment. If you do not configure a default treatment after the RAN block, the customer hears silence.

Procedure

1. Select the Output block icon ( ), right-click and click **Open**.

   OR

   Double-click the Output block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Prompts** tab.
4. Select **RAN**.
5. To silence the previous treatment, click **Silence Previous Treatment**.
6. In the **RAN Route** box, type the RAN Route for the recorded announcement.

   OR

   Click and then click **Browse** to specify a recorded announcement from the application manager data.

7. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
8. Click the **Transition** tab to configure transitions from this block.
9. Save the changes to the flow application.

Variable Definitions

Name	Description
Silence Previous Treatment	You can configure Silence Previous Treatment that gives silence after the RAN. The silence is required only if a previous block in your flow played music. To play music after the RAN, you must add it.
RAN Route	Route created and acquired on CCMA for the Avaya Media Server RAN file.

Procedure job aid

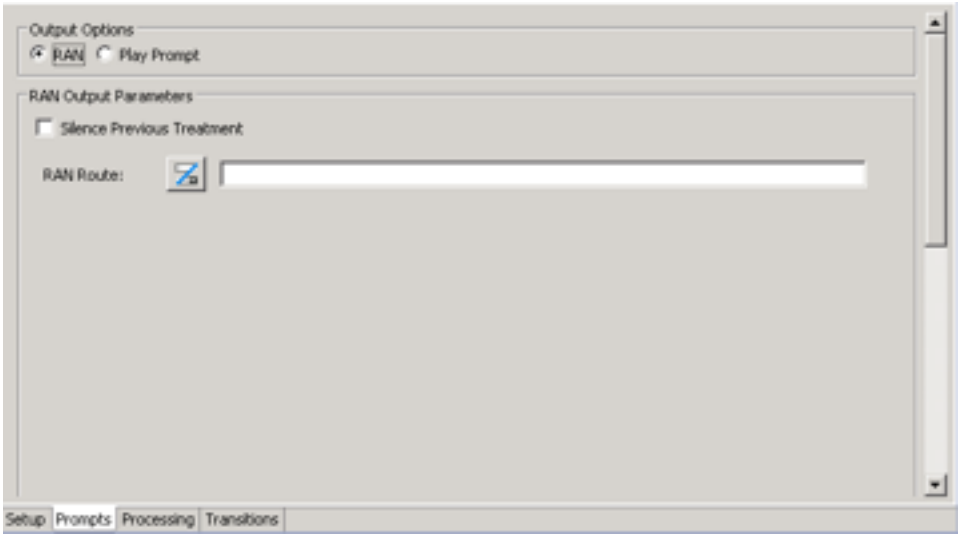


Figure 43: Output block for playing RAN

Creating a block to play a RAN in an AML-based contact center

Before you begin

- Upload a RAN file to the PABX hardware.
- Create a route on the PABX for the RAN, for more information on configuring a RAN route, see *Avaya Aura® Contact Center Configuration – Avaya CS 1000 Integration* (NN44400-512).
- Configure and acquire the route on the Contact Center Manager Administration server. For more information on configuring and acquiring routes, see *Avaya Aura® Contact Center Manager Administration – Client Administration* (NN44400-611).
- Add the Output block icon () to the flow application. See Adding blocks to the flow on page 144.

About this task

Create an Output block to play a recorded announcement (RAN) to the caller. In an AML-based contact center, upload RAN files to the RAN hardware on the PABX. You create a route for the RAN on the PABX. You configure and acquire this route on the Contact Center Manager Administration server. The route allows access to the RAN.

In Orchestration Designer, you configure an output block to play RAN using the route configured for the RAN.

Many configuration items, such as name, description, assignments and log messages are optional. If you do not complete a required field, an error message appears when you close the Output block editor.

Procedure

1. Select the Output block icon (), right-click and click **Open**.

   OR

   Double-click the Output block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Prompts** tab.
4. Select **RAN**.
5. To silence the previous treatment, click **Silence Previous Treatment**.
6. In the **RAN Route** box, type the RAN Route for the recorded announcement.

   OR

Click and then click **Browse** to specify a recorded announcement from the application manager data.

7. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
8. Click the **Transition** tab to configure transitions from this block.
9. Save the changes to the flow application.

Variable definitions

Name	Description
Silence Previous Treatment	You can configure Silence Previous Treatment that gives silence after the RAN. The silence is required only if a previous block in your flow played music. To play music after the RAN, you must add it.
RAN Route	Route created on PABX, configured and acquired on CCMA for the RAN file.

Procedure job aid

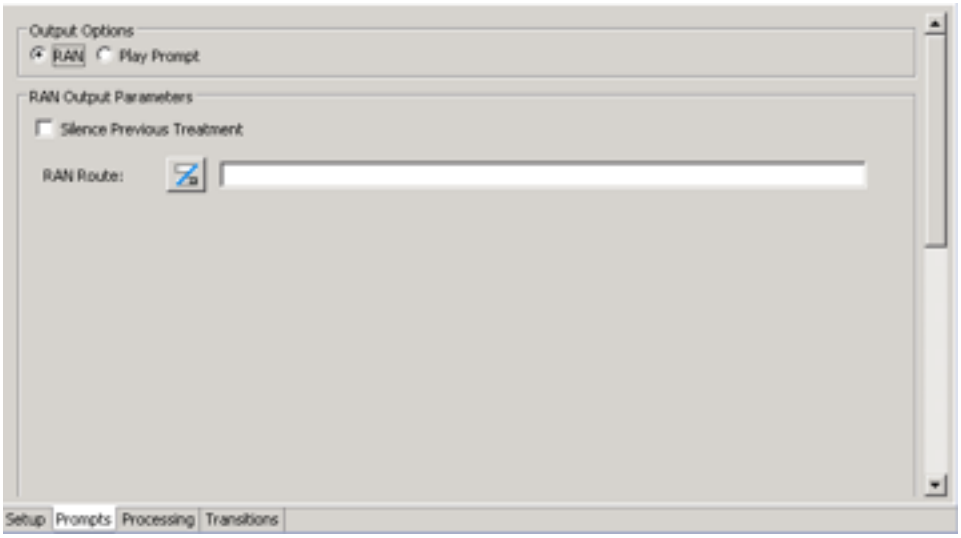


Figure 44: Output block for playing RAN

Creating a block to play prompts

Before you begin

- Add the Output block icon () to the flow application. See Adding blocks to the flow on page 144.

About this task

Create an Output block to provide an announcement to the caller or a controlled broadcast announcement.

While both an Input and an Output block can play prompts and responses to a caller, the Output block cannot accept input from a user.

In addition to the main configuration tabs, the output block also has the prompts tab. Use the prompts tab to configure the output prompt options and actions with previous call treatments that can occur before the Output block is encountered in the flow application.

Procedure

1. Select the Output block icon (), right-click and click **Open**.

   OR

   Double-click the Output block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Prompts** tab.
4. Select **Play Prompt**.
5. To play the prompt as a Broadcast Announcement, click **Play as a Broadcast Announcement**.
6. To choose a priority for the contact, select the **Priority** check box. In the **Priority** box, select the priority for the contact.
7. Select the **Continuous** box, if required.
8. In the **Access IVR DN** box, type the Access IVR DN for the announcement.

   OR

   Click  and then click **Browse** to specify the IVR DN from the application manager data.
9. To configure the prompt, under **Condition**, right-click, then choose **Add Conditional** and choose a condition.
10. If you choose a variable or condition, scroll down the **Prompts** tab to the **Selected Row Details** and complete the boxes for your prompt.

11. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
12. Click the **Transition** tab to configure transitions from this block.
13. Save the changes to the flow application.

Variable definitions

Name	Description
Access IVR DN	Select the Access IVR DN control to allow the user to select an IVR DN number to direct the call. Type a literal value, or select a variable from the application management data.
Condition Description	A description of the condition for playing the prompt or announcement.
Condition Name	The name assigned to the condition for playing the prompt or announcement.
Conditional Expression	An expression created to determine when to play the prompt or announcement.
Continuous	Choose to play the broadcast message from the beginning (Continuous box is cleared) or constantly (Continuous box is selected). You can only select continuous if you play a broadcast announcement.
Play as a Broadcast Announcement	Play a message through a voice port to all callers at the same time.
Priority	The rank of the contact where 1 is highest priority, meaning the contact is handled first and 4 is the lowest priority. You can only choose a priority if you play a broadcast announcement. Choose to select a priority or select a variable from the application management data.

Name	Description
Prompt list	Create a prompt list for an announcement and specify the time when the prompt plays: • Always—The prompt plays for every contact. • If <condition>—The prompt plays for every contact that meets the condition you configure. Type a name and a description for the condition and use the Condition builder to create the conditional expression.

Example: Playing multiple messages in languages

Assign a call variable to choose a language.



Figure 45: Output block Setup tab to configure languages

To add multiple voice prompts, right-click in the Prompts field and choose the appropriate selection from the menu.

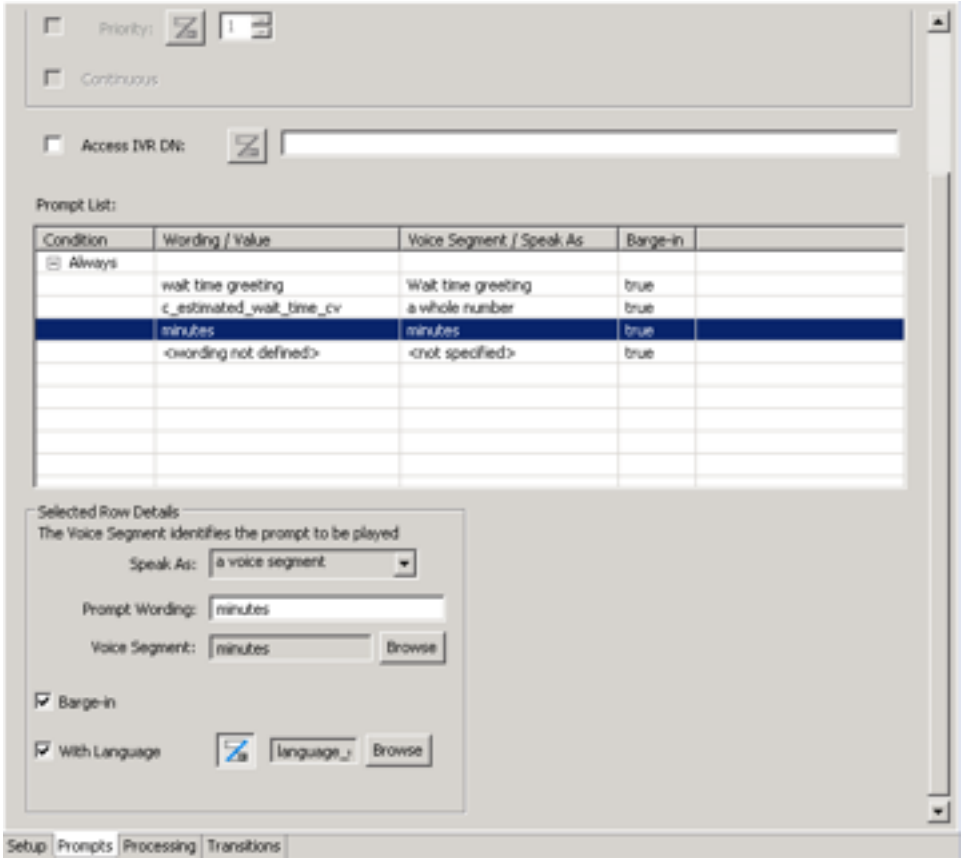


Figure 46: Output block Prompt for playing multiple prompts

Creating a host data send info block

Before you begin

- Add the host block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a host block to encapsulate Host Data Exchange commands without required detailed knowledge of the script syntax or application manager data.

The Send Info option enables the controls to identify the Provider ID integer type variables that reference the HDX application identifier that registers with Contact Center Manager Server. The Provider ID is entered using a literal text value or a variable from the application manager data. Maintain a list of send parameters by using the Add, Remove, Move Up and Move Down buttons.

In addition to the main configuration tabs, the host data block also has the host tab. Use the host tab to configure the data that is sent or received by the Host Data Exchange commands.

Procedure

1. Select the Host block icon (), right-click and click **Open**.

   OR

   Double-click the Host block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Host** tab.
4. Select **Send Info**.
5. Click  and then click **Browse** to specify the provider ID from the application manager data.
6. Under **Variables**, click **Add** to add the variables, intrinsics or application manager data to send to the Host.
7. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
8. Click the **Transition** tab to configure transitions from this block.
9. Save the changes to the flow application.

Variable definitions

Name	Description
Variables	Variables represent the data that is sent to the host. You can add up to 10 application variables or intrinsics.

Creating a host block to request information

Before you begin

- Add the Host block icon () to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a Host block to encapsulate Host Data Exchange commands without required detailed knowledge of the script syntax or application manager data.

The Request/Response uses the same Provider ID in the Send Info option to send a request or receive a response from the Contact Center Manager Server. The Provider ID is entered using a literal text value or a variable from the application manager data. You can define the request and response parameters and maintain these parameters using the Add, Remove, Move up and Move down controls.

In addition to the main configuration tabs, the host data block also has the host tab. Use the host tab to configure the data that is sent or received by the Host Data Exchange commands.

Procedure

1. Select the Host block icon (), right-click and click **Open**.

   OR

   Double-click the Host block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Host** tab.
4. Select **Request/ Response**.
5. Click  and then click **Browse** to specify the provider ID from the application manager data.
6. Beside **Request Parameters**, click **Add** to add the parameters to the list.
7. Beside **Response Parameters**, click **Add** to add the parameters to the list.
8. Select the **Timer** check box to specify a time limit.
9. In the **Timer** box, type the value in seconds for the timer.

   OR

   Click  and then click **Browse** to specify a variable to represent the timer from the variables, intrinsics, or application manager data.

10. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
11. Click the **Transition** tab to configure transitions from this block.
12. Save the changes to the flow application.

Variable definitions

Name	Description
Request	Request parameters represent the data that is sent to the host.
Response	Response parameters represent the data that is returned by the host.
Timer	Select to configure a time limit to wait for the response.

Example: Using a host block to send/receive data

Use the Host Data Exchange DIW capability to send the contact's DNIS value to a host and, based on the DNIS, retrieve the customer type for the contact. Some callers are categorized as Gold and others as Platinum customers. The type of caller is used to determine what priority to use when the contact is queued.

Use the Setup tab to make an explicit assignment of the call variable to initialize it to 0. After the host data exchange processes are complete, if your call variable still contains the null value, the host did not respond to the request.

Create the cv_response_string as a call variable to assign it a value in the variable definitions.

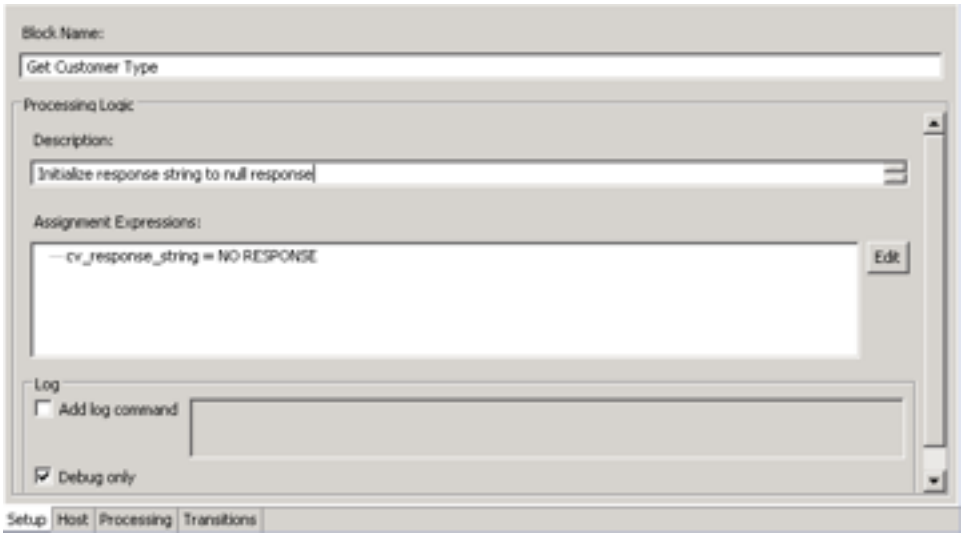


Figure 47: Setup tab showing initialization assignment string

In the Request Parameters field, include the store procedure that the host provider application uses for the DNIS lookup. The host provider also needs the DNIS value particular to the customer. The store procedure is an integer call variable.

In the Response Parameters field, include the correct stored procedure for the retrieval of the customer type, and the string used to store the customer type (GOLD or PLATINUM). The store procedure is an integer call variable. The cv_response_string is currently configured to NO RESPONSE (from the Setup tab).

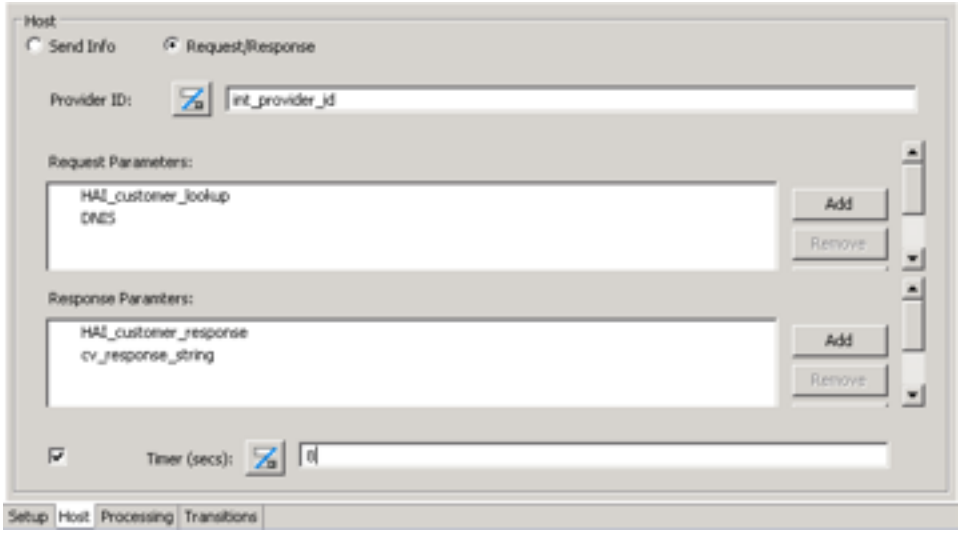


Figure 48: Host tab

Now use the processing tab to check the value in the customer type string to determine how to assign a priority. Gold customers are assigned a lower priority than the Platinum customers. If there is no response from the host application, assign the caller a lower priority than the other caller types.

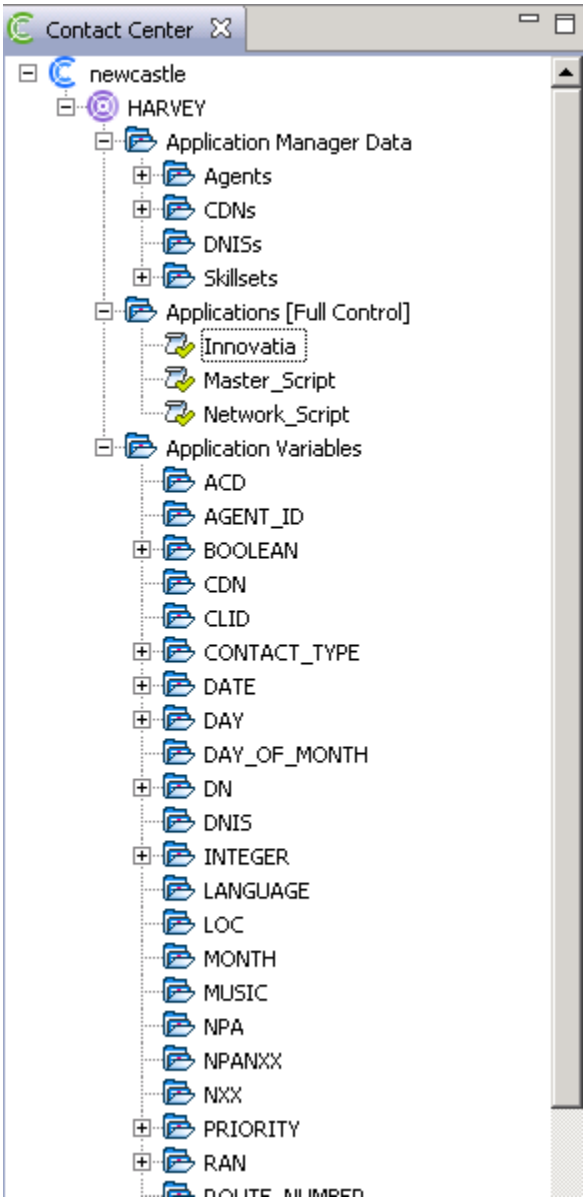


Figure 49: Processing tab

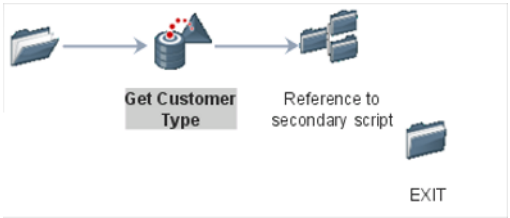


Figure 50: Sample flow

Creating a custom block

Before you begin

- Add the Custom block icon () to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a custom block to develop code where a programmer needs more control over what happens in the flow application. For example, to use a Send Info command within the context of playing a prompt and collecting data, you can use a custom block to develop the code. You must perform additional Event Handling within a custom block.

The Custom block opens a text-based editor that automatically formats and colors code as the script editor described in Script Editor operations on page 239.

A Custom block has two configuration tabs:

- Custom—Use the editor to create the automatically formatted text with indentations and colors to show formatted code.
- Transition—Configure the transitional condition for each connection from the Custom block to other blocks in the application flow after the Custom block, configure optional assignments or conditional assignments which are processed prior to a transition, and optionally configure a log message for each transition in the flow application.

Many configuration items, such as name, description, assignments, and log messages are optional. If you do not complete a required field, an error message appears when you close the Custom block editor.

Procedure

1. Select the Custom block icon (), right-click and click **Open**.

   OR

   Double-click the Custom block icon.
2. Click the **Custom** tab.
3. In the **Name** box, type a name for the Custom block.
4. In the **Description** box, type a description for the Custom block.
5. Click **TFE Editor**.
6. Type the portion of the script to add to the flow application.
   The code in the custom script is automatically validated. Errors appear in the Problems view.
7. Click the **Transition** tab to configure transitions from this block.

8. Save the changes to the flow application.

Procedure job aid

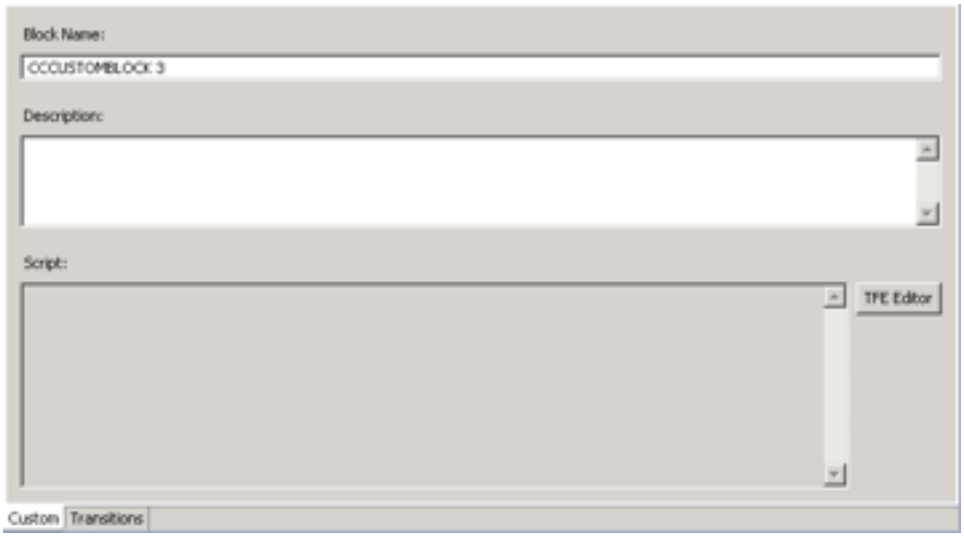


Figure 51: Custom block

Creating a reference block

Before you begin

- Add the Reference block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a Reference block to reference to another script or flow application in the Contact Center Manager Server. The treatment of the contact occurs in the referenced application and does not return to the current application.

When configuring the Reference block in the flow editor, you can double-click on the reference block to open the referenced application.

A Reference block has one configuration tab, the reference tab, where you configure the name and the referenced application.

Procedure

1. Select the Reference block icon ( ), right-click and click **Open**.

   OR

   Double-click the Reference block icon.
2. Click the **Reference** tab.
3. In the **Name** box, type a name for the Reference block.
4. In the **Script** box, select the script to reference. Click **Browse** to navigate through the list of applications.
5. Save the changes to the flow application.

Procedure job aid

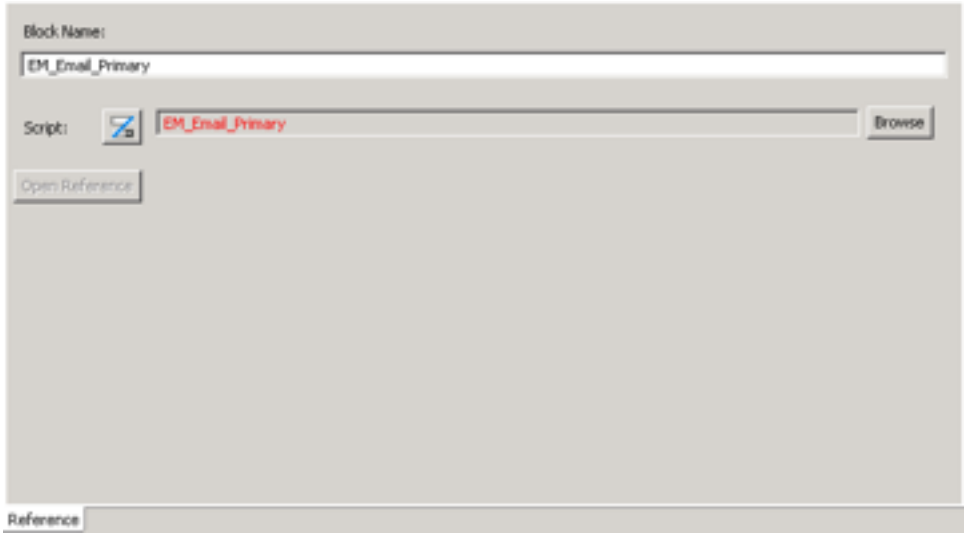


Figure 52: Reference block

Creating a block to play music

Before you begin

- Add the Treatment block icon () to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a Treatment block to play music to the caller while they wait on hold.

In addition to the main configuration tabs, the treatment block also has the treatment tab. Use the treatment tab to configure the treatment that the caller hears.

Procedure

1. Select the Treatment block icon (), right-click and click **Open**.

   OR

   Double-click the Treatment block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Treatment** tab.
4. Select **Music**.
5. If desired, select **Silence Previous Treatment**.
6. In the **Music Route** box, type the music route.

   OR

   Click  and then click **Browse** to specify a music route variable from the application manager data.
7. In the **Minimum Duration** box, type the value in seconds for the minimum duration timer.

   OR

   Click  and then click **Browse** to specify a variable to represent the timer from the variables, intrinsics, or application manager data.
8. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
9. Click the **Transition** tab to configure transitions from this block.
10. Save the changes to the flow application.

Variable definitions

Name	Description
Silence Previous Treatment	Add a silence command before the music by selecting the Silence Previous Treatment check box. If the Music block is the first block in the application, you cannot choose to silence the previous treatment because it causes contacts to default.

Procedure job aid

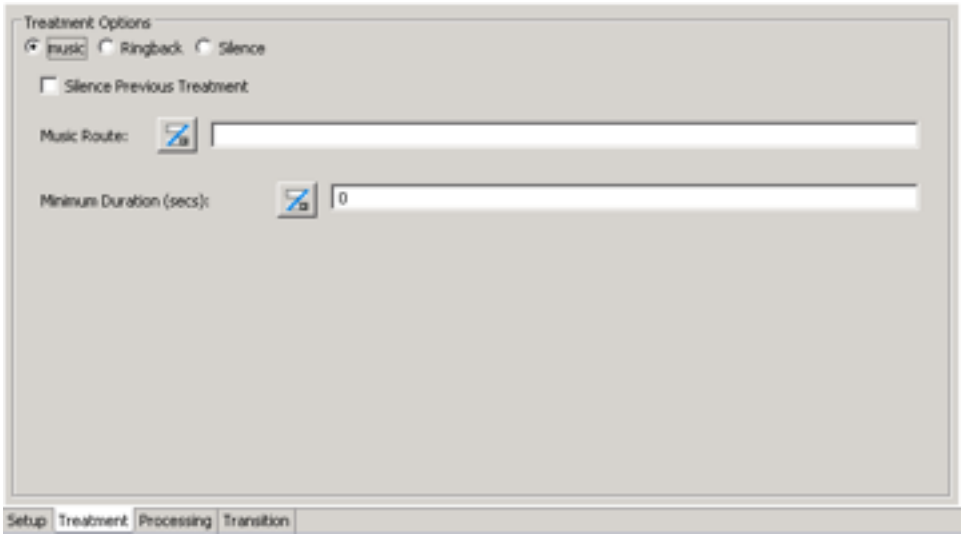


Figure 53: Treatment block to play music

Creating a ringback treatment

Before you begin

- Add the Treatment block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a ringback treatment where the caller hears a ring tone to ensure they connect to the contact center.

In addition to the main configuration tabs, the treatment block also has the treatment tab. Use the treatment tab to configure the treatment that the caller hears.

Procedure

1. Select the Treatment block icon (), right-click and click **Open**.

   OR

   Double-click the Treatment block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Treatment** tab.
4. Select **Ringback**.
5. If desired, select **Silence Previous Treatment**.
6. In the **Minimum Duration** box, type the value in seconds for the minimum duration timer.

   OR

   Click  and then click **Browse** to specify a variable to represent the timer from the variables, intrinsics, or application manager data.
7. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
8. Click the **Transition** tab to configure transitions from this block.
9. Save the changes to the flow application.

Variable definitions

Name	Description
Silence Previous Treatment	Add a silence command before the music by selecting the Silence Previous Treatment check box.

Procedure job aid

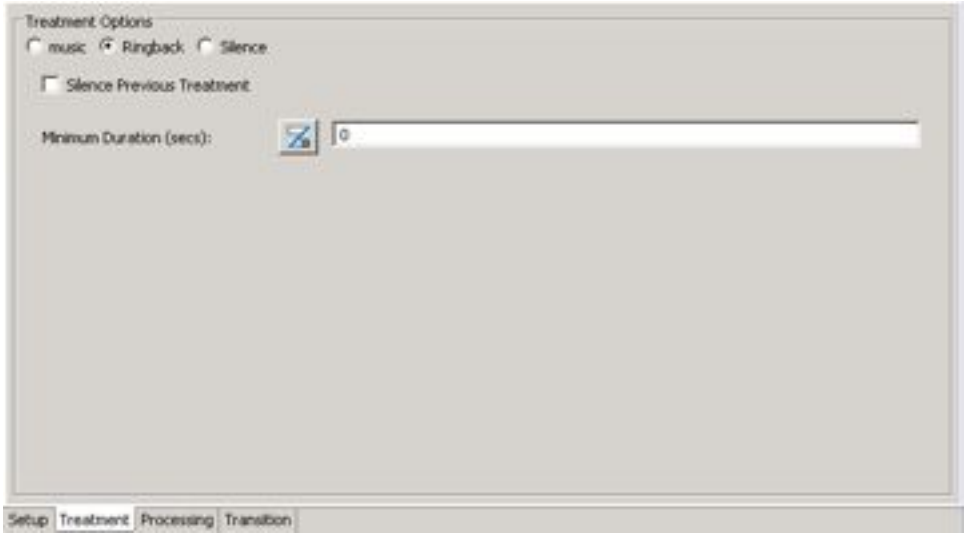


Figure 54: Treatment block to ringback

Creating a silent treatment

Before you begin

- Add the Treatment block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a silent treatment to provide a quiet moment for the caller. No music or ringback tone is provided.

In addition to the main configuration tabs, the treatment block also has the treatment tab. Use the treatment tab to configure the treatment that the caller hears.

Many configuration items, such as name, description, assignments, and log messages are optional. If you do not complete a required field, an error message appears when you close the Treatment block.

Procedure

1. Select the Treatment block icon ( ), right-click and click **Open**.

   OR

   Double-click the Treatment block icon.

2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Treatment** tab.
4. Select **Silence**.
5. In the **Minimum Duration** box, type the value in seconds for the minimum duration timer.

   OR

   Click and then click **Browse** to specify a variable to represent the timer from the variables, intrinsics, or application manager data.
6. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
7. Click the **Transition** tab to configure transitions from this block.
8. Save the changes to the flow application.

Procedure job aid

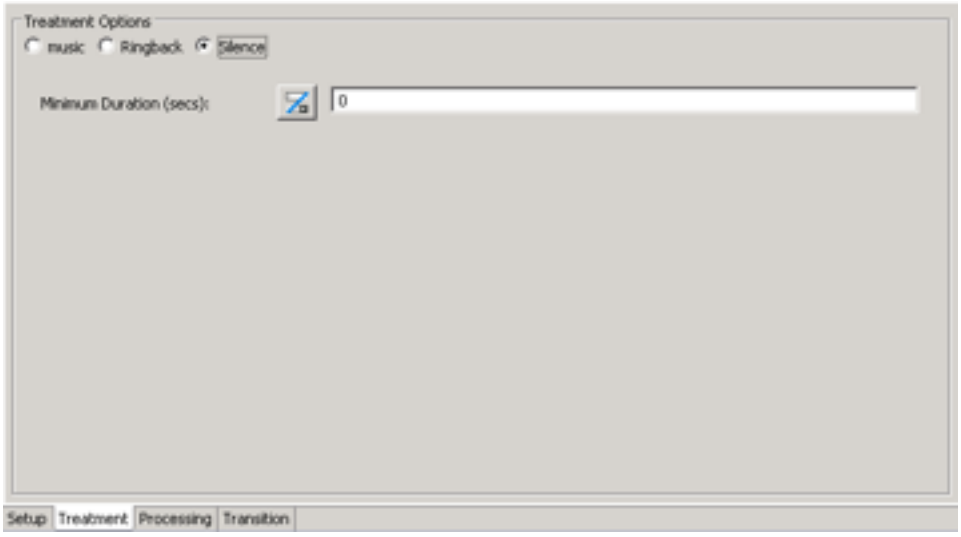


Figure 55: Treatment block to give silence

Creating a wait block

Before you begin

- Add the Wait block icon () to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a wait block to add a delay to the flow that is not already handled by blocks in the flow like the skillset queue or treatment.

A Wait block has two configuration tabs:

- Wait—Configure the name, and duration of the wait time.
- Transition—Configure the transitional condition for each connection from the Wait block to other blocks in the application flow after the Wait block, configure optional assignments or conditional assignments that are processed prior to a transition.

Procedure

1. Select the Wait block icon (), right-click and click **Open**.

   OR

   Double-click the Wait block icon.
2. Click the **Wait** tab.
3. In the **Name** box, type a name for the Wait block.
4. In the **Duration** box, select the duration of the time the caller is expected to wait.
5. Click the **Transition** tab to configure transitions from this block.
6. Save the changes to the flow application.

Variable definitions

Name	Description
Duration (secs)	The number of seconds to pause in the string of treatments presented to the caller. You can select a default wait time by choosing the minimum wait time in the Orchestration Designer preferences.

Procedure job aid

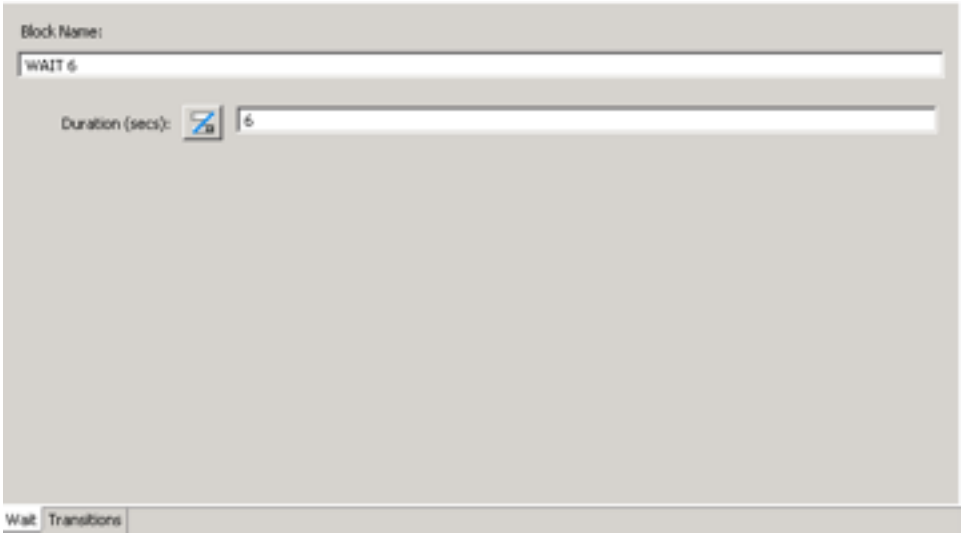


Figure 56: Wait block

Creating an IVR block

Before you begin

- Add the IVR block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create an IVR block to send a call to an Interactive Voice response (IVR) system and allows you to control the various IVR parameters depending on the switch.

On an Avaya Communication Server 1000 switch, you can configure the IVR dialed number, the treatment, the priority, and whether the IVR is interruptible.

In addition to the main configuration tabs, the IVR block also has the IVR tab. Use the IVR tab to configure the properties of the IVR application for your call.

Procedure

1. Select the IVR block icon ( ), right-click and click **Open**.

   OR

   Double-click the IVR block icon.
2. Click the **Setup** tab to configure the general details for the block.

3. Click the **IVR** tab.
4. Configure the properties for the IVR.
5. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
6. Click the **Transition** tab to configure transitions from this block.
7. Save the changes to the flow application.

Variable definitions

Name	Description
IVR properties for Avaya Communication Server 1000	Configure the properties for the IVR appropriate for an Avaya Communication Server 1000 switch: • IVR DN—The number dialed to place the call to the IVR. You can type the exact value from the application manager data, or select a variable. • Treatment—Specify the treatment the caller receives. • Priority—Set the priority by which the call is queued to the IVR where 1 is highest priority, meaning the contact is handled first and 4 is the lowest priority. • Interruptible—Select the check box to indicate the IVR session to be interrupted when an agent becomes available.

Procedure job aid



Figure 57: IVR block

Creating an IVR block to play prompts in the SIP configuration

Before you begin

- Add the IVR block icon ( ), to the flow application.

About this task

Create an IVR block to play a prompt in an Interactive Voice response (IVR) system with a SIP configuration. In addition to the main configuration tabs, the IVR block has the IVR tab. Use the IVR tab to configure the properties of the IVR application for your call.

Note:

For IVR blocks in SIP-enabled contact centers, you must use a Music or Ringback block after the IVR block to provide a default treatment. If you do not configure a default treatment after the IVR block, the customer hears silence.

Procedure

1. Select the IVR block icon ( ), right-click and click **Open**.

   OR

Double-click the IVR block icon.

2. Click the **Setup** tab to configure the general details for the block.
3. Click the **IVR** tab.
4. Select **Play Prompt**.
5. Beside **Prompt Name**, type a name or click and **Browse** to select a variable for the prompt.
6. To identify the URI of the Voice XML page, select the **Service URI** check box, and type a value or click and **Browse** to select a variable for the prompt.
7. To identify a variable for prompt substitutions, select the **Collected Digits** check box, type a value or click and **Browse** to select a variable for the prompt.
8. Select the **Agent Interruptible** check box to allow agents to interrupt the IVR.
9. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
10. Click the **Transition** tab to configure transitions from this block.
11. Save the changes to the flow application to save your changes in the IVR block.

Variable definitions

Name	Description
Prompt Name	Name of the recorded prompt that is played by the IVR.
Service URI	Identify the URI of the VXML page that the IVR calls using a text value or a variable from the application manager data.
Collected Digits	Identify the variable for the collected digits for prompts. You can choose to type a literal value or choose a variable from the application manager data.
Agent Interruptible	Select the check box to indicate the IVR session to be interrupted when an agent becomes available.
Session Based	Check when using a third party application other than Avaya Media Server for IVR treatments.

Procedure job aid



Figure 58: Play prompts (SIP only)

Creating an IVR block to play prompts and collect digits in the SIP configuration

Before you begin

- Add the IVR block icon ( ) to the flow application.

About this task

Create an IVR block to play a prompt and collect digits in an Interactive Voice response (IVR) system with a SIP configuration. In addition to the main configuration tabs, the IVR block has the IVR tab. Use the IVR tab to configure the properties of the IVR application for your call.

In Orchestration Designer applications, you can request the playback of numeric values either:

- as a full number, type '%n0' in the Prompt Name box in the IVR block
- as a sequence of digits, type '%i0' in the Prompt Name field in your IVR block

Note:

For IVR blocks in SIP-enabled contact centers, you must use a Music or Ringback block after the IVR block to provide a default treatment. If you do not configure a default treatment after the IVR block, the customer hears silence.

Procedure

1. Select the IVR block icon ( ), right-click and click **Open**.
   OR
   Double-click the IVR block icon.

2. Click the **Setup** tab to configure the general details for the block.
3. Click the **IVR** tab.
4. Select **Play and Collect**.
5. Beside **Prompt Name**, type a name or click and **Browse** to select a variable for the prompt.
6. In the **Return Value** box, type a value or click and **Browse** to select a variable to use for saving the collected digits.
7. To identify the URI of the Voice XML page, select the **Service URI** check box, and type a value or click and **Browse** to select a variable.
8. To allow the user to select the **Number of Digits**, select the check box and select the number of digits you expect the customer to type in response to the prompts. You can type a literal value for the number of digits, or click and **Browse** to choose an application variable.
9. To select a **Timeout (secs)**, select the check box and select the maximum number of seconds to wait for the customer before you play the Timeout message.
10. To select a **Terminating Character**, select the check box and select the Terminating character the customer enters when they are finished.
11. To be able to interrupt the IVR, select the **Interrupt IVR** check box.
12. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
13. Click the **Transition** tab to configure transitions from this block.
14. Save the changes to the flow application to save your changes in the IVR block.

Variable definitions

Name	Description
Interrupt IVR	Select the check box to specify if the prompt is interrupted when the caller types digits.
Number of Digits	The number of digits you expect the caller to enter. For example if your caller is entering in an account number, and your account numbers only have eight digits, your number of digits is 8.

Name	Description
Prompt Name	Name of the recorded prompt that the IVR plays.
Return Value	Store data returned from the IVR in a variable.
Service URI	Identify the URI of the VXML page that the IVR calls using a text value or a variable from the application manager data.
Session Based	Check when using a third pary application other than Avaya Media Server for IVR treatments.
Terminating character	The character the caller enters to identify that the last character of the number entered on their telephone keypad.
Timeout (secs)	The maximum delay the timer waits before playing the timeout prompt to the caller.

Procedure job aid

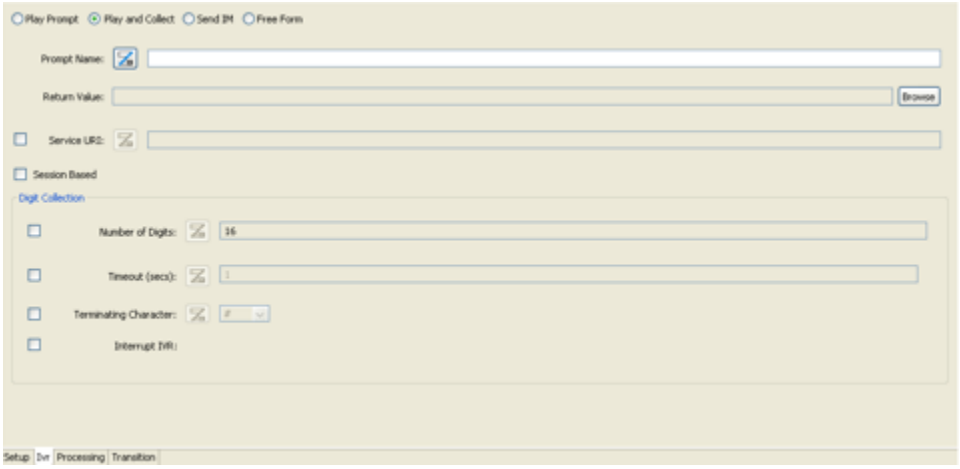


Figure 59: Play prompts and collect digits (SIP only)

Creating an IVR block to send instant messages in the SIP configuration

Before you begin

- Add the IVR block icon () to the flow application.

About this task

Create an IVR block to send instant messages in an Interactive Voice response (IVR) system with a SIP configuration.

In addition to the main configuration tabs, the IVR block has the IVR tab. Use the IVR tab to configure the properties of the IVR application for your contact.

Procedure

1. Select the IVR block icon (), right-click and click **Open**.

   OR

   Double-click the IVR block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **IVR** tab.
4. Select **Send IM**.
5. Click **Add** to add your message content in the **Message Content** box.
6. To identify the URI of the Voice XML page, select the **Service URI** check box, and type a value or click  and **Browse** to select a variable.
7. In the **Return Value** box, and type a value or click  and **Browse** to select a variable.
8. To specify a time limit on the message response, select the **Timeout** check box, and specify a maximum time to wait.
9. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
10. Click the **Transition** tab to configure transitions from this block.
11. Save the changes to the flow application to save your changes in the IVR block.

Variable definitions

Name	Description
Message Content	The content of the message. The content cannot exceed 16 message segments or 500 characters. Use the Add, Remove, Move up and Move down buttons to manage the list of message segments in the message content.
Return Value	Store data returned from the IVR in a variable.
Service URI	Identify the URI of the VXML page that the IVR calls using a text value or a variable from the application manager data.
Timeout	Identify the maximum time for waiting for an instant message. The parameter is inter digit timeout.
Content Type	The type of IM treatment. Values are text/rtf, text/plain, or text/html, depending on the release of OCS you use.

Procedure job aid

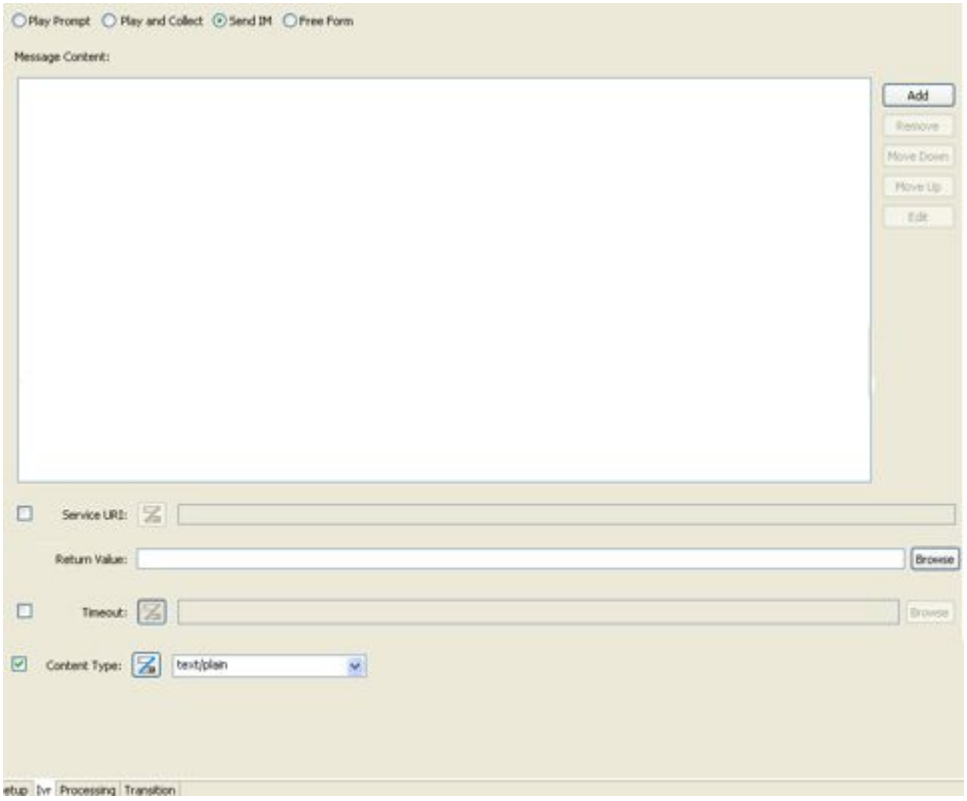


Figure 60: Send IMs (SIP only)

Creating an IVR block for free form SIP configuration

Before you begin

- Add the IVR block icon ( ) to the flow application.

About this task

Create an IVR block with custom options in an Interactive Voice response (IVR) system with a SIP configuration.

In addition to the main configuration tabs, the IVR block also has the IVR tab. Use the IVR tab to configure the properties of the IVR application for your contact.

Note:

For IVR blocks in SIP-enabled contact centers, you must use a Music or Ringback block after the IVR block to provide a default treatment. If you do not configure a default treatment after the IVR block, the customer hears silence.

Procedure

1. Select the IVR block icon (), right-click and click **Open**.

   OR

   Double-click the IVR block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **IVR** tab.
4. Select **Free Form**.
5. Beside **Service URI**, type a value or click  and **Browse** to select a variable.
6. Select the **Interruptible** check box for the IVR session to be interrupted.
7. Under **Input Parameters**, click **Add** to add variable data to pass to the IVR.
8. Under **Return Parameters**, click **Add** to add variable data to receive from the IVR.
9. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
10. Click the **Transition** tab to configure transitions from this block.
11. Save the changes to the flow application to save your changes in the IVR block.

Variable definitions

Name	Description
Service URI	Identify the URI of the VXML page that the IVR calls using a text value or a variable from the application manager data.
Input Parameters	Variable data that is passed to the IVR.
Return Parameters	Variable data received from the IVR.
Interruptible	Select the check box to allow interruptions in the IVR session when an agent becomes available.

Procedure job aid

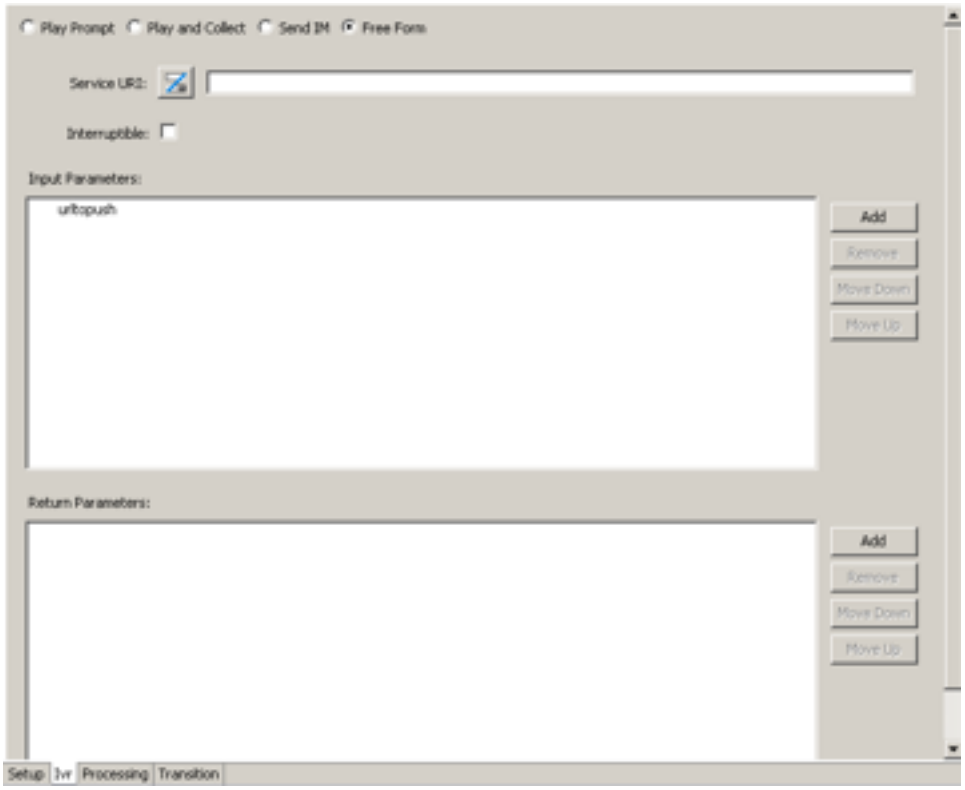


Figure 61: GIVE IVR Free Form (SIP only)

Creating a menu block

Before you begin

- Add the Menu block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a Menu block presents the caller with a menu choice that directs the call to the next block using Dual Tone Multi Frequency (DTMF) input. The DTMF corresponds to a transition taken when the value is collected.

In addition to the main configuration tabs, the menu block also has the following two tabs:

- Prompts—Configure the maximum number of retries the customer can try to enter the menu item, and the prompts for the menu items.
- Menu Configuration—Configure the list of menu choices and the actions that occur for each menu choice provided to the caller.

Procedure

1. Select the Menu block icon (), right-click and click **Open**.

   OR

   Double-click the Menu block icon.
2. Click the **Setup** tab to configure the general details for the block.
3. Click the **Prompts** tab.
4. Beside **Max Retries**, select the maximum number of retries for the user input. If the Max Retries is greater than 0, the Retry and Timeout prompts display in the Prompt List.
5. To configure the Preamble prompt, right-click **Preamble**, then choose **Add Conditional** and choose a condition.
6. If you choose a variable or condition, scroll down the **Prompts** tab to the **Selected Row Details** and complete the boxes for your prompt.
7. To configure the Initial prompt, right-click **Initial**, then choose **Add Conditional** and choose a condition.
8. If you choose a variable or condition, scroll down the **Prompts** tab to the **Selected Row Details** and complete the boxes for your prompt.
9. To configure the Retry prompt, right-click **Retry**, then choose **Add Conditional** and choose a condition.
10. If you choose a variable or condition, scroll down the **Prompts** tab to the **Selected Row Details** and complete the boxes for your prompt.
11. If you choose a variable or condition, scroll down the **Prompts** tab to the **Selected Row Details** and complete the boxes for your prompt.
12. Click the **Menu Configuration** tab.
13. Beside **Digit Entry Timeout (secs)**, select the maximum time to wait between digits.
14. In the **DTMF Input** box, select the DTMF value that the customer selects.
15. For each DTMF option, define the corresponding choice.
16. Add the DTMF option and the choice to the DTMF Input list.
17. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.

18. Click the **Transition** tab to configure transitions from this block.
19. Save the changes to the flow application.

Variable definitions

Name	Description
Barge-in	Select this check box to allow the caller to interrupt the prompt. Clear the check box to prevent the caller from interrupting the prompt or data collection.
Choice	The information that supports the choice of DTMF digit.
Condition	The specific time when the prompt is played: • Always—The prompt plays for every contact. • If <condition>—The prompt plays for every contact that meets the condition you configure. Type a name and a description for the condition and use the Condition builder to create the conditional expression.
Digit Entry Timeout (secs)	The length of time to wait to ensure the customer enters all information before processing the current collected digit.
DTMF Input	Dual-tone multi-frequency (DTMF) input to make choices on a telephone to select menu options. You can select a digit 0-9 once for each menu.
Max Retries	The maximum number of retries for accepting user input. You can specify a number, or choose a variable that represents the number of retries to configure.
Prompt Type	There are four types of prompts: • The preamble prompt plays once to the customer when the contact enters this block in the flow application. Typically, the

Name	Description
	preamble prompt is used for introductory comments. • The initial prompt is the main collection prompt that plays. The initial prompt plays to the customer once. • The retry prompt Retry1, Retry2, are played if the collected input does not match the expected results. • The timeout prompts play if there is no response from the caller. The prompt can be recorded announcement, or data gathered during the current input block.
Voice Segment/Speak As	The name of the VOICE_SEGMENT variable file on the Contact Center Manager Server used for the prompt.
Wording/Value	Voice segment can be one of the following: • Voice Segment—A prerecorded statement on the Contact Center Manager Server. • Individual digits—Echo the digits one at a time. • Whole number—Echo the digits gathered as a whole number. Use the Voice Segment box to navigate to the VOICE_SEGMENT variable file on the Contact Center Manager Server.

Example: Playing messages in different languages after prompted

Play a menu to prompt callers to choose their language to hear information (RAN) in their language.

Use the Setup tab to configure the name and description for the menu block.

Use the Prompts tab to configure a multi-lingual welcome message.

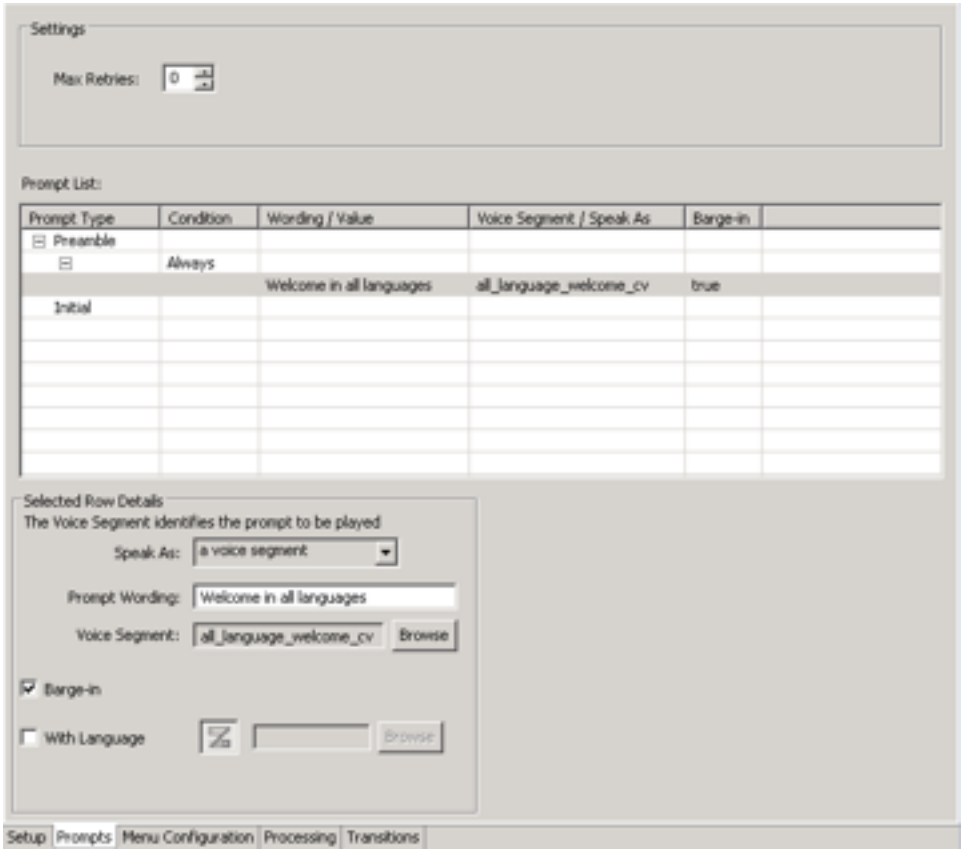


Figure 62: Menu block Prompt tab

Use the Menu Configuration tab to configure the DTMF values for each language.

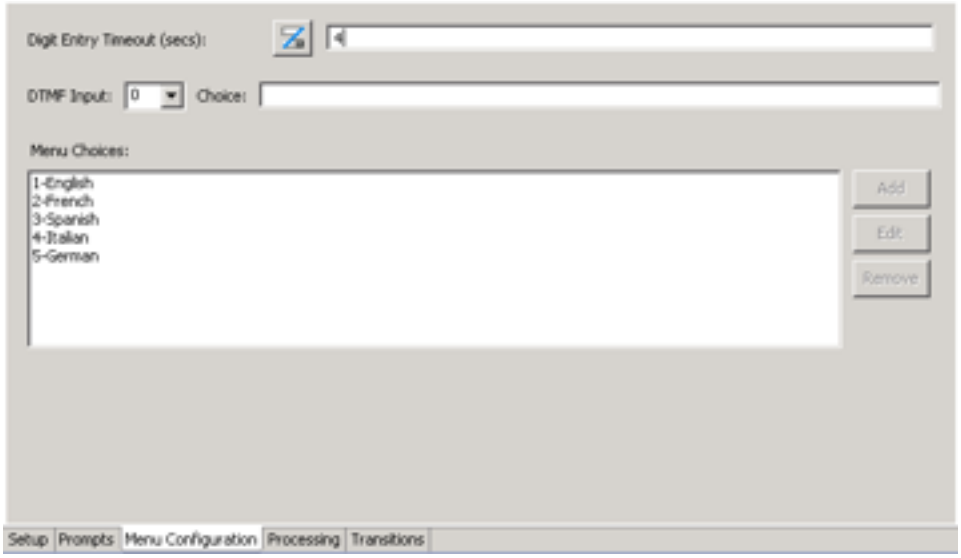


Figure 63: Menu block Menu Configuration tab

Orchestration Designer creates transitions based on the menu prompts you configure on the Menu Configuration tab.

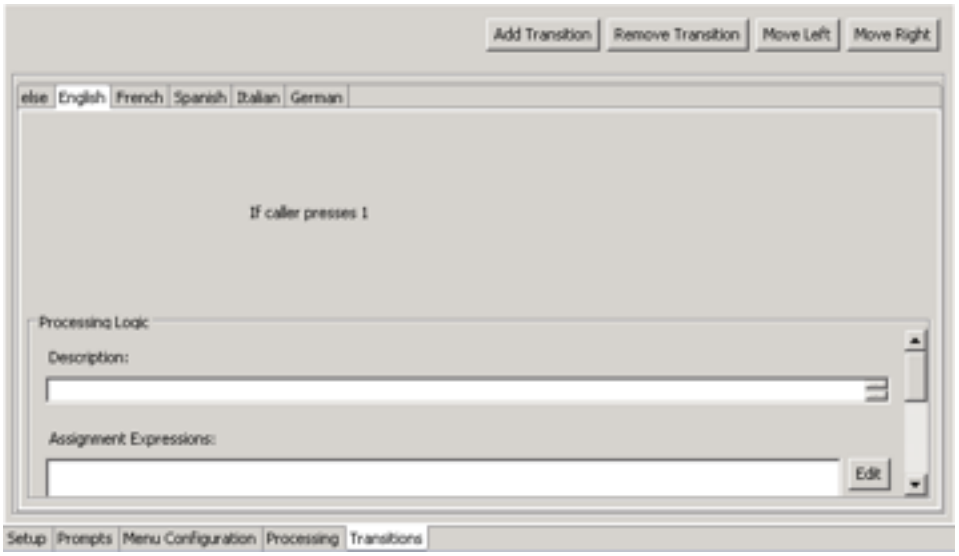


Figure 64: Menu block Transitions tab

Add the remainder of the blocks to the flow diagram, with a different path for each language choice.

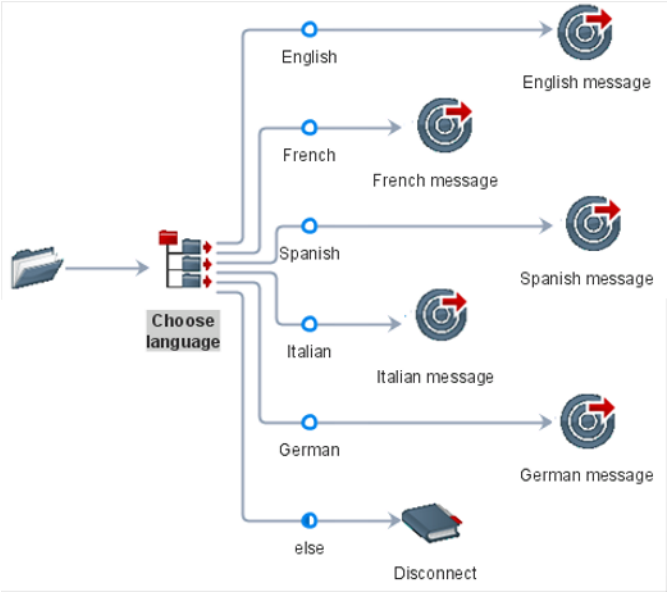


Figure 65: Menu block flow

Creating a disconnect block

Before you begin

- Add the Finish block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a disconnect block to finish routing a contact in the flow diagram and provide final treatment for the contact.

A Finish block has one configuration tab.

Procedure

1. Select the Finish block icon ( ), right-click and click **Open**.

   OR

   Double-click the Finish block icon.
2. In the **Name** box, type a name for the Finish block.
3. Select **Disconnect**.
4. Save the changes to the flow application.

Procedure job aid

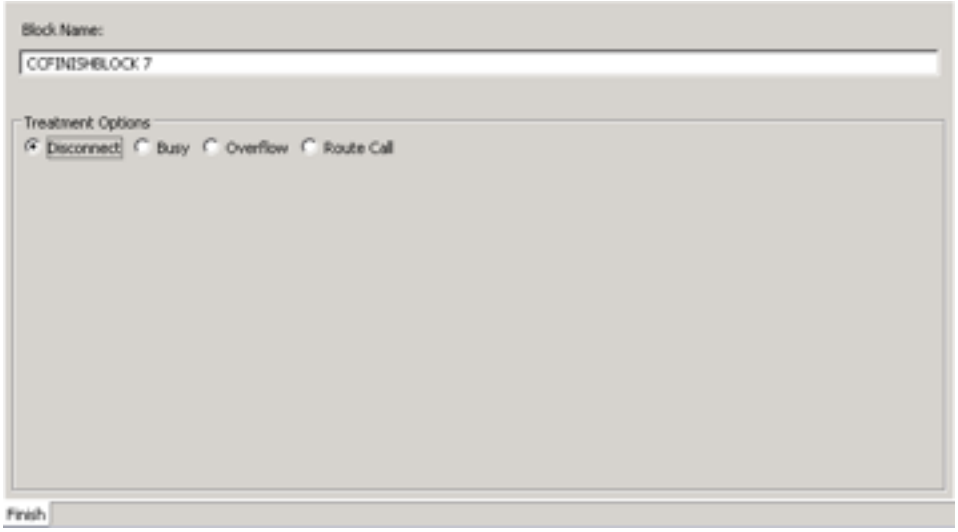


Figure 66: Finish block to disconnect

Creating a busy block

Before you begin

- Add the Finish block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a busy block that plays a busy tone to a call before the switch disconnects it.

Procedure

1. Select the Finish block icon ( ), right-click and click **Open**.
   OR
   Double-click the Finish block icon.
2. In the **Name** box, type a name for the Finish block.
3. Select **Busy**.
4. Save the changes to the flow application.

Important:

There is a Controlled option for Busy for Avaya Communication Server 1000. If you use the CONTROLLED keyword, the contact remains in the control of

Contact Center Manager, which means that if an error condition occurs on the switch (cannot give the tone to the contact, for example), the contact is returned to the queue so that the system treats it. After a command without the CONTROLLED keyword is complete, Contact Center Manager has no control over the contact.

Procedure job aid

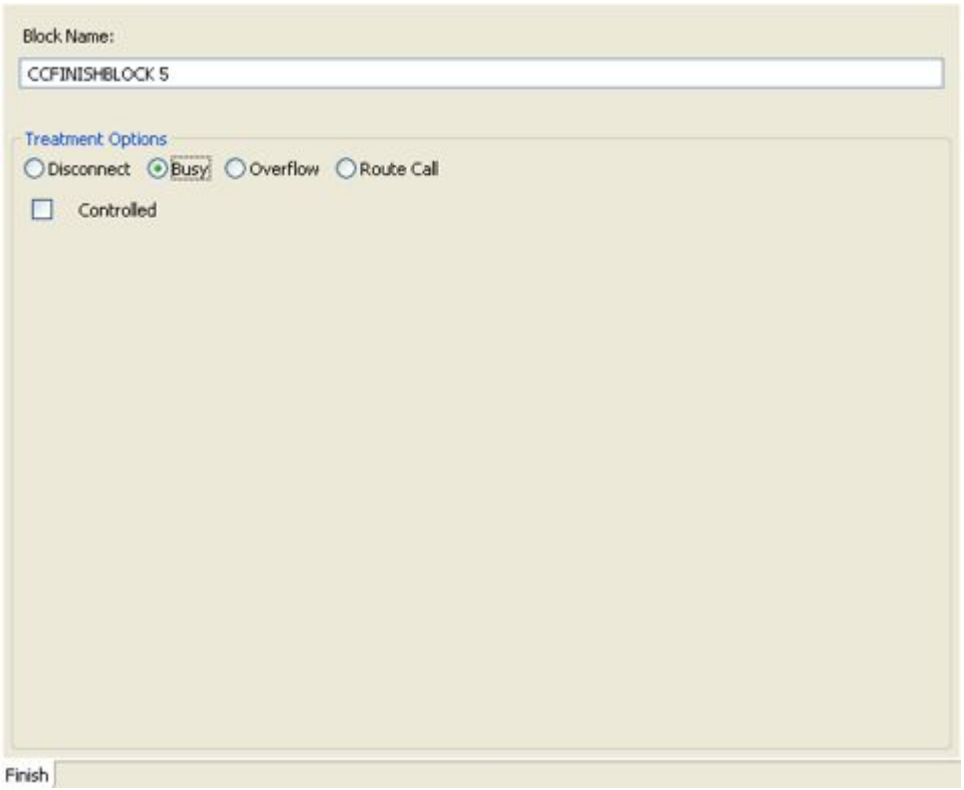


Figure 67: Finish block to define busy

Creating an overflow block

Before you begin

- Add the Finish block icon () to the flow application. See Adding blocks to the flow on page 144.

About this task

Create an overflow block that plays an overflow tone to the customer before the switch disconnects it.

Procedure

1. Select the Finish block icon (), right-click and click **Open**.

   OR

   Double-click the Finish block icon.
2. In the **Name** box, type a name for the Finish block.
3. Select **Overflow**.
4. Select the **Controlled** check box to maintain control of the contact in Contact Center Manager Server. If an error occurs on the switch, Contact Center can return to the queue.

   Avaya recommends that you select the controlled option.
5. Save the changes to the flow application.

Procedure job aid

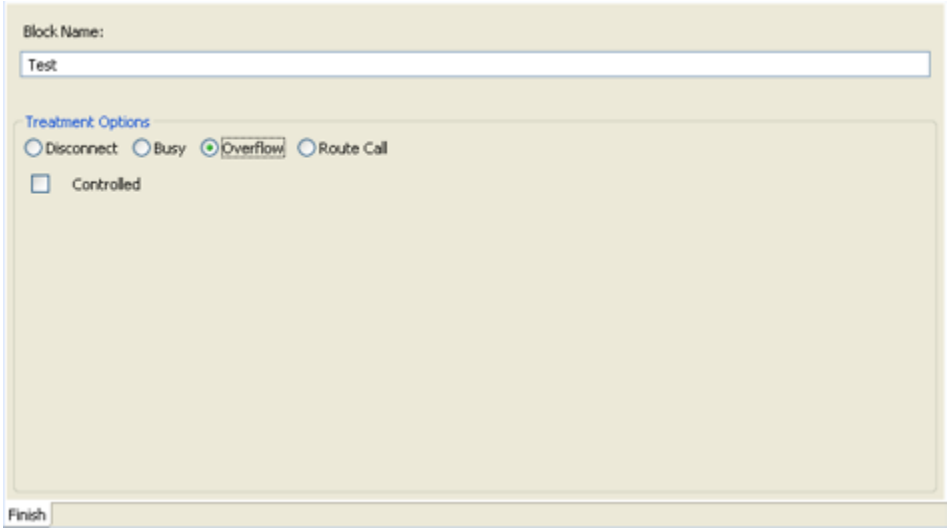


Figure 68: Finish block to overflow

Creating a route contact block

Before you begin

- Add the Finish block icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Create a route call block that routes a call to a particular dialable number (DN) such as an ACD or a personal number. You can configure a specific number or a variable from the application manager data.

Procedure

1. Select the Finish block icon ( ), right-click and click **Open**.

   OR

   Double-click the Finish block icon.
2. In the **Name** box, type a name for the Finish block.
3. Select **Route Call**.
4. Select the Default DN.

OR

In the **DN** box, type a value for the DN or click and then click **Browse** to specify a DN from the application manager data.

OR

In the **URI** box, type the value for the URI or click and then click **Browse** to specify a URI from the application manager data.

5. Save the changes to the flow application.

Variable definitions

Name	Description
Default DN	Choose the default DN for the CDN.
DN	Enter or browse to a DN variable.
URI	Enter or browse to a URI variable.

Procedure job aid

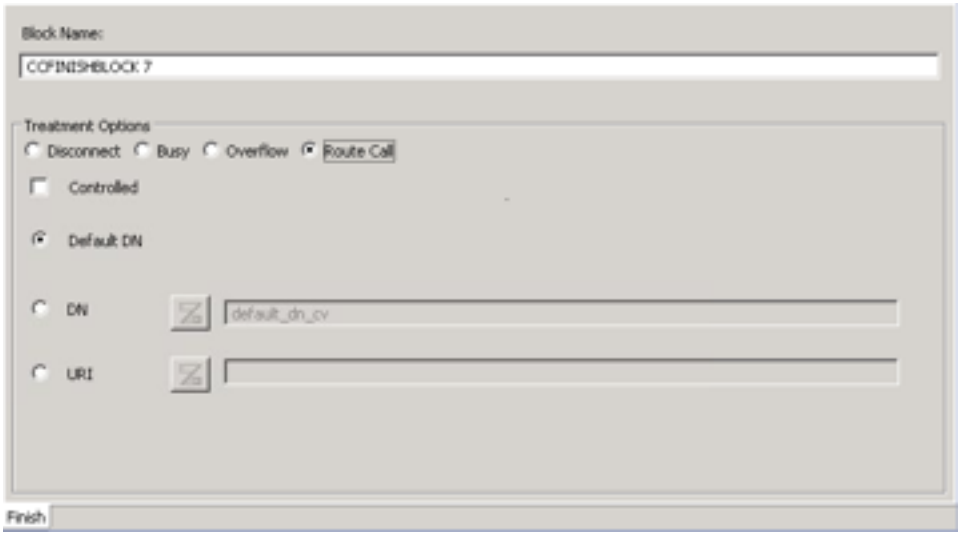


Figure 69: Finish block to route call

Using an anchor node

Before you begin

- Add the Anchor node icon ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Use an anchor node in your flow application to link a block that is outside of your main flow application path.

Procedure

1. Select the Anchor icon ( ).
2. Select the node or block before the Anchor icon, then select the Anchor icon, right-click and click **(Dis)Connect**.
3. Select the Anchor icon, then select the node or block after the Anchor icon, right-click and click **(Dis)Connect**.
4. Save the changes to the flow application.

Example: Using an anchor node

Use an anchor node to tie blocks that apply to a group of blocks in a flow application to your flow. Usually, custom blocks with code for Event Handlers are used in applications and you must include them in the flow so that validation occurs.

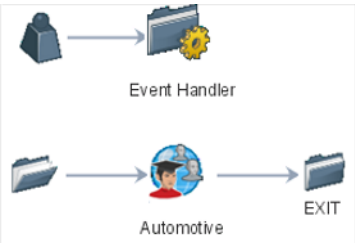


Figure 70: Anchor node

Using an anonymous node

Before you begin

- Add the Anonymous node icon () to the flow application. See Adding blocks to the flow on page 144.

About this task

Use an anonymous node in your flow application to bridge a source object and a destination object without causing links to cross in your flow application.

You can also use the anonymous node to control how the links appear between blocks appear in your flow diagram.

Procedure

1. Select the Anonymous icon ().
2. Select the node or block before the Anonymous icon, then select the Anonymous icon, right-click and click **(Dis)Connect**.
3. Select the Anonymous icon, then select the node or block after the Anonymous icon, right-click and click **(Dis)Connect**.
4. Save the changes to the flow application.

Example: Using an anonymous node

Use the anonymous node to format the blocks, nodes, and transitions so that you can read all of the text and see the transitions.

Use the anonymous node with the incremental view to adjust the position of nodes and blocks to fit in your flow diagram.

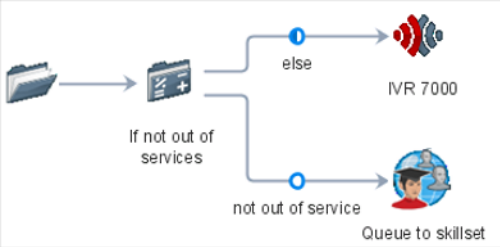


Figure 71: Before anonymous node

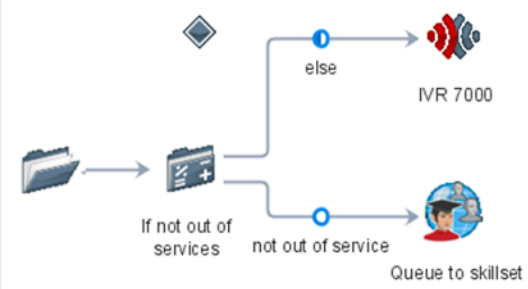


Figure 72: After anonymous node

Configuring a source shortcut node

Before you begin

- Add the source shortcut node ( ) to the flow application. See Adding blocks to the flow on page 144.

About this task

Configure a source shortcut node in the Flow Editor to point to another item in a flow application to minimize congestion and line crossings, repeat processes, or connect blocks on various pages in the flow application.

Procedure

1. Select the node or block before the Source Shortcut node ( ), then select the Source Shortcut node icon ( ), right-click and click **(Dis)Connect**.
2. Select the Source Shortcut node icon ( ), then select the Destination shortcut node icon ( ), right-click and click **(Dis)Connect**.
3. Save the changes to the flow application.

Configuring a destination shortcut node

Before you begin

- Add the destination shortcut node () to the flow application. See Adding blocks to the flow on page 144.

About this task

Configure a destination shortcut node in the Flow Editor to point to another item in a flow application to minimize congestion and line crossings in the flow application. You can also use shortcut nodes to connect blocks on different pages.

Procedure

1. Select the Destination Shortcut node icon (), then select the node or block that follows the Destination Shortcut, right-click and click **(Dis)Connect**.
2. Save the changes to the flow application.

Creating an exit node

About this task

An Exit block indicates that a contact flow ends. No other contact processing occurs after the end. Use it at the end of every flow application you create to handle contacts.

The exit node, by default, is available in every flow application you create.

Procedure

1. Select the last block of your path, then right-click the Exit block icon (), and click **(Dis)Connect**.
2. Save the changes to the flow application.

Chapter 16: Script Editor operations

When using an existing script, or creating a new script application, you can use the text editor to type commands, insert variables and intrinsics, and use application manager data to develop the content.

This chapter describes the processes by using the text editor in the Script Editor view.

Prerequisites to script editor operations

- Create a new script application or open an existing script.
- Review the script command syntax for commands you want to use in your script application.

Adding commands to script applications

Before you begin

- Use the command syntax reference section to determine the syntax and properties of each command.

About this task

Add commands to your script in the active script editor to perform the steps you require for your script application.

There are a number of shortcuts you can use to add commands and variables to your script.

The master script can contain a maximum of 100000 characters. In all other scripts, the maximum number of characters is 50000.

The script, when saved, shows errors in the Problem view.

Procedure

1. In the Script editor, type the commands that you want to use.
2. Type the start of the command, press **CTRL + SPACEBAR** to display a window from which you can select text to automatically complete the command.
3. Drag and drop application manager data from the Contact Center view or Local view to the Script editor.

4. Save the script.
   If the script is longer than the maximum number of characters, a warning message appears and the script is not saved.

Displaying or hiding line numbers

About this task

Display the line numbers to help you track the line of the script application you work with. Hide the line numbers if they distract you while you create a script application.

Procedure

1. Click **Windows** > **Preferences**.
2. Click **Contact Center**.
3. Click **Default Options**.
4. If you want to display line number, select the **Show Line Numbers** check box.
5. If you want to hide line numbers, clear the **Show Line Numbers** check box.

Going to a particular line number

About this task

Go to a particular line number in your script application if you want to make changes to a specific line or to resolve an error.

Procedure

1. Press **CTRL** + **L**.
2. In the **Go To Line** box, type the line to which you want to go.
   The cursor jumps to the line number.
3. If the line number you type is outside of the script application line range, the Go To Line box does not enable the Ok button. Type another line number.

Undoing or redoing the last change

About this task

Undo the last change you made to your script or flow application.

For example, if your change was a deletion, the undo command adds the content back. If your change was an addition, the undo command removes the content.

You can also redo the last change you made to your script or flow application.

Procedure

1. Press **CTRL** + **Z**.
2. Click **Edit** > **Undo/Redo** to review the sequence of changes.

Chapter 17: Condition and Assignment builder operations

Use the Condition and Assignment builder operation to compose conditional expressions and assignments in the blocks of your flow application. This chapter introduces you to the Condition and Assignment builder.

Prerequisites for condition and assignment builder operations

- Open the Flow Editor. See Creating a new application on page 110 or Opening an existing application on page 115.

Creating conditional and assignment expressions

Before you begin

- Create a block in your flow application. See Block creation on page 171.
- Ensure that you create a variable in the Contact Center or Local view on the Contact Center Manager Server on which you work.

About this task

Create conditional and assignment expressions in your blocks to assign specific values to a variable. Define the variable names in the Contact Center Manager Administration application or in Orchestration Designer so that variables are available in the Contact Center view of Orchestration Designer.

If you are working with applications in the Local view, you can create variables and placeholders for agent ids and copy them to the Contact Center view.

You can assign specific values using conditions based on the paths in your flow application that your contact follows. For example, if your call is received from the DNIS 2334, you can assign the Priority variable 3. The corresponding assignment Priority = 3 is placed in a block of your flow application that the call passes through to update the Priority variable.

Procedure

1. On the **Setup**, **Processing**, or **Transition** tab of the block, beside **Assignment Expressions**, click **Edit**.
2. Beside **Statements**, click **Add**.
3. Choose the Statement type that you want to add.
4. Use the controls to create your conditional or assignment expression.
5. Beside **When**, click **Add** to add the expression to the **Statements** box.
6. Click **Apply**.

Variable definitions

Name	Description
Statement type	The type of statement you want to add to your block: • Add Locked Variable—replace the existing variable with the new definition for other commands and scripts. • Add If—add an if statement to the assignment or condition expression. • Add Locked If—add the assignment to the application manager data for use in other commands. • Add Else—add a statement to apply an assignment for when the if statement is false. • Add Assignment—add the assignment for all situations. • Add Locked Assignment—add the assignment to the application manager data for use in other commands in all situations.
Controls	Use variables, application manager data, intrinsics and the following operators to configure your expressions. • Numbers • Mathematical operators: 

Name	Description
	• Relational operators:  • Application manager data 

Procedure job aid

See Procedure job aid on page 246 for information about the Condition and Assignment Builder and the statements for building expressions.

Adding assignments and expressions

Before you begin

- Create a logic statement. See Creating conditional and assignment expressions on page 243.

About this task

Add assignments and expressions to blocks in your flow application using the Condition and Assignment builder. You can use the palette to enter object such as predefined variables, user-defined variables, and intrinsics, assign the objects to a specific number, and add logic to the expressions.

Procedure

1. Beside **Assignment Expressions** on a block editor tab, click **Edit**.
2. Beside **Statements**, click **Add**.
3. Select **Add If** or **Add Locked If**.
4. Use the mathematical operations, order of operations, logical operations, and relational operations to create your expression.

Procedure job aid

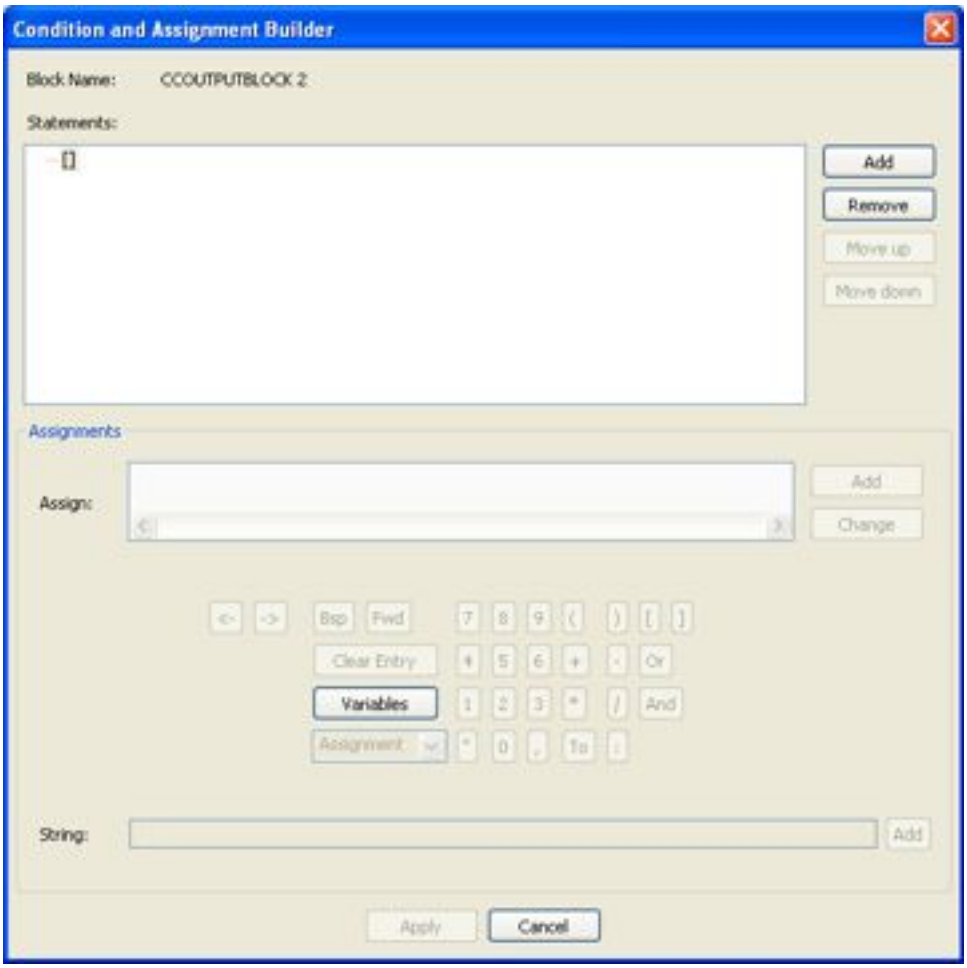


Figure 73: Condition and Assignment builder palette

Table 19: Mathematical operations

Symbol	Description
Addition (+)	Add two values of the same type. This expression can use numerical constants, variables, and expressions that return the data types Integer and Seconds.
Subtraction (–)	Subtract the second value from the first value of the same type and use numerical constants, variables, and expressions that return the data types Integer and Seconds.
Multiplication (*)	Multiply two values of the same type. This expression can use numerical constants, variables, and expressions that return the data types Integer and Seconds.

Symbol	Description
Division (/)	Divides the first value by the second value of the same type. This expression contains numerical constants, variables, and expressions that return the data types Integer and Seconds. If the result of the division is not an integer, the value is truncated, not rounded. For example, 10.7 becomes 10.

Table 20: Order of operations

When you evaluate conditional expressions, the expression with the highest precedence (importance) is evaluated first, then the one with the second highest precedence, and through to the expression with the lowest precedence.	Operations in conditional expressions are evaluated in the following order: • parentheses ( ) • multiplication * and division / • addition + and subtraction – • comparison expressions: Equal To (==), Not Equal To (!=), Greater Than (>), Greater Than Or Equal (>=), Less Than (<), Less Than Or Equal (<=) • logical expressions Not (!), And (&&), Or (||)
Notes: • When expressions appear more than once, or when two expressions with equal importance appear in the same expression, they are evaluated from left to right. • The expression NOT has precedence over the AND and OR expressions. • If an expression contains parentheses, the partial expression within the parentheses is resolved first. Perform multiplications, followed by additions, subtractions, comparison expressions, logical expressions, from left to right. Examples:	
2 * 3 + 2 * 3 = 12 2 * (3 + 2 * 3) = 18	(2 * 3 + 2) * 3 = 24 (2 * 3) + (2 * 3) = 12

Table 21: Logic operations

Symbol	Description

AND	Evaluate two or more comparative expressions and return a True value to the statement if the two expressions are both True, and a False value if either expression is False.
	Example: A customer service department is open Monday to Friday, from 08:00 a.m. to 06:00 p.m. After 06:00 p.m., and on Saturday and Sunday, the department is closed.
	`IF (TIME OF DAY = business_hours_gv)`
	`&& DAY OF WEEK = weekdays_gv) THEN`
	`  GIVE RAN open_ran_gv`
	`ELSE`
	`  GIVE RAN closed_ran_gv`

<table>
<tr><th>Symbol</th><th>Description</th></tr>
<tr><td></td><td>

```
   DISCONNECT
END IF
```

Expression	Is it a weekday?	Is it between 08:00 a.m. and 06:00 p.m.?	Give open service?
AND	yes	yes	yes
AND	yes	no	no
AND	no	no	no
AND	no	yes	no

</td></tr>
<tr><td>OR</td><td>

Evaluate two or more comparative expressions and return a True value to the statement if either, or both, of the two expressions are True, and a False value if both expressions are False.

Example: A customer service department is open Monday to Friday, from 08:00 a.m. to 06:00 p.m. After 06:00 p.m., and on Saturday and Sunday, the department is closed.

```
IF (TIME OF DAY > 18:00) || (DAY OF WEEK = SATURDAY, SUNDAY) THEN
   GIVE RAN closed_ran_gv
   DISCONNECT
ELSE
   GIVE RAN open_ran_gv
END IF
```

Expression	Is it after 06:00 p.m.?	Is it Saturday or Sunday?	Play closed recording?
OR	yes	yes	yes
OR	yes	no	yes
OR	no	no	no
OR	no	yes	yes

</td></tr>
<tr><td>NOT</td><td>

The NOT expression evaluates a conditional expression and returns a True statement to the script if the expression is False, and a False value if the expression is True.

Example: On the 1st and 15th of every month, there is a company-wide meeting that reduces the number of agents available to answer sales calls from customers. During the meetings, the company plays a recorded announcement that tells callers that there can be some delay in answering their call. However, the company does not hold the meetings if the 1st or 15th is Saturday or Sunday. Regular sales service is available on Saturday or Sunday. This example also shows how you can combine two expressions (in this case, AND and NOT).

```
IF (DAY OF MONTH = 1,15)
   &&!(DAY OF WEEK = SATURDAY, SUNDAY) THEN
   GIVE RAN meeting_ran_gv
END IF
```

Expression	Is it the 1st or 15th day of the month?	Is it Saturday or Sunday?	Give meeting recording?
AND NOT	yes	yes	no
AND NOT	yes	no	yes
AND NOT	no	no	no
AND NOT	no	yes	no

</td></tr>
</table>

Table 22: Relational operations

Symbol	Description
You can use relational expressions to compare the values of intrinsics, variables, and constants. The data type on each side of the comparison equation must be the same for the comparison to be a valid expression. Use relational expressions only with variables that return integers, seconds, or dates. For example, although you can determine that a numerical value greater is than 312, you cannot determine that a skillset is greater than another skillset. Values that can only be True or False are called Boolean values. The state of several situations at once (for example, it is later than 08:00, and there is at least one agent logged on to the skillset) can be evaluated by writing several comparison expressions, and then joining them into a logical expression consisting of a series of true or false answers.	
Equal to (==)	Compare two values of the same type to see if they are equal or if they are of the same list or range. Use Equal to (==) for numeric or text data when you compare an item to another item.
Not Equal to (!=)	Compare two values of the same type to see if the first value is different from the second value or if the value on the left is not in the list or range of values on the right. Use Not Equal to (<>) for numeric or text data when you compare an item to another item.
Greater Than (>)	Compare two values of the same type to see if the first value is greater than the second value. Use the operation only with integers, time, and dates.
Less Than (<)	Compare two values of the same type to see if the first value is less than the second value. Use the operation only with integers, time, and dates.
Greater Than or Equal (>=)	Compare two values of the same type to see if the first value is greater than or equal to the second value. Use the operation only with integers, time, and dates.
Less Than or Equal (<=)	Compare two values of the same type to see if the first value is less than or equal to the second value. Use the operation only with integers, time, and dates.
Contains	Determine if a string contains a group of characters.

Table 23: Other operators in the condition and assignment builder

Operator	Description
TO	Create a range of an expression. Example: Days of the week. DAY OF WEEK == Monday TO Friday.
APPEND	Add the current string to the existing string.
ASSIGNMENT	Assign the variable equivalent to a value.

Table 24: Editing operators

Operator	Description
Bsp	Remove the last addition to your expression.
Fwd	Move forward in the expression.
Clear Entry	Clear entire entry in the Condition box.
	Move backward one position in the expression.
	Move forward one position in the expression.

Deleting an expression

About this task

Delete an expression from the list of completed expressions when you no longer require it for the block.

Procedure

1. Under **Statements**, select the expression you want to remove.
2. Click **Remove**.

Chapter 18: Application manager data and variable configuration

Before or after you create your Avaya Aura® Contact Center applications, you can create the application manager data and application variables you plan to use for your system.

The administration of application variables and application manager data is restricted by access classes in the Contact Center view. There are four access classes:

- None
- Read Only
- Read and update
- Read, update, create and delete

In the Local view, there are no restrictions of the variables and application manager data because it is local to your machine.

Application Manager Data (AMD) types include agents, CDNs, DNISs and skillsets. You can create, modify or delete CDNs, DNISs and skillsets in the Contact Center view of Orchestration Designer. You can create, modify and delete all types of AMD in the Local view of Orchestration Designer or in Contact Center Manager Administration. You can synchronize application manager data to appear in both the Contact Center and Local views. If AMD is created in the Contact Center Manager Administration application, you must refresh Orchestration Designer to see the new resources.

Application variables can be created in the Contact Center View or in the Local View of Orchestration Designer and be synchronized in Orchestration Designer to appear in the other view.

Variables and variable values are case-sensitive. Variables are user-defined words that you can insert in an application in place of a value or a list or range of values. There are three types of variables:

- Global variables are constants that you can use in an application on the system.
- Call variables have a value that can change for each contact. These variables follow the contact through the system and pass from one application to another with the contact.
- Wild variables are integer call variables that encounter a READVAR/ SAVEVAR command. Other applications use the value of the wild variable in other applications to execute further commands.

Important:

The maximum supported number of wild variables is 30.

Some variables can have a list or range of values instead of only a single value. For example, you can create a variable for holidays that includes all holidays for the year. Variables with sets of values use a maximum of 100 items. The system tracks the first 100 items in a set of values. Item class variables must

have a single value. You must use sets with global variables rather than call variables because the ASSIGN statement supports assigning values only to an item and not a set.

The following table lists the types of variables you can use in your applications.

Table 25: Variable definitions for contacts

Data type	Class allowed	Leading zero allowed	Valid value	Example
ACD	Item	Yes	2–7 digit string	8900
AGENT ID	Item or set	Yes	list of agents	Smith, John (453)
BOOLEAN	Item	No	True or False	True
CDN	Item or set	No	1–30 alphanumeric	abc123
CLID	Item or set	Yes	1–32 digits	4165552244
CONTACT TYPE	Item	No	String values	Voice
DATE	Item or set	No	<mm>/<dd>	9/1
DAY	Item or set	No	Monday to Sunday	Tuesday
DAY OF MONTH	Item or set	No	1–31	10
DN	Item or set	Yes	1–32 digits (negative numbers are not allowed)	5552353
DNIS	Item or set	Yes	1–31 digits	3400
INTEGER	Item	No	Number from –1 999 999 999 to 1 999 999 999	22986
LANGUAGE	Item	No	French, German, English, Spanish, Portuguese, Chinese, Japanese, Italian, Russian, Korean	French
LOC	Item or set	Yes	three-digit string	123
MONTH	Item or set	No	January to December	March
MUSIC	Item	No	music route 0–511	73
NPA	Item or set	No	three-digit string	123
NPANXX	Item or set	No	six-digit string	123456
NXX	Item or set	No	three-digit string	123
PRIORITY	Item	No	numbers 1–10	5
RAN	Item	No	RAN route, 0–511	72

Data type	Class allowed	Leading zero allowed	Valid value	Example
ROUTE NUMBER	Item or set	No	route number, 0–511	234
SECONDS	Item	Yes	0–65535	10
SKILLSET	Item or set	No	list of skillsets	French_sales
STRING	Item	Yes	1–80 characters	Log message
TIME	Item or set	Yes	0:00 to 23:59	11:15
TREATMENT	Item	Yes	1–7 digit string	0123456
VOICE SEGMENT	Set	No	language, filename: segment number (must be between 0-999) **Note:** Segment numbers of 4 or more digits are not allowed. This field is case sensitive. Type the filename exactly as it appears on the Voice Prompt Editor. The maximum length of a voice segment is 120 seconds.	English file1:234
WILDCLID	Item or set	Yes	1- to 32-digit number containing wildcard @ or placeholder ? symbols	416@ or 41?

You can create application variables in the Contact Center or Local view of Orchestration Designer.

This chapter explains how to create Application Manager Data and Application Variables. It also described how to assign values to variables, how to change the values of variables, and how to delete variables. It describes these activities in Orchestration Designer, and the Contact Center Manager Administration application.

Prerequisites to variable configuration

- Create all system resources such as RAN routes, music routes, voice ports, call treatments, and agents in the Contact Center Manager Administration application.

Creating application manager data

Before you begin

- Ensure that you create a CCMS folder in a Contact Center or Local view.

About this task

Application Manager Data (AMD) types include agents, CDNs, DNISs and skillsets. You can create, modify or delete CDNs, DNISs and skillsets in Orchestration Designer. You can create placeholders for agents only in the Local view of Orchestration Designer. Create agents for the Contact Center view in the Contact Center Manager Administration application.

You can create Application Manager Data before or after you create your Avaya Aura® Contact Center applications.

You must configure agents in the Contact Center Manager Administration configuration if you need agents to be used in applications running in your contact center.

Procedure

1. In the Contact Center or Local view, expand the CCMS server.
2. Open the **Application Manager Data** folder.
3. Right-click the folder that represents the type of data to add, and click **New**.
4. Select the **Local view** or the **Contact Center view** to create your new data.
5. Expand the **CCMA** and **CCMS** folders in which to create the data item.
6. Complete the details for your Application Manager Data.
7. Click **Finish**.

Variable definitions

Name	Description
Create in Local view	Select this option to create the variable in the Local view. This option is selected by default if you used in the Local view AMD to create the new data item.
Create in Contact Center	Select this option to create the variable in the Contact Center view. This option is selected by default if you used in the Contact Center view AMD to create the new data item.
CDN Name	Specify an identifying name for the Call Distribution Network when you create CDN application manager data for use in your applications.
Number	Specify a number for the Call Distribution Network when you create CDN application manager data or the Dialed Number Identification Service number when you create DNIS application manager data for use in your applications.
DNIS Name	Specify an identifying name for the Dialed Number Identification Service number when you create DNIS application manager data for use in your applications.
Service Level Threshold	Specify a service level threshold for the DNIS.
Skillset Name	The name of the skillset to use in your applications.
First Name	The first name of an agent you plan to use in your applications. You can only create agent AMD in the Local view.
Last Name	The last name of an agent. You can only create agent AMD in the Local view.
Login ID	The logon identification for an agent. You can only create agent AMD in the Local view.
Comment	Provide a detail about the AMD resource you configure.

Renaming application manager data

Before you begin

- Create Application Manager Data in your Local or Contact Center view.

About this task

You can change the name of any resource in the Application Manager Data to clearly indicate the purpose of the data item when you use it in your applications.

Procedure

1. Right-click on the Application Manager Data item you want to change, and click **Rename**.
2. Type the new name for the Application Manager Data.
3. Click **OK**.

Viewing application manager data properties

Before you begin

- Create Application Manager Data in your Local or Contact Center view.

About this task

You can modify the properties of the application manager data.

Procedure

1. Right-click on the Application Manager Data item you want to change, and click **Properties**.
2. Review the properties for the Application Manager Data.
3. Click **OK**.

Deleting application manager data

Before you begin

- Create Application Manager Data in your Local or Contact Center view.

About this task

Remove application manager data that you no longer require. Removing extra data reduces the download times and server resources.

Procedure

1. Right-click on the Application Manager Data item you want to delete, and click **Delete**.
2. On the Confirmation message box, click **OK**.

Note:

Only deacquired applications can be deleted.

Adding a variable to Orchestration Designer

Before you begin

- Open Orchestration Designer.

About this task

Before or after you create your Avaya Aura® Contact Center applications, you can create the variables you plan to use for your system. You can create, modify, or delete variables in Orchestration Designer.

Add a variable in the Local view of Orchestration Designer to use throughout your applications. Variables are user-defined words that you can insert in an application in place of a value or a list or range of values.

Note:

You must ensure that you do not create a variable that has an existing name defined in Orchestration Designer. For a list of reserved names, see Reserved Keywords on page 461. If you create new application variables that have the same name as an existing variable, you cannot save or run your applications.

Procedure

1. Expand the Contact Center Manager Administration server.
2. Expand the Contact Center Manager Server node.
3. In the **Contact Center** view or **Local** view, expand **Application Variables**.
4. Select the type of variable you want to create.
5. Right-click and then click **New**.
6. In the **New Application Variable** dialog box, select the location to create the new variable (Local view or Contact Center view).
7. Select the CCMA and CCMS server for the new application variable.
8. In the **Application Variable Name** box, type a unique name for the variable.
9. Under **Application Variable Type**, click **Global Variable** or **Call Variable**.
   A global variable contains the same value throughout the application.
10. In the **Comment** text box, enter a description for the variable.
11. Click the **Next** button.
12. Select **Item** or **Set** as the **Class** type.
13. In the **Value** box, type the value.
14. If you are creating a set of values, for each value you want to add to the set, click **Add** to add the value to the list.
15. Click **Finish**.

Variable definitions

Name	Description
Class Type	Defines the value as a single item or a list or range of items.
Comment	Information about the variable. The comment is optional.
Global or Call Variable	The type of application variable. You can choose between: • Global variable—Variables that you can use in a script application on the system. • Call variable—Variables where the value can change for each contact.

Name	Description
Application Variable Name	The name of the variable. The variable must • be unique • cannot be the same as skillset names, language keywords, or intrinsics • begin with a letter and cannot contain spaces • valid characters include A to Z, a to z, 0 to 9 and underscore (_) • use generic names to reuse the variables in other applications • include information about the variable type in the variable name (_gv, _cv)
Type	The data type of the variable. For more information about the types of variables, see Application manager data and variable configuration on page 251.
Value	The single value (item) or list or range of items (set) for the variable. The values depend on the data type of the variable.

Changing the properties of a variable

Before you begin

- Ensure that a variable exists in your Contact Center or Local view.

About this task

Change the properties of a variable. You can update the value of the variable, add more values to a set, add comments about the variable, or change the variable type to meet the requirements of your application.

You cannot change the name of the variable. You must delete the variable, and create a new variable. You must update the variable name where it is used in your applications.

Procedure

1. Expand the Contact Center Manager Administration server.
2. Expand the Contact Center Manager Server node.
3. In the **Contact Center** view or **Local** view, expand **Application Variables**.
4. Select the type of variable you want to modify.

5. Right-click and then click **Modify**.
6. Make the change to the variable.
7. Click **Finish**.

Removing a value from a set

Before you begin

- Know which value you are removing.
- Start Orchestration Designer.

About this task

Remove a value from a list or range of values for a variable to avoid assigning one or more values to the variable when you use it in an application.

Procedure

1. Expand the Contact Center Manager Administration server.
2. Expand the Contact Center Manager Server node.
3. In the **Contact Center** view or **Local** view, expand **Application Variables**.
4. Select the type of variable you want to change.
5. Right-click and then click **Modify**.
6. In the **Comment** text box, change the comment, if required.
7. Click **Next**.
8. In the **Configured values** list, select a value that you want to remove.
9. Click **Remove** or **Remove All**.
10. Click **Finish**.

Deleting a variable

Before you begin

- Open Orchestration Designer.

About this task

Delete a variable to remove it from the list of variables for your applications when you no longer need it. The variable list may be long. To delete a variable from Contact Center Manager Administration, see Deleting a variable from Contact Center Manager Server on page 269.

Procedure

1. Expand the Contact Center Manager Administration server.
2. Expand the Contact Center Manager Server node.
3. Expand **Application Variables**.
4. Select the variable you want to delete.
5. Right-click and then click **Delete**.
6. On the confirmation message, click **OK**.

Starting the Script Variables tool in Contact Center Manager Administration

About this task

Start the Script Variables tool to list the application variables on your system.

You can also use this window to create or delete an application variable.

Procedure

1. Log on to the Contact Center Manager Administration Server.
2. In an Internet Explorer window, type http://<servername>.
3. Use your administrator login ID and password to start the application.
4. Click **Scripting**.
5. In the Scripting window, expand the system tree.
6. In the system tree, click your Contact Center Manager Server.

7. Click **Script Variables**.

Creating variables with one value

Before you begin

- Create all system resources such as RAN routes, music routes, voice ports, call treatments, and CDNs.

  For more information about setting up the system resources, see *Avaya Aura® Contact Center Configuration – Avaya CS 1000 Integration* (NN44400-512).
- Create all agents who handle contacts. For more information about setting up the skillsets and agents, see *Avaya Aura® Contact Center Manager Administration – Client Administration* (NN44400-611).
- If you plan to create voice segment variables, create your voice segments. For more information see *Avaya Aura® Contact Center Manager Administration – Client Administration* (NN44400-611).
- Understand the variable types you create. For more information, see Variable definitions for contacts on page 251.
- Start the Scripting Variables tool. See Starting the Script Variables tool in Contact Center Manager Administration on page 261.

About this task

Create application variables with a single value in Contact Center Manager Administration to create all variables before you begin to create applications.

Note:

You must ensure that you do not create a variable that has an existing name defined in Orchestration Designer. If you create new application variables that have the same name as an existing variable, you cannot save or run your applications.

Procedure

1. In the Scripting window, expand the system tree.
2. In the system tree, click your Contact Center Manager Server.
3. Click **Script Variables**.
4. In the left pane of the Script Variables window, right-click the type of variable you want to create, and then click **New**.
5. In the right pane, click **Script Variable** to minimize the list of variables.
6. Click **Script Variable Properties**.

7. Click the **General** tab.
8. In the **Name** box, type the name of the variable.
9. Select **Global Variable** or **Call Variable**.
10. Click the **Attribute** tab.
11. In the **Type** box, choose the type of variable.
12. If the **Class** box appears, select **Item**.
13. In the **Value** box, type the value you want to assign to the variable.
14. Click **Submit**.

Variable definitions

Name	Description
Name	The name of the variable. The variable must • be unique • cannot be the same as skillset names, language keywords, or intrinsics • begin with a letter and cannot contain spaces • valid characters include A to Z, a to z, 0 to 9 and underscore (_) • use generic names to reuse the variables in other • include information about the variable type in the variable name
Global or Call Variable	The type of variable. You can choose between: • Global variable—Variables that you can use in a script application on the system. • Call variable—Variables where the value can change for each contact.
Type	The data type of the variable. Possible data types include AGENT_ID, CDN, CLID. For a complete list of variable types, see Variable definitions for contacts on page 251.

Name	Description
Value	The multiple values (set) for the variable. The values depend on the data type of the variable.

Creating application variables with multiple values

Before you begin

- Create all system resources such as RAN routes, music routes, voice ports, call treatments, and CDNs.

  For more information about setting up the system resources, see *Avaya Aura® Contact Center Configuration – Avaya CS 1000 Integration* (NN44400-512).
- Create all agents and skillsets. For more information about setting up the agents and skillsets, see *Avaya Aura® Contact Center Manager Administration – Client Administration* (NN44400-611).
- If you plan to create voice segment variables, create the voice segments.
- Understand the variable types you create. For more information, see Variable definitions for contacts on page 251.
- Start the Scripting Variables tool. See Starting the Script Variables tool in Contact Center Manager Administration on page 261.

About this task

Create application variables with multiple values in Contact Center to create all variables before you begin to create applications.

Note:

You must ensure that you do not create a variable that has an existing name defined in Orchestration Designer. If you create new application variables that have the same name as an existing variable, you cannot save or run your applications.

Procedure

1. Open the Contact Center Manager Administration application, and click **Scripting**.
2. In the Scripting window, expand the system tree.
3. In the system tree, click your Contact Center Manager Server.
4. Click **Script Variables**.
5. In the left pane of the Script Variables window, right-click the type of variable you want to create, and then click **New**.

6. In the right pane, click **Script Variable** to minimize the list of variables.
7. Click **Script Variable Properties**.
8. Click the **General** tab.
9. In the **Name** box, type the name of the variable.
10. Select **Global Variable** or **Call Variable**.
11. Click the **Attribute** tab.
12. In the **Type** box, choose the type of variable.
13. In the **Class** box, select **Set**.
14. In the **Value** box, type a new value or select an available value in the list and click **Add**.
15. Click **Submit**.

Variable definitions

Name	Description
Name	The name of the variable. The variable must: • be unique • cannot be the same as skillset names, language keywords, or intrinsics • begin with a letter and cannot contain spaces • valid characters include A to Z, a to z, 0 to 9 and underscore (_) • use generic names to reuse the variables in other applications • include information about the variable type in the variable name
Global variable	Global variable—Variables that you can use in an application on the system.
Type	The data type of the variable. Possible data types include ACD, AGENT_ID, BOOLEAN. For a complete list of variable types, see Variable definitions for contacts on page 251.

Name	Description
Value	The single value (item) for the variable. The values depend on the data type of the variable.

Checking variables for referencing applications

Before you begin

- Start the Script Variables tool. See Starting the Script Variables tool in Contact Center Manager Administration on page 261.

About this task

Check the variables to see if it is referenced by an active application. If it is referenced by an active application, you can change the value of the variable or the comment.

If you want to change the properties of a variable and the variable appears in an application, you can deactivate the application or remove the reference to the variable from the referencing application.

Procedure

1. In the Scripting window, expand the system tree.
2. In the system tree, click your Contact Center Manager Server.
3. Click **Script Variables**.
   The system tree in the left pane expands to show all types of variables. The right pane shows an alphabetical list of all variables. In the Script Variables grid, you can sort all columns by clicking on the column header.
4. In the left pane of the Script Variables window, select the script variable that you want to check.
5. In the right pane, click **Script Variable Properties**.
6. Under **Referencing Scripts**, determine which scripts use the variable.
7. View the **Script Variable Properties** to change the variable value or comment.
8. Click **Submit**.

Changing a variable property

Before you begin

- Ensure that the variable is referenced by no active applications. See Checking variables for referencing applications on page 266.
- Start the Script Variables tool. See Starting the Script Variables tool in Contact Center Manager Administration on page 261.

About this task

Change a variable property such as the description, the value, or the list of values to more accurately define the variable you use in your applications.

You cannot change the variable type. If you want to change the variable type, you must delete the variable and create a new one.

Procedure

1. In the Scripting window, expand the system tree.
2. In the system tree, click your Contact Center Manager Server.
3. Click **Script Variables**.
   The system tree in the left pane expands to show all types of variables. The right pane shows an alphabetical list of all variables. In the Script Variables grid, you can sort all columns by clicking on the column header.
4. In the Script Variables window, select the script variable that you want to change.
5. Click **Script Variable Properties**.
6. In the **Script Variable Properties** property sheet, change the information in the **Comment**, and **Value** fields as required.
7. Click **Submit**.

Variable definitions

Name	Description
Comment	The description for the variable.
Value	The value for the variable.

Copying variable properties

Before you begin

- Start the Script Variables tool. See Starting the Script Variables tool in Contact Center Manager Administration on page 261.

About this task

Copy variable properties from one variable to another to create a new application variable.

Procedure

1. In the Scripting window, expand the system tree.
2. In the system tree, click your Contact Center Manager Server.
3. Click **Script Variables**.
4. In the Script Variables window, right-click the script variable that you want to copy.

   OR

   In the **Script Variables** grid, select the row for the script variable you want to copy and right-click the row.
5. Click **Copy**.
6. In the right pane, click **Script Variables**.
7. Click **Script Variable Properties**.
8. In the **Name** box, type the name of the variable.
9. Click the **Attribute** tab.
10. Change the values in the Attribute tab as required.
11. Click **Submit**.

Variable definitions

Name	Description
Name	The name of the variable.

Deleting a variable from Contact Center Manager Server

Before you begin

- Start the Script Variables tool. See Starting the Script Variables tool in Contact Center Manager Administration on page 261.
- Ensure that the variable is not referenced by an active applications. See Checking variables for referencing applications on page 266.

About this task

Delete a variable from Contact Center Manager Server to remove it from the system because you need to change the type, or because you do not require the variable.

Procedure

1. In the Scripting window, expand the system tree.
2. In the system tree, click your Contact Center Manager Server.
3. Click **Script Variables**.
4. In the Script Variables window, right-click the variable that you want to delete.

   OR

   In the **Script Variables** grid, select the row and right-click the variable that you want to delete.
5. Click **Delete**.
6. In the **Confirm Delete** message box, click **Yes**.

Refreshing Orchestration Designer

Before you begin

- Open Orchestration Designer.

About this task

If you configure application manager data or variables in the Contact Center Manager Administration environment, you must refresh Orchestration Designer application to see the new variables.

Procedure

On the **File** menu, choose **Refresh**.

Printing the variables to a file

Before you begin

- Start the Script Variables tool. See Starting the Script Variables tool in Contact Center Manager Administration on page 261.

About this task

Print the variables to a file to have a permanent record that summarizes the variables available for your applications.

Procedure

1. In the Scripting window, expand the system tree.
2. In the system tree, click your Contact Center Manager Server.
3. Click **Script Variables**.
   The system tree in the left pane expands to show all types of script variables. The right pane shows an alphabetical list of all script variables. In the Script Variables grid, you can sort all columns by clicking on the column header.
4. In the Script Variables window, right-click the Script Variables tree value. You can choose all of the script variables, or only one type.

   OR

   Select the row for the variable you want to save to a file, and right-click the row.
5. Click **Print Preview**.
6. In the Crystal Reports viewer, click the **Export Report** icon.

   The Export Report icon is the icon that appears like an arrow into an envelope on the top left-hand side of the window.
7. In the **Export Report** dialog box, in the **File Format** list, select the application for exporting the variables.
8. In the **Page Range** box, select to export all or a specific page.
9. In the **File name** box, type the name of the file to which the variables are exported.

   OR

   Click **Browse** to select the path for the file.

10. Click **OK**.

Variable definitions

Name	Description
File Format	The name of the software application to view the exported report of variables. You can choose from the following applications: • Crystal Reports • Microsoft Excel • Microsoft Excel (data only) • Microsoft Word • Rich Text Format • Adobe Acrobat (PDF)
Page Range	The collection of pages that are exported to the file. You can choose All, or specify a range of page numbers.
File name	The name of the file where the exported variables are saved.

Printing a variable to a printer

Before you begin

- Start the Script Variables tool. See Starting the Script Variables tool in Contact Center Manager Administration on page 261.

About this task

Print the variables to a printer to have a written record that summarizes the variables available for your applications.

Procedure

1. In the Scripting window, expand the system tree.
2. In the system tree, click your Contact Center Manager Server.
3. Click **Script Variables**.

4. In the Script Variables window, right-click the Script Variables tree value. You can choose all of the script variables, or only one type.

   OR

   Select the row of the script variable you want to print, and right-click the row.
5. Select **Print Preview**.
6. Select the **Print Report** icon.
7. In the **Select Printer** box, choose your printer.
8. Select the **Page Range** for the variables.
9. Click **Print**.

Variable definitions

Name	Description
Select Printer	The list of printers connected to your local computer.

Chapter 19: Threshold configuration

Use Contact Center Manager Administration to configure and view the list of threshold classes defined on your system.

A Threshold Class is a group of options that determines how statistics are treated in reports and real-time displays. Create threshold classes to reflect the expected standard of service within a contact center. Threshold classes define this standard with first- and second-level threshold values that are applied to your contact center data. When a real-time display reaches a threshold value, the Agent Desktop Displays application alerts supervisors to exceptional performance or changing conditions within the contact center.

Threshold class values alert supervisors to alarm conditions by highlighting the values that exceed desired performance levels on real-time displays. Threshold classes give supervisors a dynamic view of goals established for contact center resources.

Each script or flow application is assigned a threshold class.

Assign an application ID to each application to identify it for work force management and statistics monitoring to track individual script or flow applications. Because a contact can pass through many applications, contacts are tracked by the application ID of the primary application that they enter from the master application.

The maximum number of threshold classes you can configure for Contact Center is 505.

Prerequisites to threshold configuration

- Ensure that you have access to the application thresholds.

Creating new threshold class

About this task

Create a new threshold class, which is a collection of options that determines how statistics appear in reports and real-time displays.

Procedure

1. From Contact Center Manager Administration, choose **Configuration**.

2. Expand the desired server.
3. Select the **Threshold Classes** folder.
4. In the **Threshold Classes** window beside the asterisk, type the name of the new threshold class.
5. From the **Type** list, select the type **Application**.
6. In the Thresholds window, choose the thresholds for the new threshold by selecting the **Enabled** check box beside the threshold statistic.
7. For each threshold, select the Level 1 and Level 2 thresholds.

Variable definitions

Name	Description
Level 1	The lowest level of application threshold. For a display threshold, type the value for the low end of the normal range in the Level 1 box. For a pegging threshold, in the Level 1 box, type the upper limit value for this statistic.
Level 2	The highest level of application threshold. For a display threshold, type the value for the high end of the normal range in the Level 2 box. For a pegging threshold, do not specify an upper limit for this statistic.

Procedure job aid: Display and pegging thresholds

The two types of thresholds are display thresholds and pegging thresholds.

Display thresholds, the most common type, trigger highlighting on real-time displays. Display thresholds use both Level 1 and Level 2 values to define the low and high ends of a normal range. The values for the two threshold levels cause a real-time display field to highlight in a color specified. For example, if you configure a Level 1 threshold of ten seconds for agent time in Not Ready, the display highlights when an agent's time in the Not Ready state reaches 10 seconds. When the agent's time in Not Ready reaches the second-level threshold value, the field highlights in a different color.

Pegging thresholds count the number of calls exceeding a standard. Pegging thresholds use only Level 1 values. The Level 1 value represents a cut-off limit for the accumulation of statistics. For pegging thresholds, the system disables Level 2 thresholds.

Selecting a threshold class for an application

Before you begin

- Create an application in your Contact Center view of Orchestration Designer. See Application creation on page 43.
- Create a threshold class. See Creating new threshold class on page 273.

About this task

Select a threshold class for an application in the Scripting section. You can also modify the statistics that are monitored and the threshold values to relax or tighten the restrictions for each statistic.

Procedure

1. From the system tree in the **Scripting** application, select your Contact Center Manager Server.
2. Click **Application Thresholds**.
3. In the right-hand pane, from the list, select the application that you want to change.
4. Under **Threshold <*application name*>**, select the threshold class you want to apply to the application.
5. To make additional changes to the threshold class, select the statistic and the levels you want to monitor.
6. To disable statistics, clear the **Enabled** check box beside the statistics that you want to remove from the threshold class.
7. Press **Tab** to save your changes.

Variable definitions

Name	Description
Application name	The name of the application in Orchestration Designer Contact Center view.
Level 1	The lowest level of application threshold. For a display threshold, type the value for the low end of the normal range in the Level 1 box.

Name	Description
	For a pegging threshold, in the Level 1 box, type the upper limit value for this statistic.
Level 2	The highest level of application threshold. For a display threshold, type the value for the high end of the normal range in the Level 2 box. For a pegging threshold, do not specify an upper limit for this statistic.

Chapter 20: Web Service Integration

You can extract user information from calls and use the information in your flow or script applications as parameters to Web services. The Web services return information in the script to help route the call. The Database Integration Wizard (DIW) incorporates the Web services in the Contact Center Manager Server application.

Examples of using the Database Integration Wizard include customer-hosted Web service that Avaya Aura® Contact Center can use to retrieve information to route a call. Avaya Aura® Contact Center can also call customer-hosted Web services to alert third-party applications about contact information.

Use the procedures in this chapter to extract the information and incorporate it into your applications.

Activating the SOA Web service

About this task

To incorporate SOA Web services in the applications, you must ensure that they are activated in the Contact Center Manager Server Configuration application.

Procedure

1. Click **Start** > **All Programs** > **Avaya** > **Contact Center** > **Manager Server** > **Server Configuration**.
2. In the Server Configuration dialog box, click **WS Open Interfaces**.
3. Select the **SOA ENABLED** check box.
4. Click **Apply All** or **OK**.
5. Click **Exit**.

Importing Web services to the Database Integration Wizard

Before you begin

- Ensure that you have access to the Web Service Description Language (WSDL) document Uniform Resource Locator (URL). For more information about the WSDL URL, see SDK and documentation downloads.

Note:

At the end of the Web service ULR, add ?WSDL to locate the Web service WSDL.

About this task

Use the DIW to configure the Web services that can call the particular user information to use in your Avaya Aura® Contact Center applications.

Important:

You can only use Web services that accept simple types of variables. Multiple parameters are allowed, and you can have a maximum of 10 variables. SQL queries accept a fixed number of simple data types.

Procedure

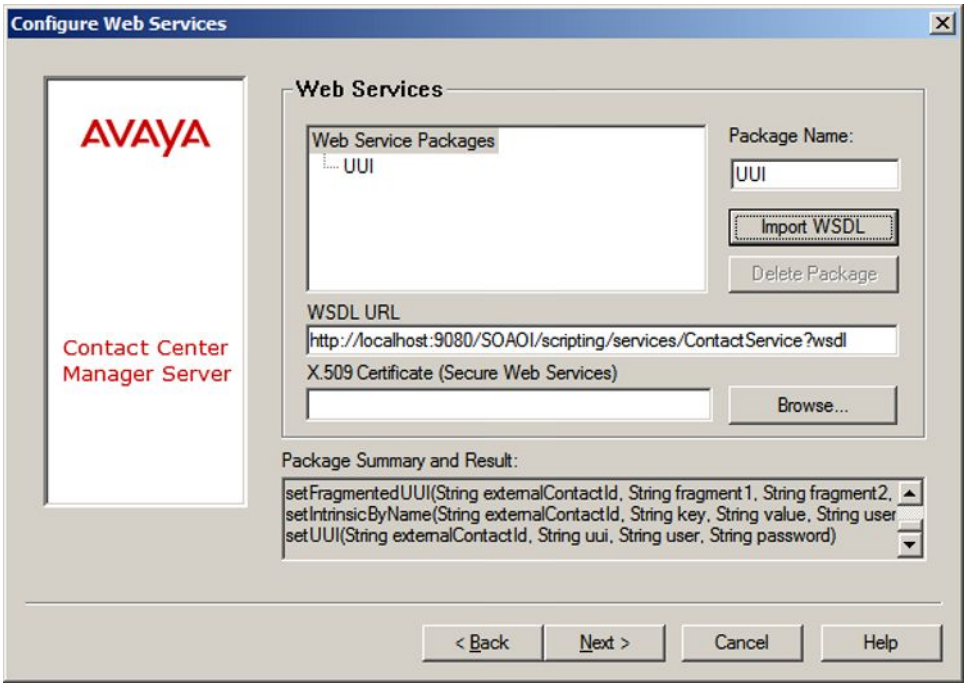
1. On the Contact Center Manager Server, click **Start** > **All Programs** > **Avaya** > **Contact Center** > **Manager Server** > **Database Integration Wizard**.
2. On the Welcome window, click **Next**.
3. On the HDX Connection window, in the **Provider ID** box, type a provider identity number for the Web service, for example type 121.
4. Click **Test Connection** to ensure the provider identity is not in use.
5. Click **Next**.
6. On the Configure Database Connections window, click **Next**.
7. On the Configure Web Services window, in the **WSDL URL** box, type the address of the WSDL document that describes the Web service.

   For example a WSDL URL is http://<*CCMS server IP address*>:9080/SOAOI/scripting/services/ContactService?wsdl
8. In the **Package Name** box, type the unique name for the Web service package.

   For example, choose UUI.
9. If the Web service uses SSL/TLS security, in the **X.509 Certificate (Secure Web Service)** box, type the location of the X.509 certificate.

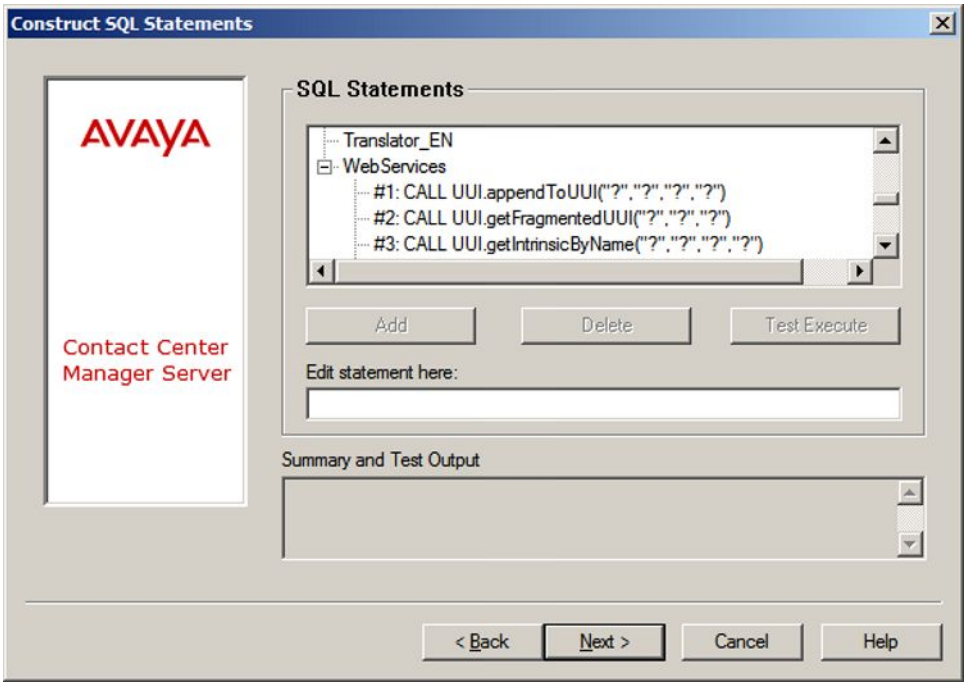
   OR

   Click **Browse** to choose the location of the X.509 certificate.
10. Click **Import WSDL**.



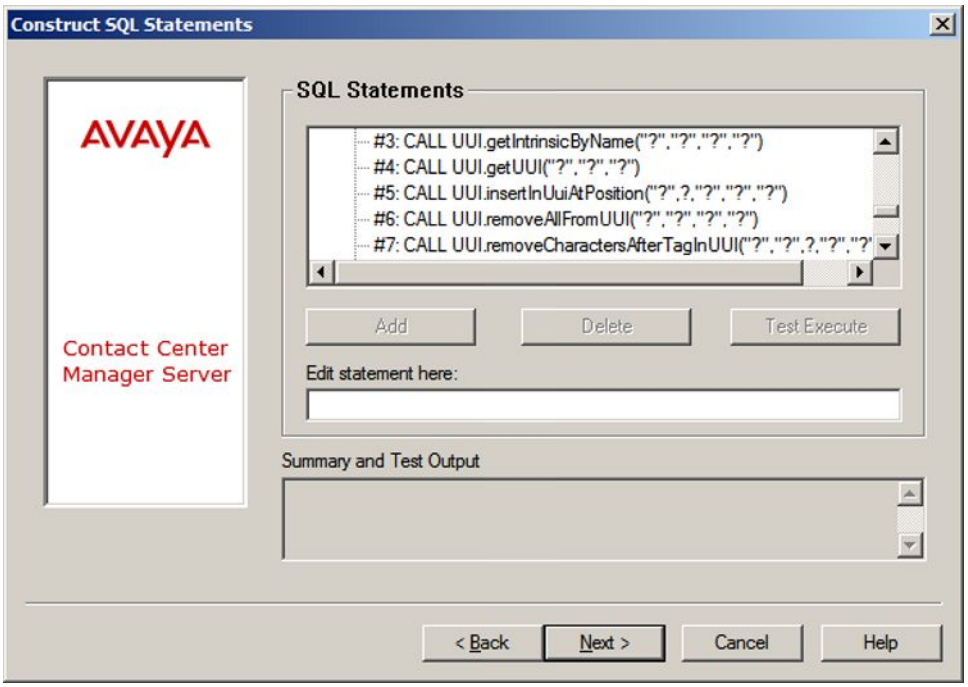
The import process can take some time, depending on the size of the WSDL document. After you import WSDL, a summary of the imported procedures appears in the Package Summary and Result box. If the import is not successful, the reason for the failure appears in the box.

11. Click **Next**.
12. On the Construct SQL Statements window, scroll down and select Web Services.



13. For the Web service you want to use, note the function number.

    For example, in the following window, we want the service UUI.getUUI. The function is number 4, which you use for the call of the Web service from the CCMS application.



14. Click **Next**.
15. On the Complete window, click **Finish**.

Procedure job aid

The following table shows all of the ContactService class services..

Service	Description
appendToUUI	Appends the specified ‘appendData’ to the end of the current UUI if no UUI is currently set, it sets the UUI to the ‘appendData’.
getFragmentedUUI	Returns the UUI property value of the contact in fragmented format if the contact exists and the login credentials are valid.
getInstrinsicByName	Returns the value of the intrinsic key specified. Throws an exception if the login parameters are not correct or the contact does not exist.

Service	Description
getUUI	Returns the UUI property value of the contact if the contact exists and the login credentials are valid.
insertInUuiAtPosition	Inserts ‘newData’ specified into the current UUI at the specified index.
removeCharactersAfterTagInUUI	Looks for the substring defined in the tag parameter in the current UUI and removes a number of characters after the tag.
replaceAllInUUI	Replaces all occurrences of ‘oldData’ found in the current UUI with the ‘newData’ specified. Returns a fail if no UUI currently set.
replaceCharactersAfterTagInUUI	Looks for the substring defined in the tag parameter in the current UUI and replaces a number of characters after the tag.
replaceFirstInUui	Replaces the occurrence of the specified 'findData' with the specified 'newData'. Returns a fail if there is no UUI currently set.
setFragmentedUUI	Sets the UUI property value of the contact by concatenating the fragments specified to get the final value. **Note:** May result in only part of the UUI being returned if the UUI is too long to fit in the fragmented result.
setIntrinsicByName	Looks for a contact based on the external id specified if the intrinsic is not immutable and an immutable intrinsic with the same key does not already exist. If an intrinsic does not exist, the intrinsic is added and the value is set. If an intrinsic already exists with the same key, then the value of this intrinsic is updated, and the original intrinsic value is returned.
setUUI	Sets the UUI property on the contact if the contact exists and the login credentials are valid.

Configuring variables in Orchestration Designer

Before you begin

- Ensure the HDX_AppID global variable is created. If not, see Adding a variable to Orchestration Designer on page 257.
- Know the unique HDX number for the Web service. For more information, see Importing Web services to the Database Integration Wizard on page 277.
- Know the other variables you need to create for your application.

About this task

In Orchestration Designer, you must ensure that the HDX application is configured to access the Web services in the Database Integration Wizard (DIW). You need to create call variables to store the information in Orchestration Designer.

Procedure

1. Expand the Contact Center Manager Administration server.
2. Expand the Contact Center Manager Server node.
3. In the Contact Center view, expand **Application Variables INTEGER**.
4. Select the **HDX_AppID** variable.
5. Right-click and then click **Modify**.
6. Ensure the variable is type **Item**, and the value is the value you used in the HDX Connection window.
7. Repeat the steps above to create all of the other variables you require for the Web service calls and retrievals.
8. Click **Finish**.

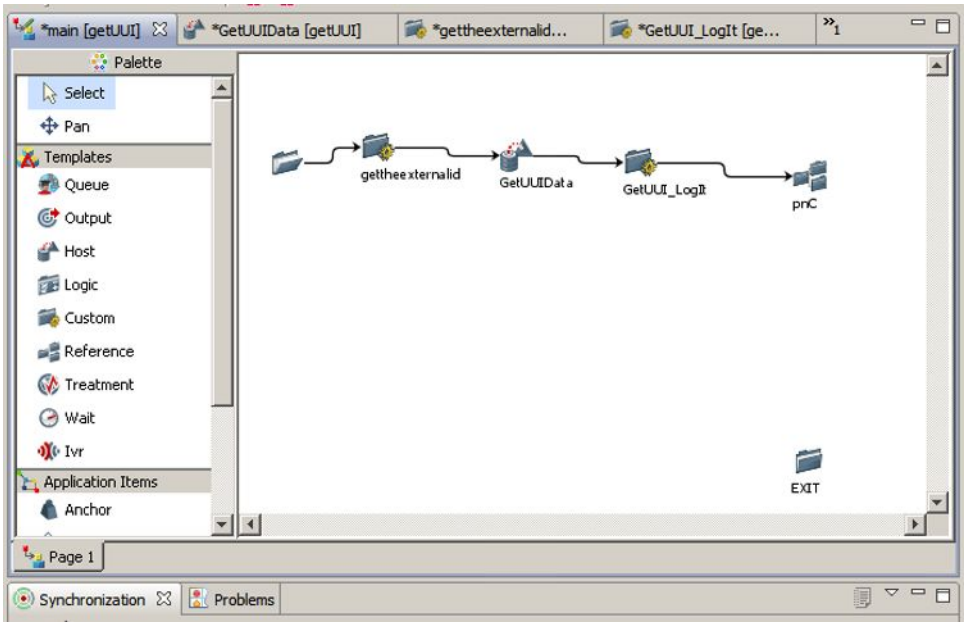
Creating required blocks to handle data

About this task

If the Web service updates contact information, you must add commands to the application. In the example of the Web service UUI.getUUI, an external call id is required to retrieve the information.

The external call id is the value of the ProviderContactID of the call which is stored within the call intrinsics. To generate the external call ID, use the command CONTACT DATA ProviderContactID in a Orchestration Designer Custom block, or in a script.

In this example, the flow appears as follows.



The first custom block, gettheexternalid, contains the command,

```
ASSIGN CONTACT DATA “ProviderContactID” TO externalid
```

The *externalid* variable must be defined in Orchestration Designer. For more information, see Adding a variable to Orchestration Designer on page 257.

The second custom block, GetUUI_LogIt, is used to trace a call. After the flow is working as expected, the logging custom block can be omitted. The following commands are used to trace the call:

```
ASSIGN CONTACT DATA "ProviderContactID" TO externalid
APPEND WSResult_cv TO im_str
LOG im_str
```

Note:

Other Web services require different commands, or different blocks in a flow application to generate the required data correctly.

Calling the Web service and collecting return values

Before you begin

- Know the call id number.
- Configure a flow application to include a custom block to get the external call ID.
- Know the username and password for the Web services you call.
- Know the Web service function number you want to use.
- Configure the variables required for calls and inputs for the Web service. See Adding a variable to Orchestration Designer on page 257.
- Add a Host block to your flow application. See Creating a Host block on page 199.
- Create a variable to capture and store the output from the Web service. See Creating a variable on page 262.

About this task

Create a Host block to call a Web service to request information using Host Data Exchange commands without required detailed knowledge of the user information.

The Request/Response uses the same Provider ID in the Send Info option to send a request or receive a response from the Contact Center Manager Server. The Provider ID is entered using a literal text value or a variable from the application manager data. You can define the request and response parameters and maintain these parameters using the Add, Remove, Move up and Move down controls.

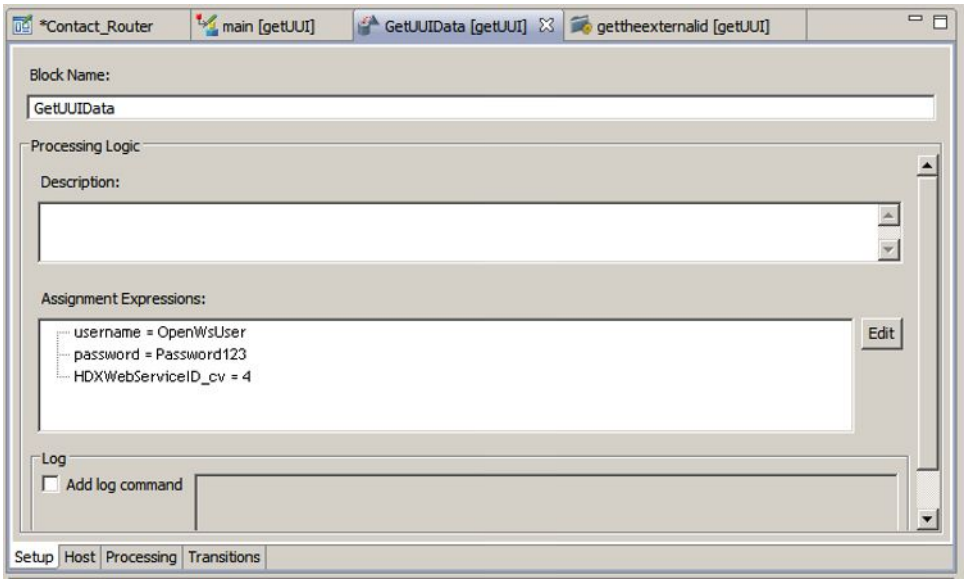
In addition to the main configuration tabs, the host data block also has the host tab. Use the host tab to configure the data that received by the Host Data Exchange commands.

Procedure

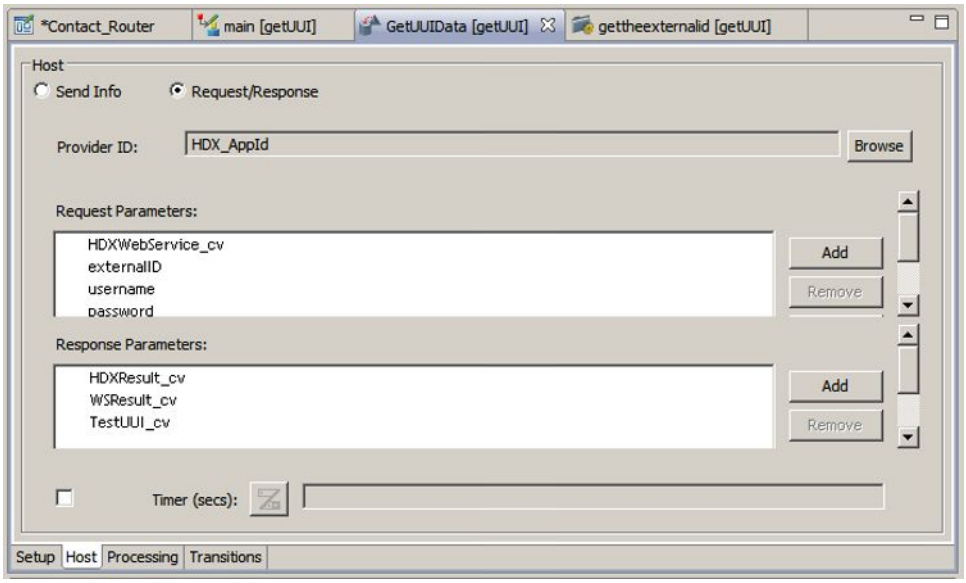
1. Select the Host block icon, right-click and click **Open**.

   OR

   Double-click the Host block icon.
2. Click the **Setup** tab.
3. Ensure the Assignment Expressions include the username and password for the Web service, and the function number for the Web service you want to call.



4. Click the **Host** tab.
5. Select **Request/Response**.
6. Specify the provider ID from the application manager data. To call the Web service, the Provider ID must be HDX_AppId.
7. Beside Request Parameters, click **Add** to add each parameter to the list. The parameters that appear in the graphic below are specific to this example. You must include the variable list required for your Web service calls.
8. Beside Response Parameters, click **Add** to add each parameter to the list. The parameters that appear in the graphic below are specific to this example. You must include the variable list required for your Web service responses.



Parameter	Description
HDX_AppID	Provider ID used to call the Web services.
HDX_QueryNum_cv	The Web service function number.
externalID	The external ID for the contact.
username	The user name for the Web service access. The user name is defined for the Server Configuration utility.
password	The password for the Web service access. The password is defined for the Server Configuration utility.
HDX_Resp_cv	The result of the execution of the Web service call. It is an indicator of success or failure.
WSResult_cv	The value of the result from the execution of the Web service.

9. To add assignments or conditional assignments that process after the block is run and before the evaluation of the transition to the next block, click the **Processing** tab and configure the assignments.
10. Click the **Transition** tab to configure transitions from this block.
11. Save the changes to the flow application.

Chapter 21: User preferences

The procedures in this chapter provide information about configuring Orchestration Designer to suit your personal preferences for creating applications.

Prerequisites for user preferences

- Start Orchestration Designer.

Configuring network connection preferences

About this task

If you have a number of Contact Center Manager Servers (CCMS) on your Contact Center Manager Administration, they are all read by Orchestration Designer when you start the application.

If you want to choose the CCMS server to read first so that you can begin to work on it, you can designate your preference for the first CCMS server by selecting it in the list. Configure the CCMS Read Delay to provide a short time to allow you to choose the preferred CCMS server to read first. The default value of the CCMS Read Delay box is 7 seconds.

You can also configure the default settings for the Contact Center Manager Administration (CCMA) application. You must choose the correct CCMA port number if your SSL is configured.

You can also select service connection values to time-out if no one is using the system.

Procedure

1. From the **Window** menu, choose **Preferences**.
2. Expand **Contact Center**.
3. Select **CCMA Connection**.
4. Configure the properties for your CCMA Connection.
5. Click **Apply**.
6. Click **OK**.

Variable definitions

Name	Description
CCMS Read Delay (seconds)	Provide a short time to allow you to choose the preferred CCMS server to read first. The default value 7 seconds.
SSL Enabled	If your Contact Center Manager Administration server is SSL-enabled, you can check the box and ensure that you specify the correct Port for your secure server.
Login Timeout (minutes)	The length of time before Orchestration Designer displays an authentication failure with CCMA's Web service if no answer is received. A value of 0 indicates an infinite timeout value.
CCMA Timeout (minutes)	The length of time before Orchestration Designer displays a communication failure with CCMA's Web service during the transfer of Application information. A value of 0 indicates an infinite timeout value.

Configuring automatic saving

About this task

Configure automatic saving to choose the interval at which Orchestration Designer projects automatically save.

Procedure

1. From the **Window** menu, choose **Preferences**.
2. In the Preferences window, select **Contact Center**.
3. Select the **Enable** check box.
4. In the **Interval (minutes)** box, specify the number of minutes between automatic saves.
5. In **Orchestration Designer Internal Logging Level**, select the level of notification that is placed in Orchestration Designer log file when the automatic save occurs.

6. Click **Apply**.
7. Click **OK**.

Variable definitions

Name	Description
Interval (minutes)	The length of the time interval between automatic saves.
Orchestration Designer GUI Tool Internal Logging Level	The level of the log entry in Orchestration Designer log when the automatic save is performed. The default log file location for Orchestration Designer is C:\Avaya\Logs\Sysops \MsiLogs. You can choose from • Error • Debug • Warning • Info

Configuring application preferences

About this task

Create application preferences in Orchestration Designer to define your global preferences for application development.

Application preferences can include:

- validation options—when and how to display the validation errors in your application
- code generation options—how to display code and debugging logs in your applications (display all or none)
- script editor options—number lines in the script editor
- default wait options—the shortest default wait time for implementing wait blocks or commands in your application
- priority values—default priorities are pre-configured in the block editor for configuring queues

Procedure

1. From the **Window** menu, select **Preferences**.
2. On the Preferences window, select **Contact Center**.
3. Select **Default Options**.
4. Select the **Default Options** you want to configure.
5. Select the **Code Generation options** you want.
6. If you want to see line numbers in the **Script editor**, select the **Show Line Numbers** check box.
7. In the **Default Wait Values**, select the number of seconds that you want to use for the default wait time in your wait commands.
8. In the **Default Priority Values**, select the priority values for the queue and IVR contacts.
9. Click **Apply**.
10. Click **OK**.

Variable definitions

Name	Description
Validate options	Choose when and how the validation occurs on each script or flow. Select the check box next to the following options: • Validate on open—Validate the script or flow application when it is opened. • Warning messages—Display warning messages in the Problem view. • Suggestions—Display suggestions for correcting the error in the Problem view.

Name	Description

Code generation options	Choose code generation options. Select the check box next to the following options:
	• Include comments—Include comments throughout the application.
	• Include debug logging—Include debugging flags throughout the application.
	`IF debug THEN`
	`  LOG 'message'`
	`END IF`
	• Sections ordered by layer—Help the user to avoid TFE critical errors due to loops in the script.

Script Editor option	Choose to have the lines numbered in the script editor. Select the check box to enable script editor lines.
Default wait values	The minimum length of time used for wait blocks or commands in your application. You can choose values for: • Default Queue Wait value • Default Network Queue Wait value
Default priority values	The default priorities assigned to calls in queue blocks or commands in your application. You can choose a queue priority for default queues from 1 to 6. This includes the agents. You can choose a queue priority for NACD queues from 1 to 4. You can choose a priority for IVR queues from 1 to 4.

Resetting the Contact Center perspective

About this task

Reset the Contact Center perspective to show the default configuration of the views in your Orchestration Designer.

The default configuration of the views in your Orchestration Designer if you launch Orchestration Designer from Contact Center Manager Administration include the Contact Center view, the Local view, the Problems view, and the Synchronization view. If you start

Orchestration Designer on a local machine, your default configuration includes only the Local view and the Problems view.

Procedure

1. On the **Window** menu, click **Reset Perspective**.
2. Click **OK**.

Refreshing data

About this task

Refresh the data such as Contact Center Manager Administration server, the Contact Center Manager Server, the agents, skillsets, CDNs, DNISs, applications, and application variables in the Contact Center view to ensure that the information about variables and applications is current.

The scope of the refresh is determined by the level that it is invoked from. For example, invoking at the Contact Center Manager Administration level refreshes all Contact Center Manager Servers, and the applications and variables on the server.

Procedure

1. Select the item you want to refresh.
2. On the **File** menu, click **Refresh**.

Chapter 22: Getting help

Use the procedures in this chapter to learn about getting help with Orchestration Designer.

Viewing online Help in Orchestration Designer

About this task

Use the online Help to view information about Eclipse and Orchestration Designer.

Procedure

1. From the **Help** menu, choose **Help Contents**.
2. On the Help window, select the topic that you require help with.

Viewing context-sensitive Help in Orchestration Designer

About this task

View context-sensitive Help to view information about the current task.

Procedure

1. With your mouse, select a window for which you need help.
2. Press **F1** to open the **Help** view.
3. Click an object on the screen to view available help topics in the **Help** view.

Chapter 23: Scripting tips

This chapter provides some general scripting guidelines.

Guidelines for providing feedback

When you handle incoming calls, Avaya recommends that your scripts provide the caller with confirmation that their contact is still being processed. For voice contacts, provide ringback or other treatments such as music or recorded announcements. All other multimedia contact can use the auto-acknowledgement features in the multimedia applications.

Consider what the user experiences

Examine your scripts to determine what the user experiences. For example, in the following script for a voice contact, if an agent does not become available, the caller receives the following feedback:

- The caller hears ringback when the call enters the queue in the general skillset.
- If no agents are available, the caller hears the entire RAN agents_busy_ran_gv.
- The caller hears music for 20 seconds.
- If, at the end of 20 seconds, no agents are available, the caller hears the entire RAN agents_still_busy_ran_gv.
- The caller hears music again until the call is answered.

The following script is an example of a voice contact script.

```
QUEUE TO SKILLSET general_sk WITH PRIORITY 3
WAIT 2 /* Check for idle agent*/
GIVE RAN agents_busy_ran_gv
GIVE MUSIC local_station_gv
WAIT 20
GIVE RAN agents_still_busy_ran_gv
```

Default call treatments

If no other treatment is given for the following commands, ringback automatically applies for voice contacts:

- QUEUE TO SKILLSET
- QUEUE TO AGENT
- SEND INFO
- SEND REQUEST
- LOG
- QUEUE TO NACD (available with the Networked Skills Based Routing feature only)
- QUEUE TO NETWORK SKILLSET (available with the Networked Skills Based Routing feature only)

QUEUE to SKILLSETS

If a voice contact receives no treatment before it is queued to a skillset by the QUEUE TO SKILLSET command, Contact Center automatically gives a ringback tone when the voice contact:

- enters the skillset queue
- is sent to an agent

The voice contact often hears only a burst of ringback in these situations, depending on the delay—or lack of delay—in providing the next treatment. Because a script cannot control that delay, it does not completely control the tones heard by the caller.

Example

This example shows how to give the voice contact a full cycle of ringback before the next treatment. This method forces a 6-second delay before giving tones to ensure a more natural-sounding ringback cycle.

```
QUEUE TO SKILLSET sales_sk
WAIT 6
GIVE MUSIC classical_music_gv
```

GIVE RAN

By default, a voice contact hears silence after a recorded announcement or after the completion of an IVR session. If you want the voice contact to hear anything other than silence you must insert a GIVE MUSIC or GIVE RINGBACK command.

GIVE MUSIC

Music resumes after all commands except GIVE SILENCE and GIVE RINGBACK. Therefore, you need not repeat the GIVE MUSIC command in a script.

Using variables to minimize script maintenance

Use variables to minimize script maintenance. Variables make your scripts easy to modify. For example, if you use the variable business_hours to represent the hours that your company is open (for example, 09:00 a.m. to 05:00 p.m.), and you change the hours to 08:00 a.m. to 06:00 p.m., you can update only the variable, instead of all the scripts.

Variables must have unique names. Do not use keywords, skillset names, or section labels to name variables. To avoid errors, define all variables before you write your scripts. If you define a variable that is not referenced by a script, delete it after you write all the scripts.

For more information about variables, see Application manager data and variable configuration on page 251.

Chapter 24: General programming commands reference

This chapter describes the basic script commands to which all Contact Center system have access. For each command, the command script syntax, the parameters, and optional segments are provided.

All commands in this chapter apply to both voice and multimedia contacts.

ASSIGN TO

Assign values to call variables.

Syntax

ASSIGN <*value*> TO <*variable*>

Parameters

<*variable*> The name of the variable to which you want to assign a value when a script executes. You can only use variables with type Item.

<*value*> The value that you want to assign to the variable. You can use an item or an expression for the value.

EXECUTE

Branch to another section in the same script.

Syntax

EXECUTE <*section_label*>

Parameter

<*section_label*> The name of the section in the current script to run.

EXECUTE SCRIPT

Branch to a different script.

After the call branches to the referenced script, it does not return to the original script.

Syntax

EXECUTE SCRIPT <*Script_Name*>

Parameter

<*Script_Name*> The name of the script that you want to run.

IF-THEN-END IF

Choose to perform one or more statements if a specified condition is met. If the condition is not met, the script skips to the next command.

You can use an IF command in a statement to introduce another condition, but each IF must have exactly one matching END IF command.

Syntax

IF <*logical_condition*> THEN <*statements*> END IF

Parameters

<*logical_condition*> The condition for which you want the script to test.

<*statements*> The actions that you want the script to take if the condition is met.

IF-THEN-ELSE-END IF

Choose to perform one or more statements if a specified condition is met. If the condition is not met, one or more other conditions are evaluated. If none of the conditions are met, the script skips to the next command.

You can use an IF command in a statement to introduce another condition, but each IF must have exactly one matching END IF command.

Syntax

IF <*logical_condition*> THEN <*statement1*> ELSE <*statement2*> END IF

Parameters

<*logical_condition*> The condition which you want the script to test.

<*statement1*> The action that you want the script to take if the condition is met.

<*statement2*> The action that you want the script to take if the condition is not met.

QUIT

End the execution of a script if commands remain to be executed.

Scripts stop executing automatically when no commands remain to execute.

Syntax

QUIT

Restriction

The QUIT command cannot be the first command in a script.

SECTION

Define a section of commands. You can jump to a section within in the same script using the EXECUTE command.

Syntax

SECTION <*section_name*>

Parameters

<*section_name*> is the name of the section.

Example

The following SECTION command creates a simple loop that requeues the contact if the contact is not queued.

```
SECTION Wait_Loop
   WAIT wait_delay_gv
   IF NOT QUEUED THEN
      EXECUTE Requeue_Call
   END IF
   GIVE RAN please_wait_ran_gv
   EXECUTE Wait_Loop
```

WAIT

Suspend an application for a period of time before executing the next application command. Avaya recommends that you run a WAIT command with at least 2 seconds after queuing a contact to a skillset or to an agent. The 2-second pause provides time for the contact to be answered by the agent before the command is executed.

You enter the WAIT time in seconds format. If you enter a WAIT time of 0, the system resets the value to 2 seconds. Zero is the only value that receives special handling. The system uses timers when processing calls in a WAIT state. Therefore, the WAIT time is accurate to +/– 1 second. For example, if you specify a WAIT time of 5 seconds, the WAIT time can be from 4 to 6 seconds.

Syntax

WAIT <*time_in_seconds*>

Parameter

<*time_in_seconds*> The amount of time, in seconds, that you want the script to pause. You can replace this parameter with a variable of type seconds or with a numerical constant.

Example 1

In this example, calls are queued to the sales skillset. After a 2-second delay, callers hear a recorded announcement asking them to wait.

```
QUEUE TO SKILLSET sales_sk
WAIT 2
GIVE RAN please_wait_ran_gv
```

Example 2

The placement of the WAIT command, particularly in the master application, can affect report statistics.

For example, if the main application contains the following:

```
GIVE RINGBACK
WAIT 6
IF CDN = test_cdn THEN
   EXECUTE Sales_App
END IF
```

And the primary script performs the following:

```
QUEUE TO SKILLSET Sales_Sk
WAIT 2
```

If a call arrives and agents are available in Sales_Sk, then the Application Report shows an Average Delay of at least 6 to 8 seconds, depending on the timing queue, and the Skillset Report shows a delay of 0 to 2 seconds.

Avaya recommends that you qualify the GIVE RINGBACK command as shown in the following script so that only transferred calls are affected by the WAIT statement—new calls are not affected.

```
IF TRANSFERRED THEN
   GIVE RINGBACK
   WAIT 6
END IF
IF CND = test_cdn THEN
   EXECUTE Sales_App
END IF
```

Chapter 25: Basic contact processing commands reference

Basic contact processing commands describes the basic processing that you can use in your applications. The contact processing commands describe the functions for each contact and represent a skillset or agent interaction with the contact.

Some commands in this chapter apply only to voice contacts. Other commands apply to both voice and multimedia contacts.

APPEND

Add alphanumeric characters to the variable in a voice or multimedia contact.

Syntax

APPEND <*Variable*> TO <*ContactVariable*>

APPEND <*LiteralStringConstant*> TO <*ContactVariable*>

Parameters

<*Variable*>, <*ContactVariable*> must be of type String.

Example

If you assign the language French to the variable language_gv and voice to the variable contact_type_gv using the command:

APPEND language_gv TO contact_type_gv

The new variable language_gv becomes Frenchvoice.

CHANGE PRIORITY IN AGENT

Change the priority of a voice or multimedia contact that is currently in an agent's queue command only when the voice or multimedia contact is queued by QUEUE TO AGENT command.

The CHANGE PRIORITY IN AGENT command applies only to local agents.

Syntax

CHANGE PRIORITY IN AGENT [<*agentID*> | <*agentID_list*>] TO PRIORITY <*priority*>

Parameters

<*agentID*> or <*agentID_list*> The ID of the agent to whom the contact is queued, or a list of IDs the agents to whom the contact is queued.

<*priority*> The new priority with which you want the contact queued to the skillset.

Restriction

The CHANGE PRIORITY IN AGENT command cannot be the first command in an application.

CHANGE PRIORITY IN SKILLSET

Change the priority of a voice or multimedia contact that is currently in a skillset to which it is queued. The priority of the voice or multimedia contact changes in all skillsets when the command is executed.

The CHANGE PRIORITY IN SKILLSET command applies only to local skillsets.

Syntax

CHANGE PRIORITY IN SKILLSET [<*skillset*> | <*skillset_list*>] TO PRIORITY <*priority*>

Parameters

<*skillset*> or <*skillset_list*> The skillset, or list of skillsets, to which the contact is queued. You can replace this parameter with a variable that returns a skillset or skillset list.

<*priority*> The new priority with which you want the contact queued to the skillset.

Restriction

The CHANGE PRIORITY IN SKILLSET command cannot be the first command in an application.

CONTAINS

The CONTAINS keyword searches for substrings within a main string. The CONTAINS keyword applies to all text-based communications in Contact Center.

Use CONTAINS in logical expressions such as IF and WHERE. It is only applicable for use with string type variables or literal strings. When the variable is a string variable, up to 80 digits can be returned and evaluated by the CONTAINS keyword.

```
IF stringvar CONTAINS substringvar
   THEN statements [ELSE statements]
END IF
```

Syntax

CONTAINS

Restrictions

In both cases stringvar is a variable of type string. Both call and global variables are acceptable in CONTAINS keyword searches.

```
ASSIGN “This is an example” TO string1
```

IF string1 CONTAINS “is” the script returns TRUE

IF string1 CONTAINS “anexample” the script returns FALSE

The CONTAINS keyword also supports the wildcard character asterisk *

IF string1 CONTAINS “is*example”the script returns TRUE

IF string1 CONTAINS “th*AN*pl” the script returns TRUE

The search string may contain multiple wildcards and the comparison is case insensitive. Since the algorithm is a substring search there is no need for leading or trailing wildcards.

Example

```
WHERE string2 CONTAINS
   VALUE string1: String1
   VALUE string2: String2
   DEFAULT: String3
END WHERE
```

Each VALUE or string literal may also contain multiple wildcards and is case insensitive.

You can use the intrinsic SIP_INITIAL_TEXT to search the initial message from a customer.

If the search string is blank, then a positive match results.

CONVERT

Convert script variables of type ACD, CDN, CLID, DN, DNIS, and INTEGER to STRING. The string operations APPEND, CONTAINS, and MATCHES only work with string variables. This command allows a primitive conversion for complex routing decisions in scripting.

You cannot convert variables of type SKILLSET, AGENT_ID, or any other types of variables not listed above.

Syntax

CONVERT <*simple variable*> TO STRING <*call variable*>

Parameters

<*simple variable*> A global or call variable of type ACD, CDN, CLID, DN, DNIS, INTEGER, or STRING.

<*call variable*> The new call variable of type STRING after the conversion.

Example

Create variables as follows: myCDN_cv, myCLID_cv, convertedString_cv.

```
ASSIGN CDN TO myCDN_cv
CONVERT myCDN_cv TO STRING convertedString_cv
IF convertedString_cv MATCHES “3000 3001 3002” THEN
   LOG “CDN matches a CDN in the group 3000 to 3002”
END IF
ASSIGN CLID TO myCLID_cv
CONVERT myCLID_cv TO STRING convertedString_cv
IF covertedString_cv CONTAINS “35366” THEN
   LOG “This call is from Kerry”
END IF
```

DISCONNECT

Disconnect a voice or multimedia contact. You can disconnect a voice or multimedia contact at any time when it is in the CDN or ROUTE POINT queue, except when it is presented to or answered by an agent.

This command applies to all voice and multimedia contacts but for application efficiency, Avaya recommends that you hold multimedia contacts in the application even if a skillset goes out of service or if an agent logs off. If the DISCONNECT command is used for multimedia contacts, Contact Center Multimedia automatically resubmits the contact to the application immediately.

Syntax

DISCONNECT

Restriction

DISCONNECT must be the last command in the application.

GIVE BUSY

Provide a busy tone to a voice contact before the switch disconnects it. A voice contact can receive a busy tone any time that it is in the CDN queue, except when it is presented to, or answered by, an agent.

Syntax

GIVE BUSY {CONTROLLED}

CONTROLLED keyword is optional on the Avaya Communication Server 1000 switch.

Parameters

If you use the CONTROLLED keyword, the contact remains in the control of Contact Center, which means that if an error condition occurs on the switch (cannot give the tone to the contact, for example), the contact is returned to the queue so that the system treats it. After a command without the CONTROLLED keyword is complete, Contact Center has no control over the contact.

Restriction

This treatment must be the first treatment that the contact receives; otherwise, for some trunk types, the switch does not disconnect the contact automatically after giving the busy tone. These trunks stay connected until the caller disconnects the contact.

Because some commands give an automatic ringback tone, you must analyze the application to ensure these commands do not execute before the GIVE BUSY command. For more information about commands that automatically generate a ringback treatment, see Guidelines for providing feedback on page 295.

Do not use this command in a network application if you purchased Networked Skills-Based Routing.

GIVE MUSIC

Play music from a specified music route to a voice contact that is waiting in queue.

Syntax

GIVE MUSIC <*music_route*>

Parameters

<*music_route*> The variable of type music or a numeric constant that represents the music route that contains the music you want to play to the contact.

Restriction

Do not use the GIVE BUSY or GIVE OVERFLOW command immediately after GIVE MUSIC.

Other information

Use WAIT command immediately after GIVE MUSIC to control the duration of the music. If a contact is given music followed by a GIVE RAN command, the music resumes after the RAN finishes.

Music stops when another treatment or action with an unspecified duration is given (such as GIVE RINGBACK or ROUTE CALL). Music is suspended during presentation to an agent and resumes if the contact returns to the queue (if the agent does not answer).

In applications that use the GIVE IVR command, you must use GIVE MUSIC or GIVE RINGBACK after the GIVE IVR command to provide a default treatment. If you do not configure default treatment after the GIVE IVR command, the customer hears silence.

GIVE OVERFLOW

Provide an overflow tone to a voice contact before the call disconnects from the switch. The contact is given an overflow tone at any time that it is in the CDN queue, except when it is presented to, or answered by, an agent.

An overflow tone is a fast busy tone.

Syntax

GIVE OVERFLOW {CONTROLLED}

CONTROLLED keyword is optional on the Avaya Communication Server 1000 switch.

Parameters

If you use the CONTROLLED keyword, the contact remains in the control of Contact Center, which means that if an error condition occurs on the switch (cannot give the tone to the contact, for example), the contact returns to the queue so that the system treats it. After a command without the CONTROLLED keyword are complete, Contact Center has no control over the contact.

Restrictions

This treatment must be the first treatment that a contact receives; otherwise, for some trunk types, the contact is not disconnected automatically by the switch after the busy tone. This can cause trunks to remain connected until the user disconnects the voice contact.

Because some statements give an automatic ringback tone, you must analyze the application carefully to ensure that these commands are not executed before the GIVE OVERFLOW command. Issuing the ringback tone after the GIVE OVERFLOW command ensures that the overflow tone is always the first treatment given to the contact. For more information about commands that automatically generate a ringback treatment, see Guidelines for providing feedback on page 295.

Do not use this command in a network application if you purchased Networked Skills-Based Routing.

GIVE RAN

Provide a recorded announcement (RAN) to a voice contact through the specified RAN trunk. The RAN is interrupted if an agent becomes available to take the voice contact. Otherwise, the RAN is completed and the next command in the application is executed.

If the GIVE RAN command is the first treatment in the application, and there is a delay before the RAN is available, then the contact hears ringback until the announcement plays.

Syntax

GIVE RAN <*ran_route*>

Parameters

<*ran_route*> The variable of type RAN or the route number of the RAN route that contains the recorded announcement that you want to play to the contact.

Restrictions

Do not use GIVE BUSY or GIVE OVERFLOW commands immediately after the GIVE RAN command.

Other information

For Avaya Communication Server 1000 switches, the GIVE CONTROLLED BROADCAST feature allows a maximum of 50 callers to listen to the same port for an announcement. This feature requires Avaya CallPilot®.

For GIVE RAN commands in SIP-enabled contact centers, you must use GIVE MUSIC or GIVE RINGBACK after the GIVE RAN command to provide a default treatment. If you do not configure a default treatment after the GIVE RAN command, the customer hears silence.

GIVE RINGBACK

Provide a ringback tone in a voice contact (a ringing sound) to a caller.

When the voice contact is presented to an agent, the switch automatically provides the ringback tone.

Syntax

GIVE RINGBACK

Restriction

Do not use GIVE BUSY or GIVE OVERFLOW commands immediately after the GIVE RINGBACK command.

Other information

You can use this command with the WAIT command if you want to control the duration of the ringback tone. Use this command before a contact is queued against one or more destinations (that is, to skillsets or agents). If a contact is queued by the application without a specific treatment given first, then Contact Center automatically applies ringback (while the contact is queued). For more information about which commands give an automatic ringback tone, see Guidelines for providing feedback on page 295.

GIVE SILENCE

Provide silence to a voice contact. Use GIVE SILENCE to turn off either music or ringback.

Syntax

GIVE SILENCE

Restriction

The GIVE SILENCE command can potentially be the first treatment for a contact even if it is not the first command in the application. This can happen, for example, if an IF-THEN-ELSE or WHERE-EQUALS command leads to a GIVE SILENCE. In that case, you receive an error, and the contact is rejected and routed to the default ACD-DN of the CDN.

MATCHES

The MATCHES keyword searches for strings within a main string. The MATCHES keyword applies to all text-based communications in Contact Center.

Use MATCHES in logical expressions such as IF and WHERE. It is only applicable for use with string type variables or literal strings. When the variable is a string variable, up to 80 digits can be returned and evaluated by the MATCHES keyword.

You can use either call or global variables in MATCHES keyword searches.

You can use multiple MATCHES keyword separated by spaces.

Syntax

MATCHES

Example

```
ASSIGN “My name is Joe” TO im_str
IF im_str MATCHES “Joseph Jos* Joe Joey”
```

```
   THEN LOG “im_str MATCHES Joseph”
END IF
WHERE im_str MATCHES
   VALUE “Joseph Jos* Joe Joey”:
      LOG “im_str MATCHES Joseph”
   VALUE “Michael Mic* Mike”:
      LOG “im_str MATCHES Michael”
   DEFAULT:
      LOG “No match found”
END WHERE
```

If the search string is blank, then a positive match results.

QUEUE TO AGENT

Deliver a voice or multimedia contact to a local agent with agent-based routing capability.

Queuing to specific agents at other sites in a network is not supported.

Syntax

QUEUE TO AGENT [<*agent_id*> | <*agent_id_list*>] {WITH PRIORITY <*priority*>}

Syntax: LONGEST IDLE AGENT

QUEUE TO AGENT LONGEST IDLE AGENT [<*skillset*> | <*skillset_list*>] {WITH PRIORITY <*priority*>}

Parameters

<*agent_id*> or <*agent_id_list*> The ID of the agent, or the list of agent IDs, to which you want the contact queued.

<*priority*> The priority with which you want the contact queued to the specific agent.

<*skillset*> or <*skillset_list*> The skillset, or list of skillsets, from which the longest idle agent is taken.

Restrictions

Do not use the GIVE BUSY or GIVE OVERFLOW commands immediately after the QUEUE TO AGENT command.

Use a WAIT command with at least 2 seconds after a QUEUE TO AGENT command.

LONGEST IDLE AGENT intrinsic

If you include the LONGEST IDLE AGENT intrinsic, the contact is presented to the agent who is idle for the longest period from all of the skillsets listed. The criteria used to determine which agent is idle for the longest period is defined for all skillsets. The definition is one of the following:

- the amount of idle time since the last contact was taken by the agent
- the total amount of idle time since the agent logged on
- the idle time since the last status change

Use LONGEST IDLE AGENT only if you are sure that an idle agent is available. If no idle agents are available, Contact Center does not run the QUEUE TO AGENT command.

The QUEUE TO AGENT WITH PRIORITY option for queueing to specific agents at other sites in a network is not supported. You must use the QUEUE TO NETWORK SKILLSET command to queue the contact to network agents.

QUEUE TO SKILLSET

Queue voice or multimedia contacts locally at the Contact Center site. Queue local contacts against local and network skillsets, provided that local agents are assigned to the network skillsets. Locally queued contacts are answered only by local agents. Queue the contacts according to the number of agents logged on to the skillsets or by the length of time the agents are idle at the skillsets.

Syntax

QUEUE TO SKILLSET [<*skillset*> | <*skillset_list*>] {WITH PRIORITY <*priority*>}

Syntax: MOST LOGGED AGENTS

QUEUE TO SKILLSET MOST LOGGED AGENTS [<*skillset*> | <*skillset_list*>] {WITH PRIORITY <*priority*>}

Syntax: LONGEST IDLE AGENT

QUEUE TO SKILLSET [<*skillset*> | <*skillset_list*>] BY LONGEST IDLE AGENT {WITH PRIORITY <*priority*>}

Parameters

<*skillset*> or <*skillset_list*> The skillset, or list of skillsets, to queue the contact.

<*priority*> The priority with which the contact is queued to the specific skillset.

Restrictions

Do not use the GIVE BUSY or GIVE OVERFLOW commands immediately after the QUEUE TO SKILLSET command.

Use a WAIT command with at least 2 seconds after a QUEUE TO SKILLSET command.

A QUEUE TO SKILLSET command cannot include both the MOST LOGGED AGENTS intrinsic and the BY LONGEST IDLE AGENT option.

Skillset lists

Queue contacts to up to 20 local skillsets simultaneously by specifying all skillsets to which contacts are queued in a single QUEUE TO SKILLSET command or by using a separate QUEUE TO SKILLSET command for each skillset.

The QUEUE TO SKILLSET command randomly evaluates each skillset. For each skillset, if an agent is available, the contact is presented to that agent. If no agents are available, the contact is queued to that skillset.

MOST LOGGED AGENTS intrinsic

If you use the MOST LOGGED AGENTS intrinsic, the skillset to which the contact is queued is based on the number of agents logged on to the skillset. The QUEUE TO SKILLSET command finds the skillset among all of the skillsets listed that has the most agents logged on at the time and queues the contact only to that one skillset.

BY LONGEST IDLE AGENT option

If you use the BY LONGEST IDLE AGENT option, and two or more agents (with the same priority for the skillset) are idle in the listed skillsets, the agent is selected based on the Global Skillset setting for Agent Preference.

The following settings are available:

- Longest total time in Idle state since login.
- Longest time in Idle state since the last status change between Not Ready, Logging out, Answering, Programming, or Reserve.
- Longest total time since last CDN or ACD contact including time spent idle, busy on DN contacts, and Not Ready.

When you use multiple skillsets, the BY LONGEST IDLE AGENT option selects the most qualified agent from all of the skillsets, based on the global setting for Agent Preference. When no idle agents are available, the system queues the contact to all skillsets listed in the command, and the next available agent is presented with the contact BY LONGEST IDLE AGENT has no affect.

The priority of the idle agents for the skillsets takes precedence over the Agent Preference setting, which means that the system always routes the contact to the highest priority agent, regardless of the Agent Preference setting.

When you choose multiple skillsets, the system randomly queries the skillset and the contact is presented to the longest idle agent for that skillset.

WITH PRIORITY option

Specify the priority with which the contact is queued. Assign a priority of 1 to 10, with 1 representing the highest priority and 10 being the lowest. A contact is assigned a priority 10 by default. Contacts with high priorities are presented to agents before the contacts with low priorities.

REMOVE FROM AGENT

Remove a voice or multimedia contact from the specified local agent. Use this command if a contact must be answered within a specific duration.

If the voice or multimedia contact is still queued after this amount of time passes, the contact can be retrieved and requeued.

Syntax

REMOVE FROM AGENT [<*agent_ID*> | <*agent_ID_list*>]

Parameters

<*agent_ID*> or <*agent_ID_list*> The single agent ID or list of agent IDs of the agent from which you want the contact removed.

REMOVE FROM SKILLSET

Remove a queued voice or multimedia contact from a local skillset.

Skillset

REMOVE FROM SKILLSET [<*skillset*> | <*skillset_list*>]

Parameters

<*skillset*> or <*skillset_list*> The skillset or list of skillset, from which you want the contact removed.

ROUTE CALL

Send a voice or multimedia contact to a destination specified by the parameter.

Syntax

ROUTE CALL [<*dn*> | DEFAULT DN] {CONTROLLED}

CONTROLLED is optional.

Parameters

<*dn*> The dialable internal or external number that represents the directory number (DN) to which you want the contact routed. Examples of dialable numbers are an ACD-DN or a personal DN.

If you specify the default DN, the contact is sent to the default ACD-DN configured on the switch for the CDN from which the contact entered the system.

If you use the CONTROLLED keyword, the contact remains in the control of Contact Center, which means that if an error condition occurs on the switch (cannot give the tone to the contact, for example), the contact is returned to the queue so that it can be further treated by the system. When a command without the CONTROLLED keyword is completed, Contact Center has no control over the contact.

Restrictions

For Avaya Communication Server 1000 switches, you cannot use Vacant Number Routing (VNR) with the ROUTE CALL command.

No commands can follow the ROUTE CALL command.

If the ROUTE CALL command is intended to be the first command in the script, then it must be preceded by WAIT 2.

Use the ROUTE CALL command with caution in a network application. If the DN specified is invalid, the ROUTE CALL command can cause phantom contacts.

Chapter 26: Advanced script command reference

This chapter describes the advanced script commands that you can use if you purchased the Avaya Contact Center application software package. For each command, the syntax, parameters, and optional segments are described for each command.

EVENT HANDLER

Give instructions to manage certain conditions that can occur during a call or RAN response failure. The Event Handler can manage unsolicited or unexpected events and failed responses that can interrupt a suspended state.

If an unexpected response occurs, the Event Handler runs statements to handle the event. The statements are placed at the beginning of the script and do not run unless the event occurs. If the event occurs, execution of the script moves to the Event Handler and executes the statement for that event. The script execution does not return to the body of the script.

If a failed treatment response occurs, the Event Handler command is performed, and then the script execution normally returns to the next statement in the main body of the script after the treatment command that failed.

A script monitors the call after it is answered by an agent. If the agent places the call on hold, the Event Handler can play music to the caller.

Syntax

EVENT HANDLER

EVENT <*event_a*>: <*statements*>
EVENT <*event_b*>: <*statements*>
EVENT <*event_c*>: <*statements*>
…

END HANDLER

Parameters

<*event_a*>, <*event_b*>, <*event_c*>: The events that you want the Event Handler to address.

<*statements*> The action that you want the script to take if the event occurs.

Restrictions

The Event Handler must be the first command in the script.

This command applies to voice contacts only.

Use only the following commands under the Call On Hold event:

- GIVE MUSIC
- IF-THEN-END-IF
- IF-THEN-ELSE-END-IF
- QUIT
- WAIT
- LOG
- SEND INFO

Use only the following commands under the Call Abandon event:

- IF-THEN-END-IF
- IF-THEN-ELSE-END-IF
- QUIT
- WAIT
- LOG
- SEND INFO

You can use all commands with the Response Broadcast Fail, IVR Response Fail, and RAN Response Fail events, except the SECTION command. After the failed response is addressed, the execute of the script leaves the Event Handler and returns to the original place in the script. The command that is immediately after the failed treatment command is executed.

The Event Handler applies only to the script in which it appears. If you want Event Handler conditions to apply after the call is sent to a referenced script, you must repeat the EVENT HANDLER command at the beginning of each script.

Examples of failed responses

Replace the event parameters with:

- RAN Response Fail
- IVR Response Fail
- Broadcast Announcement Response Fail

LOG

Record a message to the Event Browser. The LOG command is typically used for script testing because it slows response time if executed for every call in a high-traffic contact center.

Use the LOG command in the following situations:

- in test scripts to which only a few calls are made
- to identify errors

Syntax

LOG <*“msg_character_string”*>

Parameters

<*“msg_character_string”*> The text message entry as you want it to appear in the script log file.

You must enclose the text within quotation marks, or use a variable of type string without quotation marks.

Restriction

Search for event code 48467 in the Event Browser to locate messages created by the LOG command.

Do not use the LOG command for normal call processing. New events overwrite the old events in the log file. Events logged by this command reduce the amount of historical data in the log file.

Do not use GIVE BUSY or GIVE OVERFLOW commands immediately after the LOG command.

Limitations

The LOG command is shown as an event in the Event Browser up to 10 times. After this, the event is not visible because the throttling feature prevents reoccurring events from filling up the Event Browser log. The default setting for the throttling threshold is 3. The throttling feature is reset every hour, after which the event is visible again.

READVAR and SAVEVAR

Enable a call to change the value of a variable and pass the updated value to other calls.

READVAR

READVAR uses an existing integer call variable as a parameter, and then reads the current value of the call variable from the Wild Variable table. If the value does not exist in the Wild Variable table, READVAR reads the value from the Call Variable table.

You must terminate a READVAR block by using a SAVEVAR command with limited commands allowed in the block.

SAVEVAR

SAVEVAR saves the current value of the call variable to the Wild Variable table.

You can use the following commands only between READVAR and SAVEVAR:

- ASSIGN
- LOG
- IF-THEN-ELSE-END IF

Syntax

READVAR <*integer_call_variable*>

<*optional statements*>

SAVEVAR

Parameters

<*integer_call_variable*> The variable into which you want to read the value.

<*optional statements*> One of the following commands with appropriate syntax:

- ASSIGN
- LOG
- IF-THEN-ELSE-END IF

WHERE-EQUALS

Test for a condition that can have more than one result. Possible results include advancing to another command, running another script, or terminate the execution of the script. Use the DEFAULT clause to specify a command to use when none of the results are possible.

After finishing the statements in the applicable VALUE clause, the script leaves the WHERE-EQUALS command and continues executing the next commands after the END WHERE command, unless the statements transfer the call to another script.

Syntax

WHERE <*value_expr*> EQUALS

VALUE <*value_1*>: <*statements*>
VALUE <*value_2*>: <*statements*>
VALUE <*value_3*>: <*statements*>
DEFAULT<*statements*>

END WHERE

Parameters

<*value_expr*> The expression that you want the script to evaluate. You can replace the parameter with an item, an intrinsic, or a formula. If you use a variable in this parameter, it must be of type item.

<*value_1*>, <*value_2*>, <*value_3*> The values for the expression that you want the script to address. Replace these parameters with an item, a list, or a range.

<*statements*> The action that you want the script to take if the expression is equal to the specified value.

Limitations

The number of value clauses is unlimited; however, for effective script execution, Avaya recommends that you do not use more than 50 value clauses, and that you break value clauses into groups where possible.

Chapter 27: Network commands

Use the Network commands if your Contact Center system has the Network Skill-Based Routing (NSBR) option to send voice contacts from one contact center to another. When a contact is sent between sites through the network, the contact is processed by the network application at the target site only if the contact returns to the queue.

Using network skillsets

A network skillset is common to all Contact Center Manager Servers in a network. An administrator at the Network Control Center (NCC) must create a network skillset, and the NCC automatically propagates the network skillset to all sites. When a script queues a contact to a network skillset, the contact can be routed to a server on the network.

If a site has a local skillset with the same name as a network skillset, the server converts the local skillset to a network skillset. Agents assigned to the local skillset remain assigned to the network skillset. The existing scripts continue to function as before (that is, the QUEUE TO SKILLSET command continues to queue calls locally), and other sites can queue contacts to the skillset. However, unless you modify your applications to include the network skillset, contacts for that skillset that arrive at your site cannot be queued to any other site.

To use NSBR, you must modify your existing applications when a network skillset is propagated to your site. You can write your scripts to route calls to both the local skillset (QUEUE TO SKILLSET) and the network skillset (QUEUE TO NETWORK SKILLSET).

If Local Node Inclusion (LNI) is defined for the network skillset, then only the QUEUE TO NETWORK SKILLSET command is required. Contacts are automatically queued to both the local and target sites. Consider the order in which contacts are queued to local and network skillsets. Often, it is more efficient to first queue a contact locally, and then wait a specific amount of time before queuing to a network skillset. If a local agent is available and qualified to handle the contact, the contact is immediately presented to the agent.

The Local Node Inclusion (LNI) option is available for the Agent Reservation methods when defining the network skillset as shown in the following table.

Agent Reservation method	Description
AVERAGE SPEED ANSWER	The server waits a configurable amount of time. During this time, it examines the agent reservation notifications received from the other sites to identify the reserved agents with the highest priority for the skillset, and determine which of these agents is at the site with the

Agent Reservation method	Description
	fastest average speed of answer for the skillset. It then routes the call to the site with the fastest average speed of answer. This method distributes calls for a given skillset to the most efficient sites in the network.
LONGEST IDLE AGENT	The server waits a configurable amount of time. During this time, it examines the agent reservation notifications received from the other sites to identify the reserved agents with the highest priority for the skillset, and to determine which of these high-priority agents is idle for the longest time. It then routes the call to the site with the longest idle agent. This method helps distribute call load across the network.
FIRST BACK	The server routes the call to the first site from which it receives an agent reservation notification. Because the server does not wait to hear from slower sites, but queues calls to the site that responds most quickly, calls are answered more quickly with this method.

Agent priority takes precedence over agent idle time in a network routing decision.

Sending calls to a remote site

Three commands are available to send calls to a remote site:

- ROUTE CALL on page 316
- QUEUE TO NETWORK SKILLSET on page 329
- QUEUE TO NACD on page 333

Routing contacts after an NSBR command failure

If a network routing command fails, you can ensure the call is routed to the appropriate site by using NACD routing, which must be configured at the switch. For information about how to configure NACD routing, see *Avaya Aura® Contact Center Configuration – Avaya CS 1000 Integration* (NN44400-512).

A network routing command can fail for the following reasons:

- The NCC server cannot communicate with the server in Contact Center at the source site.
- The WAN link between the Contact Center Manager Server at the source site and the destination site is not transmitting data.

To determine if NACD routing is needed, use the QUEUED, PRIORITY IN QUEUE, or PRIORITY IN NETWORK QUEUE intrinsic after a QUEUE TO SKILLSET or QUEUE TO NETWORK SKILLSET command. If either of these commands successfully queues the

contact, the intrinsic (QUEUED, PRIORITY IN QUEUE, or PRIORITY IN NETWORK QUEUE) returns a value of True.

Limitations of the network script

The network application handles error conditions and can perform only a limited number of treatments on the contact. This restriction is in effect in case the contact returns to the queue because the reserved agent becomes unreserved after the contact is presented.

The network script handles error conditions and can perform only a limited number of treatments on the call. This restriction is in effect in case the contact returns to the queue because the reserved agent becomes unreserved after the contact is presented.

Subscripts for the network application do not generate contact statistics. Therefore, the network script is the only application from which incoming contact statistics are collected.

The network CDNs process only network contacts, so you must ensure that local contacts do not enter the network CDN. Local contacts that enter the contact center though a network CDN are not handled properly.

Network application restrictions

Do not use the following commands or intrinsics in the network script:

- GIVE BUSY
- GIVE BUSY CONTROLLED
- GIVE OVERFLOW
- GIVE OVERFLOW CONTROLLED
- QUEUE TO NACD
- QUEUE TO NETWORK SKILLSET
- CHANGE PRIORITY IN NACD
- CHANGE PRIORITY IN NETWORK SKILLSET
- REMOVE FROM NACD
- REMOVE FROM NETWORK SKILLSET
- PRIORITY IN NETWORK QUEUE

If a command in this chapter is used in either the network application or any of the subscripts it references, your server can stop responding or contact execution can stop. You must ensure that none of the applications called by the network application (either primary or secondary) include any of the commands listed in this section.

These commands are rejected during script validation. When a contact enters the network and the agent becomes unavailable, the network application is executed. If the network script

executes a subscript that contains any of these commands and the commands run, further application execution stops, the contact remains in a queue, and the contact continues to experience previous treatments.

However, because a contact is networked, it already has ringback. You do not need to use GIVE BUSY or OVERFLOW to treat to these contacts.

For information about these commands in SIP-enabled contact centers, see Contact processing for SIP on page 359.

CHANGE PRIORITY IN NETWORK SKILLSET

Change the priority of a contact in a network skillset. The priority of the contact changes within the network skillsets listed in the command. This command applies only to network skillsets.

Syntax

CHANGE PRIORITY IN NETWORK SKILLSET [<*network_skillset*> | <*network_skillset_list*>] TO PRIORITY <*priority*>

Parameters

<*network_skillset*> or <*network_skillset_list*> The network skillset, or list of network skillsets, to which the contact is queued.

<*priority*> The new priority with which you want the contact queued to the specified network skillset.

Restrictions

The CHANGE PRIORITY IN NETWORK SKILLSET command cannot be the first command in an application.

Do not use this command in the network script (if you purchased the Network Skill-Based Routing feature) or any of its subscripts.

Do not insert the GIVE BUSY or GIVE OVERFLOW command immediately after the QUEUE TO NETWORK SKILLSET command. Currently, this applies only to voice contacts.

Use a WAIT command with at least 4 seconds after a QUEUE TO NETWORK SKILLSET command.

The amount of time given with the WAIT command depends on your system response time. For example, if you use Virtual Network Services, you may require a longer wait time.

Example

In this example, if the age of the call queued to the network skillset (skillset_N1_sk) is greater than 60 seconds, the priority of the contact increases from 3 to 1.

```
IF (AGE OF CALL > 60) THEN
   CHANGE PRIORITY IN NETWORK SKILLSET skillset_N1_sk TO
      PRIORITY 1
END IF
```

QUEUE TO NETWORK SKILLSET

Queue a contact on a network-wide basis to specified skillsets. You can simultaneously queue contacts to multiple skillsets. However, to queue a contact to local and network skillsets, you must use the QUEUE TO SKILLSET command for the local skillsets, unless you enable the include source node option for the network skillset.

Syntax

QUEUE TO NETWORK SKILLSET [<*network_skillset*> | <*network_skillset_list*>] {WITH PRIORITY <*priority*>}

Optional

The WITH PRIORITY segment is optional.

Parameters

<*network_skillset*> or <*network_skillset_list*> The network skillset, or list of network skillsets, to which you want the contact queued.

<*priority*> The priority with which you want the contact queued to the specified network skillset.

WITH PRIORITY option

You can specify the priority with which the contact is queued. You can assign a priority of 1 to 6, with 1 being the highest priority and 6 being the lowest. If you do not specify a priority, the contact is queued with the default priority of 6. Contacts with high priorities are presented to the agents before contacts with lower priorities.

Restrictions

Do not insert the GIVE BUSY or GIVE OVERFLOW command immediately after the QUEUE TO NETWORK SKILLSET command. This restriction applies only to voice contacts.

Use a WAIT command with at least 4 seconds after a QUEUE TO NETWORK SKILLSET command. The amount of time given with the WAIT command depends on your system response time. For example, if you are using Virtual Network Services, you may require a longer wait time.

Do not use this command in the network script (if you purchased the NSBR feature) or any of its subscripts.

Queuing to a list of network skillsets

If you specify a list of skillsets with the QUEUE TO NETWORK SKILLSET command, contacts are always randomly queued to the skillsets. You can configure the number of target nodes, from 3 to 20.

Example

In this example, if the local skillset (local_skill_main_sk) is out of service, the contact is queued to the network skillset (skillset_N1_sk) and to a local backup skillset.

```
IF OUT OF SERVICE local_skill_main_sk THEN
   QUEUE TO NETWORK SKILLSET skillset_N1_sk
   WAIT 4
   QUEUE TO SKILLSET backup_sk
   WAIT 4
END IF
```

REMOVE FROM NETWORK SKILLSET

Remove a queued contact from a network skillset or skillsets. This command applies only to network skillsets.

Syntax

REMOVE FROM NETWORK SKILLSET [<*network_skillset*> | <*network_skillset_list*>]

Parameters

<*network_skillset*> or <*network_skillset_list*> The network skillset, or list of network skillsets, from which you want the contact removed. You can also replace this parameter with an intrinsic returning a skillset.

Restrictions

The REMOVE FROM NETWORK SKILLSET command cannot be the first command in an application.

Do not use this command in the network script (if you purchased the Network Skill-Based Routing feature) or in subscripts.

Examples

Example 1 In this example, the contact is queued to the network skillset (skillset_N1_sk). If the contact is not answered within 60 seconds, the contact is queued to a local backup skillset.

```
QUEUE TO NETWORK SKILLSET skillset_N1_sk
WAIT 4GIVE RAN agents_busy_ran_gv
GIVE MUSIC pop_music_gv
SECTION Check_Age
   WAIT 20
   IF (AGE OF CALL > 60) THEN
      EXECUTE Get_Out
   END IF
   EXECUTE Check_Age
SECTION Get_Out
   REMOVE FROM NETWORK SKILLSET skillset_N1_sk
   QUEUE TO SKILLSET backup_sk
   WAIT 2
SECTION WaitLoop
```

Example 2 This example shows a network script when the Local Node Inclusion (LNI) option is selected. The Local Node Inclusion option enabled for network skillset at the source site. The contact is queued to the local and remote sites. After 6 seconds, check if the contact is queued anywhere and if not send the contact to an emergency number. If the contact is queued somewhere, the caller hears the assigned music source. Every 30 seconds, check the condition of the contact to see if it is still queued, if not then perform the same check as when the contact entered the application.

If the contact is queued somewhere, check the priority of the contact in the local skillset. If it has no priority in the local skillset, check the condition of the skillset, if it is staffed. If the local skillset is staffed then requeue the contact locally. If the contact has a priority in a local skillset, ignore the local priority check and verify the contact priority in the network. If the contact is not queued to the network, the contact is requeued. If the contact is queued to the network the network priority check is ignored and repeat the wait check every 30 seconds, until the contact is answered or the caller abandons.

```
QUEUE TO NETWORK SKILLSET skillset_cv WITH PRIORITY prio_cv
WAIT 6
IF NOT QUEUED THEN
   ROUTE CALL emerg_dn
END IF
GIVE MUSIC music_cv
SECTION WaitLoop
   WAIT wait_timer30
   IF NOT QUEUED THEN
      QUEUE TO NETWORK SKILLSET skillset_cv WITH
         PRIORITY prio_cv
      WAIT 6
      IF NOT QUEUED THEN
         ROUTE CALL emerg_dn
      END IF
   END IF
   IF (PRIORITY IN QUEUE skillset_cv = 0)THEN
      IF NOT OUT OF SERVICE skillset_cv THEN
         QUEUE TO SKILLSET skillset_cv WITH
            PRIORITY prio_cv
      WAIT 2
      END IF
   IF (PRIORITY IN NETWORK QUEUE skillset_cv = 0) THEN
      QUEUE TO NETWORK SKILLSET skillset_cv WITH
         PRIORITY prio_cv
      WAIT 2
   END IF
EXECUTE WaitLoop
```

CHANGE PRIORITY IN NACD

Change the priority of a voice contact in an NACD ACD-DN and in the queues at the target NACD sites.

Syntax

CHANGE PRIORITY IN NACD <*acd_dn*> TO PRIORITY <*priority*>

Parameters

<*acd_dn*> The number of the ACD-DN to which the contact is queued. This parameter must be a single ACD-DN or, if you use a variable, an ACD type variable.

<*priority*> The new priority with which you want the contact queued to the specified ACD-DN.

Restrictions

The CHANGE PRIORITY IN NACD command cannot be the first command in a voice application.

Do not use this command in the network script (if you purchased the NSBR feature) or any subscripts.

Example

In this example, if the age of the call queued to the NACD queue is greater than 60 seconds, the priority of the call increases from 3 to 1.

```
IF (AGE OF CALL > 60) THEN
   CHANGE PRIORITY IN NACD nacd_queue TO PRIORITY 1
END IF
```

Limitations of this command

To use a network ACD, you configure a local ACD-DN on the switch with up to 20 target switches. Each target has an associated timer. For example, your local ACD-DN can be configured as follows.

Local ACD-DN	Target switch	Timer (seconds)
4500	63434500	0
	63444500	16
	63454500	20

In this example, with the QUEUE TO NACD command, the voice contact is queued to 63434500 immediately, to 63444500 after 16 seconds, and to 63454500 after 20 seconds. When you issue the CHANGE PRIORITY IN NACD command, the priority changes only for those target sites to which the contact is not yet queued.

Example

```
QUEUE TO NACD nacd_queue WITH PRIORITY 4|
WAIT 10
CHANGE PRIORITY IN NACD nacd_queue TO PRIORITY 2
WAIT 2
```

In this example, the contact is queued to the local ACD-DN, which has three target sites. After 10 seconds, the command to change the priority of the contact in the NACD queues executes. Because of the timer, the contact is already queued to the first target site. Therefore, the priority of the contact at this site remains the same. The contact is not yet queued to the second and third sites, however, so the priority for these target sites does change.

Local AD-DN	Targt switch	Tmer	Rsult
4500	63434500	0	Priority 4
	63444500	16	Priority 2
	63454500	20	Priority 2

QUEUE TO NACD

If the network routing command fails, use the QUEUE TO NACD command to ensure the contact is routed to the appropriate site.

Use the QUEUE TO NACD command to queue a contact to one or more remote target ACD-DNs that are defined in the NACD routing table for the specified ACD-DN in the command.

With this command, contacts are queued on the remote switch rather than on Contact Center. You can use Contact Center to:

- remove contacts from the queue
- add contacts to the queue
- change the priority of the NACD contact
- provide treatments such as music, RAN, IVR, and broadcast announcements for contacts

Syntax

QUEUE TO NACD <*acd_dn*> {WITH PRIORITY <*priority*>}

Optional

The WITH PRIORITY segment is optional.

Parameters

<*acd_dn*> The number of the ACD-DN to which you want the contact queued. This parameter must be a single ACD-DN or, if you use a variable, an ACD type variable.

<*priority*> The priority with which you want the contact queued to the specified ACD-DN.

WITH PRIORITY option

You can assign the contact a priority from 1 to 4, with 1 being the highest priority and 4 being the lowest. Although a priority of 5 or 6 can be specified, the actual priority within the NACD queues defaults to 4 (the lowest possible for NACD).

Restrictions

The WITH PRIORITY option applies to voice contacts only.

Do not insert the GIVE BUSY or GIVE OVERFLOW command immediately after the QUEUE TO NACD command.

Use a WAIT command with at least 2 seconds after a QUEUE TO NACD command.

Do not use this command in the network script (if you purchased the Network Skill-Based Routing feature) or any subscripts.

Example

In this example, a contact is queued locally to the sales skillset and to an NACD.

```
QUEUE TO SKILLSET sales_sk
WAIT 6
QUEUE TO NACD nacd_queue
```

Add a brief pause between the QUEUE TO SKILLSET and QUEUE TO NACD commands. This pause can prevent an error if an agent from the local skillset becomes unavailable before the contact is answered. If this happens, the QUEUE TO NACD command fails because Contact Center cannot queue to NACD while presenting a contact to a local agent.

REMOVE FROM NACD

Remove a contact from the ACD-DN to which the contact is queued. The contact is then removed from all targets in the NACD table for that ACD-DN.

Syntax

REMOVE FROM NACD <*acd_dn*>

Parameters

<*acd_dn*> The number of the ACD-DN from which you want the contact removed. This parameter must be a single ACD-DN or, if you use a variable, it must be an ACD type variable.

Restrictions

The REMOVE FROM NACD command cannot be the first command in a voice script.

Do not use this command in the network script (if you purchased the NSBR feature) or any of its subscripts.

Example

A contact is queued to an NACD queue when it first arrives. If, after 60 seconds, the contact is not answered, it is removed from the NACD and queued to local sales and service skillsets.

```
QUEUE TO NACD nacd_queue
WAIT 2
GIVE MUSIC pop_music_gv
SECTION Check_Age
   WAIT 20
   IF (AGE OF CALL > 60) THEN
      EXECUTE Get_Out
   END IF
   EXECUTE Check_Age
SECTION Get_Out
   REMOVE FROM NACD nacd_queue
```

```
QUEUE TO SKILLSET sales_sk, service_sk
WAIT 2
```

Chapter 28: Voice processing for Avaya Communication Server 1000

Use voice processing commands to communicate with voice contacts using recorded announcements (RAN), interactive voice response (IVR) systems, broadcast announcements, and voice prompts.

You can play messages to voice contacts or, if your contact center has an IVR system, you can collect information from the voice contact so agents can service the contact more efficiently.

Choosing the most efficient commands

This section contains information to help you decide which voice processing commands are the most efficient for your contact center.

Typical uses of voice processing commands

The following sections describe the typical uses of different types of voice processing commands.

GIVE RAN Use the GIVE RAN command in the following situations:

- when you want to play a message to a voice contact
- when your contact center has RAN equipment from a previous contact center installation
- if RAN equipment is less expensive than a voice processing system and other voice processing functions are not required

GIVE IVR Use the GIVE IVR command in the following situations:

- when you use a third-party voice processing system for announcements or voice menus (voice contact interaction)
- when you use Avaya Self Service or any third-party application with Avaya CallPilot® for voice menus (voice contact interaction)

GIVE CONTROLLED BROADCAST ANNOUNCEMENT Use the GIVE CONTROLLED BROADCAST ANNOUNCEMENT command in the following situations:

- when you use Contact Center Voice Services on Avaya CallPilot® and ACCESS to simultaneously play the same message to multiple contacts
- because controlled broadcast announcements use ports more efficiently than RANs, and because it is easier to control the played message

The GIVE CONTROLLED BROADCAST feature is not supported on a CS 1000E platform. For more information, see Avaya Product Bulletin P-2006-0312_Global.

Voice sessions Use the OPEN VOICE SESSION and END VOICE SESSION commands when you use Contact Center Voice Services on Avaya CallPilot® to provide:

- custom messages (for example, to play a call expected wait time)
- call interaction (for example, digit collection)

Contact center configuration

Your contact center configuration and resources determine which commands you can use.

If you have	You can use
Contact Center Voice Services on Avaya CallPilot®	GIVE IVR GIVE CONTROLLED BROADCAST ANNOUNCEMENT OPEN/END VOICE SESSION
a third-party voice processing system	GIVE IVR

Playing messages

If callers only listen to the message, as opposed to making selections, you can use the following commands:

- GIVE RAN
- GIVE IVR
- GIVE CONTROLLED BROADCAST ANNOUNCEMENT
- OPEN/END VOICE SESSION with PLAY PROMPT

Interactive voice sessions

If callers enter information during the voice session, you can use the following commands:

- OPEN/END VOICE SESSION with COLLECT DIGITS
- GIVE IVR with Host Data Exchange commands

For information about Host Data Exchange commands, Host Data Exchange commands on page 353.

Single connection or broadcast

If all voice contacts, or many contacts, must hear the same announcement, use any of the listen only commands. However, port use is more efficient if you use the BROADCAST command rather than the one-call-to-one-port commands. With BROADCAST, you can sustain higher contact rates with fewer ports.

Use the one-contact-per-port commands to play custom messages to contacts (for example, EXPECTED WAIT TIME) if input is collected, or if you have a third-party voice processing system.

The GIVE CONTROLLED BROADCAST ANNOUNCEMENT command connects multiple contacts for each port.

The following commands connect one contact for each port:

- GIVE IVR
- OPEN/END VOICE SESSION

Start or Stop and CONTINUOUS modes

Start/Stop operation means the caller is guaranteed to hear the message from the beginning to the end because callers connect to the system at the beginning of the message.

Continuous operation means the message plays repeatedly. When a contact enters the system, the caller hears the message from the point it plays at the time. That is, the caller can hear the message from middle to end and then hear the beginning of the message when it starts over.

These commands can operate in Start/Stop mode:

- GIVE RAN
- GIVE IVR
- GIVE CONTROLLED BROADCAST ANNOUNCEMENT
- OPEN/END VOICE SESSION

The GIVE CONTROLLED BROADCAST ANNOUNCEMENT command also supports the continuous mode.

The CONTROLLED BROADCAST CONTINUOUS command connects the call immediately upon arrival and continues the script only after the caller hears one full cycle of the message.

Impact of commands

Some commands use more contact center system resources than others. One parameter that determines the impact of the different voice processing commands on the system is the amount of messaging necessary to run the statements. Commands that interact with Avaya CallPilot® use more system CPU resources than those that do not:

- Lower-usage commands are GIVE RAN and GIVE IVR.
- Higher-usage commands are GIVE CONTROLLED BROADCAST ANNOUNCEMENT and OPEN/END VOICE SESSION.

Cluster the voice segments in as few PLAY PROMPT statements as possible (within the maximum allowed). Clustering voice segments uses fewer resources than using multiple PLAY PROMPT statements, and achieves the same effect.

Examples

The following examples demonstrate how to properly use the PLAY PROMPT command.

Best practice:

PLAY PROMPT NUMBERBYDIGIT 1234

Not recommended:

PLAY PROMPT number 1 number 2 number 3 number 4

Avoid:

PLAY PROMPT number 1

PLAY PROMPT number 2

PLAY PROMPT number 3

PLAY PROMPT number 4

GIVE IVR

Use the GIVE IVR (interactive voice response) command to deliver a call to a voice messaging system, such as Avaya CallPilot®, or Avaya IVR. This command is used only when Contact

Center does not control the voice session. No statistics are pegged against activities that take place while the call is in the voice messaging system.

The voice messaging system voice ports must be configured as agents for the queue that is specified through the parameter ivr_dn.

Execution of the application is suspended until the IVR session is complete.

The IVR session is not interrupted by an agent becoming idle unless the INTERRUPTIBLE keyword is used.

Syntax

GIVE IVR {INTERRUPTIBLE} <*ivr_dn*> {WITH PRIORITY <*priority*>} {WITH TREATMENT <*vsdn*>}

Parameters

<*ivr_dn*> The switch ACD-DN in which you want the call placed.

<*priority*> The priority with which you want the call queued to the voice services queue.

<*vsdn*>(service)/<*vsdn*> The voice service DN for the treatment that you want to give the call.

INTERRUPTIBLE keyword

When you write your application, you can queue a contact first and then use the GIVE IVR command to deliver the contact to a voice messaging system. If you use the INTERRUPTIBLE keyword, the IVR session is interrupted when an agent becomes available to take the contact. If you do not use the INTERRUPTIBLE keyword, the voice session continues until it is complete, even if an agent becomes available.

The contact holds its position in all queues it was in prior to the GIVE IVR command.

Do not use the INTERRUPTIBLE keyword for IVR sessions that include menus. If you do this, an agent becoming available can interrupt the contact while the caller is listening to the menu options.

Longest waiting calls and INTERRUPTIBLE IVR sessions Use the INTERRUPTIBLE keyword to ensure that voice contacts waiting for the longest amount of time are presented to the first available agent. If you do not use the INTERRUPTIBLE keyword, the longest waiting contact may be held in an IVR session, while contacts that enter the system later are presented to available agents.

WITH PRIORITY option

You can specify the priority with which the contact is queued. You can assign a priority of 1 to 4, with 1 being the highest priority and 4 being the lowest. A contact is assigned a priority of 4 by default.

WITH TREATMENT option

You can specify the treatment that a contact receives. You can apply different voice services, such as menus and announcements, to the call depending on the treatment specified. To configure voice service treatments, use Avaya CallPilot®. If you do not specify a treatment, the system applies the default treatment DN defined for the IVR ACD DN.

Restrictions

Do not insert the GIVE BUSY or GIVE OVERFLOW command immediately after the GIVE IVR command.

Do not use the WITH TREATMENT option with the GIVE CONTROLLED BROADCAST ANNOUNCEMENT and OPEN VOICE SESSION commands. Use of this option with these commands can cause the switch to lose voice ports.

Example

In this example, the contact is queued to the sales skillset and then receives IVR treatment. The IVR session is interrupted if an agent becomes available to answer the contact before the IVR session is complete. Otherwise, the caller hears music while waiting in a queue.

```
QUEUE TO SKILLSET sales_sk
WAIT 2
GIVE IVR INTERRUPTIBLE ivr_dn WITH TREATMENT agents_busy
GIVE MUSIC pop_music_gv
```

COLLECT DIGITS

Collect digits from the caller through voice ports. Script execution is suspended until the digit collection is complete. If the call is queued before the COLLECT DIGITS command is executed, digit collection is interrupted when an agent becomes available.

An enhanced GIVE IVR command replaces the COLLECT DIGITS command in SIP-enabled contact centers. For information about the GIVE IVR command in SIP-enabled contact centers, see Contact processing for SIP on page 359.

The maximum number of digits that can be collected is 16. You must also define a DN type call variable in which to store the digits.

Important:

This command is used only within an OPEN VOICE SESSION and END VOICE SESSION section.

Syntax

COLLECT <*max_num_of_digits*> DIGITS INTO <*call_digit_var*>

{NO TYPE AHEAD}

{INTER DIGIT TIMER <*timer*>}

{WITH TERMINATING CHARACTER <*terminating_char*>}

Parameters

<*max_num_of_digits*> The maximum number of digits, up to 16, that you want to collect. The script continues automatically when the specified number of digits is collected. The terminating character, if you use it, counts as a digit. For example, if you collect 8 digits from the call, you must specify the maximum number of digits as 9.

<*call_digit_var*> The call variable in which you want to store the collected di<timer> The maximum time, in seconds, that you want the script to wait between each digit collected. This variable must be a DN type call variable.

<*terminating_char*> The number (from 0–9) or symbol (* or #) that, when collected, terminates the digit collection before the maximum number of digits is reached.

The symbols asterisk (*) and pound (#) are terminating characters only. They are not used for any other purpose.

NOT TYPE AHEAD keyword

Use the NO TYPE AHEAD keyword to tell the system to discard digits pressed by the caller before this command is executed. Use this option, for example, if the caller entered digits during a preceding PLAY PROMPT command and you want to discard them.

INTER DIGIT TIMER command

By default, the voice port waits until the maximum number of digits is collected or until the INTER DIGIT TIMER command times out. If you specify no timer, the command uses the default value of 10 seconds.

WITH TERMINATING CHARACTER option

You can specify a terminating character to end the digit collection even if the maximum number of digits is collected. The terminating character is not included in the variable with the digits.

Ending digit collection

Digit collection ends under the following conditions:

- The INTER DIGIT TIMER expires.
- The terminating character is received.
- The maximum number of possible digits is reached.
- An agent becomes available to take the call (if the call is already queued before digit collection began).

OPEN VOICE SESSION and END VOICE SESSION

Obtain control over a voice port on the voice processing system. After a voice port is secure, the call is connected to the port automatically and the actions specified between the OPEN VOICE SESSION and END VOICE SESSION commands are performed.

You need Contact Center Voice Services on Avaya CallPilot® to use the OPEN VOICE SESSION and END VOICE SESSION commands.

Syntax

OPEN VOICE SESSION {<*access_ivrdn*>}

<*statement1*>

<statement2>

...

END VOICE SESSION

Parameters

<access_ivrdn> The ACCESS IVR DN configured on the Avaya CallPilot® system.

Important:

Avaya recommends that you use the default ACCESS IVR DN configured in the Global Settings property page for the server, rather than specifying an ACCESS IVR DN in the script. From the Contact Center Manager Administration Launchpad, choose Configuration, <server name>, Global Settings.

<statement1>, *<statement2>*, … The action that you want the server to perform on the call.

Restriction

This command applies to voice contacts only.

Possible actions

You can use any of the following commands inside the OPEN VOICE SESSION and END VOICE SESSION commands:

- ASSIGN TO
- ROUTE CALL
- IF-THEN-END-IF
- IF-THEN-ELSE-END-IF
- DISCONNECT
- LOG
- PLAY PROMPT
- COLLECT DIGITS
- SEND INFO
- SEND REQUEST
- GET RESPONSE
- WHERE-EQUALS

Example

Example 1 In this example, a voice session begins in which the caller receives a welcome message and is informed that no agents are currently available to answer the contact. The caller is asked to respond to a menu option by entering a digit from the phone. The digit is collected into a variable named menu_choice_cv.

```
OPEN VOICE SESSION
   PLAY PROMPT VOICE SEGMENT welcome_vs
   VOICE SEGMENT all_agents_busy_vs
   VOICE SEGMENT leave_message_vs
```

```
   COLLECT 1 DIGITS INTO menu_choice_cv
END VOICE SESSION
```

Example 2 In this example, a voice session begins in which the caller receives a menu option in the language of their choice. If they do not select options 1 or 2, they are prompted to enter their choice again. The voice session ends and, based on the collected data, the appropriate language and skillset is assigned to the call.

```
OPEN VOICE SESSION
   PLAY PROMPT WITH LANGUAGE lang_cv VOICE SEGMENT
   gen_info_main_menu_vs
   COLLECT 1 DIGITS INTO menu_choice_cv INTER DIGIT TIMER 2
   IF menu_choice_cv <> 1 OR menu_choice_cv <> 2 THEN
      ASSIGN 11 TO menu_choice_cv
      PLAY PROMPT WITH LANGUAGE lang_cv VOICE SEGMENT
      gen_info_main_menu_vs
      COLLECT 1 DIGITS INTO menu_choice_cv INTER
      DIGIT TIMER 2
   END IF
END VOICE SESSION
WHERE menu_choice_cv EQUALS
   VALUE 1: ASSIGN French TO lang_cv
      ASSIGN Gen_Info_Fr TO skillset_cv
   VALUE 2:    ASSIGN English TO lang_cv
      ASSIGN Gen_Info_En TO skillset_cv
   DEFAULT: ASSIGN French TO lang_cv
      ASSIGN Gen_Info_Fr TO skillset_cv
END WHERE
QUEUE TO SKILLSET skillset_cv
```

Syntax when using the GIVE CONTROLLED BROADCAST ANNOUNCEMENT and OPEN VOICE SESSION commands

The following examples demonstrate the differences between using the GIVE CONTROLLED BROADCAST ANNOUNCEMENT and OPEN VOICE SESSION commands.

The syntax for GIVE CONTROLLED BROADCAST ANNOUNCEMENT is:

```
GIVE CONTROLLED BROADCAST ANNOUNCEMENT {access acddn}
PLAY PROMPT VOICE SEGMENT x
VOICE SEGMENT y
```

The syntax for OPEN VOICE SESSION is:

```
OPEN VOICE SESSION {access acddn}
PLAY PROMPT VOICE SEGMENT x
END VOICE SESSION
```

The GIVE CONTROLLED BROADCAST feature is not supported on a CS 1000E platform. For more information, see Avaya Product Bulletin P-2006-0312_Global.

PLAY PROMPT

Play a voice prompt or to speak a number to the caller through a voice port. Voice prompts can be one of the following:

- a list of voice segments identified by voice segment variables
- a list of numbers
- numbers interspersed with voice segments

If the call is queued, the voice prompt is interrupted when an agent is available to take the call.

You need Contact Center Voice Services on Avaya CallPilot® to use the PLAY PROMPT command.

Important:

Use this command only within an OPEN VOICE SESSION and END VOICE SESSION command. The PLAY PROMPT command is similar to the PLAY PROMPT command used as part of the GIVE CONTROLLED BROADCAST ANNOUNCEMENT command, but it includes different optional parameters.

Syntax

PLAY PROMPT

{WITH LANGUAGE <*language*>}

{NO TYPE AHEAD}

{[NUMBER | NUMBERBYDIGIT] <*number*>}

{VOICE SEGMENT <*voice_seg*>}

You must include at least one (to a maximum of 50) NUMBER, NUMBERBYDIGIT, or VOICE SEGMENT clause with this command. You can place these commands in any order.

Parameters

<*language*> The language in which you want the prompt played.

<*number*> The number that you want played.

<*voice_seg*> The voice segment that you want played. The maximum length of a voice segment is 2 minutes (120 seconds) for Avaya CallPilot®.

WITH LANGUAGE option

You can record voice segments in various languages and then use the WITH LANGUAGE option to specify in which language you want the voice segment played. For example, if your contact center provides service in several languages, you can record three versions of your main voice menu: English, Spanish, and French. If you do not use the WITH LANGUAGE option, the system uses the default language.

Important:

If no other language is specified in the script, the default language is played. The default language is system-dependent based on regional settings of the operating system on the server. If the system default language is other than English and you want to play any of the system defined voice segments, such as c_zero_gv which has a predefined value English:file:1:1, you must specify in the script PLAY PROMPT WITH LANGUAGE English.

NO TYPE AHEAD keyword

By default, the voice prompt is interrupted if the caller presses a digit on the phone keypad. To configure the prompt so that it is not interrupted by the caller entering digits, include the NO TYPE AHEAD option.

Numbers

You can record numbers as voice segments and then play them to calls. Contact Center comes with number variables predefined, but you must record the voice segments. For Contact Center Voice Services on Avaya CallPilot®, you use Application Builder to record voice segments. For Contact Center Voice Services, you use the Voice Prompt Editor.

For more information about Application Builder, see the Avaya CallPilot® Application Builder Guide. For more information about Voice Prompt Editor, see *Avaya Aura® Contact Center Manager Administration – Client Administration* (NN44400-611).

A number can be an integer constant or a script variable of Integer or DN type.

You can play numbers to calls as a full number or as individual digits. For example, the number 1234 can be spoken as either “one thousand two hundred thirty-four” or “one two three four”. To indicate how you want the number spoken, use either the NUMBER keyword or the NUMBERBYDIGIT keyword.

NUMBER keyword If you want a number to be spoken as full words, it must be preceded by the NUMBER keyword. In this case, the number 1234 is spoken as *one thousand two hundred thirty-four*.

Note:

DN variables containing leading zeros are not supported when using the NUMBER keyword.

NUMBERBYDIGIT keyword If you want numbers to be spoken as digits, they must be preceded by the NUMBERBYDIGIT keyword. In this case, the number 1234 is spoken as *one two three four*.

Note:

DN variables containing leading zeros are supported when using the NUMBERBYDIGIT keyword.

Voice segments

Voice segments contain recorded voice data that you can play to calls. To do so, you must record the voice segment, and then create a voice segment variable. You can then use that variable in your script to identify the voice segment that you want the caller to hear.

For Contact Center Voice Services on Avaya CallPilot®, you use Application Builder to create, record, and manage voice segments. For Contact Center Voice Services, you use the Voice Prompt Editor.

In Avaya CallPilot®, the maximum length of a voice segment is 2 minutes (120 seconds). Also, ensure you do not delete a voice segment as all voice prompts shuffle up, and their file numbers change. Avaya CallPilot® does not have this issue.

Example

This example demonstrates a bank balance lookup using the PLAY PROMPT command. Full IVR system are better equipped for this type of application (rather than a voice session) because IVR systems are specially designed for this type of work.

Use voice sessions in the contact center to make routing and queuing decisions, rather than to replace IVR.

```
OPEN VOICE SESSION
   PLAY PROMPT VOICE SEGMENT enter_account_number_vs
   COLLECT 10 DIGITS INTO account_num_cv INTER DIGIT TIMER 5
   WITH TERMINATING CHARACTER #
   SEND REQUEST appl_id account_num
   GET RESPONSE appl_id balance_cv
   PLAY PROMPT VOICE SEGMENT acct_balance1_vs
   NUMBERBYDIGIT account_num_cv
   VOICE SEGMENT is_vsNUMBER balance_cv
   VOICE SEGMENT dollars_please_hold_vs
END VOICE SESSION
```

GIVE CONTROLLED BROADCAST ANNOUNCEMENT

Broadcast an announcement to multiple callers at the same time using a single voice port (up to 50 callers). If an agent becomes available to take the contact while the broadcast message is playing, the message is interrupted.

You need Contact Center Voice Services on Avaya CallPilot® to use the GIVE CONTROLLED BROADCAST ANNOUNCEMENT command.

The GIVE CONTROLLED BROADCAST feature is not supported on a CS 1000E platform. You can use Open Voices sessions with one caller per access channel for small contact centers on CS 1000E instead of GIVE CONTROLLED BROADCAST. For more information, see Avaya Product Bulletin P-2006-0312_Global.

Important:

Ensure that you correctly configure the value of the “Maximum Ports With Queuing For Broadcast” option in the Global Settings window on the client. If you configure this value to be greater than the actual number of ports in the IVR ACDDN, call processing suspends at this statement even if no ports are free for the call. The calls queue until a connection to the port becomes free. However, if you configure this value equal to or less than the number of ports in the IVR ACD-DN, the caller only hears the broadcast if a port with free capacity is available to connect the call (if all ports are used, the call skips this statement and goes to the next statement in the script).

Syntax

GIVE CONTROLLED BROADCAST ANNOUNCEMENT {<*access_ivrdn*>}

{WITH PRIORITY <*priority*>}

PLAY PROMPT

{WITH LANGUAGE <*language*>}

{CONTINUOUS}

{[NUMBER | NUMBERBYDIGIT] <*number*>}

{VOICE SEGMENT <*voice_seg*>}

You must include at least one NUMBER, NUMBERBYDIGIT, or VOICE SEGMENT clause with this command. You can include up to 50 NUMBER, NUMBERBYDIGIT, or VOICE SEGMENT clauses with this command, in any order.

Parameters

<access_ivrdn> The ACCESS IVR DN configured on the Avaya CallPilot® system.

Important:

Avaya recommends that you use the default ACCESS IVR DN configured in the Global Settings property page for the server, rather than specifying an ACCESS IVR DN in the script. From the Contact Center Manager Administration Launchpad, choose Configuration > (server name) > Global Settings.

<*priority*> The priority with which you want the call queued.

<*language*> The language in which you want the broadcast announcement presented.

<*number*> The number that you want broadcast.

<*voice_seg*> The voice segment that you want played to the call. The maximum length of a voice segment is 2 minutes (120 seconds) for Avaya CallPilot®.

The GIVE CONTROLLED BROADCAST ANNOUNCEMENT command must always contain the PLAY PROMPT command (with voice segments and numbers). The other parameters are optional.

<access_ivrdn> parameter

You can specify the ACCESS IVR DN in which the call is placed. If no IVR DN is specified, the default, which is configured using Contact Center Manager Administration, is used.

WITH PRIORITY option

You can specify the priority with which the contact is queued. You can assign a priority of 1 to 4, with 1 being the highest priority and 4 being the lowest. A contact is assigned a priority of 4 by default.

PLAY PROMPT option

For the PLAY PROMPT option parameters, specify a list of voice segments identified by voice segment variables, a list of numbers, or numbers interspersed with voice segments.

CONTINUOUS option

By default, broadcast announcements always start from the beginning (that is, in Start or Stop mode). As an option, announcements can be given in continuous mode, meaning the announcement is played continuously. Calls are connected to voice ports as they enter the contact center; therefore, depending on the arrival time, the contacts may or may not start at the beginning of the announcement. Every contact hears one complete play of the announcement.

WITH LANGUAGE option

You can record voice segments in various languages and then use the WITH LANGUAGE option to specify in which language you want the voice segment played. For example, if your contact center provides service in several languages, you can record three versions of your main voice menu: English, Spanish, and French. If you do not use the WITH LANGUAGE option, the system uses the default language.

Important:

If no other language is specified in the script, the default language is played. The default language is system-dependent based on regional settings of the operating system on the server. If the system default language is other than English and you want to play any of the system defined voice segments, such as c_zero_gv which has a predefined value English:file:1:1, you must specify in the script PLAY PROMPT WITH LANGUAGE English.

Restrictions

Do not insert the GIVE BUSY or GIVE OVERFLOW command immediately after the GIVE CONTROLLED BROADCAST ANNOUNCEMENT command.

Numbers

You can record numbers as voice segments and then play them to calls. Contact Center provides several predefined variables, but you must record the voice segments. For Contact Center Voice Services on Avaya CallPilot®, you use Application Builder to record voice segments. For Contact Center Voice Services, you use the Voice Prompt Editor.

A number can be an integer constant or a script variable of Integer or DN type.

You can play numbers to calls as a full number or as individual digits. For example, the number 1234 can be spoken as either *one thousand two hundred thirty-four* or *one two three four*. To indicate how you want the number spoken, use either the NUMBER keyword or the NUMBERBYDIGIT keyword.

NUMBER keyword If you want a number to be spoken as full words, it must be preceded by the NUMBER keyword. In this case, the number 1234 is spoken as *one thousand two hundred thirty-four or one two three four*.

NUMBERBYDIGIT keyword If you want numbers to be spoken as digits, they must be preceded by the NUMBERBYDIGIT keyword. In this case, the number 1234 is spoken as *one two three four*.

Voice segments

Voice segments contain recorded voice data that you can play to calls. To do so, you must record the voice segment and then create a voice segment variable. You can then use that variable in your script to identify the voice segment that you want the caller to hear.

For Contact Center Voice Services on Avaya CallPilot®, you use Application Builder to create, record, and manage voice segments. For Contact Center Voice Services, you use the Voice Prompt Editor.

The maximum length of a voice segment is 2 minutes (120 seconds) for Avaya CallPilot®.

Example

In this example, the script checks whether the call CLID is included in the VIP_list_gv variable. If it is, the caller hears a special broadcast announcement. Otherwise, the caller hears a general announcement.

```
IF (CLID = VIP_list_gv) THEN
   GIVE CONTROLLED BROADCAST ANNOUNCEMENT
   PLAY PROMPT VOICE SEGMENT gold_card_welcome_vs
ELSE
   GIVE CONTROLLED BROADCAST ANNOUNCEMENT
   PLAY PROMPT VOICE SEGMENT general_welcome_vs
END IF
```

WITH LANGUAGE EXAMPLE

This script segment determines the language and skillset based on the DNIS information received from the call and assigns the appropriate value to call variables.

```
WHERE DNIS EQUALS
   VALUE 1215:ASSIGN English TO lang_pref_cv
      ASSIGN Sales_English TO pref_skillset_cv
   VALUE 1216:ASSIGN Japanese TO lang_pref_cv
      ASSIGN Sales_Japanese TO pref_skillset_cv
   VALUE 1217:ASSIGN German TO lang_pref_cv
      ASSIGN Sales_German TO pref_skillset_cv
END WHERE
QUEUE TO SKILLSET pref_skillset_cv
WAIT 2
GIVE CONTROLLED BROADCAST ANNOUNCEMENT
PLAY PROMPT WITH LANGUAGE lang_pref_cv
VOICE SEGMENT welcome
GIVE MUSIC general_music
...
```

Chapter 29: Host Data Exchange commands

Contact Center provides a data exchange interface for third-party applications (or hosts) to send data to and receive data from an active script. The third-party application programming interface is referred to as the Host Data Exchange (HDX) API.

The commands used to interact with the third-party application are referred to as Host Data Exchange (HDX) commands. HDX commands process both voice and multimedia contacts. Typically, a host application queries data in a third-party database based on data available in the script—such as CallID or DNIS—and returns data to the script to enable more intelligent call routing. The HDX commands operate in one of two modes:

- Data is sent from the script to the host without waiting for a response.
- Data is sent to the host and the script expects a response from the host. The script waits for the response. In this mode, the potential exists for degradation of contact center operation due to the performance of the host application.

When using HDX commands, you must consider traffic engineering. Every contact accessing a script containing HDX commands is affected by the following items.

- Application response time—A slow third-party application response time can slow down all contact processing in Contact Center.
- Application design—Databases may need special indexing and optimization to handle a high number of requests at the same time. For example, if the application can handle only one request at a time, and each request takes 4 seconds to process, the script can handle a maximum of 900 calls per hour.
- LAN interference—LAN interference includes backup and restore procedures and large file transfers.

HDX commands pass data from the script to the host and receive data from the host to the script. Data passed from the script to the host can be an intrinsic or a variable, or both. Both call and global variables can be sent to the host; however, not all intrinsics and data types are supported as HDX command parameters. Data received by the script from the host is received only into call variables. Using invalid data types results in script compilation failure. The provider ID can be a call or global variable of type Integer. The following variable types are supported as parameters in HDX commands:

- Integer
- String
- DN
- CLID

- DNIS
- Agent ID
- Skillset
- CDN
- ACD
- LOC
- NPA
- NPANXX
- NXX

Internally, all data types in the HDX are strings. When passed from the script to the HDX, non string types are converted to a string representation. For example, when a skillset call variable assigned in the script with the skillset SALES is passed to the HDX, it is represented in the HDX as the string SALES rather than the Skillset Name. When the script receives a skillset from a host application, the host must pass it as a valid string representation of the skillset. The string representation of the skillset is converted back to the skillset for use with the script for queuing.

Every call that enters Contact Center has a call ID associated with it. The call ID is sent automatically with every SEND INFO or SEND REQUEST command. You can use the call ID to show information about the call (such as skillset or call-entered data) in a screen pop on the agents' desktops.

SEND INFO

Use the SEND INFO command to send data to and start a third-party application, such as a screen-pop application. You must use the SEND REQUEST and GET RESPONSE commands to request and receive information.

Syntax

SEND INFO <*provider_ID*> [<*variable*> | <*list_of_variables*>]

Parameters

<*provider_ID*> The unique identifier integer type variable that the HDX application uses to register with Contact Center. Replace this parameter with an integer type variable. No two applications at the same site can have the same provider ID. The default provider ID for IVR CTI applications is 1. HDX supports a maximum of 10 provider applications.

<*variable*> or <*list_of_variables*> The variable, or list of variables (up to 10), that contains the data you want to send to the third-party application. Lists of variables are single variable values separated by commas. If you use more than one variable, separate the variables with a comma.

Restrictions

Do not insert the GIVE BUSY or GIVE OVERFLOW command immediately after the SEND INFO command.

Example

In this example, calls entering CDN number 5553750 initiate a request to a third-party application (such as a database) for the call priority. After the priority is returned, the call is queued to the skillset customer_service_sk with that priority.

If a call enters any other CDN, the CDN number is sent to the third-party application and the call is queued to skillset general_information_sk.

```
WHERE CDN EQUALS
   VALUE 5553750:
   SEND REQUEST app_ID cv_caller_acct_num
   GET RESPONSE app_ID priority_cv
   QUEUE TO SKILLSET customer_service_sk WITH PRIORITY
      priority_cv
   WAIT 2
   DEFAULT:
   SEND INFO app_ID cv_acct_type
   QUEUE TO SKILLSET general_information_sk
   WAIT 2
END WHERE
GIVE RAN agents_busy_ran_gv
GIVE MUSIC soft_music_gv
```

TAPI Call Data attachment

You can also use TAPI call data attachment to attach script and call data to a call, and pass that information to a TAPI server. The first data parameter of the SEND INFO command is interrogated. If the text in the parameter is %TAPI%, then the data in the remaining parameters is sent to TAPI. The parameters of the SEND INFO command are attached to a call using the published TAPI IVR/CallData interface.

For example, the following lines of code send the CLID and the agent ID to the TAPI application.

```
ASSIGN "%TAPI%" TO HAI_STRING_cv
SEND INFO HAI_AppId HAI_STRING_cv, CLID, HAI_AGENTID_cv
```

The SEND INFO command can update a database while attaching data to a call. The first data parameter of the SEND INFO command is interrogated. If the text in the parameter is %TAPI %, the data in the remaining parameters is sent to TAPI. Otherwise, the SEND INFO command is assumed to be for database access and the information in the first parameter is used to identify the SQL statement number.

For example, the SEND INFO can also be used to attach TAPI data to a call based on the assumption that SEND INFO otherwise operates like SEND REQUEST.

```
EVENT HANDLER
    EVENT CALL ABANDON: ASSIGN "abandon" TO string_cv
    SEND INFO HAI_AppId, HAI_SQLNO_cv, CLID, abandon_cv
END HANDLER
```

The first data parameter of the SEND INFO command is interrogated. If the text in the parameter is %TAPI%, then the data in the remaining parameters is sent to TAPI. The parameters of the SEND INFO command are attached to a call using the published TAPI IVR/CallData interface.

Example of script syntax:

```
ASSIGN 5005 TO HAI_AppId
ASSIGN "%TAPI%" TO HAI_STRING_cv
SEND INFO HAI_AppId HAI_STRING_cv, CLID
```

SEND REQUEST

Use the SEND REQUEST command to request specific data from a third-party application. Call and global variables are used in this command to identify the data that you request from the host. These variables are read-only; that is, the host application cannot change the values.

A GET RESPONSE command must always follow the SEND REQUEST command. Only comments can separate these commands in a script.

Syntax

SEND REQUEST <*provider_ID*> [<*variable*> | <*list_of_variables*>]

Parameters

<*provider_ID*> The unique identifier integer type variable that the HDX application uses to register with Contact Center. Replace this parameter with an integer type variable. No two applications at the same site can have the same provider ID. The default provider ID for IVR CTI applications is 1. HDX supports a maximum of 10 provider applications.

<*variable*> or <*list_of_variables*> The variable, or list of variables (up to 10), that contains the data that you request from the third-party application. If you use more than one variable, separate each variable with a comma.

Restrictions

Do not use the GIVE BUSY or GIVE OVERFLOW command immediately after the SEND REQUEST command.

Example

The following portion of script sends a call CLID to a third-party application and then receives the preferred banker for that customer. The first line of the script assigns a default value of 12345 to the variable named personal_banker_cv. Then the SEND REQUEST command sends the customer's CLID to the third-party application.

The GET RESPONSE command retrieves the preferred banker for the customer from the third-party application. The default value of the variable personal_banker_cv is replaced with the value retrieved from the third-party application.

If the value retrieved from the third-party application is returned within 2 seconds, the call is presented to the appropriate agent. Otherwise, the call is queued to the sales skillset.

When you replace a specific agent with an agent variable, you need not modify scripts if the agent leaves the contact center and is replaced by another agent.

```
ASSIGN 12345 TO personal_banker_cv
SEND REQUEST app_ID cv_caller_acct_num
GET RESPONSE app_ID TIMER 2 personal_banker_cv
IF NOT LOGGED OUT AGENT personal_banker_cv THEN
   QUEUE TO AGENT personal_banker_cv
   WAIT 2
ELSE
   QUEUE TO SKILLSET sales_sk
   WAIT 2
END IF
GIVE RAN agent_busy_ran_gv
GIVE MUSIC soft_music_gv
```

GET RESPONSE

Use the GET RESPONSE command to obtain the response from the SEND REQUEST command previously sent to a third-party application. The script validation fails if the command is not preceded by a SEND REQUEST command.

You can specify one or more call variables in which to store the data in the response message.

All variables to be returned from a third-party application using the GET RESPONSE command must have the initial value configured to a default value. The value returned replaces the default value.

Syntax

GET RESPONSE <*provider_ID*> {TIMER <*timer*>} [<*variable*> | <*list_of_variables*>]

Optional

The TIMER segment is optional. The default value is 10 seconds.

Parameters

<*provider_ID*> The unique identifier integer type variable that the HDX application uses to register with Contact Center. Replace this parameter with an integer type variable. No two applications at the same site can have the same provider ID. The default provider ID for IVR CTI applications is 1. HDX supports a maximum of 10 provider applications.

<*timer*> The maximum amount of time, in seconds, to wait for a response. Specify an amount of time from 0 to 20 seconds.

<*variable*> or <*list_of_variables*> The call variable, or list of variables (up to 10), in which you want to store the data received from the third-party application. Do not use global variables in the variable list—only call variables are allowed. However, you can use either global or call variables for the application_ID and the timer.

TIMER option

You can specify the duration to wait for a response from the host, to a maximum of 20 seconds. If the server does not receive a response in the specified time, the command fails and is stopped. If you do not specify a timer, the default value is 10 seconds.

The only way to detect that the timer expired in the script is to initialize the call variables with specific values (using the ASSIGN TO command), and test them after the GET RESPONSE command to see if they changed.

Example

The following section of script sends a call CLID to a third-party application, and then receives the preferred skillset for that customer. The first line of the script initializes the variable named pref_skillset_cv. Then the SEND REQUEST command sends the customer's CLID to the third-party application. The GET RESPONSE command retrieves the preferred skillset for the customer from the third-party application and replaces the value initially assigned to the variable pref_skillset_cv with the value retrieved from the third-party application.

If the value retrieved from the third-party application is returned within 10 seconds, the call is queued to the appropriate skillset.

```
ASSIGN Sales_sk TO pref_skillset_cv
/* initialize the call variable in case of failed response */
SEND REQUEST ivr_hdxID pref_skillset_gv
GET RESPONSE ivr_hdxID pref_skillset_cv
QUEUE TO SKILLSET pref_skillset_cv
WAIT 2
GIVE RAN agents_bsy_ran
GIVE MUSIC soft_music
QUIT
```

Chapter 30: Contact processing for SIP

This chapter describes the scripting command support in a SIP-enabled Contact Center.

Configuration guidelines and general scripting considerations

This section describes the configuration details relevant to call flow operation.

Host Data Exchange (HDX) is supported for SIP. For more information, see Host Data Exchange commands on page 353.

RAN and MUSIC routes

RAN and MUSIC route configuration does not affect call flow operation but is described here as it relates to RAN and MUSIC command handling in a SIP-enabled contact center.

Current optional configuration

Currently, when you use the GIVE RAN and GIVE MUSIC script commands, Avaya recommends that you configure the RAN or MUSIC route number as a script variable.

Important:

RAN and MUSIC routes do not apply to IM contacts.

The following figure shows a music script variable in Contact Center Manager Administration.

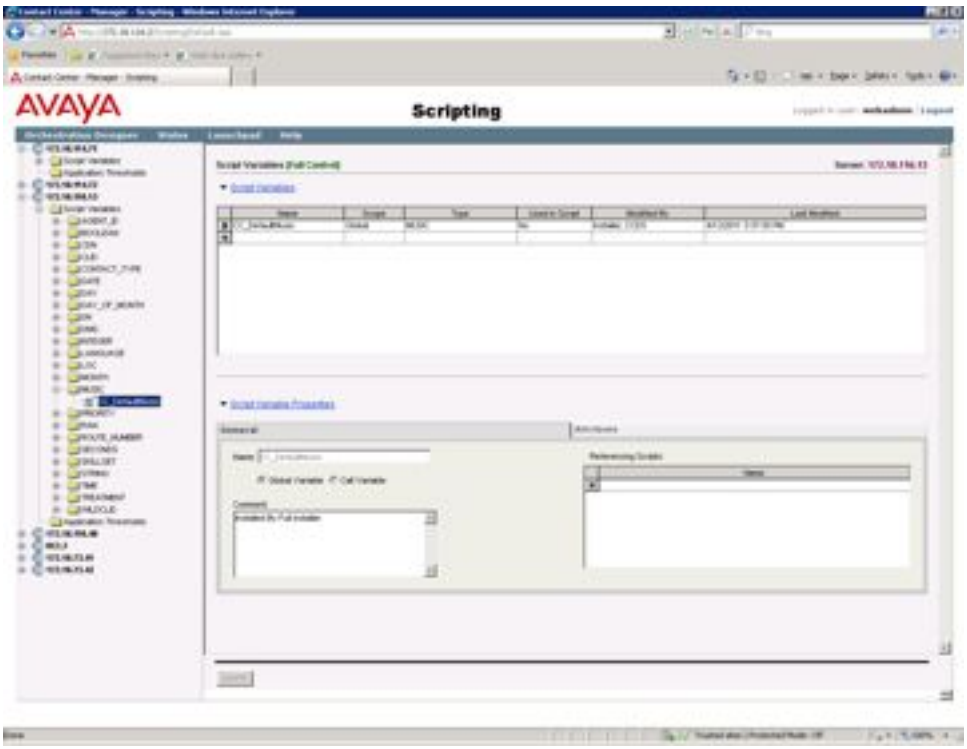


Figure 74: Script variables

SIP Contact Center mandatory configuration

To support announcement and music service in Contact Center, you must configure the route information (route number and route name) in Contact Center Manager Administration. The route name is a string, which must be identical to the announcement file name or music content group (for example, myAnnouncement.wav or pop) stored on the Avaya Media Server. Enforcing the configuration of the announcement file means that the announcement and music file names are displayed on the generated RAN and MUSIC reports.

The following figure shows the route number and name of music and announcement files in Contact Center Manager Administration.

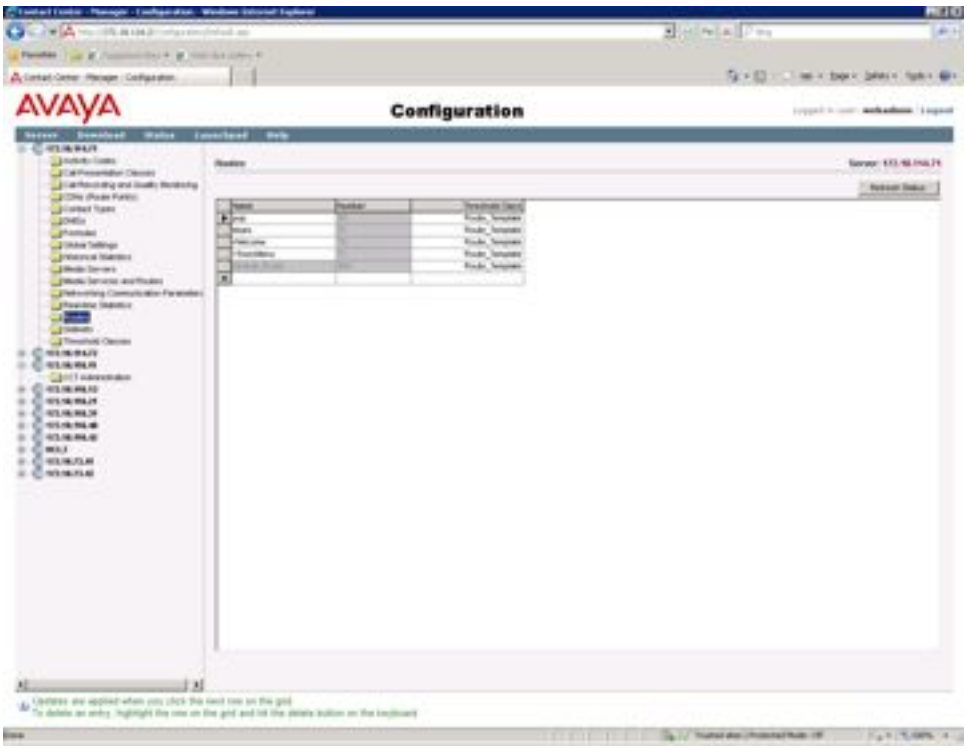


Figure 75: Music and announcement routes

Example: Using the MUSIC variable type with the GIVE MUSIC command

```
/* This script demonstrates the use of the GIVE MUSIC using the music variable type.
To use this sample script:
(1) Create the following variable in CCMA scripting-
Global Variable:        Variable Type
pop_music_route         MUSIC (12)
(2) Configure the following route in CCMA configuration-
Routes
Number:      Name:
12           pop
*/
GIVE RINGBACK
WAIT 5
QUEUE TO SKILLSET Default_Skillset
GIVE MUSIC pop_music_route
WAIT 5
DISCONNECT
```

Important:

On Avaya Communication Server 1000 installations of Contact Center Manager Server, music is automatically reapplied by the switch after the initial GIVE MUSIC command if, for example, the call is returned to queue. This means the GIVE MUSIC command does not need to be reissued in scripts.

In SIP-enabled Contact Center Manager Server installations, music is automatically reapplied by the switch after the initial GIVE MUSIC command if the call is returned to queue. However, in all other scenarios, GIVE MUSIC must be manually reapplied at appropriate points in the script call flow as required. For example, Avaya Communication Server 1000 Contact Center Manager Server installations automatically reinsert MUSIC (if MUSIC was

the last command issued) between RANs but in SIP-enabled Contact Center Manager Server installations, GIVE MUSIC must be manually reapplied.

Content groups with the GIVE MUSIC command

Avaya Media Server supports the concept of content groups. Content groups are collections of compatible music audio files. For example, use the content group “pop” to deploy a collection of various pop music songs. Each song is submitted as an individual .wav file within the content group. For information about creating content groups and deploying recordings to them, see *Avaya Aura® Contact Center Commissioning* (NN44400-312).

When Avaya Media Server receives an request for a content group, such as pop, Avaya Media Server selects at random any .wav file in the content group and begins streaming playback of that file.

When the first file is finished, Avaya Media Server selects another recording in the content group folder and begins playback. Avaya Media Server continues playback until Contact Center Manager Server scripting terminates the request for music. This occurs, for example, when a call is offered to a selected agent or a WAIT timeout is followed by an invocation to a new treatment.

Important:

The GIVE MUSIC <*route number*> command does not work if the route number corresponds to a specific music file rather than a content group.

GIVE_IVR

In the SIP-enabled contact center, the GIVE IVR (Interactive Voice Response) command provides treatments to callers by playing recorded announcements or by collecting caller information from VXML scripts residing on the media server. This command is extended to include instant message treatments.

Each invocation of the GIVE IVR command pulls resources real-time on both the Contact Center Manager Server and the Avaya Media Server server terminating the GIVE IVR request. If you have multiple treatments in a sequence of invocations of GIVE IVR, Avaya recommends compounding these treatments into a single GIVE IVR command. For example, instead of invoking two instances of GIVE IVR to play a prompt and collect digits, use a single invocation passing parameters to support both functions. You significantly optimize the performance of the overall solution and, in particular, Avaya Media Server usage performance.

Note:

In applications that use the GIVE IVR command, you must use GIVE MUSIC or GIVE RINGBACK after the GIVE IVR command to provide a default treatment. If you do not configure default treatment after the GIVE IVR command, the customer hears silence.

Syntax

GIVE IVR {INTERRUPTIBLE}

{SERVICE URI <*service_uri*>}

WITH VXML TREATMENT <*treatment_uri*>

{PARAMETERS <*variable_p1*>,… <*variable_p10*>}

{RETURNS <*variable_r1*>}

Keywords denoted with parentheses {} are optional.

INTERRUPTIBLE keyword

INTERRUPTIBLE is an optional keyword used by Contact Center Manager Server to interrupt the media treatment, for example, when an agent becomes available to take the contact.

When the INTERRUPTIBLE option is used, if the contact is queued before the GIVE IVR command is used, then the IVR session is interrupted when an agent becomes available to take the contact.

SERVICE URI

The SERVICE URI keyword is an optional keyword that expects a predefined script variable (<*service_uri*>) of type String. The <*service_uri*> script variable represents any valid SIP address of the following format:

- sip: username@domain

The name and value of the <service_uri> script variable are defined in Contact Center Manager Administration scripting. The name of the script variable can be any value; there is no validation on the script variable name. For example:

- cv_service_uri represents sip:__sip-dialog@Avaya.com
- cv_dialog_sip_add represents sip:__sip-dialog@sip6.avaya.com

The SIP address represents the media service at which the invoked VXML treatment can be serviced. The SIP address must have a media server (or proxy) associated with it through Contact Center Manager Administration configuration. A default SIP address is configured on the Default Media Services tab of the Server Configuration utility on Contact Center Manager Server and this address is used, if a SIP address is not specified in the variable name.

The Default Media Services tab only appears in the Server Configuration utility if Contact Center Manager Server is installed with a SIP-enabled switch.

WITH VXML TREATMENT option

The WITH VXML TREATMENT keyword of the GIVE IVR command expects a predefined script variable (<treatment_uri>) of type String. The <*treatment_uri*> script variable represents any valid VXML file to invoke.

The VXML script name must exactly match in Contact Center Manager Administration scripting and the Avaya Media Server.

The name and value of the <treatment_uri> script variable are defined in Contact Center Manager Administration scripting. The name of the script variable can be any value; there is no validation on the script variable name. For example:

- cv_treatment_uri represents InitialTreatment.vxml
- cv_vxml_treatment represents CollectTreatment.vxml

When invoking the GIVE IVR command against the Avaya Media Server Contact Center application, the <treatment_uri> script variable name must be voicexml and the script variable value must be either PlayPrompt.vxml or PlayAndCollect.vxml.

The Avaya Media Server is case-sensitive so all values must be assigned correctly.

PARAMETERS keyword

The optional PARAMETERS keyword of the GIVE IVR command uses a list of 1 to 10 predefined script variables (<*variable_p1*>,...<*variable_p10*>) of type string, integer, or digit that are inputs to the VXML treatment identified by the WITH VXML TREATMENT keyword. The names of these variables are fixed and the names and functions of the variables are defined in the following sections.

PARAMETERS and Avaya Media Server When the Contact Center Manager Server scripting engine operates with Avaya Media Server, the VXML scripts must use at least one parameter from the list of predefined valid parameters. For example, one parameter that must be passed to PlayPrompt.vxml is the name of the .wav file to play in voice contacts.

The following table lists the parameters to be used with the VXML scripts. These Contact Center script variables are automatically created when SIP is enabled on the Contact Center Manager Server. The script variable names and values are case-sensitive. You must use the variable names as they are specified in the table.

Variable name	Description	Script	Type	Default value
contenttype	Identify the type of formatting (plain, rtf, or html) for the text in the instant message.	call	string	text/plain
im_str	A text string to store values returned by the customer to a treatment.	call	string	“str_val”
im_int	A numerical value to store values returned by the customer to a treatment.	call	int	0
interdigittimeout	The maximum time, in seconds, the script waits for digit information from the voice contact or instant message contact. The script waits until either the required number of digits is collected or the timer expires.	call	call	10
notypeahead	By default, the audio prompt is interrupted if the caller presses a digit. Use the notypeahead option to ensure the prompt is not	call	string	false

Variable name	Description	Script	Type	Default value
	interrupted by the caller entering digits. This string value is true or false.			
numberofdigits	The maximum number of digits that the script collects from the caller.	call	string	1
prompttoplay	The file that the script invokes. For example, filename.wav.	call	string	hello.wav
termchar	Any character that terminates the digit collection before the maximum number of digits is reached. The terminating character is not returned to the script.	call	string	9
vars	Any string value.	call	string	99
vxmlfrom	A SIP address of the format username@domain. The vxmlfrom parameter identifies the SIP address that the recipient sees as the originator of the received IM or Web page.	call	string	contactcenter@cc.avaya.com
vxmlto	A SIP address in the format username@domain. The vxmlto parameter identifies the SIP address to which the IM or Web page is sent.	call	string	customer@domain.com

When the SIP-enabled Contact Center Manager Server scripting engine is the invoking party, a list of parameters (with default values) are predefined as Contact Center Manager Server scripting engine variables. Depending on the target VXML script, these parameters are optional or mandatory. For example, the interdigittimeout parameter can have a value configured in the Contact Center Manager Server scripting engine, but it can also have a default value to which the VXML scripts reverts to if the parameter is not included with the GIVE IVR command.

When SIP is selected as the Contact Center Manager Server switch integration option during installation, the following script variables are automatically detected. The default values indicate for what the variables are used.

Name	Script	Type	Default value
c_estimated_wait_time_cv	call	integer	199
c_play_and_collect_gv	global	string	PlayAndCollect.vxml
c_play_only_gv	global	string	PlayPrompt.vxml
c_position_in_queue_cv	call	integer	299

Name	Script	Type	Default value
serviceuri	call	string	sip:__sip-dialog@sipserver.com
c_sip_digits_int_cv	call	integer	123
c_sip_digits_str_cv	call	string	123
c_sip_from_add_cv	call	string	customer@domain.com
c_sip_to_add_cv	call	string	contactcenter@cc.avaya.com
voicexml	call	string	PlayPrompt.vxml

Two script variables (c_sip_digits_int_cv and c_sip_digits_str_cv) are used to store collected digit information returned from the PlayAndCollect.vxml script.

PlayAndCollect.vxml returns collected digit information only. You can manipulate the returned digits in Contact Center Manager Server scripting engine by defining the returned digits script variable as either an integer or a string as follows:

- As an integer according to the definition of c_sip_digits_int_cv. When the variable is an integer variable, only nine digits can be extracted and manipulated. In this instance, the returned digits parameter is used in conjunction with the GIVE IVR vars parameter (which is defined as an integer) and the prompttoplay parameter to create an announcement (see script example below). This plays back the collected digits to the customer for validation.

  The %i0 represents the individual digits entered.

```
/*This GIVE IVR collects the customers account number.*/
ASSIGN c_play_and_collect_gv TO voicexml
ASSIGN “EnterYourAccountNumber.wav” TO prompttoplay
GIVE IVR
SERVICE URI serviceuri
WITH VXML TREATMENT voicexml
PARAMETERS prompttoplay, notypeahead, numberofdigits,
interdigittimeout
RETURNS c_sip_digits_int_cv
/*This GIVE IVR command plays back the digits that were
entered.*/
ASSIGN c_sip_digits_int_cv TO vars
ASSIGN c_play_only_gv TO voicexml
ASSIGN “TheAccountNumberYouEnteredWas.wav+%i0” TO
prompttoplay
GIVE IVR
SERVICE URI serviceuri
WITH VXML TREATMENT voicexml
PARAMETERS prompttoplay, vars
```

- As a string according to the definition of c_sip_digits_str_cv. When the variable is a string variable, up to 80 digits can be returned and manipulated in scripting. When the variable is defined as a string it cannot be used in conjunction with the GIVE IVR vars parameter.

  Text can be assigned to the message buffer and sent multiple times during the execution of a script. If the message buffer data is larger than 500 characters, the command succeeds, but the data is truncated. You can use the following commands to manipulate the message buffer:

- ASSIGN MESSAGE on page 368
- APPEND MESSAGE on page 369
- SEND MESSAGE on page 369
- CONTAINS on page 370
- MATCHES on page 370

RETURNS keyword

The optional RETURNS keyword of the GIVE IVR command is a predefined script variable (<variable_r1>) of type string, integer, or digit that is returned to Contact Center Manager Server by the VXML treatment identified by the WITH VXML TREATMENT keyword.

The GIVE IVR command compiles any user-defined return parameters, but you must declare and assign values to return parameter script variables before using the GIVE IVR command.

If the GIVE IVR command times out, a default value is returned to the Contact Center Manager Server. For voice contacts, the default value returned is -1. For IM contacts, the default value returned is TIMEOUT. You can configure the timeout time using the interdigittimeout variable.

GIVE IVR syntax rules

The following syntax rules apply only to SIP scripts that execute on Avaya Media Server. You must use these rules with the GIVE IVR command to maintain functional compatibility with the Avaya Media Server Contact Center Service-provided VXML files. The Contact Center Manager Server scripting engine compiler does not check for compliance to these rules.

- The voicexml variable must be present and the value of the variable must be PlayPrompt.vxml or PlayAndCollect.vxml.

  The variable voicexml must be defined in Contact Center Manager Administration Scripting and the value (PlayPrompt.vxml or PlayAndCollect.vxml) must be stored on the Avaya Media Server. The voicexml parameter is used as follows:

  ```
  GIVE IVR WITH VXML TREATMENT voicexml
  ```

- If you use PlayAndCollect.vxml as the value of the voicexml script variable in Contact Center Manager Administration, you must also use the PARAMETERS keyword with prompttoplay, numberofdigits, or termchar. PlayAndCollect.vxml is stored on the Avaya Media Server. The parameters for the treatment are used as follows:

  ```
  GIVE IVR WITH VXML TREATMENT voicexml
  PARAMETERS prompttoplay, numberofdigits
  or
  GIVE IVR WITH VXML TREATMENT voicexml
  PARAMETERS prompttoplay, termchar
  or
  GIVE IVR WITH VXML TREATMENT voicexml
  PARAMETERS prompttoplay, numberofdigits, termchar
  ```

- If you use the prompttoplay parameter, you must use a value that is a combination of template values separated by a plus sign (+).

  ```
  ASSIGN “YourCurrentExpectedWaitTime.wav
        +Three.wav+SecondsPleaseHold.wav” TO prompttoplay
  ```

Instant message commands

Use the following commands to compose and send instant message commands using the message buffer to maximize the effectiveness of the message text.

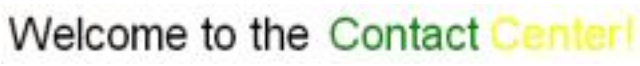
To support longer instant messages, a message buffer can contain a string argument with a maximum of 500 characters. One message buffer exists per contact.

Text messages can contain text formatting such as a different font, color, or boldface, however, the formatting decreases the number of characters that can be used for the text. To format the instant message, begin typing the message into an application such as WordPad, format the message, and save it. Open the formatted text to Notepad.

The following script shows an example with plain text formatting.

```
ASSIGN MESSAGE “Please enter your query.<BR>”, “You can also enter the department
if you know its name”
SEND MESSAGE
PARAMETERS vxmlfrom, vxmlto
RETURNS im_str
```

The following shows an example with rich text formatting.




```
/*Rich text is supported for scripted IMs*/
ASSIGN “text/rtf” TO contenttype
ASSIGN MESSAGE “{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fswiss
\fprq2\fcharset0”, “Arial;}{\f1\fswiss\fcharset0 Arial;}} {\colortbl ;
\red0\green128\blue0;”,“\red255\green255\blue0;} \viewkind4\uc1\pard\f0\fs28 ”,
“Welcome to the \b Avaya\b0  \cf1 Contact\cf0  \cf2 Center!”, “\cf0\f1\fs20\par }”
SEND MESSAGE
PARAMETERS vxmlfrom, vxmlto, contenttype
RETURNS im_str
```

The default contenttype is text/plain. The initial ASSIGN command assigns the rtf text type to contenttype so that the message can be viewed with the fonts and colors in the text message. Each excerpt of the message is chunked in 80 character strings.

ASSIGN MESSAGE

Create or replace the current message string in the message buffer with the new string. The instant message text cannot exceed 500 characters.

ASSIGN MESSAGE parameter1, <parameter2>, ... <parameter15>

The parameters can be a literal string constant, a call or global variable of type string, integer, or DN.

Example:

```
ASSIGN MESSAGE “Welcome to the Avaya SIP Contact Center. ”,
“If you know the department (e.g. Sales, Support, HR), then please enter it now.”
```

The message buffer contains: Welcome to the Avaya SIP Contact Center. If you know the department (e.g. Sales, Support, HR), then please enter it now.

Use the SEND MESSAGE command to send the text in the current buffer.

You can use carriage returns in the instant message to create a line break. In scripts, the carriage return appears as
.

**ASSIGN MESSAGE “Welcome”
 “To Contact Center”**

The above command appears to the customer as follows:

```
Welcome
To Contact Center
```

APPEND MESSAGE

Add the current message string to the message buffer. The instant message text cannot exceed 500 characters.

APPEND MESSAGE parameter1, <parameter2>, ... <paramater15>

The parameters can be a literal string constant, a call or global variable of type string, integer, or DN.

Example:

```
APPEND MESSAGE “Otherwise, state your query.”
```

The message buffer contains: Welcome to the Avaya SIP Contact Center. If you know the department (e.g. Sales, Support, HR), then please enter it now. Otherwise, state your query.

SEND MESSAGE

Send the statement currently in the message buffer to the customer. The message in the buffer remains intact between SEND statements, so the same message can be used multiple times without redefining it.

SEND MESSAGE

You can use the SEND MESSAGE command to send instant messages to experts in the contact center. The following code from the instant message script shows how to determine what agents in the contact center are available to handle the instant message. By using the VXMLTO call variable, you can send the instant message to the expert. If you do not use the VXMLTO command, the message goes to the customer as usual.

```
/*Enter the URI of the expert*/
ASSIGN “sip:joe_expert@avaya.com” TO vxmlto
```

```
ASSIGN MESSAGE “An expert is required for a Contact Center call.”,
“Are you available to take the call (y/n)?”
SEND MESSAGE
PARAMETERS vxmlfrom, vxmlto
RETURNS im_str
WHERE im_str CONTAINS
   VALUE “y” : ...
```

You cannot send messages to CDNs.

CONTAINS

The CONTAINS keyword searches for substrings within a main string. The CONTAINS keyword applies to all text-based communications in Contact Center.

See CONTAINS on page 307.

MATCHES

The MATCHES keyword searches for strings within a main string. The MATCHES keyword applies to all text-based communications in Contact Center.

See MATCHES on page 312.

ROUTE CALL URI

Send a string to a Uniform Resource Identifier (URI) destination specified by the parameter.

ROUTE CALL URI <*any string*> {CONTROLLED}

CONTROLLED is optional.

<*any string*> The string that represents the URI to which you want the contact routed.

If you use the CONTROLLED keyword, the contact remains in the control of Contact Center, which means that if an error condition occurs on the switch, the contact is returned to the queue so that it can be further treated by the system. When a command without the CONTROLLED keyword is completed, Contact Center has no control over the contact.

No commands can follow the ROUTE CALL URI command.

Chapter 31: Intrinsics overview

Intrinsics contain system-wide information about skillsets, agents, time, traffic, and call type. You can use intrinsics in your scripts to access system information. The script then uses this information in formulas and decision-making statements.

Contact Center creates and maintains intrinsics automatically. Intrinsics are available only to query data about the system within scripts, not to modify data. Any script can use information from intrinsics throughout the system.

Important:

Intrinsics are pieces of data that may be available on a contact depending on factors such as the contact type. If a treatment is performed on a contact, or the contact is transferred, conferenced or routed to an agent, the intrinsic value may not be available. If the value is not available, the system may return BAD or INVALID as the name.

Types of intrinsics

There are several types of intrinsics for use in scripts:

- Call intrinsics on page 375
- Open Queue intrinsics (see Multimedia intrinsics on page 391)
- Skillset intrinsics on page 403
- Time intrinsics on page 421
- Traffic intrinsics on page 425

Return value

The data that an intrinsic gathers from the system and inserts into the script is referred to as the return value of the intrinsic. For example, the following section of a script instructs the system to queue calls to the support skillset if the number of agents in the service skillset (LOGGED AGENT COUNT) is less than five.

```
IF (LOGGED AGENT COUNT service_sk < 5) THEN
   QUEUE TO SKILLSET support_sk
END IF
```

If, at 2:00 p.m., three agents from the service skillset are logged on, then the return value for the intrinsic is 3. Therefore, incoming calls are queued to the support skillset. Suppose later,

at 3:30 p.m., nine agents from the service skillset log on. Now, the return value for the intrinsic is 9, and calls are not queued to the support skillset.

Intrinsic returns no value

If an intrinsic cannot return a valid value, it can return no value at all. For example, in the following statement:

```
QUEUE TO AGENT LONGEST IDLE AGENT service_sk, general_sales_sk
```

If all agents in the service_sk and general_sales_sk skillsets are busy on active calls or are in Not Ready mode, then the intrinsic LONGEST IDLE AGENT does not return a value. As a result, Contact Center cannot queue the call to an agent.

To prevent the call from being queued to the default skillset, ensure that the call is queued before the end of the script.

Intrinsic usage

This section provides examples of how you can use intrinsics in your scripts for voice and multimedia contacts.

Decision making within a script based on skillset intrinsics

This script checks whether the number of idle agents in the service skillset is greater than the number of idle agents in the support skillset. If the service skillset has more idle agents, the contact is queued to the service skillset. Otherwise, the contact is queued to the support skillset.

```
IF (IDLE AGENT COUNT service_sk > IDLE AGENT COUNT support_sk)
THEN
   QUEUE TO SKILLSET service_sk
   WAIT 2
ELSE
   QUEUE TO SKILLSET support_sk
   WAIT 2
END IF
```

Decision making within a script based on time intrinsics

This script first checks whether the time of day is between 05:00 p.m. and 08:00 a.m. and that the day of the week is not Saturday or Sunday. If all of these conditions are true, then the Night_Section statement is executed.

```
IF (TIME OF DAY = dinner_hour_gv) AND
   (DAY OF WEEK < > SATURDAY, SUNDAY)THEN EXECUTE
      Night_Section
END IF
```

Decision making within a script based on traffic intrinsics

This script checks to see if the number of voice contacts in the system is greater than 50. If it is greater, the caller hears a busy tone.

```
IF (TOTAL ACTIVE CALLS > 50) THEN
   GIVE BUSY
   WAIT 20
   DISCONNECT
END IF
```

If Open Queue is enabled then the TOTAL ACTIVE CALLS is the total number of calls and Open Queue contacts in the system. In this scenario there is potential for a large number of contacts to be waiting in the system relative to what normally is expected for telephony. Therefore, use TOTAL ACTIVE CALLS with caution in an Open Queue enabled system.

Decision making within a script based on call intrinsics

This script checks to see if the CLID, in the case of the Avaya Communication Server 1000, of the voice contact is 416-555-1212. If it is, then the contact is queued to the skillset gold_skills_sk. Otherwise, the contact is queued to the skillset general_skills_sk.

```
IF (CLID = 4165551212) THEN
   QUEUE TO SKILLSET gold_skills_sk
   WAIT 2
ELSE
   QUEUE TO SKILLSET general_skills_sk
   WAIT 2
END IF
IF (DNIS = 4165551212) THEN
   QUEUE TO SKILLSET gold_skills_sk
   WAIT 2
ELSE
   QUEUE TO SKILLSET general_skills_sk
   WAIT 2
END IF
```

Chapter 32: Call intrinsics

Call intrinsics are specific to each call.

AGE OF CALL

This intrinsic returns the age of the contact at the time the intrinsic is executed.

This intrinsic applies to both voice and multimedia contacts.

Syntax

AGE OF CALL

Example

The following example is a section of script that queues the voice contact to the sales skillset, and then plays music. The section Check_Age checks the age of the voice contact every 20 seconds, until the voice contact is more than 2 minutes old. If the voice contact is greater than 2 minutes old, it is queued to a backup skillset.

```
QUEUE TO SKILLSET sales_sk WITH PRIORITY 2
WAIT 2
GIVE MUSIC pop_music_gv
SECTION Check_Age
   WAIT 20
   IF (AGE OF CALL > 120) THEN
      EXECUTE Requeue
   END IF
   EXECUTE Check_Age
SECTION Requeue
   QUEUE TO SKILLSET backup_sk WITH PRIORITY 1
   WAIT 2
   /* additional call handling steps */
```

CALL DATA

This intrinsic holds caller-entered data. You must have an Avaya IVR system in place to use this intrinsic in your scripts.

Syntax

CALL DATA <*data number*> | <*variable (1-10)*>

Parameters

- <*data number*> A number from 1 to 10. You can use up to 10 call data intrinsics in your scripts.
- <*variable*> The name of the variable in which to store the caller-entered data. The type for this variable must be digit. Lists of variables are single variables separated by commas.

Return type

This intrinsic returns a digit to the script.

This intrinsic can only return digits to the script. It cannot return letters or symbols.

Examples

Example 1 In this example, this portion of the script checks the value of the caller-entered data intrinsic, Call Data 1. The value of this intrinsic is determined by the information entered by the caller in an IVR session. If the caller presses 1 on their keypad, then the call is queued to General Sales. If the caller presses 2, then the call is queued to the Business Travel skillset, and so on. If the caller presses any other number, or no number at all, the call is automatically queued to the General Sales skillset.

```
...
WHERE CALL DATA 1 EQUALS
   VALUE 1: QUEUE TO SKILLSET General_Sales
   VALUE 2: QUEUE TO SKILLSET Business_Travel
   VALUE 3: QUEUE TO SKILLSET Gold_Clients
   DEFAULT: QUEUE TO SKILLSET skillset_A
END WHERE
```

Example 2 Karen calls BestAir Airlines to inquire about European vacation packages. Before her call is answered by an agent, she receives IVR treatment. She hears a prompt such as:

“For information about business travel, press 1. For information about vacations, press 2.”

Karen presses 2. This information is stored in a call data intrinsic and is sent to Contact Center where it can be used in a script to route and treat the call appropriately.

```
WHERE CALL DATA 1 EQUALS
   VALUE 1: QUEUE TO SKILLSET Business_Travel_sk
   VALUE 2: QUEUE TO SKILLSET Vacations_sk
   DEFAULT: QUEUE TO SKILLSET Default_sk
END WHERE
```

Using an Avaya IVR

You can use a front-end IVR system to collect information from callers. This system plays voice menus to callers, and collects their responses. After the callers choose to talk to an agent, Avaya IVR sends the caller-entered data to Contact Center and then transfers the call to a CDN that is controlled by Contact Center.

Call handling with IVR

If you use Avaya IVR, you can configure your system to handle calls as follows:

- The incoming call enters the switch and is queued to a DN that is connected to the front-end IVR system.
- After an IVR voice channel (configured on the switch as a phone) becomes available, the call is routed to the IVR system.
- The caller receives IVR treatment, which typically involves playing voice prompts and collecting responses. The responses are collected into the Call Data intrinsic.
- After the IVR session is finished, the IVR system sends the caller-entered data to Contact Center over the contact center server subnet, and then transfers the call to a CDN that is monitored and controlled by Contact Center.
- Call processing continues as configured in the script for that CDN.

Scripting recommendation

Several seconds can pass after the start of the call that is transferred from the IVR system to the CDN. The switch expects the call to receive treatment from Contact Center within the time specified in a switch-configurable timer; otherwise, the switch gives the call default treatment. Avaya recommends that you write your script to give the call an immediate treatment.

Example

```
IF CONSULTED AND (CLID = mail_ports) THEN
   GIVE RINGBACK /* First treatment 8?
   WAIT 20 /* Wait for the transfer to complete */
   QUIT /* Code not reach here */
END IF
```

CALL FORWARD

This intrinsic indicates if the call is forwarded (where the return value is True) or not (where the return value is False).

This intrinsic is useful, for example, if a caller dials an agent DN directly, but the agent is not available to take the call. If the call is forwarded to the contact center, you can play an announcement stating that the call was forwarded.

This intrinsic applies to voice contacts only.

Syntax

CALL FORWARD

Example

In this example, forwarded calls receive a recorded announcement that tells the caller that their call is forwarded to the contact center.

```
IF CALL FORWARD THEN
   GIVE RAN forward_to_call_center_ran_gv
END IF
```

CALL FORWARD BUSY

This intrinsic indicates whether the call is forwarded because the phone at the original destination is busy (where the return value is True) or not (where the return value is False).

This intrinsic applies to voice contacts only.

Syntax

CALL FORWARD BUSY

Return type

This intrinsic returns a True or False value to the script.

Example

In this example, forwarded calls due to a busy condition receive a recorded announcement stating that the call is forwarded.

```
IF CALL FORWARD BUSY THEN
   GIVE RAN busy_forwarded_ran_gv
END IF
```

CALL FORWARD DO NOT DISTURB

This intrinsic indicates whether the call is forwarded due to a do not disturb condition at the destination (where the return value is True) or not (where the return value is False).

This intrinsic applies to voice contacts only.

Syntax

CALL FORWARD DO NOT DISTURB

Example

In this example, calls forwarded due to a do not disturb condition at the destination receive a recorded announcement stating that the call is forwarded.

```
IF CALL FORWARD DO NOT DISTURB THEN
   GIVE RAN sleeping_ran_gv
END IF
```

CALL FORWARD NO ANSWER

This intrinsic indicates whether the call is forwarded because no one answered the phone at the original destination (where the return value is True) or not (where the return value is False).

This intrinsic applies to voice contacts only.

Syntax

CALL FORWARD NO ANSWER

Example

In this example, calls forwarded due to no answer at the destination are given a recorded announcement stating that the call is forwarded.

```
IF CALL FORWARD NO ANSWER THEN
   GIVE RAN forward_no_answer_ran_gv
END IF
```

CDN

The value of this intrinsic is the controlled directory number (CDN) of the current call.

This intrinsic applies to voice contacts only.

Syntax

CDN

Example

In this example, the script executes the GoldCard, Reg_Customers, or Promotions script, depending on the CDN of the call.

```
WHERE CDN EQUALS
   VALUE 5551230: EXECUTE SCRIPT Gold_Customers
   VALUE 5551231: EXECUTE SCRIPT Reg_Customers
   VALUE 5551232: EXECUTE SCRIPT Promotions
END WHERE
```

CLID

The value of this intrinsic is the calling line identification (CLID) of the current call.

This intrinsic applies to voice contacts only.

Use the provider.exe tool to verify the CLID string sent by the switch. Make sure you generate test calls from several regions of the country.

Syntax

CLID

Wildcards and placeholders

The following free-format expressions in CLID comparisons provide flexible digit string evaluation:

- wildcard—Use the at symbol (@) to represent a string of digits (zero or more).
- placeholder—Use a question mark (?) to represent a single-digit position.

The following rules apply to the use of wildcards and placeholders:

- Use wildcards and placeholders only with the CLID intrinsic.
- Use only one wildcard in any one CLID string. For example, the string @345@ is not valid.
- Use wildcards only at the beginning or the end of a string.
- Do not use wildcards and placeholders in ranges of CLIDs. For example, you cannot use 333@ .. 339@. However, you can use wildcards and placeholders in lists of CLIDs.
- Only use equal (=) and not equal (< >) operators with wildcards and placeholders. Greater than and less than operations on wildcard expressions give ambiguous results. A validation error is generated for scripts that use this construct.
- Use only variables of the WILDCLID data type for wildcard and placeholder characters.
- The statement CLID = @ always returns a True value.
- A string with placeholders has only a return value of True for CLIDs with the same number of digits as there are placeholders. For example, the statement CLID = ??? true for all CLID strings of three digits, and false for all other strings. This type of expression is useful if you want to filter out strings of a fixed length.
- Both wildcards and placeholders can occur in a CLID digit string. All rules apply to placement.

Examples

In this example, calls with CLIDs in the vip_list_gv variable are handled in a special way.

```
IF (CLID = vip_list_gv) THEN
   EXECUTE Special_Handling
END IF
```

The following example shows how to use a wildcard in a CLID digit string. All calls starting with 305 are queued to the VIP skillset with the priority of 1 and hear a special recorded announcement.

```
IF (CLID = 305@) THEN
   QUEUE TO SKILLSET VIP_sk WITH PRIORITY 1
   WAIT 2
   GIVE RAN you_are_special_ran_gv
END IF
```

The following example shows how to use a placeholder and wildcard to indicate that all CLIDs starting with 305 through to 395 are queued to the VIP skillset with a priority of 1 and hear a special recorded announcement.

```
IF (CLID = 3?5@) THEN
   QUEUE TO SKILLSET VIP_sk WITH PRIORITY 1
   WAIT 2
   GIVE RAN you_are_special_ran_gv
END IF
```

CONFERENCED

This intrinsic is True when the call is a consultative call. A consultative call is created when you start to conference in another party. When the conference is established, this intrinsic is False.

This intrinsic applies to voice contacts only.

Syntax

CONFERENCED

Example

In this example, the conference consultation call receives a recorded message stating that the conference is starting. The other callers in the conference do not hear this message. Only the caller initiating the conference hears it.

```
IF CONFERENCED THEN
   GIVE RAN in_conference_ran_gv
END IF
```

CONSULTED

The return value of this intrinsic indicates if the call was transferred or conferenced (where the return value is True) or not (where the return value is False).

Syntax

CONSULTED

Example

In this example, this script plays a recorded announcement to callers who are transferred or conferenced.

```
IF CONSULTED THEN
   GIVE RAN 24 /* “Your call has been transferred” */
END IF
```

DIALED DN

The value of this intrinsic is the number that the caller originally dialed if the call is forwarded to the CDN using a switch feature. This intrinsic applies only to forwarded calls. You can use the dialed DN to transfer a call to Contact Center.

This intrinsic applies to voice contacts only.

Syntax

DIALED DN

Example

In this example, calls are queued to the appropriate skillset depending on the dialed DN.

```
WHERE DIALED DN EQUALS
   VALUE 2512: QUEUE TO SKILLSET sales_sk WITH PRIORITY 1
   VALUE 2603: QUEUE TO SKILLSET sales_sk WITH PRIORITY 2
   VALUE 2776: QUEUE TO SKILLSET service_sk WITH PRIORITY 1
   DEFAULT: QUEUE TO SKILLSET service_sk WITH PRIORITY 2
END WHERE
```

The DIALED DN intrinsic can handle the blind transfer of a call back to the contact center by an agent. The dialed DN of the agent's transfer call is preserved for the original call when the transfer is complete so the caller can be redirected to a new treatment or skillset.

DIRECT CALL

The value of this intrinsic indicates if the call is a direct call (where the return value is True) or not (where the return value is False). Use the DIRECT CALL intrinsic with phones that have the forwarding option.

This intrinsic applies to voice contacts only.

Syntax

DIRECT CALL

Example

In this example, all calls that are not direct calls receive a recorded announcement.

```
IF NOT DIRECT CALL THEN
   GIVE RAN you_were_forwarded_ran_gv
END IF
```

DNIS

The value is the first number that the caller dialed to enter the system.

This intrinsic is called the Dialed Number Identification Service (DNIS) and is specific to the trunk used. Not all trunks are configurable to support DNIS; therefore, the DNIS intrinsic is empty for any trunk that is not configured on the switch to provide DNIS.

This intrinsic applies to voice contacts only.

If you use Virtual Network Services (VNS) trunks, the DNIS feature is not supported. This means that a DNIS number cannot be forwarded from one site to another in a VNS multisite contact center. If you want to send DNIS numbers from one site to another, you must configure Meridian Customer Defined Networking (MCDN) between sites.

When used for an incoming network call, the DNIS value is the same as the value at the source node. The answering agent phone can display the original source DNIS information, as long as the DNIS name and number are programmed in both the source and target Contact Center (under Switch Configuration). Use the provider.exe tool to verify the DNIS string sent by the switch. Make sure you generate test calls to various DNIS numbers. For more information, see Host Data Exchange commands on page 353.

Syntax

DNIS

Example

In this example, calls are queued to the appropriate skillset depending on the DNIS.

```
WHERE DNIS EQUALS
   VALUE 5552512: QUEUE TO SKILLSET sales_sk WITH PRIORITY 1
   VALUE 5552603: QUEUE TO SKILLSET sales_sk WITH PRIORITY 2
   VALUE 5552776: QUEUE TO SKILLSET service_sk WITH PRIORITY
   1
   DEFAULT: QUEUE TO SKILLSET service_sk WITH PRIORITY 2
END WHERE
WAIT 2
GIVE RAN agents_busy
GIVE MUSIC soft_music
```

INTERNATIONAL CALL

The value of this intrinsic indicates whether the call is international (where the return value is True) or not (where the return value is False).

This intrinsic applies to voice contacts only.

Syntax

INTERNATIONAL CALL

Example

The following script gives calls that are identified as international a choice of languages in which the callers can receive service.

```
IF INTERNATIONAL CALL THEN
   OPEN VOICE SESSION
   PLAY PROMPT VOICE SEGMENT choose_language_vs
   /* “For service in English, please press 1, followed
   by the pound key. For service in French, please
   press 2, followed by the pound key. For service in
   German, please press 3, followed by the pound
   key.” */
      COLLECT 1 DIGITS INTO language_choice_cv WITH
      TERMINATING CHARACTER #
   END VOICE SESSION
END IF
```

WITH TERMINATING CHARACTER option

You can specify a terminating character to end the digit collection even if the maximum number of digits is collected. The terminating character is not included in the variable with the digits.

Ending digit collection

Digit collection ends under the following conditions:

- The INTER DIGIT TIMER expires.
- The terminating character is received.
- The maximum number of possible digits is reached.
- An agent becomes available to take the call (if the call is already queued before digit collection began).

LOC

The value is the location code (LOC) number of the call. The LOC is the first three digits for private network calls.

This intrinsic applies to voice contacts only.

Syntax

LOC

Example

In this example, calls are played the appropriate recorded announcement based on the LOC number of the individual call.

```
WHERE LOC EQUALS
   VALUE 512: GIVE RAN marketing_ran_gv
   VALUE 603: GIVE RAN sales_ran_gv
```

```
   VALUE 776: GIVE RAN accounting_ran_gv
   DEFAULT: GIVE RAN company_ran_gv
END WHERE
```

NETWORK CALL

The value indicates if the call is a network call (where the return value is True) or not (where the return value is False). You can use the NETWORK CALL intrinsic in the network script to restrict local users from dialing network CDNs.

Syntax

NETWORK CALL

Example

In this example, if the call is not a network call, the caller receives a recorded announcement asking the caller to phone the local number.

```
IF NOT NETWORK CALL THEN
   GIVE RAN wrong_number_ran_gv
   DISCONNECT
END IF
```

NPA

The value is the number plan area (NPA) or area code of the current call.

This intrinsic applies to voice contacts only.

Syntax

NPA

Example

This script executes different sections of the script based on the area code of the caller.

```
WHERE NPA EQUALS
   VALUE 416: EXECUTE Toronto_Section
   VALUE 514: EXECUTE Montreal_Section
   VALUE 613: EXECUTE Ottawa_Section
   DEFAULT: EXECUTE Ontario_Section
END WHERE
```

NXX

The value is the local exchange code (NXX) of the current call.

This intrinsic applies to voice contacts only.

Syntax

NXX

Example

This script executes different sections of the script based on the local exchange of the caller.

```
WHERE NXX EQUALS
   VALUE 491, 492, 493: EXECUTE North_Section
   VALUE 290, 291, 293: EXECUTE East_Section
   DEFAULT: EXECUTE All_Section
END WHERE
```

NPANXX

The value is the number plan area or area code and the local exchange code (NPANXX) of the current call.

This intrinsic applies to voice contacts only.

Syntax

NPANXX

Example

This script executes different sections of the script based on the area code and the local exchange code of the caller.

```
WHERE NPANXX EQUALS
   VALUE 416491, 416492, 416493: EXECUTE North_Section
   VALUE 416290, 416291, 416293: EXECUTE East_Section
   DEFAULT: EXECUTE All_Section
END WHERE
```

ON HOLD

The value indicates whether the call is on hold (where the return value is True) or not (where the return value is False).

This intrinsic applies to voice contacts only.

Syntax

ON HOLD

Example

This script plays music if a call is on hold.

```
EVENT HANDLER
   EVENT CALL ON HOLD: GIVE MUSIC pop_music_gv
END HANDLER
```

QUEUED

The value indicates whether the call is queued in any local skillsets, network skillsets, or agent queues, or in an NACD queue (where the return value is True) or not (where the return value is False).

This intrinsic applies to voice and non voice contacts.

Syntax

QUEUED

Examples

This script uses the QUEUED command to avoid an endless loop and ensures that the initial QUEUE TO SKILLSET command worked.

```
IF OUT OF SERVICE sales_sk THEN
   EXECUTE Help_Me_Now
END IF
QUEUE TO SKILLSET sales_sk WITH PRIORITY 3
WAIT 3
/* This section repeats a recorded announcement every 30 seconds.It also checks
whether the call is queued before playing the announcement. */
SECTION Play_2nd_RAN
   WAIT 30
   IF NOT QUEUED THEN
      EXECUTE Help_Me_Now
   END IF
   GIVE RAN agents_still_busy_ran_gv
EXECUTE Play_2nd_RAN
SECTION Help_Me_Now
...
```

This example shows the QUEUED intrinsic when the networking option is enabled and the include local node option is selected. This example is used to test if a call is queued. The QUEUED intrinsic returns a Boolean value of true if the call is queued to a skillset (local or network) or an agent.

```
QUEUE TO NETWORK SKILLSET Sales_SK
WAIT 10
IF NOT QUEUED THEN
   QUEUE TO NETWORK SKILLSET Service_SK
   WAIT 4
   IF NOT QUEUED THEN
      GIVE IVR mmail WITH TREATMENT sorry_closed
```

```
      DISCONNECT
   END IF
END IF
```

ROUTE NUMBER

The value is the route point of the current call. If some routes are more expensive than others, you can use this intrinsic to ensure that calls from expensive routes are treated quickly.

This intrinsic applies to voice contacts only.

Syntax

ROUTE NUMBER

Example

In this example, a variable named route_variable_cv represents the route number of the current call. If the route number of the current call is 321, then the caller hears a recorded announcement.

```
ASSIGN ROUTE NUMBER TO route_variable_cv
IF (route_variable_cv = 321) THEN
   GIVE RAN calling_californians_ran_gv
END IF
```

TRANSFERRED

The value indicates whether the call is transferred (where the return value is True) or not (where the return value is False).

The value for the TRANSFERRED intrinsic is True only for the call ID that is created when you initiate a transfer. When the transfer is complete, the intrinsic value is False.

This intrinsic applies to voice contacts only.

Syntax

TRANSFERRED

Example

In this example, transferred calls receive a recorded announcement (named please_complete_transfer_ran_gv) that tells the agent to complete the transfer. That is, the agent who performs the transfer hears the please_complete_transfer_ran_gv announcement, but the original call (which is on hold during the transfer) does not. After the transfer is complete, the script restarts at the top of the master script, and the original caller then hears a recorded

announcement (hello_again_ran_gv) because the TRANSFERRED intrinsic returns to a false state after transfer completion.

```
IF TRANSFERRED THEN
   SECTION Transfer_Loop
      /*force agent to complete transfer*/
      GIVE RAN please_complete_transfer_ran_gv
      WAIT 20
      EXECUTE Transfer_Loop
ELSE
   GIVE RAN hello_again_ran_gv
END IF
```

Chapter 33: Multimedia intrinsics

Multimedia intrinsics are specific to multimedia routing with the Open Queue feature installed and configured. Contact Center Multimedia creates a number of contact data intrinsics for each multimedia contact. The intrinsics are used within the sample multimedia scripts to correctly route the contact.

%AGENT%AGENT

For scheduled contacts or transferred contacts, this intrinsic returns the agent assigned to the contact by Contact Center Multimedia based on actions performed by an agent on the Agent Desktop.

For reply e-mail contacts and new e-mail contacts from existing customers, this intrinsic returns the preferred agent assigned to that customer.

An agent ID is sent in the contact data under the following conditions:

- An agent saves an e-mail as pending with a scheduled presentation time.
- An agent chooses a disposition code for an outbound contact that reschedules the contact to be presented at a later time.
- An agent transfers a contact to another agent.
- An agent creates a scheduled callback contact from the Agent Desktop.
- If a reply e-mail arrives from a customer, the preferred agent for that customer is available as the agent ID in the scripts.
- If a new e-mail arrives from an existing customer, the preferred agent for that customer is available as the agent ID in the scripts.
- An agent pulls a closed contact (pull-mode). In this case, a new contact is created and sent to Contact Center Manager Server for processing by scripts.

Syntax

The intrinsic must be used in conjunction with the CONTACT DATA intrinsic and must be assigned to a script call variable before use.

ASSIGN CONTACT DATA “%AGENT%AGENT” TO contact_agent_cv

Example

This example stores the %AGENT%AGENT and %PRIORITY%PRIORITY intrinsics in call variables and then queues the contact to the agent indicated by %AGENT%AGENT with a priority of %PRIORITY%PRIORITY.

```
ASSIGN CONTACT DATA “%AGENT%AGENT” TO contact_agent_cv
ASSIGN CONTACT DATA “%PRIORITY%PRIORITY” TO contact_priority_cv
```

```
QUEUE TO AGENT contact_agent_cv WITH PRIORITY    contact_priority_cv
WAIT 2
```

%DATE%DATE

This intrinsic returns the date on which a scheduled contact is presented to the agent. This date is called the Callback Date.

If you want scheduled contacts to occur at the correct date and time, ensure that your script delays these contacts until that date and time and then queues the contacts appropriately. The contacts can be delayed in the scripts in one of two ways:

- using a loop in conjunction with the intrinsics %DATE%DATE, %TIME%TIME, and NEXTYEAR
- using a WAIT statement in conjunction with the intrinsic %INTEGER%WAITTIME

Important:

The use of a WAIT statement is the preferred method.

Syntax

The intrinsic must be used with the CONTACT DATA intrinsic and must be assigned to a script call variable before use.

ASSIGN CONTACT DATA “%DATE%DATE” TO date_cv

Example

In this example, a check determines if the current date is equal to the callback date and if the current time is greater than the callback time. If so, the system executes the section Queue_Contact. Otherwise, the contact remains in a loop checking the date and time.

```
ASSIGN CONTACT DATA “%DATE%DATE” TO contact_cbdate_cv
ASSIGN CONTACT DATA “%TIME%TIME” TO contact_cbtime_cv
SECTION Callback_loop
/*************************************************************
Loop until the Current Date / Time is greater than or equal to the Callback Date /
Time then Queue the Contact using the script
Queue_Multimedia_Contact.
*************************************************************/
IF (DATE > contact_cbdate_cv) THEN
   EXECUTE SCRIPT Queue_Multimedia_Contact
   ELSE
   IF ((DATE = contact_cbdate_cv)
      AND (TIME OF DAY >= contact_cbtime_cv)) THEN
      EXECUTE SCRIPT Queue_Multimedia_Contact
   ELSE
      WAIT 60
      EXECUTE Callback_loop
   END IF
END IF
```

%INTEGER%WAITTIME

This intrinsic returns the number of seconds from the time the contact is created until the callback time. Use this intrinsic as the value in a WAIT statement in the scripts to delay queuing of a contact until the callback time.

If you want scheduled contacts to be presented at the correct date and time, ensure that your script delays these contacts until that date and time and then queues the contacts appropriately. The contacts can be delayed in the scripts in one of two ways:

- using a loop with the intrinsics %DATE%DATE, %TIME%TIME, and NEXTYEAR
- using a WAIT statement the intrinsic %INTEGER%WAITTIME

Important:

The use of the WAIT statement is the preferred method.

Syntax

The intrinsic must be used in the CONTACT DATA intrinsic and must be assigned to a variable before use.

ASSIGN CONTACT DATA “%INTEGER%WAITTIME” TO contact_wait_time_cv

Example

This example stores the %INTEGER%WAITTIME, %SKILLSET%SKILLSET, and %PRIORITY%PRIORITY intrinsics in call variables. It then waits for the required number of seconds until the callback time before queuing the contact to the skillset indicated by %SKILLSET%SKILLSET with a priority of %PRIORITY%PRIORITY.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO contact_skillset_cv
ASSIGN CONTACT DATA “%PRIORITY%PRIORITY” TO contact_priority_cv
ASSIGN CONTACT DATA “%INTEGER%WAITTIME” TO contact_wait_time_cv
WAIT contact_wait_time_cv
QUEUE TO SKILLSET contact_skillset_cv WITH PRIORITY
contact_priority_cv
WAIT 2
```

%PRIORITY%PRIORITY

This intrinsic returns the priority assigned to this contact by Contact Center Multimedia based on the rules configured or based on the actions performed by an agent on the Agent Desktop.

Avaya recommends that you queue multimedia contacts with the priority %PRIORITY %PRIORITY, because this priority is determined by Contact Center Multimedia.

Syntax

The intrinsic must be used in the CONTACT DATA intrinsic and must be assigned to a script call variable before use.

ASSIGN CONTACT DATA “%PRIORITY%PRIORITY” TO contact_priority_cv

Example

This example stores the %SKILLSET%SKILLSET and %PRIORITY%PRIORITY intrinsics in call variables and then queues the contact to the skillset indicated by %SKILLSET%SKILLSET with a priority of %PRIORITY%PRIORITY.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO contact_skillset_cv
ASSIGN CONTACT DATA “%PRIORITY%PRIORITY” TO contact_priority_cv
QUEUE TO SKILLSET contact_skillset_cv WITH PRIORITY
contact_priority_cv
WAIT 2
```

%SKILLSET%SKILLSET

This intrinsic returns the skillset ID assigned to this contact by Contact Center Multimedia based on the rules configured or based on the actions performed by an agent on the Agent Desktop.

Syntax

The intrinsic must be used in the CONTACT DATA intrinsic and must be assigned to a call variable before use.

ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO contact_skillset_cv

Example

This example stores the %SKILLSET%SKILLSET and PRIORITY%PRIORITY intrinsics in call variables and then queues the contact to the skillset indicated by %SKILLSET%SKILLSET with a priority of %PRIORITY%PRIORITY.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO contact_skillset_cv
ASSIGN CONTACT DATA “%PRIORITY%PRIORITY” TO contact_priority_cv
QUEUE TO SKILLSET contact_skillset_cv WITH PRIORITY
contact_priority_cv
WAIT 2
```

%TIME%TIME

This intrinsic returns the time of day on which a scheduled contact is presented to an agent. This time is called the callback time.

If you want scheduled contacts to be presented at the correct date and time, ensure that your script delays these contacts until that date and time and then queues the contacts appropriately. The contacts can be delayed in the scripts in one of two ways:

- using a loop the intrinsics %DATE%DATE, %TIME%TIME, and NEXTYEAR
- using a WAIT statement the intrinsic %INTEGER%WAITTIME

Important:

The use of a WAIT statement is the preferred method.

Syntax

The intrinsic must be used in the CONTACT DATA intrinsic and must be assigned to a script call variable before use.

ASSIGN CONTACT DATA “%TIME%TIME” TO time_cv

Example

This is the same example as shown for the %DATE%DATE intrinsic. In this example, a check determines if the current date is equal to the callback date and if the current time is greater than the callback time. If so, the system executes the section Queue_Contact. Otherwise, the contact remains in a loop checking the date and time.

```
ASSIGN CONTACT DATA “%DATE%DATE” TO contact_cbdate_cv
ASSIGN CONTACT DATA “%TIME%TIME” TO contact_cbtime_cv
SECTION Callback_loop
/************************************************************
Loop until the Current Date / Time is greater than or equal to the
Callback Date / Time then Queue the Contact using the script
Queue_Multimedia_Contact.
************************************************************/
IF (DATE > contact_cbdate_cv) THEN
   EXECUTE SCRIPT Queue_Multimedia_Contact
ELSE
   IF ((DATE = contact_cbdate_cv)
      AND (TIME OF DAY >= contact_cbtime_cv)) THEN
      EXECUTE SCRIPT Queue_Multimedia_Contact
   ELSE
      WAIT 60
      EXECUTE Callback_loop
   END IF
END IF
```

CONTACTSUBTYPE

Contacts in the Contact Center Multimedia database have a contact type which is one of the main contact types, e-mail, outbound, voice, Web communications and video.

Contacts also have a subtype. For example, you can create fax, SMS, voice mail, and white mail contacts in the Multimedia database using the e-mail rules configuration. These contact types are all subtypes of the e-mail contact type. In scripting, a fax message has a CONTACTSUBTYPE of fax and a CONTACT TYPE of e-mail.

Similarly, you can create scheduled callback contacts in the Multimedia database using the Agent Desktop or Web services interface. This contact type is a subtype of the outbound contact type. In scripting, the scheduled callback has a CONTACTSUBTYPE of scheduled callback and a CONTACT TYPE of outbound.

This intrinsic returns the subtype of the contact.

The following table shows all contact subtypes and the corresponding contact types.

CONTACT SUBTYPE	CONTACT TYPE
fax	e-mail
SMS	e-mail
voice mail	e-mail
white mail	e-mail
scanned documents	e-mail
other	e-mail
e-mail	e-mail
scheduled callback	outbound
outbound	outbound
Web communications	Web communications
video	video
Contacts must be queued to skillsets that correspond to a contact type. For example, scheduled callbacks are queued to outbound (OB) skillsets while fax contacts are queued to e-mail (EM) skillsets.	

Syntax

The intrinsic must be used in conjunction with the CONTACT DATA intrinsic and must be assigned to a variable before use.

ASSIGN CONTACT DATA “CONTACTSUBTYPE” TO contact_subtype_cv

Example

This example checks the CONTACTSUBTYPE of scheduled callback and executes separate primary scripts for scheduled callbacks and outbound campaign contacts.

```
IF CONTACT TYPE = c_contact_type_outbound_gv THEN
ASSIGN CONTACT DATA “CONTACTSUBTYPE” TO contact_subtype_cv
   IF contact_subtype_cv = “Scheduled Callback” THEN
      EXECUTE SCRIPT OB_ScheduledCallback_Primary
   ELSE
      EXECUTE SCRIPT OB_Outbound_Primary
   END IF
END IF
```

CUSTOMERID

This intrinsic returns the customer ID associated with contact.

Syntax

The intrinsic must be used in conjunction with the CONTACT DATA intrinsic and must be assigned to a script call variable before use.

ASSIGN CONTACT DATA “CUSTOMERID” TO contact_custID_cv

Example

The following example assigns the customer ID to a call variable and then increases the priority of the contact if that customer is identified as a priority customer.

```
ASSIGN CONTACT DATA “CUSTOMERID” TO contact_custID_cv
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO contact_skillset_cv
ASSIGN CONTACT DATA “%PRIORITY%PRIORITY” TO contact_priority_cv
IF contact_custID_cv = “5004” THEN
   /* identify priority Customer */
   ASSIGN 1 TO contact_priority_cv
END IF
QUEUE TO SKILLSET contact_skillset_cv WITH PRIORITY
contact_priority_cv
WAIT 2
```

CUSTOMFIELD1

This intrinsic returns the text value of the first custom field associated with a contact. Information about the first custom field is only available as an intrinsic within scripting. Other custom fields are not available.

Syntax

The intrinsic must be used in conjunction with the CONTACT DATA intrinsic and must be assigned to a variable before use.

ASSIGN CONTACT DATA “CUSTOMFIELD1” TO contact_customfield_cv

Example

This example checks for a CUSTOMFIELD1 value of Ref_123456 and executes a specific primary script for contacts that have this custom field value.

```
IF CONTACT TYPE = c_contact_type_outbound_gv THEN
   ASSIGN CONTACT DATA “CUSTOMFIELD1” TO contact
      _customfield_cv
   IF contact_customfield_cv = “Ref_123456” THEN
      EXECUTE SCRIPT OB_ref123456_Primary
   ELSE
      EXECUTE SCRIPT OB_Outbound_Primary
   END IF
END IF
```

FROMADDRESS

This intrinsic returns the e-mail address from which an e-mail contact was sent. This intrinsic is available for e-mail contacts only.

Syntax

The intrinsic must be used in conjunction with the CONTACT DATA intrinsic and must be assigned to a script call variable before use.

ASSIGN CONTACT DATA “FROMADDRESS” TO contact_fromAddress_cv

Example

The following example assigns the from address value to a call variable and then increases the priority of the contact if the customer is identified as a priority customer.

```
ASSIGN CONTACT DATA “FROMADDRESS” TO contact_fromAddress_cv
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO contact_skillset_cv
ASSIGN CONTACT DATA “%PRIORITY%PRIORITY” TO contact_priority_cv
IF CONTACT TYPE = c_EMAIL_gv THEN
   /* Identify a Priority Customer by their e-mail address */
   IF (contact_fromAddress_cv = “johnsmith@online.com”) THEN
      ASSIGN 1 TO contact_priority_cv
   END IF
END IF
QUEUE TO SKILLSET contact_skillset_cv WITH PRIORITY
contact_priority_cv
WAIT 2
```

NEXTYEAR

This intrinsic indicates whether the callback date and time values for the contact are in the current year or the next year.

If you want scheduled contacts to be presented at the correct date and time, ensure that your script delays these contacts until that date and time and then queues the contacts appropriately. The contacts can be delayed in the scripts in one of two ways:

- using a loop in conjunction with the intrinsics %DATE%DATE, %TIME%TIME, and NEXTYEAR
- using a WAIT statement in conjunction with the intrinsic %INTEGER%WAITTIME

Important:

The use of a WAIT statement is the preferred method.

Syntax

The intrinsic must be used in conjunction with the CONTACT DATA intrinsic and must be assigned to a script call variable before use.

ASSIGN CONTACT DATA “NEXTYEAR” TO callback_nextyear_cv

Example

This example checks if the NEXTYEAR intrinsic has the value YES. If so, the script remains in a loop until the date changes to January 1, and then the script runs another section that checks the callback date and time.

```
ASSIGN CONTACT DATA “NEXTYEAR” TO callback_nextyear_cv
SECTION Next_Year_Loop
IF CALLBACK_NEXTYEAR_cv = “YES” THEN
   IF DATE = Jan 01 THEN
      ASSIGN “NO” TO callback_nextyear_cv
      EXECUTE Callback_DateTime_Check
   ELSE
      WAIT 60
      EXECUTE Next_Year_Loop
   END IF
END IF
SECTION Callback_DateTime_Check
```

QUEUETYPE

This intrinsic allows for flexibility when you decide how to queue a contact by identifying groups of contacts based on either of the following:

- The type of contact (for example, a reply e-mail from a customer).
- Actions performed on the contact using the Agent Desktop (for example, a contact that is transferred to an agent).

These contact types must be queued appropriately depending on the needs of multimedia contact center.

Syntax

The intrinsic must be used in conjunction with the CONTACT DATA intrinsic and must be assigned to a script call variable before use.

ASSIGN CONTACT DATA “QUEUETYPE” TO contact_queuetype_cv

Value	Description and recommended queuing method
SKILLSET	Queue this contact to the skillset indicated by the intrinsic %SKILLSET %SKILLSET with a priority of 1. Configure the QueueType value when a multimedia contact is transferred to a skillset by using the Contact Center Agent Desktop. This contact already reached the front of one Skillset Queue and is queued with high priority to the skillset it is being transferred to, hence the priority of 1. If you do not require transferred contacts to have a high priority, queue this contact with a priority of %PRIORITY%PRIORITY instead.
AGENT	Queue this contact to the agent indicated by the intrinsic %AGENT %AGENT with a priority of 1. This QueueType value is configured when one of the following actions occurs on the Contact Center Agent Desktop: • A multimedia contact is transferred to an agent. • An agent saved an e-mail contact as pending with a scheduled callback time. • An agent created a scheduled callback contact from the Contact Center Agent Desktop. • An agent saved an outbound contact with a disposition code that initializes a scheduled callback time for the contact. The contact reached the front of one Skillset Queue and is queued with High Priority to the relevant agent or next skillset. It is assigned the priority of 1.
AGENTPULL	Queue this contact to the agent indicated by the intrinsic %AGENT %AGENT with a priority of 1. Configure the QUEUETYPE value when a multimedia contact is pulled from the queue by using the Contact Center

Value	Description and recommended queuing method
	Agent Desktop. The agent requires this contact to be presented immediately.
REPLYMAIL	This QUEUETYPE value indicates that the contact is a reply e-mail contact from a customer. This situation normally occurs where a customer replies to an e-mail message sent from Contact Center Agent Desktop by an agent. It can also occur where a new e-mail message from an existing customer includes a string that matches the System Delivery Failure rule. There are two queuing options in this case: • If you want reply e-mail contacts from customers to be queued to the preferred agent for that customer, queue this contact to the agent indicated by the intrinsic %AGENT%AGENT with a priority of %PRIORITY%PRIORITY%. • If you want reply e-mail contacts from customers to be queued to a skillset, queue this contact to the skillset indicated by the intrinsic %SKILLSET%SKILLSET with a priority of %PRIORITY%PRIORITY.
NEWMAIL	This QUEUETYPE value indicates the contact is a new e-mail contact from an existing customer. There are two queuing options in this case: • If you want new e-mail contacts from existing customers to be queued to the preferred agent for that customer, queue this contact to the agent indicated by the intrinsic %AGENT%AGENT with a priority of %PRIORITY%PRIORITY%. • If you want new e-mail contacts from existing customers to be queued to a skillset, queue this contact to the skillset indicated by the intrinsic %SKILLSET%SKILLSET with a priority of %PRIORITY%PRIORITY.
NOQUEUE	In this case, no specific queuing method is recommended by Contact Center Multimedia because no actions occur on this contact by using the Agent Desktop. This QUEUETYPE value is configured for new contacts (other than new e-mail from existing customers that have a QUEUETYPE value of NEWMAIL). Queue these contacts to the skillset indicated by the intrinsic %SKILLSET%SKILLSET with a priority of %PRIORITY %PRIORITY.

Example

This example assigns the QUEUETYPE intrinsic to a variable and then queues the contact differently based on the value of the QUEUETYPE.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO contact_skillset_cv
ASSIGN CONTACT DATA “%AGENT%AGENT” TO contact_agent_cv
ASSIGN CONTACT DATA “%PRIORITY%PRIORITY” TO contact_priority_cv
ASSIGN CONTACT DATA “QUEUETYPE” TO contact_queuetype_cv
WHERE contact_queuetype_cv EQUALS
   VALUE “SKILLSET”: QUEUE TO SKILLSET contact_skillset_cv
      WITH PRIORITY 1
   VALUE “AGENT”: QUEUE TO AGENT contact_agent_cv WITH
      PRIORITY 1
   VALUE “AGENTPULL”: QUEUE TO AGENT contact_agent_cv WITH
      PRIORITY 1
   VALUE “REPLYMAIL”: QUEUE TO AGENT contact_agent_cv WITH
```

```
      PRIORITY contact_priority_cv
   VALUE “NEWMAIL”: QUEUE TO SKILLSET contact_skillset_cv
      WITH PRIORITY contact_priority_cv
   VALUE “NOQUEUE”: QUEUE TO SKILLSET contact_skillset_cv
      WITH PRIORITY contact_priority_cv
   DEFAULT: QUEUE TO SKILLSET contact_skillset_cv WITH
      PRIORITY contact_priority_cv
END WHERE
WAIT 2
```

TOADDRESS

This intrinsic returns the e-mail address that an e-mail contact was sent to. It is available for e-mail contacts only.

Syntax

The intrinsic must be used in conjunction with the CONTACT DATA intrinsic and must be assigned to a script call variable before use.

ASSIGN CONTACT DATA “TOADDRESS” TO contact_toAddress_cv

Example

This example assigns the to address and customer ID values to call variables and then uses these call variables to check if a priority customer sent an e-mail to a general mailbox. If so, the priority of the contact is increased.

```
ASSIGN CONTACT DATA “TOADDRESS” TO contact_toAddress_cv
ASSIGN CONTACT DATA “SKILLSET” TO contact_skillset_cv
ASSIGN CONTACT DATA “PRIORITY” TO contact_priority_cv
ASSIGN CONTACT DATA “CUSTOMERID” TO contact_custID_cv
IF CONTACT TYPE = c_EMAIL_gv THEN
   /* Identify a Priority Customer who sent an E-mail to a
   general mailbox */
        IF ((contact_toAddress_cv = “info@mycompany.com”)
      AND (contact_custID_cv = “5004”)) THEN
      ASSIGN 1 TO contact_priority_cv
        END IF
END IF
QUEUE TO SKILLSET contact_skillset_cv WITH PRIORITY contact_priority_cv
WAIT 2
```

Chapter 34: Skillset intrinsics

Skillset intrinsic elements are based on information about skillsets. You can use the value returned from the intrinsic to queue, for example, commands, conditional command. Skillset intrinsics return skillsets, integer values, and agent IDs.

By default, the skillset of a multimedia contact is shown to the agent when an incoming contact is presented to the agent on the Agent Desktop. Voice call skillsets do not appear when the call is present to the agent because the skillset information is not available as an intrinsic.

If you want the skillset information to appear on the Agent Desktop for voice calls before the agent accepts the call, then you must add the skillset information as call data to the voice call before you queue it to the appropriate skillset. The Agent Desktop displays the information in the call data when the call is presented.

Important:

To pass the network skillset information across the network, the Network/IVR tab must be configured in the Database for each Communication Control Toolkit (CCT) server in the network environment. To add the skillset information to the call data, you must configure the TAPI connection in the Database Integration Wizard on Contact Center Manager Server, and then modify your scripts to add the appropriate voice skillset name to the TAPI call data for the call.

ANSWERED COUNT

The ANSWERED COUNT intrinsic is the number of incoming contacts answered during the preceding 10 minutes. This intrinsic applies to both voice and multimedia contacts.

If you specify a list of skillsets, then the sum of answered contact counts for the skillsets is returned.

ANSWERED CALL COUNT is supported.

Syntax

ANSWERED COUNT [<skillset> | <skillset_list>]

Parameters

Enter information for the following parameter:

- <*skillset*> or <*skillset_list*> The skillset, or list of skillsets, that you want to include in the return value.

Examples

In this example, if the number of answered calls for the service skillset is less than that of the support skillset, then incoming calls are queued to the service skillset. Otherwise, calls are queued to the support skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

```
IF (ANSWERED COUNT service_sk < ANSWERED CALL COUNT support_sk)THEN
   QUEUE TO SKILLSET service_sk
ELSE
   QUEUE TO SKILLSET support_sk
END IF
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC pop_music_gv
SECTION WaitLoop
```

In this multimedia example, if the number of answered contacts for the e-mail skillset sent in the contact data is greater than that of the Promo2 e-mail skillset, then incoming contacts are queued to the Promo2 e-mail skillset. Otherwise, contacts are queued to the e-mail skillset sent in the contact data.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
IF (ANSWERED COUNT skillset_cv > ANSWERED COUNT EM_Promo2_sk
THEN
   ASSIGN EM_Promo2_sk TO skillset_cv
END IF
QUEUE TO SKILLSET skillset_cv
WAIT 2
SECTION WaitLoop
```

AVERAGE SPEED ANSWER

The AVERAGE SPEED ANSWER is the calculated average speed of contact answering in the given priority coming into the system. This data is based on the same calculations as the real-time supervisor data displays using the real-time moving window time frame of 10 minutes. This intrinsic applies to both voice and multimedia contacts.

If you specify a list of skillsets, then the minimum AVERAGE SPEED ANSWER for the list of skillsets is returned. If you omit the WITH PRIORITY segment, then the return value includes calls of all priorities.

Syntax

AVERAGE SPEED ANSWER [<*skillset*> | <*skillset_list*>] {WITH CALL PRIORITY <*priority*>}

Parameters

Enter information for the following parameters:

- <*skillset*> or <*skillset_list*> The skillset, or list of skillsets, that you want to include in the return value.
- <*priority*> The priority of the calls that you want to track.

Examples

In this example, if calls are answered more quickly, on average, by the service skillset than by the support skillset, then incoming calls are queued to the service skillset. Otherwise, calls are queued to the support skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

```
IF (AVERAGE SPEED ANSWER service_sk < AVERAGE SPEED ANSWER
   support_sk) THEN
      QUEUE TO SKILLSET service_sk
ELSE
   QUEUE TO SKILLSET support_sk
END IF
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC pop_music_gv
SECTION WaitLoop
```

In this multimedia example, if contacts are answered less quickly, on average, for the e-mail skillset sent in the contact data than for the Promo2 e-mail skillset, then incoming contacts are queued to the Promo2 e-mail skillset. Otherwise, contacts are queued to the e-mail skillset sent in the contact data.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
IF (AVERAGE SPEED ANSWER skillset_cv > AVERAGE SPEED ANSWER
   EM_Promo2_sk THEN
      ASSIGN EM_Promo2_sk TO skillset_cv
END IF
QUEUE TO SKILLSET skillset_cv
WAIT 2
SECTION WaitLoop
```

EXPECTED WAIT TIME

The EXPECTED WAIT TIME is the predicted wait time of the current contact in the skillset at the moment the intrinsic is executed. This value can change over time, depending on contact traffic. This intrinsic is calculated by the contact processing executor using real-time data. If the contact is not yet in the skillset indicated, an average expected wait time for the skillset is returned.

If you specify a skillset list, then the returned value is the minimum EXPECTED WAIT TIME of all the skillsets.

The purpose of the EXPECTED WAIT TIME intrinsic is to estimate, based on historical information, the length of time for the current contact to be answered by a particular skillset. This information can then be used to announce to the contact the amount of time to wait before being answered, or it can provide a different treatment to the contact. This intrinsic applies to both voice and multimedia contacts.

Important:

Use this intrinsic to play the expected wait time to voice contacts only if the wait is unusually long for your contact center. For example, if the normal wait time for your contact center is 2 minutes, but a burst of traffic increased the wait time to 5 minutes, use this intrinsic to warn voice contacts of the long wait. Give the caller options at this point (for example, the choice of leaving a message or continuing to wait).

Syntax

EXPECTED WAIT TIME [<*skillset*> | <*skillset_list*>]

Parameters

Enter information for the following parameter:

- <*skillset*> or <*skillset_list*> The skillset, or list of skillsets, that you want to include in the return value.

Examples

In the following voice example, the contact is first queued to the sales skillset. If the EXPECTED WAIT TIME for the contact is greater than 4 minutes, the caller hears a recorded announcement stating that a long wait is expected. If the EXPECTED WAIT TIME is greater than 1 minute but less than 4 minutes, the recorded announcement indicates an average wait time. An EXPECTED WAIT TIME less than 1 minute uses a short wait announcement.

```
QUEUE TO SKILLSET sales_sk
WAIT 4
ASSIGN EXPECTED WAIT TIME sales_sk TO exp_wait_cv
IF (exp_wait_cv > 240) THEN
   GIVE RAN long_wait_ran_gv
ELSE
   IF (exp_wait_cv > 60) THEN
      GIVE RAN average_wait_ran_gv
   ELSE
      GIVE RAN short_wait_ran_gv
   END IF
END IF
```

In this multimedia example, if the expected wait time for the e-mail skillset sent in the contact data is greater than 3600 seconds, then incoming contacts are queued to the Promo2 e-mail skillset. Otherwise, contacts are queued to the E-mail skillset sent in the contact data.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
ASSIGN Promo2 e-mail TO backup_skillset_cv
ASSIGN EXPECTED WAIT TIME skillset_cv TO exp_wait_cv
IF (exp_wait_cv < 3600) THEN
   QUEUE TO SKILLSET skillset_cv
   WAIT 2
ELSE
   QUEUE TO SKILLSET backup_skillset_cv
   WAIT 2
END IF
SECTION WaitLoop
```

IDLE AGENT

Use this intrinsic to check whether the specified agent is currently idle. If the agent is idle, the value returned is True. Otherwise, the value returned is False.

This intrinsic applies to both voice and multimedia contacts.

Syntax

IDLE AGENT <*agent_ID*>

Parameters

Enter information for the following parameter:

- <*agent_ID*> The logon ID of the agent that you want to track.

Examples

In this example, if the agent represented by the agent variable agent_4 is idle and available, the call goes to that agent. Otherwise, the call is queued to the sales skillset.

```
IF IDLE AGENT agent_4 THEN
   QUEUE TO AGENT agent_4
   WAIT 2
ELSE
   QUEUE TO SKILLSET sales_sk
   WAIT 2
END IF
```

In this multimedia example, if the preferred agent sent in the contact data is idle and available, the contact goes to that agent. Otherwise, the contact is queued to the skillset sent in the contact data.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
ASSIGN CONTACT DATA “%AGENT%AGENT” TO agent_cv
IF IDLE AGENT agent_cv THEN
   QUEUE TO AGENT agent_cv
   WAIT 2
ELSE
   QUEUE TO SKILLSET skillset_cv
   WAIT 2
END IF
```

IDLE AGENT COUNT

The IDLE AGENT COUNT is the current number of idle agents in the skillset list.

If you specify a list of skillsets, then the return value is the maximum IDLE AGENT COUNT of all the skillsets in the list.

This intrinsic applies to both voice and multimedia contacts.

Syntax

IDLE AGENT COUNT [<*skillset*> | <*skillset_list*>]

Parameters

Enter information for the following parameter:

- <*skillset*> or <*skillset_list*> The skillset, or list of skillsets, that you want to include in the return value.

Examples

This sample script first queues the call to the sales skillset. If the call is not answered immediately, the script checks if more than two idle agents are in the service skillset. If so, the call is queued to the service skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

```
QUEUE TO SKILLSET sales_sk
WAIT 2
IF (IDLE AGENT COUNT service_sk > 2) THEN
   QUEUE TO SKILLSET service_sk
   WAIT 2
END IF
GIVE RAN agents_busy_ran_gv
GIVE MUSIC soft_music_gv
SECTION WaitLoop
```

The following sample script first queues the multimedia contact to the skillset sent in the contact data. If the contact is not answered after 600 seconds, the script checks if there are more than two idle agents in the EM_Promo2_sk skillset. If so, the contact is also queued to the EM_Promo2_sk skillset.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
QUEUE TO SKILLSET skillset_cv
WAIT 600
IF (IDLE AGENT COUNT EM_Promo2_sk > 2) THEN
   QUEUE TO SKILLSET EM_Promo2_sk
   WAIT 2
END IF
```

LOGGED AGENT COUNT

The LOGGED AGENT COUNT is the number of currently logged-on agents in the skillset list.

If you specify a list of skillsets, then the return value is the maximum of the LOGGED AGENT COUNT for the skillsets. This value is not the actual number of logged-on agents in all skillsets.

Use this intrinsic in loops to ensure that agents are logged on to a skillset while the contact is waiting to be answered.

This intrinsic applies to both voice and multimedia contacts.

Do not use the LOGGED AGENT COUNT intrinsic to test whether a skillset is in service. Although agents are logged on to a skillset, it does not mean that the skillset is in service. (You can place a skillset into out-of-service or transition mode from the Skillset configuration window under Configuration in Contact Center Manager Administration.) Instead, use the OUT OF SERVICE intrinsic to test whether a skillset is in service.

Syntax

LOGGED AGENT COUNT [<*skillset*> | <*skillset_list*>]

Parameters

Enter information for the following parameter:

- <*skillset*> or <*skillset_list*> The skillset, or list of skillsets, that you want to include in the return value.

Examples

This example shows how to test if enough agents are within a skillset to handle the current call volume. If enough agents are not available, the caller receives a busy tone.

```
IF (QUEUED COUNT sales_sk) > (2 * LOGGED AGENT COUNT sales_sk)
   THEN GIVE BUSY
END IF
QUEUE TO SKILLSET sales_sk
WAIT 2
```

This example shows how to test if there are enough agents within a skillset to handle the current multimedia contact volume. If there are not enough agents, the contact is disconnected. The contact is sent back to Contact Center Manager Server.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
IF (QUEUED COUNT skillset_cv) > (2 * LOGGED AGENT COUNT
   skillset_cv) THEN
   DISCONNECT
END IF
QUEUE TO SKILLSET skillset_cv
WAIT 2
```

LOGGED OUT AGENT

The LOGGED OUT AGENT intrinsic tests whether agents are logged off.

This intrinsic applies to both voice and multimedia contacts. This intrinsic only tests for agents at the local site. You cannot use this intrinsic to test for agents logged off network sites.

Syntax

LOGGED OUT AGENT <*agent ID*> | <*agent ID list*>

Parameters

Enter information for the following parameter:

- *<agent ID>* | *<agent ID list>* The ID, or list of IDs, of the agents you want to include in the return value.

Examples

In this example, the LOGGED OUT AGENT intrinsic is used to test if the preferred agent is available. If not, the contact is queued to a backup skillset.

```
IF LOGGED OUT AGENT agent_4 THEN
   QUEUE TO SKILLSET backup_sk
   WAIT 2
ELSE
   QUEUE TO AGENT agent_4
   WAIT 2
END IF
```

In this multimedia example, the LOGGED OUT AGENT intrinsic is used to test if the preferred agent is logged on. If not, the multimedia contact is queued to the skillset sent in the contact data.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
ASSIGN CONTACT DATA “%AGENT%AGENT” TO agent_cv
IF LOGGED OUT AGENT agent_cv THEN
   QUEUE TO SKILLSET skillset_cv
   WAIT 2
ELSE
   QUEUE TO AGENT agent_cv
   WAIT 2
END IF
```

LONGEST IDLE AGENT

The LONGEST IDLE AGENT intrinsic is the ID of the agent in the skillset list who is idle for the longest time.

The LONGEST IDLE AGENT is determined by either the idle time of the agent since the last contact disconnected or the total idle time of the agent since logging on. The algorithm is a system parameter configured by the administrator in the Global configuration screen under Configuration in Contact Center Manager Administration.

If you specify a list of skillsets, then the return value is the LONGEST IDLE AGENT for all the skillsets.

This intrinsic applies to both voice and multimedia contacts.

If you queue the contact to only one skillset, you need not use this intrinsic. The server automatically queues the contact to the agent who is idle for the longest time.

Syntax

LONGEST IDLE AGENT [*<skillset>* | *<skillset_list>*]

Parameters

Enter information for the following parameter:

- <*skillset*> or <*skillset_list*> The skillset, or list of skillsets, that you want to include in the return value.

BY LONGEST IDLE AGENT option

If you use the BY LONGEST IDLE AGENT option, and one or more agents are idle in the listed skillsets, the agent is selected based on the length of time the agent is idle.

The agent priorities in each skillset are considered, so only the front of each idle agent queue (for all the listed skillsets) is inspected for the longest idle agent. This option does not override the agent priorities within each skillset.

Examples

In this example, the call is queued to the skillset with the longest idle agent. If

the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

```
QUEUE TO SKILLSET sales_sk, service_sk BY LONGEST IDLE AGENT
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC pop_music_gv
SECTION WaitLoop
```

In this multimedia example, the contact is queued to the skillset with the longest idle agent.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
ASSIGN EM_Promo2_sk TO backup_skillset_cv
QUEUE TO SKILLSET skillset_cv, backup_skillset_cv BY
   LONGEST IDLE AGENT
WAIT 2
SECTION WaitLoop
```

MOST LOGGED AGENTS

The MOST LOGGED AGENTS intrinsic identifies the skillset with the most logged-on agents at the time the intrinsic is executed.

This intrinsic applies to both voice and multimedia contacts.

Script syntax

MOST LOGGED AGENTS <*skillset_list*>

Parameter

Enter information for the following parameter:

- *<skillset_list>* The list of skillsets that you want to include in the return value.

Examples

In this example, the value of the skillset with the most logged-on agents is assigned to a call variable named skillset_cv. The voice contact is then queued to the skillset represented by this variable. If the voice contact is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music. After this, the section named WaitLoop is executed. Every 30 seconds, this section checks whether the voice contact was answered and whether agents are available in the required skillset to answer the voice contact.

```
/* Always assign the skillset to a skillset call variable (skillset_cv) so that you
can check where the call was queued to */
ASSIGN MOST LOGGED AGENTS service_sk, support_sk TO skillset_cv
QUEUE TO SKILLSET skillset_cv
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC classical_music_gv
SECTION WaitLoop
   WAIT 30
   IF NOT QUEUED THEN
      IF OUT OF SERVICE skillset_cv THEN
         GIVE RAN sorry_closed_ran_gv
         DISCONNECT
      ELSE
         QUEUE TO SKILLSET skillset_cv
         WAIT 2
      END IF
   END IF
   GIVE RAN agents_still_busy_ran_gv
EXECUTE WaitLoop
```

In this multimedia example, the value of the skillset with the most logged-on agents is assigned to a call variable named skillset_cv. The contact is then queued to the skillset represented by this variable. Every n seconds, this section checks whether the contact is answered and if there are agents available in the required skillset to handle the multimedia contact.

```
/* Always assign the skillset to a skillset call variable (skillset_cv) so that you
can check where the contact was queued to */
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
ASSIGN EM_Promo2_sk TO backup_skillset_cv
ASSIGN MOST LOGGED AGENTS skillset_cv, backup_skillset_cv TO
   skillset_cv
QUEUE TO SKILLSET skillset_cv
WAIT 2
SECTION WaitLoop
WAIT waittime_cv
IF NOT QUEUED THEN
   IF OUT OF SERVICE skillset_cv THEN
      ASSIGN 120 TO waittime_cv
   ELSE
      QUEUE TO SKILLSET skillset_cv
      ASSIGN 60 TO waittime_cv
   END IF
END IF
EXECUTE WaitLoop
```

OLDEST

The OLDEST intrinsic is the age of the contact (with the specified priority, if you choose to include it) that waited longest in the given skillset queue when the intrinsic is executed.

If you specify a list of skillsets, then the return value is the maximum value of the OLDEST values for all the skillsets.

This intrinsic applies to both voice and multimedia contacts.

See AGE OF CALL on page 375 to compare.

Syntax

OLDEST [<*skillset*> | <*skillset_list*>] {WITH CALL PRIORITY <*priority*>}

The syntax OLDEST CALL is supported.

Optional

The WITH CALL PRIORITY segment is optional.

Parameters

Enter information for the following parameters:

- <*skillset*> or <*skillset_list*> The skillset, or list of skillsets, that you want to include in the return value.
- <*priority*> The priority of the contacts that you want to track.

Examples

In this example, if the oldest voice contact queued to the sales skillset is more than 5 minutes old, then the current call is queued to a backup skillset. Queuing the call to a backup skillset prevents more calls from being queued to the sales skillset. If the oldest call queued to the sales skillset is less than 5 minutes old, the script continues to queue the current call to that skillset.

```
IF (OLDEST sales_sk > 300) THEN
   QUEUE TO SKILLSET backup_sk
   WAIT 2
ELSE
   QUEUE TO SKILLSET sales_sk
   WAIT 2
END IF
```

In this multimedia example, if the oldest contact queued to the skillset sent in the contact data is more than 5 minutes old, then the current contact is queued to a backup skillset. This prevents further contacts from being queued to the skillset sent in the contact data. If the oldest contact queued to the skillset sent in the contact data is less than 5 minutes old, the script continues to queue the current contact to that skillset.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
ASSIGN EM_Promo2_sk TO backup_skillset_cv
IF (OLDEST skillset_cv > 300) THEN
   QUEUE TO SKILLSET backup_skillset_cv
```

```
   WAIT 2
ELSE
   QUEUE TO SKILLSET skillset_cv
   WAIT 2
END IF
```

OUT OF SERVICE

Use the OUT OF SERVICE intrinsic to test whether skillsets are out of service. If you specify a list of skillsets, then the OUT OF SERVICE intrinsic returns a True value if all skillsets are out of service. Otherwise, the intrinsic returns a False value.

A skillset is out of service when

- it is placed into OUT OF SERVICE mode in the Skillset Properties property sheet (in either night service or transition mode)
- all agents are logged off the skillset
- all agents are in standby mode in this skillset

This intrinsic applies to both voice and multimedia contacts.

Syntax

OUT OF SERVICE [<*skillset*> | <*skillset_list*>]

Parameters

Enter information for the following parameter:

- <*skillset*> or <*skillset_list*> The skillset, or list of skillsets, that you want to include in the return value.

Examples

In this example, if the sales skillset is out of service, then incoming calls receive night service treatment. Otherwise, they queue to the sales skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

```
IF OUT OF SERVICE sales_sk THEN
   EXECUTE Night_Section
END IF
QUEUE TO SKILLSET sales_sk
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC pop_music_gv
SECTION WaitLoop
```

In this multimedia example, if the skillset sent in the contact data is out of service, then incoming contacts run a section that checks the condition of the skillset every 1800 seconds. Otherwise, the contact queues to the skillset.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
IF OUT OF SERVICE skillset_cv THEN
   EXECUTE Night_Section
END IF
QUEUE TO SKILLSET sales_sk
```

```
WAIT 2
QUIT
SECTION Night_Section
   IF OUT OF SERVICE skillset_cv THEN
      WAIT 1800
      EXECUTE Night_Section
   ELSE
      QUEUE TO SKILLSET skillset_cv
      WAIT 2
      QUIT
   END IF
```

The Out of Service intrinsic checks only the condition of the local skillset, even if you select the include local node option. If no agents are available for the skillset on this node, but there are agents with that skillset on other nodes, the skillset is considered to be out of service when the command is executed. Only the local node skillset is checked. If no agents are logged on, or the skillset was placed into out-of-service mode using the client, callers immediately receive a closed message.

```
IF OUT OF SERVICE Sales_SK THEN
   GIVE RAN closed_ran
   DISCONNECT
ELSE
   QUEUE TO NETWORK SKILLSET Sales_SK
   WAIT 4
END IF
```

The system returns a value of type Boolean that indicates whether all specified local skillset queues at this node are out of service.

POSITION IN QUEUE

The POSITION IN QUEUE intrinsic is the position of the contact in the skillset queue at the time the intrinsic is executed.

The contact must be queued to the skillset before this intrinsic is used.

If you specify a list of skillsets, then the return value is the minimum value of the POSITION IN QUEUE values for all of the skillsets.

The position is calculated using the number of contacts queued in front of the current contact. This number includes all contacts with priorities equal to or higher than the priority of the current contact. For example, if the current contact is priority 2, then the calculation includes all priority 2 and priority 1 contacts.

This intrinsic applies to both voice and multimedia contacts.

Avaya recommendations:

- Use this intrinsic only for contact centers that do not assign priority levels.
- Do not use this intrinsic to play the queue position to a voice contact. Even in single-priority systems, a voice contact’s position in queue can become higher rather than lower.

Syntax

POSITION IN QUEUE [<*skillset*> | <*skillset_list*>]

Parameters

Enter information for the following parameter:

- <*skillset*> or <*skillset_list*> The skillset, or list of skillsets, that you want to include in the return value.

Examples

In this example, the contact is already queued to the service skillset. This section of the script tests the contact current position in the skillset queue. If it is more than 5, the priority is raised to priority 1 (unless it is already priority 1).

```
IF (POSITION IN QUEUE service_sk > 5)
   AND (PRIORITY IN QUEUE service_sk < > 1) THEN
       CHANGE PRIORITY IN SKILLSET service_sk TO PRIORITY 1
END IF
```

In this multimedia example, the contact is already queued to the skillset sent with the contact data. This section of the script tests the contact's current position in the skillset queue. If it is more than 5, the priority is raised to priority 2 (unless it is already priority 2).

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
ASSIGN CONTACT DATA “%PRIORITY%PRIORITY” TO priority_cv
QUEUE TO SKILLSET skillset_cv WITH PRIORITY priority_cv
WAIT 2
SECTION WaitLoop
WAIT looptimer_gv
IF (POSITION IN QUEUE skillset_cv > 5)
   AND (PRIORITY IN QUEUE skillset_cv < > 2) THEN
       CHANGE PRIORITY IN SKILLSET skillset_cv TO PRIORITY 2
END IF
EXECUTE WaitLoop
```

PRIORITY IN QUEUE

The PRIORITY IN QUEUE intrinsic is the priority of the current contact in the queue. This value can range from 1 to 10, with 1 being the highest priority and 10 being the lowest priority. A value of 0 indicates the contact is not currently in the queue.

This intrinsic applies to both voice and multimedia contacts.

The contact must be queued to the skillset before this intrinsic is used. The PRIORITY IN QUEUE intrinsic is used to check if the call is queued locally, even if the include local node option is selected for the network skillset.

Syntax

PRIORITY IN QUEUE <*skillset*>

Parameters

Enter information for the following parameter:

- <*skillset*> The skillset that you want to include in the return value.

Description

You can use the priority value in two ways.

You can use <> and = to make a direct comparison of the priority value. For example:

```
IF (PRIORITY IN QUEUE sales_sk = 0) THEN
IF PRIORITY IN QUEUE Customer_Service_Eng <> 1 THEN
```

Less than or greater than operations are not valid with a priority value.

```
IF (PRIORITY IN QUEUE Local_Sales_Sk > 0) THEN
```

You can declare a call variable of type PRIORITY and assign a priority to it and then perform the required comparison.

For example

```
ASSIGN 2 to cv_PriorityInQueue
IF (PRIORITY IN QUEUE Local_Sales_Sk > cv_PriorityInQueue) THEN
...
END IF
```

Examples

In this example, the script first checks if the contact is queued in the sales skillset. If this is true and agents are available to answer the contact in the service skillset, the contact is queued to the service skillset. Otherwise, the contact receives a message indicating that the contact center is closed.

```
IF (PRIORITY IN QUEUE sales_sk = 0) THEN
   IF NOT OUT OF SERVICE service_sk THEN
      QUEUE TO SKILLSET service_sk
      WAIT 2
   ELSE
      GIVE RAN sorry_closed_ran_gv
      DISCONNECT
   END IF
END IF
```

In this multimedia example, the script first checks if the contact is queued in the skillset sent with the contact data. If this is true and there are agents available to answer the contact in the skillset, the contact is requeued to the skillset. Otherwise, the contact is disconnected and Contact Center Multimedia requests the contact.

```
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
IF (PRIORITY IN QUEUE skillset_cv = 0) THEN
   IF NOT OUT OF SERVICE skillset_cv THEN
      QUEUE TO SKILLSET skillset_cv
      WAIT 2
   ELSE
      DISCONNECT
   END IF
END IF
```

Use DISCONNECT with caution as the multimedia contact is sent back to Contact Center Manager Server as a new contact and the time in queue is reset.

In this example, a secondary script is used to increase the priority of calls in the script.

Primary script:

```
QUEUE TO SKILLSET Customer_Service_Eng WITH PRIORITY 6
GIVE RAN ran_cv
EXECUTE SCRIPT Secondary_One
```

Secondary script Increase Priority

```
ASSIGN 3 TO prio_cv
IF PRIORITY IN QUEUE Customer_Service_Eng > prio_cv THEN
   CHANGE PRIORITY IN SKILLSET Customer_Service_Eng TO
   PRIORITY prio_cv
END IF
```

PRIORITY IN NETWORK QUEUE

The PRIORITY IN NETWORK QUEUE intrinsic is the priority of the current call in the network skillset queue. This value can range from 1 to 10, with 1 being the highest priority and 10 being the lowest priority. A value of 0 indicates the contact is not currently in the queue.

The call must be queued to the network skillset and then you must give a wait time of at least 4 seconds to ensure the target site can respond to the source site before this intrinsic is used.

Syntax

PRIORITY IN NETWORK QUEUE <*skillset*>

Parameters

Enter information for the following parameter:

- <*skillset*> The network skillset that you want to include in the return value.

Restriction

Do not use this intrinsic in the Network script (if you purchased the NSBR feature).

Description

You can use the priority value in two ways.

You can use <> and = to make a direct comparison of the priority value.

For example

```
IF (PRIORITY IN NETWORK QUEUE sales_sk = 0) THEN
   IF PRIORITY IN NETWORK QUEUE Customer_Service_Eng <> 1 THEN
```

Less than or greater than operations are not valid with a priority value.

```
IF (PRIORITY IN NETWORK QUEUE Local_Sales_Sk > 0) THEN
```

You can declare a call variable of type PRIORITY and assign a priority to it and then perform the required comparison.

For example

```
ASSIGN 2 to cv_PriorityInQueue
IF (PRIORITY IN NETWORK QUEUE Local_Sales_Sk > cv_PriorityInQueue) THEN
   ...
END IF
```

Examples

In the following example, if the priority of the call in the network skillset service_sk is 0, then the call is queued to the network skillset support_sk.

```
IF (PRIORITY IN NETWORK QUEUE service_sk = 0) THEN
   QUEUE TO NETWORK SKILLSET support_sk
   WAIT 10
END IF
```

In the following example, the include local node option is selected. The value returned is the priority (1 to 10) of the call in the remote node network skillset queue. This intrinsic returns a 0 (zero) if the call is not queued remotely. You can use this command to determine if the call is queued to a remote node.

```
IF PRIORITY IN NETWORK QUEUE Sales_Sk = 0 THEN
   QUEUE TO NETWORK SKILLSET Sales_Backup_Sk
   WAIT 4
END IF
```

QUEUED COUNT

The QUEUED COUNT intrinsic is the number of contacts outstanding against the skillsets in the given contact priority at the time the intrinsic is executed. If you specify a list of skillsets, then the return value is the maximum value of the QUEUED COUNT value for all skillsets.

This intrinsic applies to both voice and multimedia contacts.

This count includes only contacts not yet serviced and does not include contacts currently being presented to an agent.

QUEUED CALL COUNT is supported.

Syntax

QUEUED COUNT [<*skillset*> | <*skillset_list*>] {WITH CALL PRIORITY <*priority*>}

Optional

The WITH CALL PRIORITY segment is optional.

Parameters

Enter information for the following parameters:

- <*skillset*> or <*skillset_list*> The skillset, or list of skillsets, that you want o include in the return value.
- <*priority*> The priority of the calls that you want to track.

Examples

In this example, if the number of calls queued to the sales skillset is greater than twice the number of agents logged on to the sales skillset, then the caller is given a busy tone. Otherwise, the call is queued to the sales skillset. If the call is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

```
IF (QUEUED COUNT sales_sk) > (2 * LOGGED AGENT COUNT sales_sk)
   THEN GIVE BUSY
END IF
QUEUE TO SKILLSET sales_sk
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC soft_music_gv
SECTION WaitLoop
```

In this multimedia example, if the number of contacts queued to the skillset sent with the contact data is greater than twice the number of agents logged on to the skillset, the contact is disconnected and Contact Center Multimedia sends the contact back to Contact Center Manager Server to be requeued. Otherwise, the contact is queued to the skillset.

```
ASSIGN CONTACT DATA "%SKILLSET%SKILLSET" TO skillset_cv
IF (QUEUED COUNT skillset_cv) > (2 * LOGGED AGENT COUNT
   skillset_cv) THEN
      DISCONNECT
END IF
QUEUE TO SKILLSET skillset_cv
WAIT 2
SECTION WaitLoop
```

Use DISCONNECT with caution as the multimedia contact is sent back to Contact Center Manager Server as a new contact and the time in queue is reset.

Chapter 35: Time intrinsics

Time intrinsics are based on system time information, including the time of day, day of week, and day of year.

DAY OF WEEK

The value returned by this intrinsic is the current day of the week. You can use this intrinsic to determine whether the current day is a specific day or is in a list or range of days.

This intrinsic applies to both voice and multimedia contacts.

Syntax

DAY OF WEEK

Example

In this example, voice contacts coming into the contact center between 05:00 p.m. and 08:00 a.m. on Saturday and Sunday receive closed call treatment. Otherwise, voice contacts are queued to the general skillset. If the voice contact is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

You can use a variable to represent a specific period of time. For example, the variable named weekend_gv represents the days from Saturday to Sunday, and a variable named closed_hours_gv represents the hours from 05:00 p.m. to 08:00 a.m.

```
IF (DAY OF WEEK = weekend_gv) AND (TIME OF DAY =
   closed_hours_gv)
   THEN
      EXECUTE Closed_Section
END IF
QUEUE TO SKILLSET general_sk
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC pop_music_gv
SECTION WaitLoop
```

DATE

The value returned by this intrinsic is the current date. You can use this intrinsic to determine whether the current date is a specific date or is in a range of dates.

This intrinsic applies to both voice and multimedia contacts.

Syntax

DATE

Format

You can specify the month by using either the first three characters of the month (for example, Apr), or by spelling out the entire month (for example, September). In addition, the month can appear before or after the day.

Example

The following example uses both specific dates and a range of dates. Voice contacts coming in to the contact center on May 31, July 4, and December 25 to January 1 receive holiday call treatment.

You can use a variable to represent a specific period of time. For example, the variable named christmas_holiday_gv represents the days from December 25 to January 1.

```
IF (DATE = May 31, 4 July, christmas_holiday_gv) THEN
   EXECUTE Holiday_Section
END IF
```

DAY OF MONTH

The value returned by this intrinsic is the current day of the month. You can use this intrinsic to determine whether the current day is a specific day of the month (the 15th, for example), or is in a range of days (the 2nd through the 10th, for example).

This intrinsic applies to both voice and multimedia contacts.

Syntax

DAY OF MONTH

Format

Use a number or a list of numbers from 1 to 31.

Examples

BestAir Airlines has monthly flight specials. Voice contacts calling BestAir between the 27th and the 31st of each month hear a recorded announcement that gives information about the upcoming specials for the month.

You can use a variable to represent a specific period of time. For example, the variable named monthly_special_gv represents the days between the 27th and the 31st of each month.

```
IF (DAY OF MONTH = monthly_special_gv) THEN
   GIVE RAN specials_ran_gv
END IF
```

In this example, on the 1st and 15th day of each month, a section of the script called Pay_Day_Section is executed.

```
IF (DAY OF MONTH = 1,15) THEN
   EXECUTE Pay_Day_Section
END IF
```

MONTH OF YEAR

The value returned by this intrinsic is the current month of the year. You can use this intrinsic to determine whether the current month is a specific month or is in a range of months.

This intrinsic applies to both voice and multimedia contacts.

Syntax

MONTH OF YEAR

Examples

In this example, voice contacts entering the contact center between the months of December and February hear a recorded announcement describing winter sales.

You can use a variable to represent a specific period of time. For example, the variable named winter_months_specials_gv represents the months of December to February.

```
IF (MONTH OF YEAR = winter_months_specials_gv) THEN
   GIVE RAN winter_specials_ran_gv
END IF
```

In this example, callers hear a different recorded announcement depending on the month in which they contact the contact center. For example, in January, they hear a recorded announcement describing sales highlights for the month. If there is no holiday in a particular month, callers hear a default recorded announcement.

```
WHERE MONTH OF YEAR EQUALS
   VALUE January: GIVE RAN jan_highlights_ran_gv
   VALUE October: GIVE RAN oct_highlights_ran_gv
   VALUE November: GIVE RAN nov_highlights_ran_gv
   VALUE December: GIVE RAN dec_highlights_ran_gv
   DEFAULT: GIVE RAN default_ran_gv
END WHERE
```

TIME OF DAY

The value returned by this intrinsic is the current time of day. You can use this intrinsic to determine the exact time or whether the current time of day is in a specified range of time.

This intrinsic applies to both voice and multimedia contacts.

Syntax

TIME OF DAY

Format

hh:mm

Examples

In this example, voice contacts coming in to the contact center between 05:00 p.m. and 08:00 a.m. receive night contact treatment. Otherwise, voice contacts are queued to the general skillset. If the voice contact is not answered immediately, the caller hears a message indicating that all agents are busy, followed by music.

You can use a variable to represent a specific period of time. For example, the variable named closed_hours_gv represents the hours from 05:00 p.m. to 08:00 a.m.

```
IF (TIME OF DAY = closed_hours_gv) THEN
   EXECUTE Night_Section
END IF
QUEUE TO SKILLSET general_sk
WAIT 2
GIVE RAN agents_busy_ran_gv
GIVE MUSIC soft_music_gv
SECTION WaitLoop
```

To be meaningful, time comparisons using an exact time includes a greater than operator (>) or a less than operator (<).

For example

```
IF (TIME OF DAY = 08:00)
IF (TIME OF DAY < 08:00)
```

The first expression is true for 1 minute; the second is true from midnight until 7:59 a.m.

Chapter 36: Traffic intrinsics

Time intrinsics are based on the system traffic level information.

CALL RATE

The value returned by this intrinsic is the number of voice and non voice contacts that entered the system during the previous 10-minute period.

This intrinsic applies to both voice and multimedia contacts.

This intrinsic does not differentiate between voice and non voice contacts. The intent, for a future release, is to have CALL RATE reflect voice contacts only.

Syntax

CALL RATE

Example

In this example, when the call rate for the 10-minute period exceeds 50 voice contacts, callers hear a recorded announcement that states that the contact center experiences a high level of traffic.

```
IF (CALL RATE > 50) THEN
   GIVE RAN busy_ran_gv
   ...
END IF
```

CONTACT RATE

This intrinsic represents the number of voice and non voice contacts that enter the system during the previous 10-minute period.

This intrinsic applies to both voice and multimedia contacts.

This intrinsic does not differentiate between voice and non-voice contacts. The intent, for a future release, is to have CONTACT RATE reflect non-voice contacts only.

Syntax

CONTACT RATE

Example

```
IF CONTACT TYPE = g_EMAIL_gv THEN
   IF CONTACT RATE > 50 THEN
      IF TOTAL ACTIVE CONTACTS > 2500 THEN
      /* Special Handling */
      END IF
   END IF
END IF
```

TOTAL ACTIVE CALLS

The value returned by this intrinsic is the total number of voice contacts in the system at the moment the intrinsic is executed.

Active calls include all voice contacts in the system that are currently queued, are presented to agents, or are answered by agents. Active calls do not include abandoned, rejected, defaulted, or completed calls.

This intrinsic, TOTAL ACTIVE CALLS, reflect voice contacts only.

Syntax

TOTAL ACTIVE CALLS

Example

In this example, if the number of active voice contacts exceeds 200, then callers receive overflow treatment. You can restrict the number of active voice contacts to 200 to preserve shared phone lines used by other staff in your contact center.

```
IF (TOTAL ACTIVE CALLS >= 200) THEN
   GIVE OVERFLOW
END IF
```

TOTAL ACTIVE CONTACTS

The value returned by this intrinsic is the number of non-voice or multimedia contacts in the system at the time the intrinsic is executed.

Active contacts include all multimedia contacts in the system that are currently queued, are presented to agents, or are answered by agents. Active contacts do not include abandoned, rejected, defaulted, or completed contacts.

This intrinsic, TOTAL ACTIVE CONTACTS, reflect non-voice or multimedia contacts only.

Syntax

TOTAL ACTIVE CONTACTS

Example

```
IF CONTACT TYPE = g_EMAIL_gv THEN
   IF CONTACT RATE > 50 THEN
      IF TOTAL ACTIVE CONTACTS > 2500 THEN
         /* Special Handling */
      END IF
   END IF
END IF
```

Chapter 37: Open queue intrinsics

Open queue intrinsics are specific to multimedia routing on the Avaya Communication Server 1000 with the Open Queue feature installed and configured.

AGE OF CONTACT

This intrinsic represents the length of time the contact is in the system and is the non-voice equivalent to AGE OF CALL.

The AGE OF CONTACT intrinsic resets when the DISCONNECT command is used in a multimedia script because the DISCONNECT command sends the multimedia contact from Contact Center Multimedia to Contact Center Manager Server to be requeued.

Syntax

AGE OF CONTACT

Example

In this example, an e-mail contact receives special handling if the age of the contact is greater than 500 seconds.

```
IF CONTACT TYPE = g_EMAIL_gv THEN
   IF AGE OF CONTACT > 500 THEN
      /* Special Handling */
   END IF
END IF
```

CONTACT DATA

Use the CONTACT DATA intrinsic to return specific pieces of custom information associated with the contact. The intrinsic information is associated with the contact at time of contact creation and is stored with the contact in key or value string pairs. To retrieve the intrinsic value, the script must have explicit knowledge of the name of the intrinsic key.

The CONTACT DATA intrinsic is used with the ASSIGN TO script command. The value returned by CONTACT DATA is assigned to the variable specified in the ASSIGN TO command. This is the only use of CONTACT DATA. The CONTACT DATA key name is a string, which returns a string type by default; however, a specific number of alternative return data types are also supported. A convention is defined for identifying the specific data types: a defined string is used in the key name to specify the alternative return type.

Expected data may not be returned in CONTACT DATA. To handle these conditions, it is good practice to initialize the variable values used to store CONTACT DATA intrinsic values with suitable defaults in the script.

Syntax

ASSIGN CONTACT DATA <*key name*> TO <*call variable*>

Parameters

<*key name*>: The string name of the key used to store the intrinsic data. The key is unique for the specific piece of information. The following strings at the start of the key name are used to specify alternative return types:

- %AGENT%
- %DATE%
- %INTEGER%
- %PRIORITY%
- %SKILLSET%
- %TIME%

These strings are case-insensitive. However, the full key name must exactly match the key name entered when the contact is created.

<*call variable*>: A variable of a suitable type to store the value of the key. The default type for the variable is string. If an alternative data type is specified using the defined string at the start of the key name, then the type of the variable must match. Using an inappropriate variable for the returned type results in a script activation failure because of the incompatible types in the assignment.

Examples

This example specifies skillset and priority return types for direct queuing using information stored with the contact.

```
ASSIGN CONTACT DATA “%SKILLSET%mySkillset” TO skillset_cv
ASSIGN CONTACT DATA “%PRIORITY%myPriority” TO priority_cv
QUEUE TO SKILLSET skillset_cv WITH PRIORITY priority_cv
WAIT 2
```

This example specifies date and time return types for a callback using information stored with the contact.

```
ASSIGN CONTACT DATA “%TIME%myTime” TO time_cv
ASSIGN CONTACT DATA “%DATE%myDate” TO date_cv
SECTION wait_loop
IF (((date_cv = DATE)
   AND (time_cv > TIME OF DAY))
   OR (date_cv > DATE)) THEN
   WAIT 60
   EXECUTE wait_loop
ELSE
   EXECUTE wait_complete
END IF
SECTION wait_complete
```

This example specifies an integer return type defining the length of time a contact waits before being queued.

```
ASSIGN CONTACT DATA “%SKILLSET%mySkillset” TO skillset_cv
ASSIGN CONTACT DATA “%INTEGER%myWaitTime” TO integer_cv
WAIT integer_cv
QUEUE TO SKILLSET skillset_cv
WAIT 2
```

This example specifies an agent return type for direct queuing using information stored with the contact.

```
ASSIGN CONTACT DATA “%AGENT%myAgent” TO agent_cv
QUEUE TO AGENT agent_cv
WAIT 2
```

This example specifies a string for direct queuing using account type information stored with the contact.

```
ASSIGN CONTACT DATA “myAccountType” TO string_cv
WHERE string_cv EQUALS
   VALUE “GOLD”: QUEUE TO SKILLSET gold_ss
   VALUE “PLATINUM”: QUEUE TO SKILLSET platinum_ss
   VALUE “BRONZE”: QUEUE TO SKILLSET bronze_ss
   DEFAULT: QUEUE TO SKILLSET general_ss
END WHERE
WAIT 2
```

Contact Type

Numeric representation of the contact type.

ROUTE POINT

The ROUTE POINT intrinsic represents the anchor point for a contact and is a multimedia equivalent to the CDN for a voice contact to represent the route point for all contact types.

The ROUTE POINT is a software-only address you define and acquire in Contact Center Manager Administration. In Contact Center Multimedia, this intrinsic maps to multimedia skillsets. It does not exist in the switch.

This intrinsic is used for filtering in the master script. It executes primary scripts based on the ROUTE POINT of the contact, rather than just the contact type.

Syntax

ROUTE POINT

Example

In this example, contacts are first routed based upon contact types. Voice contacts receive treatment based upon their CDN, and e-mail and outbound contacts receive treatment based upon their ROUTE POINT.

```
/* Title: Master Script */
GIVE RINGBACK
WHERE CONTACT TYPE EQUALS
   VALUE c_VOICE_gv: EXECUTE CDNCheck
   VALUE c_EMAIL_gv: EXECUTE SCRIPT RoutepointCheck
   VALUE c_OUTBOUND_gv: EXECUTE SCRIPT RoutepointCheck
END WHERE
SECTION CDNCheck
WHERE CDN EQUALS
   VALUE 1001: EXECUTE SCRIPT Cust_Service_Eng
   DEFAULT: ROUTE CALL DEFAULT DN
END WHERE
SECTION RoutepointCheck
WHERE ROUTE POINT EQUALS
   VALUE “7000”: EXECUTE SCRIPT EM_Email_App
   VALUE “8000”: EXECUTE SCRIPT OB_Outbound_App
   DEFAULT: QUIT
END WHERE
/* Primary Script for E-mail Contacts */
ASSIGN CONTACT DATA “%SKILLSET%SKILLSET” TO skillset_cv
ASSIGN CONTACT DATA “PRIORITY%PRIORITY” TO priority_cv
ASSIGN EM_Default_Skillset TO default_skillset_cv
IF NOT OUT OF SERVICE skillset_cv THEN
   QUEUE TO SKILLSET skillset_cv WITH PRIORITY priority_cv
   WAIT 2
ELSE
   IF NOT OUT OF SERVICE default_skillset_cv THEN
      QUEUE TO SKILLSET default_skillset_cv WITH
      PRIORITY priority_cv
      WAIT 2
   ELSE
      EXECUTE NightCheck
   END IF
END IF
QUIT
SECTION NightCheck
WAIT looptimer
IF NOT OUT OF SERVICE skillset_cv THEN
   QUEUE TO SKILLSET skillset_cv WITH PRIORITY priority_cv
   WAIT 2
ELSE
   IF NOT OUT OF SERVICE default_skillset_cv THEN
      QUEUE TO SKILLSET default_skillset_cv WITH
      PRIORITY priority_cv
      WAIT 2
   ELSE
      EXECUTE NightCheck
   END IF
END IF
EXECUTE NightCheck
```

URI

Uniform Resource Indentifier (URI) represents the name of the location of an associated route point. In its current implementation, the URI is a free form string without validation and does not need to be unique to a route point. Any useful string can be entered as the URI value. The URI is available to make routing decisions in the script using the URI intrinsic.

Syntax

URI

Example

In this example, a contact is queued to an e-mail skillset if the contact type is e-mail, if the route point matches the route point global variable, and if the URI matches the global variable URI.

```
IF CONTACT TYPE = c_EMAIL_gv THEN
   IF ROUTE POINT = g_ROUTEPOINT_gv THEN
      IF URI = g_URI_gv THEN
         QUEUE TO SKILLSET EM_URI
      END IF
   END IF
END IF
```

Chapter 38: SIP intrinsics

Use this chapter to identify SIP intrinsics.

Important:

Intrinsics are pieces of data that may be available on a contact depending on factors such as the contact type. If a treatment is performed on a contact, or the contact is transferred, conferenced or routed to an agent, the intrinsic value may not be available.

SIP intrinsics are string types of name and value parameters used to extract intrinsic values. In SIP-enabled contact centers, the intrinsic value is made available using the new script command:

CONTACT DATA <*keyname*>

For example,

```
ASSIGN CONTACT DATA “Intrinsic1” TO string1
```

Contact Type

Numeric representation of the contact type.

CmfContactID

The unique internal ID for the contact in Contact Management Framework (CMF).

Provider

The mechanism that created the contact, represented by a short string. If the contact was created in open queue, the string is OQ. SIP contacts use the string SIP.

ProviderContactID

The external ID for the contact in Contact Management Framework (CMF).

SIP_CALL_ID

The SIP Call ID is the alphanumeric identification of the initial customer session. While it can be used as a unique key for parameter storage or retrieval to a remote database using HDX commands, a SIP Call ID is not maintained end-to-end as the contact traverses multiple proxies.

SIP_INITIAL_TEXT

The SIP Initial text is the first 80 characters of the first customer message to the contact center. You can use the CONTACT DATA command to retrieve the information for CONTAINS.

You can use the ASSIGN command to assign the first 80 characters of a customer's first message to the SIP_INITIAL_TEXT intrinsic.

```
ASSIGN CONTACT DATA "SIP_INITIAL_TEXT" TO c_sip_digits_str_cv
```

SIP_REQUEST_URI

The SIP Request URI is received in the initial INVITE intrinsic. The format of the URI is user@domain. This intrinsic is equivalent to the CDN.

SIP_FROM_ADDRESS

The SIP Contact Source is the value received in the From address in the SIP INVITE message. Its value is a URI that identifies the logical initiator of the request. For example, bcustomer@cc.avaya.com.

SIP_TO_ADDRESS

The value received in the To address in the SIP INVITE message. Its value is a URI that identifies the logical recipient of the request. For example, cdn1000@cc.avaya.com.

SIP_LOCATION

The value populated from the P-location SIP header if it is available. For example, P-Location: SM;origlocname="Banna";termlocname="Ardfert".

SIP_PREFERRED_LANGUAGE

The language associated with the contact source address. In the case of an Avaya Communication Server 1000 originated contact, it is based on the received locale header. For generic SIP solutions, the locale is the SIP accept-language header in the SIP INVITE message. If no language indication is received in the SIP message, this intrinsic returns en_us (US English).

The following ISO 639-1 Language Codes apply to the system:

Language	ISO Language Code	Country
Traditional Chinese	zh_cn	China
Japanese	ja_jp	Japan
English	en_us	Unites States
French	fr_fr	France
Korean	ko_kr	Korea
Simplified Chinese	zh_tw	China
German	de_de	Germany
Spanish	es_es	Spain

Example

```
IF CONTACT DATA SIP_PREFERRED_LANGUAGE = fr_fr THEN
   ASSIGN “French_welcome.wav” to prompttoplay
ELSE
   ASSIGN “default_welcome.wav” to prompttoplay
END IF
GIVE IVR
SERVICE URI serviceuri
```

```
WITH VXML TREATMENT voicexml
PARAMETERS prompttoplay
```

SIP_SUBJECT

The string value entered by the caller as the subject of the call. It is not mandatory for the caller to enter a header. If nothing is entered, the intrinsic contains an empty space.

SIP_USER_AGENT

Contains the details of the agent used by the customer to make the call. For example, the agent can use UCCP/2.0.6362.0 or OC/2.0.6362.0.

SIP_CALL_TYPE

Provides the primary contact type (media type) of the contact. In SIP Contact Center, this is normally either voice (c_contact_type_voice_gv) or instant message (c_contact_type_im_gv).

System-defined global variables are created for CONTACT TYPE:

- c_contact_type_voice_gv
- c_contact_type_im_gv

Example

The following example uses the CONTACT TYPE intrinsic.

```
GIVE RINGBACK
WAIT 5
WHERE CONTACT TYPE EQUALS
    VALUE c_contact_type_voice_gv:
    QUEUE TO SKILLSET Voice_Skillset
    VALUE c_contact_type_im_gv:
    QUEUE TO SKILLSET IM_Skillset
END WHERE
```

SIP_MAIN_CONTACT_ID

The cmf ID of the main contact used in transfers or conferences.

Skillset

The skillset to which to route the contact.

SIP_RETURNED_TEXT_n or SIP_RETURNED_DIGITS_n

The intrinsics used for storing the RETURNS <return_value> parameter of the GIVE IVR command. If the GIVE IVR command is used for an instant message, use the SIP_RETURNED_TEXT_n intrinsic. For voice, use the SIP_RETURNED_DIGITS_n intrinsic. The n represents the instance of GIVE IVR encountered as the script executes.

Chapter 39: P-Intrinsics SIP Header

Avaya Aura® Contact Center supports the custom P-Intrinsics private header. The Session Initiation Protocol (SIP) includes a number of message headers in each SIP message. These headers contain information that enables the receiver to understand and use the message properly. In a contact center solution, SIP headers may be used to transfer small amounts of call-related information between SIP-enabled applications. The application receiving this SIP message reads these headers and performs some action based on the contents of the headers. SIP header information can provide additional data about a call that applications can use to process that call.

You can use P-Intrinsics header information to pass context information between SIP-enabled applications. Avaya Aura® Contact Center parses the P-Intrinsics SIP header information and uses it to create Contact Intrinsics or Call Attached Data. You can use P-Intrinsics in conjunction with User-to-User (UUI) information if backwards compatibility with existing applications is required.

SIP private headers (P-Headers) are purely informational. They do not create new commands and they do not interfere with the regular transmission of SIP messages. SIP private headers are used only to pass extra information that the receiving application can use. Avaya Aura® Contact Center supports the P-Intrinsics SIP header in incoming SIP INVITE messages.

Components that support this private header include front-end IVRs systems such as Avaya Aura® Experience Portal and Avaya Voice Portal, SIP proxies such as Avaya Aura® Session Manager, or other SIP-enabled entities in the call flow.

P-Intrinsics information is not restricted by legacy limitations like UUI. P-Intrinsics information can grow in size, depending on other headers in the call, and on the call flow path. It can also be used to inject call attached data. It is therefore more flexible than UUI data. You can use both headers together, and customers can retain backwards compatibility with applications that already use UUI data.

Typical solution using P-Intrinsics

A front-end Avaya Aura® Experience Portal system uses XML speech applications and SIP header information to integrate with Avaya Aura® Contact Center. A self-service Voice XML speech application running on the Avaya Aura® Experience Portal – Application Server answers customer calls and gathers call-associated information based on customer's answers and inputs. Experience Portal then transfers the customer call, complete with this call-associated information stored in the P-Intrinsics SIP header, to Avaya Aura® Contact Center.

Contact Center uses the P-Intrinsics header to generate Contact Intrinsic and/or Call Attached Data specific to that call. If this call is ultimately answered by an agent, the agent can use the call-related Contact Intrinsic data to access customer details. The agent may receive the

Contact Intrinsic data in a screen pop, or they may need to access these details manually using Avaya Aura® Agent Desktop.

P-Intrinsics reduce the amount of time the agents spend on each call, improve the customer experience, and make Contact Center more efficient.

P-Intrinsics format

The P-Intrinsics SIP header uses both XML-formatting and hex-encoding. The amount of P-Intrinsics information associated with a call depends on the other SIP headers in the call and on the call flow path. Typically, Avaya Aura® Contact Center supports up to 10 ASCII key-value pairs of P-Intrinsics. Avaya Aura® Contact Center supports a key name of up to 25 characters and a value size of up to 80 characters.

If your solution has an Avaya Aura® Communication Manager in the incoming call path, the Refer-To header for blind transfers is limited to 1500 bytes overall.

If your solution has an Avaya Aura® Communication Manager in the call path, for improved P-Intrinsics support, Avaya recommends Communication Manager Release 6.0.1 SP8 or later.

Single intrinsic example

To insert a single intrinsic, start with:

```
<cc><i>CUSTOMER_SESSION_ID=12345</i></cc>
```

Do not use spaces, tabs, or carriage returns in XML.

When this single intrinsic example is encoded as hexadecimal, it looks like this:

```
3c63633e3c693e435553544f4d45525f53455353494f4e5f49443d31323334353c2f693e3c
2f63633e
```

To include the above in a P-Intrinsics header:

P-Intrinsics:
3c63633e3c693e435553544f4d45525f53455353494f4e5f49443d31323334353c2f693e3c2f
63633e

Compound intrinsics example

To insert more than one intrinsic use compound intrinsics. This compound example includes five intrinsics and call attached data:

```
<cc>
 <i>AAEP_FIRSTNAME=John</i>
 <i>AAEP_LASNAME=Doe</i>
 <i>AAEP_CUSTOMERID=12345</i>
 <i>AAEP_CUSTOMERPRIORITY=1</i>
 <i>AAEP_PREFERREDAGENT=3307</i>
 <cad>http://MyCRMSystem.avaya.com/accountid=12345</cad>
</cc>
```

For clarity this compound XML example is shown with carriage returns and tabs included, but you must write it all on one line without spaces, tabs, or carriage returns before you encode it into hexadecimal.

For example:

```
<cc><i>AAEP_FIRSTNAME=John</i><i>AAEP_LASNAME=Doe</
i><i>AAEP_CUSTOMERID=12345</i><i>AAEP_CUSTOMERPRIORITY=1</
i><i>AAEP_PREFERREDAGENT=3307</i><cad>http://MyCRMSystem.avaya.com/
accountid=12345</cad></cc>
```

When this compound example is encoded as hexadecimal, it looks like this (408 bytes):

```
3c63633e3c693e414145505f46495253544e414d453d4a6f686e3c2f693e3c693e414145505f5355524
e414d453d446f653c2f693e3c693e414145505f435553544f4d455249443d31323334353c2f693e3c69
3e414145505f435553544f4d45525052494f524954593d313c2f693e3c693e414145505f50524546464
5525245444147454e543d333330373c2f693e3c6361643e687474703a2f2f4d7943524d73797374656d
2e61766179612e636f6d2f6163636f756e7469643d31323334353c2f6361643e3c2f63633e20
```

Sample Voice XML applications using P-Intrinsics

Avaya Aura® Contact Center supplies sample applications that demonstrate how to transfer call context information using UUI and P-Intrinsics SIP headers.

Important:

Avaya Aura® Contact Center supplies generic sample applications for demonstration purposes. You must review the sample application code and customize it to your solution before deploying it on a contact center production server.

For more information, and to download the complete Avaya Aura® Experience Portal and Avaya Aura® Contact Center sample files, see Orchestration Designer Sample Applications on www.avaya.com/devconnect.

For more information, and to download the complete Avaya Voice Portal and Avaya Aura® Contact Center sample files, see Dialog Designer Sample Applications on www.avaya.com/devconnect.

Chapter 40: Troubleshooting applications

This chapter provides troubleshooting information and tips if you experience problems in your application writing process.

Application execution problems

The following sections describe problems with application execution.

Application does not validate

If the application executes a series of commands but then ignores any of the subsequent commands, you may not have included both an opening and a closing marker with comments inserted in the application. Comments in your applications must begin with an opening marker (/*) and end with a closing marker (*/). If you do not include both an opening and closing marker, the application does not validate.

Callers are caught in an endless loop

When using loops to give recorded announcements (GIVE RAN) to callers waiting in queue, application writers often forget to include a test to determine if the contact is answered. If queuing fails or if the contact is disconnected for some reason, the caller continues to hear a recorded announcement. However, the contact is never answered.

You must include a test, such as using the QUEUED or OUT OF SERVICE intrinsics, inside loops. Use the results of these tests to provide default treatments to the contact to ensure that it is successfully routed.

In this example, a voice contact is queued to the sales skillset, and then the caller is given a recorded announcement. Every 30 seconds, a loop is used to test whether the contact is in a queue or if agents are logged on to the sales skillset.

```
QUEUE TO SKILLSET sales_sk
WAIT 10
SECTION LoopForever
   IF NOT QUEUED THEN
      IF OUT OF SERVICE sales_sk THEN
         ROUTE CALL auto_att
      ELSE
         QUEUE TO SKILLSET sales_sk
         WAIT 2
      END IF
```

```
   END IF
   GIVE RAN please_wait_ran_gv
   WAIT 30
EXECUTE LoopForever
```

Calls are routed to a default ACD-DN because of a failure

If one of the commands below is the first command in an application, Contact Center does not take control of the voice contact. The switch routes the contact to the default ACD-DN configured for the CDN. The contact is not queued to the default skillset or RAN. Contact Center logs an error to the alarm monitor and event browser.

- WAIT
- QUIT
- GIVE SILENCE
- REMOVE FROM AGENT
- CHANGE PRIORITY IN AGENT
- REMOVE FROM SKILLSET
- CHANGE PRIORITY IN SKILLSET

The following commands in this list apply only if you purchased the NSBR option:

- REMOVE FROM NETWORK SKILLSET
- CHANGE PRIORITY IN NETWORK SKILLSET

The voice contact must first be given treatment using any other command before any of the preceding commands are used in an application.

Voice processing problems

The following sections describe problems with voice processing.

GIVE IVR treatment fails

If a GIVE IVR treatment to a voice messaging system such as Avaya CallPilot® fails, you must ensure the following details are correct:

Proposed Solution

Procedure

1. On Contact Center, the IVR DN is configured correctly and is acquired.
2. On the switch (in LD 23), the IVR ACD-DN is configured with IVR=YES and ALOG = YES.
3. On both the switch and the voice messaging system, voice ports are logged on and available.

Note:

Ports configured on the switch for the ACD DN must also be configured on Contact Center and Avaya CallPilot®. The configuration on the switch must match the configuration on Avaya CallPilot®.

GIVE CONTROLLED BROADCAST ANNOUNCEMENT command fails

If your applications fail after the GIVE CONTROLLED BROADCAST ANNOUNCEMENT command, ensure that you configure the value of the Number of IVR Ports Reserved for Broadcast variable in the IVR ACD-DN Global Settings window on the client.

If you configure the GIVE CONTROLLED BROADCAST ANNOUNCEMENT value to be greater than the number of ports in the IVR ACD-DN, contact processing suspends at this statement, even if no ports are free for the contact. The contacts queue until a connection to the port becomes free.

If you configure this value equal to or less than the number of ports in the IVR ACD-DN, the caller hears the announcement only if there is a port with free capacity to connect the voice contact. If all ports are used, the contact skips this statement and goes to the next statement in the script.

Application does not execute voice processing commands

When the application executes, it ignores the voice processing commands and continues to run after the voice processing commands. This behavior can be caused by a configuration problem on the switch, on Contact Center Manager Server, on Avaya CallPilot®.

Voice processing commands do not run consistently

When applications use the CONTROLLED BROADCAST or voice sessions commands, some callers hear the prompts, and others do not. This behavior can be caused by a configuration problem on the switch, on Contact Center Manager Server, onAvaya CallPilot®.

Callers hear silence instead of voice processing treatments

The script executes the voice processing commands, but callers hear silence instead of the prompts and announcements specified in the application. This behavior can be caused by a configuration problem on the switch, on Contact Center Manager Server, on Avaya CallPilot®. This behavior can also be caused by servers that are not synchronized or voice ports that are not in a correct state.

Callers hear the wrong message

Callers hear a message different from the one indicated by the voice processing command. This behavior can be caused by a configuration problem on the switch, on Contact Center Manager Server, on Avaya CallPilot®. This behavior can also be caused by the voice prompt variable referring to the wrong voice prompt file.

Callers hear only one of multiple voice processing treatments

At first, callers hear the correct voice processing treatment. Then, after some traffic, the script no longer runs the voice processing commands. This behavior can be caused by a configuration problem on the switch, on Contact Center Manager Server, or on Avaya CallPilot®.

Voice contacts suspend in the application at the voice processing statement

Applications advance to the voice processing commands, but do not continue beyond them. The application stops until the caller hangs up or an agent answers the voice contact (if the voice contact was previously queued). This suspension can be caused by a configuration problem on the switch, on Contact Center Manager Server, or on Avaya CallPilot®.

Callers wait too long to hear voice processing

The caller hears too many cycles of ringback before the message is played. This behavior can be caused by a configuration problem on the switch, on Contact Center Manager Server, or on Avaya CallPilot®.

Callers hear “Your voice session cannot be completed” message

The caller hears the message “Your Voice Session cannot be completed” instead of the prompts specified in the script. This behavior can be caused by a configuration problem on the switch, on Contact Center Manager Server, or on Avaya CallPilot®.

Caller hears voice prompts but is not presented to an agent

The caller hears the voice prompts, but is not connected to an agent. This behavior occurs if a transfer fails after a GIVE IVR command to a third-party voice processing system.

If the GIVE IVR command is used with a third-party IVR application or an Avaya CallPilot® voice menu to transfer a voice contact to a DN that is busy, Contact Center loses control of the contact. When you use the GIVE IVR command, always transfer the contact to an ACD-DN.

Callers hear a broadcast announcement too many times

The caller hears a broadcast announcement repeated several times. This behavior occurs only if the application repeats the same voice segment within the GIVE CONTROLLED BROADCAST command. For example:

```
GIVE CONTROLLED BROADCAST
   PLAY PROMPT VOICE SEGMENT welcome_vs
        VOICE SEGMENT welcome_vs
```

Networking problems

The following sections describe problems with network processing.

Application execution stops in the network application

This situation occurs only if you purchased the NSBR option. When a contact is networked and the agent at the destination site becomes unavailable for some reason (such as the agent

using the Not Ready key on the phone), the network application is executed. If the network application references a subscript that contains any of the following commands or intrinsic, application execution stops, the contact remains in the queue, and the caller continues to hear any previous treatments. Contact Center logs an error to the alarm monitor and event browser:

- GIVE BUSY
- GIVE BUSY CONTROLLED
- GIVE OVERFLOW
- GIVE OVERFLOW CONTROLLED
- QUEUE TO NACD
- QUEUE TO NETWORK SKILLSET
- CHANGE PRIORITY IN NACD
- CHANGE PRIORITY IN NETWORK SKILLSET
- REMOVE FROM NACD
- REMOVE FROM NETWORK SKILLSET
- PRIORITY IN NETWORK QUEUE (intrinisc)

QUEUE TO NETWORK SKILLSET command does not successfully route contact

This situation occurs only if you purchased the NSBR option. If you attempt to route a contact to another Contact Center site using the QUEUE TO NETWORK SKILLSET command, yet the contact does not arrive at the site, the network communication link between the sites may not be functioning properly.

You can use the PRIORITY IN NETWORK QUEUE intrinsic to determine if the contact was successfully routed to another Contact Center site.

Phantom voice contacts

A phantom call is a voice contact that is no longer active on the switch, but which the Statistical Data Manager or the call processing subsystem treats as though it is active. That is, the server continues to collect statistics for the voice contact and provide treatments for it.

Diagnosis

If you have voice contacts in your system for an extended period of time, those contacts can be phantom contacts. However, other conditions, such as incorrectly written applications, can make contacts remain in the system for a long time.

Voice contacts that are not treated correctly in the application show as Waiting in the real-time statistics but not in the skillset real-time statistics. For example, contacts that are queued to a skillset that is out of service or contacts that are unqueued show as Waiting in the real-time statistics but not in the skillset real-time statistics.

Check your applications to ensure that there are no infinite loops and that no contacts are queued to skillsets that are out of service. If no problems exist in your scripts, the oldest contacts in the system can be phantom contacts.

Possible causes

Phantom voice contacts can occur if the system is configured incorrectly. Make sure that you

Proposed Solution

Procedure

1. Install all of the latest patches.
2. Dedicate voice ports to Contact Center; the server must not share voice ports with other applications.
3. Create separate ACD-DNs for GIVE IVR and ACCESS voice ports.

Validation errors

This section contains a list of errors you can receive when you validate your application. Validations occur when you save your application and appear in the Problems view of Orchestration Designer.

Error message	Description
An end-of-file is encountered before the comment is terminated with an end-of-comment “*/”.	An end-of-comment symbol (*/) is missing. Insert the symbol at the end of commented text.
Syntax error or unsupported command.	A syntax error can occur when the validator cannot understand a command because of a missing or incorrect keyword. If you cannot see the error in the line of the application indicated in the Problems view, check the previous line for errors. For the correct syntax of commands, use the Script Command Reference section of this document.

Error message	Description
An end-of-file is encountered in the script in the middle of a statement.	There is an incomplete command in the line indicated by the Problems view. For the correct syntax of the command, use the Script Command Reference section of this document.
A carriage return is encountered in a string literal before the closing quotation marks.	A literal string is a string of characters surrounded by quotation marks. For example, “This is a string.” There is a hard return inside a string of quoted text. Remove the hard return.
An end-of-file is encountered in a string literal before the closing quotation marks.	A literal string is a string of characters surrounded by quotation marks. For example, “This is a string.” A closing quotation mark (”) is missing from the line indicated by the Problems view.
Literal strings cannot be longer than 80 characters.	A literal string is a string of characters surrounded by quotation marks. For example, “This is a string.” There are too many characters in the string of text Reduce the number of characters in the string to 80 or fewer.
Valid integer constants cannot be greater than 1999999999 or less than – 1999999999.	Ensure that the number on the line indicated by the Problems view is valid.
WildCLID constants cannot be longer than 32 digits.	An incorrect WildCLID constant is specified in the line indicated by the Problems view. Make sure that the WildCLID constant is 1–32 digits in length.
CDN or DNIS range is 1–30 digits.	Ensure that the CDN or DNIS number on the line indicated by the Problems view is 1–30 digits in length.
Agent IDs can be maximum 16 digits, skillsets 30.	An invalid agent ID or skillset is specified in the line indicated by the Problems view. Make sure that the agent ID is not longer than 16 digits or that the skillset is not longer than 30 alphanumeric characters.
Max day of month is 31.	An invalid day of month is specified in the line indicated by the Problems view. Make sure that the day of month is between 1 and 31.
Time of day ranges from 00:00 to 23:59.	An invalid time of day is specified in the line indicated by the Problems view. Make sure that the time of day is from 00:00 to 23:59.
Time format was incorrect (three-digit hours, and so on).	An invalid time format is specified in the line indicated by the Problems view. Make sure that you specify the time in the format hh:mm.
Valid music, RAN, route number is 0 to 511.	An invalid music, RAN, or route number is specified in the line indicated by the Problems view. Make sure that the number is from 0– 511.
DNs or CLIDs can be max 32 digits.	An invalid DN or CLID number is specified in the line indicated by the Problems view. Make sure that the number is 1 to 32 digits in length.

Error message	Description
Valid priority 1 to 10.	An invalid priority is specified in the line indicated by the Problems view. Make sure that the priority is between 1 and 10.
IDs (variable names) can be 30 characters max.	An invalid variable name is specified in the line indicated by the Problems view. Make sure that the variable name is from 1 to 30 characters in length.
Incorrect format for WildCLID was used (Wildcards @/? in WildCLID were used improperly.)	An invalid WildCLID is specified in the line indicated by the Problems view.
Char not valid in language ({, |, and so on) was encountered.	An invalid character is in the line indicated by the Problems view. Remove invalid characters.
Sets (lists and ranges) must contain elements of the same type, and variables cannot (currently) be mixed with constants, even if they are of the same type.	The list of values in the line indicated by the Problems view contains values of different types, or mixes constants with variables. Make sure that all of the values are the same type, and remove either the constants or the variables.
The variable specified is not the expected type.	The variable in the line indicated by the Problems view is not the correct type. Use a different variable, or change the variable type. For more information, see Application manager data and variable configuration on page 251.
An EXECUTE statement references a non-existent label.	An EXECUTE statement in the line specified by the Problems view references a section that does not exist. Make sure that the section exists and that it is referenced correctly (that is, use exact spelling) by the EXECUTE statement.
A label is defined in more than one place.	A section name is repeated in the application. Rename one of the sections.
In the Event Handler, no event can be handled more than once.	An event is repeated in the EVENT HANDLER statement. Remove the repeated event.
This statement cannot be the first statement in an application.	An invalid first statement is in the application. The following commands cannot be used as the first command in an application. • WAIT • QUIT • GIVE SILENCE • REMOVE FROM AGENT • CHANGE PRIORITY IN AGENT

Error message	Description
	• REMOVE FROM SKILLSET • CHANGE PRIORITY IN SKILLSET
A list (variable or constant) cannot be used in a prompt.	You can use only a variable of class Item or a single constant number for a prompt. If you use a variable for the prompt indicated (by line number) by the Problems view, make sure its class is Item.
Day in date is invalid (three digits, and so on).	An invalid day is specified in the date. Make sure that the correct day is specified. The day can contain no more than two digits. Valid date formats are May 4, 16 July, Dec 13.
Invalid date for month specified.	The date specified for the month in the line indicated by the Problems view is invalid. For example, Feb. 30 and Nov. 31 are invalid. Make sure that you specify the correct date.
Max 20 agents IDs in list.	Too many agent IDs are in the list specified by the Problems view. Make sure that the list contains no more than 20 agent IDs.
Too many skillsets are listed for this command.	Too many skillsets are listed in the line indicated. Skillset-related commands can use no more than 20 skillsets in the list of skillsets. If the command relates to an NSBR feature, no more than 10 skillsets can be listed.
Invalid type in Unary Minus operation.	A Unary Minus operation is an operation in which you change the sign of a mathematical expression using a negative sign (–) in front of the expression. (For example, ASSIGN –variable2 TO variable1.) The variable being negated must be type Integer or Seconds.
Invalid type (left side) in Addition operation.	The value on the left side of the Addition operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see Condition and Assignment builder operations on page 243.
Invalid type (right side) in Addition operation.	The value on the right side of the Addition operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see Condition and Assignment builder operations on page 243.
Invalid type (left side) in Subtraction operation.	The value on the left side of the Subtraction operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see Condition and Assignment builder operations on page 243.
Invalid type (right side) in Subtraction operation.	The value on the right side of the Subtraction operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see Condition and Assignment builder operations on page 243.
Invalid type (left side) in Multiplication operation.	The value on the left side of the Multiplication operation is invalid. The left and right sides must be type Integer or

Error message	Description
	Seconds. For more information, see Condition and Assignment builder operations on page 243.
Invalid type (right side) in Multiplication operation.	The value on the right side of the Multiplication operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see Condition and Assignment builder operations on page 243.
Invalid type (left side) in Division operation.	The value on the left side of the Division operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see Condition and Assignment builder operations on page 243.
Invalid type (right side) in Division operation.	The value on the right side of the Division operation is invalid. The left and right sides must be type Integer or Seconds. For more information, see Condition and Assignment builder operations on page 243.
Sets (left side) are not allowed in Addition operations.	A set (that is, more than one value) is included in the Addition operation. This is invalid. Ensure that there is only one value. If a variable is on the left side of the equation, ensure that the class of the variable is Item.
Sets (right side) are not allowed in Addition operations.	A set (that is, more than one value) is included in the Addition operation. This is invalid. Ensure that there is only one value. If a variable is on the right side of the equation, ensure that the class of the variable is Item.
Incompatible types in Less Than or Equal To relational expression.	The values in the Less Than or Equal To expression cannot be compared. Use only types Integer and Seconds.
Incompatible types in Greater Than relational expression.	The values in the Greater Than expression cannot be compared. Use only types Integer and Seconds.
Incompatible types in Greater Than or Equal To relational expression.	The values in the Greater Than or Equal To expression cannot be compared. Use only types Integer and Seconds.
Constants in range must be of same type.	Both ends in the range are not the same type. (For example, Monday..March is an invalid range because both ends of the range are not the same type.) Make sure that all values in the range are the same type.
In a WHERE-EQUALS statement the Where expression must be of the same type as the value lists.	Either the Where statement or one of the value statements in the WHERE-EQUALS command contains data of the wrong type. The Where value must be the same type as the values in the values list.
The Where expression in the WHERE-EQUALS statement cannot be a set.	The Where statement in the line indicated by the Problems view contains invalid data. Make sure that the value in the Where statement is a single value only. You cannot use sets of values. If you use a variable, make sure that its class is Item.

Error message	Description
Sets cannot be used on the left side of the Equal To operator.	The Equal To operator in the line indicated by the Problems view contains invalid data. Make sure that the value on the left side of the operator is a single value only. You cannot use sets of values. If you use a variable, make sure that its class is Item.
Sets cannot be used on the left side of the Not Equal To operator.	The Not Equal To operator in the line indicated by the Problems view contains invalid data. Make sure that the value on the left side of the operator is a single value only. You cannot use sets of values.
WildCLID constants cannot be assigned to call variables in an ASSIGN TO command.	The ASSIGN TO command in the line indicated by the Problems view contains an invalid value. Do not use a WildCLID with the ASSIGN TO command.
A set cannot be assigned to a call variable in an ASSIGN TO command.	The ASSIGN TO command in the line indicated by the Problems view contains an invalid value. Make sure that only a single value is assigned to the call variable. You cannot use sets of values.
Variable cannot be in a set.	One of the following problems is indicated: • The variable class is Set, but the variable is an invalid type (for example, Boolean). • The command cannot use a variable whose class is Set, a constant list, or a range.
Unknown variable type.	The variable in the line indicated by the Problems view is not defined or is an unknown type. Make sure that the variable is properly created and is the correct type for the command. If this error appears after validating the application, it can indicate a product problem. If this problem persists, report this error to your Avaya customer support representative.
Incompatible types in ASSIGN TO command.	The values in the ASSIGN TO command in the line indicated by the Problems view are not compatible.
<MaxDigits> cannot be a set type.	The maximum digits parameter must be a single value. Make sure that you specify only one value (not a list of values). If you use a variable to indicate the number of digits, make sure that the class is Item.
An undefined skillset is used.	An invalid skillset is specified in the line indicated by the Problems view. Make sure that the skillset exists and is properly referenced in the application. If you use a variable, make sure that it is created, that it is type Skillset, and that its value is a valid skillset.
An undefined agent ID is used.	One of the following problems occurred:

Error message	Description
	• A variable in the line indicated by the Problems view is not defined. This variable is expected to be type Agent. • No agent is defined with the agent logon ID specified in the line indicated by the Problems view. Make sure that the agent exists and is assigned the correct ID. If you use a variable, ensure that it is defined correctly and is type Agent.
An undefined application is referenced in the EXECUTE SCRIPT command.	An invalid script name is in the EXECUTE SCRIPT command in the line indicated by the Problems view. Make sure that the script exists and that it is properly referenced (that is, it is spelled correctly) in the EXECUTE SCRIPT command.
An undefined identifier is encountered.	A name (or an identifier) is used in the application, but no skillset, variable, or script is defined in the system with this name.
Illegal statement is used in Event statement.	An invalid statement is in the Event Handler. For a list of valid statements that you can use in the Event Handler, see EVENT HANDLER on page 319.
Division by zero is not allowed.	An error in the division operation is in the line indicated by the Problems view. You cannot divide by zero.
An EXECUTE/SECTION label should not be the same as a variable name or a skillset name.	Change the name of the SECTION label in the line indicated by the Problems view so that it is not the same as a skillset or variable name. The system is case-insensitive, so the label names must differ by more than just the case of the letters.
Only call variables can be used on the left side of the ASSIGN TO command.	The variable used with the ASSIGN TO command in the line indicated by the Problems view is a global variable, which is not correct. You must delete the variable and create it again, this time choosing Call as the variable type.
The variables used in third-party statements (SEND INFO, SEND REQUEST, GET RESPONSE) cannot be sets.	The variable class for the SEND INFO, SEND REQUEST, or GET RESPONSE variable is Item.
A GET RESPONSE statement must appear immediately after a SEND REQUEST statement.	A GET RESPONSE statement is missing after the SEND REQUEST statement.
No more than 10 variables are allowed as parameters in each third-party statement.	Too many variables are listed in the SEND INFO, SEND REQUEST, or GET RESPONSE command in the line number indicated by the Problems view. Remove as many variables as necessary.

Error message	Description
Third-party variables can be of the following types: DN, CLID, ACD, CDN, DNIS, LOC, NPA, NXX, NPANXX, String, Integer.	The variable in the line indicated by the Problems view is the wrong type. Change the type to one the following: DN, CLID, ACD, CDN, DNIS, LOC, NPA, NXX, NPANXX, String, Integer.
Invalid use of keywords or punctuation.	The line indicated by the Problems view contains an invalid keyword or invalid punctuation. For the correct syntax of a command, use the Script Command Reference in this document.
Missing parentheses.	A parenthesis is missing in the line indicated by the Problems view. Insert the parenthesis where appropriate.
Only variables of Boolean type are allowed.	The variable in the line indicated by the Problems view is a Boolean-type variable.
Invalid Boolean value. Must be True/1 or False/0.	In a WHERE-EQUALS command, if you use a boolean-type variable for the expression, the Value clauses must also be Boolean. However, you can also use a constant 1 or 0 to represent True or False. The constants 1 and 0 are the only valid constants.
TIMER variable must be seconds type.	The timer variable in the line indicated by the Problems view is a Seconds-type variable.
Only variables of DN and ACD type are allowed. Variables cannot be sets.	The variable type in the line indicated by the Problems view is invalid. The variables indicated must be the type DN or ACD. The class must be Item.
Statement is part of the Advanced Script Elements package. Package not purchased.	The command in the line indicated by the Problems view is an advanced command.
Advanced Scripts Package Query failed.	A server installation problem. Try to validate the application. If the problem persists, contact your Avaya customer support representative.
No SEND REQUEST before GET RESPONSE.	A SEND REQUEST statement is missing before the GET RESPONSE statement.
Invalid value assigned to this variable. Check limit for variables of this type.	The value assigned to the variable in the line indicated by the Problems view is invalid. Change the value of the variable. For more information, see Application manager data and variable configuration on page 251.
Range of seconds is 0 to 65535.	An incorrect number of seconds is specified in the line indicated by the Problems view. Make sure that the number of seconds you specify is from 0 to 65535.
Treatment range is 1 to 7 digits.	An incorrect treatment number is in the line indicated by the Problems view. Make sure that the treatment number is 1 to 7 digits in length.

Error message	Description
ACD range is 2 to 7 digits.	There is an incorrect ACD number in the line indicated by the Problems view. Make sure that the ACD is 2 to 7 digits in length.
Variables used in this context must be Boolean.	An invalid variable type is in the line indicated by the Problems view. Make sure that the variable is a Boolean-type (that is, has a True or False value) variable.
Numbers cannot be greater than 32 digits.	The number in the line indicated by the Problems view is not valid. Make sure that the number is no longer than 32 digits.
Agent ID expression cannot be compared with constant ranges. Use variable instead.	An expression or a statement with a return type of Agent ID cannot be compared with a list of constant Agent ID values or a range of Agent ID values. That is, IF LONGEST IDLE AGENT skillset1 = 1111, 2222 cannot be used. Use a set type variable instead.
Statement is part of Host Data Exchange API feature. Feature not purchased or error accessing database.	A third-party statement (SEND INFO, SEND REQUEST, GET RESPONSE) was encountered in the application. Either you did not purchase the feature or a database error occurred when the system accessed the features purchased.
Call Center Manager feature not purchased or error accessing database.	You did not purchase the package that includes the basic features for the system to work or a database error occurred when the system accessed the features purchased.
Intrinsic not supported for ICM protocol.	An intrinsic statement that is not supported for the ICM protocol was encountered in the script.
Call Data index out of range. Valid range is 1 to 10.	The value of the Call Data intrinsic is incorrect. Ensure that the value is an integer between 1 and 10.
Incomplete comment. Check that a previous comment was terminated properly.	“/*” specifies the beginning of a comment and “*/” specifies the end of a comment. This error is logged when a comment is not terminated properly or when another comment starts before a previous comment ends.
A Wildcard constant/variable cannot be used on the LHS of an Equal comparison statement.	In an IF statement comparison, a wildcard (a constant or a variable) cannot be on the left side of the equal sign (=). That is, you cannot use IF @23 = c_wildcard THEN...
A Wildcard constant/variable cannot be used on the LHS of a Not Equal comparison statement.	In an IF statement comparison, a wildcard (a constant or a variable) cannot be on the left side of the not equal symbol (<>). That is, you cannot use IF @23 <> c_wildcard THEN...
A Wildcard constant/variable cannot be used in the <exp> in a WHERE <exp> EQUALS statement.	In a WHERE statement, a wildcard (a constant or a variable) cannot be in the expression. That is, you cannot use WHERE c_wildcard EQUALS statement.

Appendix A: Reserved Keywords

The keywords in this section list all the reserved words that you cannot use.

Reserved keywords

Consult this table for a list of keywords that are valid regardless of the switch that you use.

%AGENT%	%DATE%
%PRIORITY%	%SKILLSET%
%TIME%	
ABANDON	ACD
ACTIVE	AGE
AGE OF CONTACT	AGENT
AGENTS	AHEAD
AND	ANSWER
ANSWERED	APPEND
APPEND MESSAGE	APPLICATION
APR	APRIL
ASSIGN	ASSIGN_CC_QUEUETOHIGH
ASSIGN_CC_QUEUETOLOW	ASSIGN_DEFAULT_SKILLSETS
ASSIGNED	AUG
AUGUST	AVERAGE
BUSY	BY
CALL	CALLS
CCCUSTOMBLOCK_2	CCCUSTOMBLOCK_3
CCCUSTOMBLOCK_4	CCCUSTOMBLOCK_5
CCCUSTOMBLOCK_6	CCCUSTOMBLOCK_7
CCCUSTOMBLOCK_8	CCFINISHBLOCK_14
CCFINISHBLOCK_18	CCFINISHBLOCK_9

CCIVRBLOCK_0	CCIVRBLOCK_15
CCIVRBLOCK_9	CCLOGICBLOCK_1
CCLOGICBLOCK_11	CCLOGICBLOCK_17
CCLOGICBLOCK_18	CCLOGICBLOCK_19
CCQUEUEBLOCK_12	CCQUEUEBLOCK_2
CCTREATMENTBLOCK_0	CCWAITBLOCK_12
CCWAITBLOCK_13	CCWAITBLOCK_16
CDN	CHANGE
CHARACTER	CHECK_ACTIVE_CONTACTS
CHECK_AGENT_AVAILABLE	CHECK_CDN_VALUE
CHECK_CLID_VALUE	CHECK_CONTACT_TYPE
CHECK_COUNT	CHECK_COUNTER
CHECK_DEFAULT_IN_SERVICE	CHECK_DNIS_VALUE
CHECK_IF_QUEUED	CHECK_NOT_QUEUED
CHECK_QUEUED	CHECK_QUEUE_TYPE
CHECK_SKILLSET_IN_SERVICE	CHECK_SKILLSET_OUT_OF_SERVICE
CHECK_TOTAL_ACTIVE_CONTACTS	CHECK_TRANSFERRED
CHECK_TRANSFER_STATE	CHECK_VOICE_CONTACT
CHINESE	CHK_DFLT_SS_IN_SERVICE
CHK_QUEUED	CHK_QUEUED_BOOLEAN
CK_TREATMENT_GIVEN	CLID
CONSULTED	CONTACT DATA
CONTACT TYPE	CONTAINS
CONTINUOUS	COUNT
CUSTOMERID	
DATA	DATE
DAY	DEC
DECEMBER	DEFAULT
DELAY	DEQUEUE
DIALED	DIGIT
DIGITS	DIRECT
DISCONNECT	DISCONNECT_0

DISTURB	DN
DNIS	DO
ELSE	EMERGENCY_ANNOUNCEMENT
END	END_SCRIPT
ENGLISH	EQUALS
EVENT	EXECUTE
EXPECTED	EXIT
EXIT_1	
FAIL	FALSE
FEB	FEBRUARY
FLOW_END	FOR
FOREVER	FORWARD
FRENCH	FRIDAY
FROM	FROMADDRESS
GERMAN	GET
GIVE	GIVERINGBACK
GIVE_FIRST_RAN	GIVE_MUSIC
GREETING_ANNOUNCEMENT	
HANDLER	HI_PR_SKILLSET_IN_SERVICE
HOLD	HOLIDAYS_ANNOUNCEMENT
ID	IDLE
IF	IN
INCOMING	INFO
INIT_VARS	INTER
INTO	IVR
JAN	JANUARY
JAPANESE	JUL
JULY	JUN
JUNE	
LANGUAGE	LEAST
LOC	LOCATION
LOG	LOGGED

LONGEST	LOPRIOR_SKILLSET_IN_SERVICE
MAR	MARCH
MAY	MONDAY
MONTH	MOST
MULTIMEDIA	MUSIC
NACD	NAME
NEXTYEAR	NIGHT
NO	NON
NOT	NOT_QUEUED
NOV	NOVEMBER
NPA	NPANXX
NUMBER	NUMBERBYDIGIT
NXX	
OCT	OCTOBER
OF	OLDEST
ON	OPEN
OR	OUT
OUT_OF_HOURS_ANNOUNCEMENT	OUT_OF_SERVICE_ANNOUNCEMENT
OUT_OF_SERVICE_SKILLSETS	OVERFLOW
PASSWORD	PLAY_MUSIC
PORTUGUESE	POSITION
PRESENT	PRIORITY
PRIORITY_1	PROVIDE_FIRST_RAN
PROVIDE_MUSIC	
Q_HI_PRIORITY_SKILLSET	Q_LOW_PRIORITY_SKILLSET
QUEUE	QUEUE_TO_AGENT
QUEUE_TO_DEFAULT	QUEUE_TO_DEFAULT_SKILLSET
QUEUE_TO_DEF_SKILLSET	QUEUE_TO_SKILLSET
QUEUED	QUEUED_0
QUEUES	QUEUETYPE
QUIT	
RAN	RATE

RECEIVED	RELEASED
REMOVE	REQUEST
REROUTE_TO_DN	RESPONSE
RETRIEVE	RETURN
RINGBACK	RINGING
ROUTE	ROUTE_CONTACT
ROUTE_DEFAULT_DN	ROUTE_TO_DN
ROUTE_TO_PRIMARY	ROUTEPOINT
SATURDAY	SCRIPT
SECOND_RAN	SECTION
SEND_WAIT_MESSAGE	SEND
SEND MESSAGE	SEP
SEPTEMBER	SERVICE
SESSION	SET
SET_DEFAULTS	SILENCE
SKILLSET	SKILLSET_QUEUING
SKILLSET_QUEUING_LOOP	SPANISH
SPEED	SUCCESS
SUNDAY	SUSPEND_CONTACT
TERMINATING	THEN
THURSDAY	TIME
TIMEOUT	TIMER
TO	TOADDRESS
TOTAL	TRUE
TUESDAY	TYPE
URI	
VALUE	VOICE
WAIT	WAIT_0
WAIT_2	WAIT_6
WAIT_DELAY	WAIT_TIME_120_SEC
WAIT_TIME_300_SEC	WAIT_TIME_600_SEC
WAIT_TIME_60_SEC	WAIT_TIME_900_SEC

WAITING	WEDNESDAY
WEEK	WHERE

Reserved keywords for Avaya Communication Server 1000 PBX

The words in this list are applicable only if your switch is an Avaya Communication Server 1000 PBX. If you use this switch, consult both this list and the list that includes all other keywords.

ANNOUNCEMENT	INTERNATIONAL	PROMPT
BROADCAST	INTERRUPTIBLE	SEGMENT
COLLECT	NETWORK	TRANSFERRED
CONFERENCED	NONCONTROLLED	TREATMENT
CONTROLLED	PLAY	VOICE

Appendix B: Icons Rebranded in Orchestration Designer Release 6.2

The tables in this appendix list the changes in the icons and block items from Avaya Aura® Contact Center Service Creation Environment Release 6.0/6.1 to the rebranded Avaya Aura® Contact Center Orchestration Designer Release 6.2.

View and Editor icons changes

The following table shows changes in view or editor icons.

Table 26: View/editor icons

New Icon	Old Icon	Description
		Contact Center View
		Local View
		Synchronization View

General icons changes

The following table shows the changes in general icons.

Table 27: General icons

New Icon	Old Icon	Description
		Script
		Application

Flow Operation icons changes

The following table shows the changes in flow operation icons.

Table 28: Flow operation icons

New Icon	Old Icon	Description
		Zoom In
		Zoom Out
		Select Zoom Window
		Zoom Reset
		Fit To Contents
		Incremental Layout
		Full Layout
		Cut
		Paste
		Delete

Flow Node icons changes

The following table shows the changes in the flow node icons.

Table 29: Flow node icons

New Icon	Old Icon	Description
		Queue block - Contact routing to skill-set, agent.
		Input block - Prompt or collect digits from user.

New Icon	Old Icon	Description
		Output block - Play RAN announcement or Voice Segments as Broadcast/Voice sessions.
		Host block - Send or request and retrieve data from external database or web service.
		Logic block - Create a logical decision point such as IF_THEN_ELSE or WHERE_EQUALS statements.
		Custom block - Create advanced scripting code.
		Reference block - Execute another application or script.
		Treatment block - Provides music, ring back or silence.
		Wait block - Suspends processing and extends the previous operation for the defined time.
		IVR block - Provide interactive response such as IVR treatment, IM, Web push and so on.
		Menu block - Provide DTMF Menu.
		Anchor - Allow event handler custom block to exist in flow without connecting to another block.
		Anonymous - Allow connection lines between blocks to be moved for visual presentation.
		Source shortcut - Jump to a destination block in the flow.
		Destination shortcut - Continue execution from a Source.
		Finish block - Allow contact to be disconnected, given busy, overflow, or routed to a DN.
		Exit block - End of contact execution.

Appendix C: Commands and intrinsics

The tables in this appendix list all the commands and intrinsics and compares their use with Avaya Communication Server 1000, Open Queue, and SIP.

Commands–General

The following table contains the general commands and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
APPEND	Add a literal string or sting call variable to the existing variable.	Yes	Yes	Yes	
APPEND MESSAGE	Add the current message string in the message buffer with the new string.	No	No	Yes	
ASSIGN MESSAGE	Set or replace the current message string in the message buffer with the new string.	No	No	Yes	
ASSIGN TO	Assign values to contact variables.	Yes	Yes	Yes	
EXECUTE	A GOTO label type instruction where the label is in the current script.	Yes	Yes	Yes	
CONTAINS	Search a string for another string, including wildcards.	Yes	Yes	Yes	
EXECUTE SCRIPT	A GOTO script type instruction.	Yes	Yes	Yes	
IF-THEN-END IF	Conditional execution.	Yes	Yes	Yes	

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
IF-THEN-ELSE-END IF	Conditional execution.	Yes	Yes	Yes	
QUIT	Stops script execution.	Yes	Yes	Yes	
SECTION	Label	Yes	Yes	Yes	
SEND MESSAGE	Send the statement currently in the message buffer.	No	No	Yes	
WAIT	Pause script execution for this contact/call for a specified number of seconds.	Yes	Yes	Yes	

Commands–Call Procedures

The following table contains the call procedures commands and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
CHANGE PRIORITY IN AGENT	Change the priority of a call/contact queued to an agent's queue using the QUEUE TO AGENT command.	Yes	Yes	Yes	
CHANGE PRIORITY IN SKILLSET	Change the priority of a call/contact in a skillset queue.	Yes	Yes	Yes	
DISCONNECT	Disconnect the contact/call.	Yes	Yes	Yes	
GIVE BUSY	Give a busy tone to the caller.	Yes	No	Yes	For SIP: Tone is localized through the Default Locale Setting.
GIVE MUSIC	Give music to the caller.	Yes	No	Yes	

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
GIVE OVERFLOW	Give an overflow tone to the caller.	Yes	No	Yes	For SIP: Tone is localized through the Default Locale Setting.
GIVE RAN	Give a recorded announcement to the caller.	Yes	No	Yes	For SIP: RAN is localized through the Default Locale Setting.
GIVE RINGBACK	Give ringback to the caller.	Yes	No	Yes	For SIP: Tone is localized through the Default Locale Setting.
GIVE SILENCE	Give silence to the caller.	Yes	No	Yes	
QUEUE TO AGENT	Queue the call to a local agent identified by the agent ID.	Yes	Yes	Yes	
QUEUE TO SKILLSET	Queue the call to a local skillset.	Yes	Yes	Yes	
REMOVE FROM AGENT	Remove the call from the queue of the agent specified by the agent ID.	Yes	Yes	Yes	
REMOVE FROM SKILLSET	Remove the call from the local skillset queue.	Yes	Yes	Yes	
ROUTE CALL	Route the call to a DN.	Yes	No	Yes	For SIP: This command is supported only for numeric targets. For example, it does not support SIP addresses; aliases only.

Commands–Advanced

The following table contains the advanced commands and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
EVENT HANDLER	A specific handler for unsolicited events (Call on-hold/call abandon) and failure events (RAN/IVR/ Broadcast announcement response failures).	Yes	No	Yes	
LOG	Logs a text message to the event browser.	Yes	Yes	Yes	
READVAR/ SAVEVAR	Allows contact/call variables to retain their value across calls.	Yes	Yes	Yes	
WHERE-EQUALS	Switch type statement.	Yes	Yes	Yes	

Commands–Networking

The following table contains the networking commands and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
CHANGE PRIORITY IN NACD	Change the priority of the call in the specified Network ACD.	Yes	No	No	
CHANGE PRIORITY IN NETWORK SKILLSET	Change the priority of the call in the specified network skillset.	Yes	Yes	Yes	
QUEUE TO NACD	Queue call to specified Network ACD.	Yes	No	No	For SIP: NACD features are not applicable.
QUEUE TO NETWORK SKILLSET	Queue the call to the specified network skillsets.	Yes	Yes	Yes	

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
REMOVE FROM NACD	Remove the call from the specified Network ACD.	Yes	No	No	For SIP: NACD features are not applicable.
REMOVE FROM NETWORK SKILLSET	Remove the call from the specified network skillsets.	Yes	Yes	Yes	

Commands–Voice Processing

The following table contains the voice processing commands and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
COLLECT DIGITS	Collect digits from the caller using IVR.	Yes	No	No	For SIP: Replaced by GIVE IVR enhancements.
GIVE CONTROLLED BROADCAST ANNOUNCE—MENT	Broadcast an announcement to the caller using IVR.	Yes	No	No	For SIP: Broadcast not supported.
GIVE IVR	Deliver the call to IVR.	Yes	No	No	For SIP: This command is replaced by GIVE IVR with integrated VXML service invocation and open VXML service invocation support.
GIVE IVR WITH VXML	VXML treatment for a SIP CC contact.	No	No	Yes	
OPEN/END VOICE SESSION	Open/Close the IVR voice port.	Yes	No	No	For SIP: This command is replaced by GIVE IVR enhancements.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
PLAY PROMPT	Play a prompt to the caller using IVR.	Yes	No	No	For SIP: This command is replaced by GIVE IVR enhancements.

Commands–HDX

The following table contains the HDX commands and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
GET RESPONSE	Response to a GET REQUEST.	Yes	Yes	Yes	
SEND INFO	Send data to a third-party application.	Yes	Yes	Yes	
SEND REQUEST	Request for data from a third-party.	Yes	Yes	Yes	

Intrinsics–Skillset

The following table contains the skillset intrinsics and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
ANSWERED COUNT	Number of calls answered in the preceding 10 minutes.	Yes	Yes	Yes	
AVERAGE SPEED ANSWER	Average speed for answering calls in the preceding 10 minutes.	Yes	Yes	Yes	
EXPECTED WAIT TIME	Real-time expected wait time for the call based on the last 10 minutes of activity.	Yes	Yes	Yes	

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
IDLE AGENT	Returns the idle status of the specified agent.	Yes	Yes	Yes	
IDLE AGENT COUNT	Number of idle agents for the skillset.	Yes	Yes	Yes	
LOGGED AGENT COUNT	Number of agents logged on to the skillset.	Yes	Yes	Yes	
LOGGED OUT AGENT	Returns the logged out status of the specified agent.	Yes	Yes	Yes	
MOST LOGGED AGENTS	Returns the skillset in a skillset list with the most agents logged on.	Yes	Yes	Yes	
OLDEST	Returns the age of the oldest contacts in a skillset.	Yes	Yes	Yes	
OUT OF SERVICE	Checks whether the skillset is out of service.	Yes	Yes	Yes	
POSITION IN QUEUE	Returns the position of the contact in the skillset queue.	Yes	Yes	Yes	
PRIORITY IN QUEUE	Return the priority of the contact in the specified skillset queue.	Yes	Yes	Yes	
PRIORITY IN NETWORK QUEUE	Return the priority of the contact in the specified network skillset.	Yes	Yes	Yes	
QUEUED COUNT	Number of contacts queued against the skillset at the defined priority.	Yes	Yes	Yes	

Intrinsics–Time

The following table contains the time intrinsics and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
DATE	Returns the date to the script as a DATE data type.	Yes	Yes	Yes	
DAY OF MONTH	Returns the day of month to the script as a DAY OF MONTH data type.	Yes	Yes	Yes	
DAY OF WEEK	Returns the day to the script as a DAY OF WEEK data type.	Yes	Yes	Yes	
MONTH OF YEAR	Returns the month of year to the script as a MONTH OF YEAR data type.	Yes	Yes	Yes	
TIME OF DAY	Returns the time to the script (hh:mm) as a Time data type.	Yes	Yes	Yes	

Intrinsics–Traffic

The following table contains the traffic intrinsics and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
CALL RATE	Number of calls that entered the system in the last 10 minutes.	Yes	No	Yes	
CONTACT RATE	Number of contacts that entered the system in the last 10 minutes.	No	Yes	No	
TOTAL ACTIVE CALLS	Total number of active calls in the system.	Yes	No	Yes	
TOTAL ACTIVE CONTACTS	Total number of active contacts in the system.	No	Yes	No	

Intrinsics–Call

The following table contains the call intrinsics and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
AGE OF CALL	Age of the contact.	Yes	No	Yes	
CALL FORWARD	The call is forwarded.	Yes	No	No	For SIP: Not supported in SIP Contact Center.
CALL FORWARD BUSY	The call is forwarded because the original destination was busy.	Yes	No	No	For SIP: Not supported in SIP Contact Center.
CALL FORWARD DO NOT DISTURB	The call is forwarded because the Do Not Disturb was enabled at the original destination.	Yes	No	No	For SIP: Not supported in SIP Contact Center.
CALL FORWARD NO ANSWER	The call is forwarded because the Call was not answered at the original destination.	Yes	No	No	For SIP: Not supported in SIP Contact Center.
CDN	Controlled DN of the call.	Yes	No	Yes	For SIP: This is an alias for the SIP URI configured in CCMA.
CLID	Calling line identification of the call.	Yes	No	No	
CONFERENCED	Consultation call for a conference.	Yes	No	No	For SIP: Not supported in SIP Contact Center.
CONSULTED	Returns true if the call is a consultation call.	No	No	No	
DIALED DN	Original number dialed where the call	Yes	No	Yes	

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
	was forwarded to a CDN.				
DIRECT CALL	A direct call to a CDN.	Yes	No	No	
DNIS	Dialed Number Identification Service.	Yes	No	Yes	For SIP: This command numeric alias for DNIS URI configured through CCMA. (SIP To: Address).
INTERNATIONAL CALL	Returns the status of the call as an international call.	Yes	No	No	
LOC	Returns the Location Code of the call.	Yes	No	No	
NETWORK CALL	Indicates if the call is a networked call.	Yes	No	No	For SIP: Not supported in SIP Contact Center.
NPA	Returns the Number Plan Area.	Yes	No	No	
NPANXX	Returns the Number Plan Area and the Local Exchange Code.	Yes	No	No	
NXX	Returns the Local Exchange Code.	Yes	No	No	
ON HOLD	Returns true if the call is on hold.	Yes	No	Yes	
QUEUED	Returns true if the call is already queued.	Yes	Yes	Yes	
ROUTE NUMBER	Returns the route number of the call.	Yes	No	No	
TRANSFERRED	Returns true if the call is transferred.	Yes	No	No	

Intrinsics–Multimedia

The following table contains the multimedia intrinsics and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
AGE OF CONTACT	Age of the contact.	No	Yes	Yes	
CONTACT DATA	Specific data associated with a contact at the time of contact creation. Uses multimedia-related key names: • %AGENT% • %DATE% • %PRIORITY% • %SKILLSET% • %TIME%	Yes	Yes	Yes	
CONTACT TYPE	Contact type for the contact. Returned as CONTACT_ TYPE.	Yes	Yes	Yes	
ROUTE POINT	Route point of the contact.	No	Yes	Yes	For SIP: This command returns the same SIP route point.
URI	URI of the route point.	No	Yes	Yes	The command returns the URI as a string.

Intrinsics–SIP

The following table contains the SIP intrinsics and a brief description.

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
ContactType	Numeric representation of the contact type.	No	No	Yes	
CmfContactID	Unique internal ID of this contact in contact management framework.	No	No	Yes	
PROVIDER	What component created the contact (Open Queue, SIP)	No	No	Yes	
PROVIDERCONTACTID	External ID of this contact in contact management framework.	No	No	Yes	
SIP_REQUEST_URI	Request URI is received in the initial INVITE intrinsic.	No	No	Yes	
SIP_FROM_ADDRESS	Value received in the From address in the SIP INVITE message.	No	No	Yes	
SIP_TO_ADDRESS	Value received in the To address in the SIP INVITE message.	No	No	Yes	
SIP_LOCATION	Value received from the SIP header, if available.	No	No	Yes	
SIP_CALL_ID	Identification of the initial customer session.	No	No	Yes	
SIP_PREFERRED _LANGUAGE	Language associated with the contact source address.	No	No	Yes	
SIP_SUBJECT	String entered by the caller as the subject of the call.	No	No	Yes	
SIP_USER_AGENT	Details about the agent used by the customer to make the call.	No	No	Yes	
SIP_CALL_TYPE	Contact type (media type) of the SIP contact.	No	No	Yes	

Command	Description	Avaya CS 1000	Open Queue	SIP	Notes
SIP_MAIN_CONTACT	The cmf ID of the main contact used in transfers or conferences.	No	No	Yes	
Skillset	The skillset to which to route the contact.	Yes	Yes	Yes	
SIP_RETURNED_TEXT_n or SIP_RETURNED_DIGITS_n	Used to store the RETURNS parameter of the GIVE IVR command.	No	No	Yes	

Index

Special Characters

%AGENT%AGENT 391
%DATE%DATE 392
%i0 216
%INTEGER%WAITTIME 393
%n0 216
%PRIORITY%PRIORITY 393
%SKILLSET%SKILLSET 394
%TIME%TIME 395

A

about block symbols 145
about contact center view 98
about flow editor 100
about instant messages 368
about integrated reporting 91
about intrinsics 371
about local view 99
about Orchestration Designer 15
about problems 101
about script editor 100
about session management timers 97
about synchronization view 99
about thresholds 273
accelerator keys 146
accepting updates 139, 140
access classes for application manager data 251
ACD route call to 233
ACD-DN routing error 446
activating applications 120
activating SOA Web service 277
active applications 35
adding applications to Contact Center view 134
adding applications to Local view 136
adding blocks to applications 144
adding blocks using accelerator keys 146
adding commands 239
adding data to the Local view 136
adding expressions 245
adding flow pages 164
adding prompts 30
adding resources for contact center 21
adding variables to Local view 136
adjusting arrangement of flow 236
AGE OF CALL 375
AGE OF CONTACT 429
age of contact intrinsic 413
agent assigned intrinsic 391
agent assignment 181
agent check for idle intrinsic 407
agent creation 254
agent desktop custom field intrinsic 397
agent idle count intrinsic 407
agent longest idle intrinsic 410
agents changing priority command 305
agents logged off intrinsic 409
agents number logged intrinsic 408
agents queue to command 313
agents remove assignment command 315
agents skillset queue command 314
AML and contact center integrated reporting 94
AML IVR routing 212
AML RAN 192
anchor node about 145
anchor node creation 235
anchor node example 235
annoucments recorded command 311
announcement command 348
announcement error 447
announcements in flow 190, 192
announcements recorded example 64
anonymous node about 145
anonymous node example 236
anonymouse node creation 236
ANSWERED COUNT 403
APPEND 305
APPEND MESSAGE 369
application changes 47
application changing properties 116
application creation 43, 110
application creation procedures 43
application demonstration 51
application manager data access classes 251
application manager data adding to OD 257
application manager data creation 254
application manager data deletion 257
application manager data refreshing list in Orchestration Designer 269
application manager data renaming 256
application manager data view properties 256
application network error 449
application pathways 156, 159

application planning 37
application preferences 289
application resolving errors 130
application sample 51
application upgrades 33
application validation errors 445
applications activating 120
applications adding blocks 144
applications adding notes 165
applications adding pages 164
applications adding to Contact Center view 134
applications adding to Local view 136
applications changing to Contact Center view 135
applications check variable references 266
applications checking for errors 125
applications closing 123
applications connecting pages 237, 238
applications converting scripts to flows 124
applications copying and pasting 160
applications deactivating 121
applications default 25
applications deleting 123
applications deleting from Contact Center view 135
applications deleting items 161, 162
applications deleting multiple items 162
applications deleting pages 164
applications disconnecting blocks 163
applications error checking 125
applications existing in Orchestration Designer 35
applications exporting 119
applications fitting in window 166
applications for routing 29
applications generating documents for 126
applications importing 119
applications layout 163
applications line number 240
applications moving in window 168
applications opening existing 115
applications printing 127
applications removing from Local view 138
applications renaming 121
applications renaming blocks and nodes 151
applications renaming pages 165
applications resizing 167
applications saving 122
applications script adding 239
applications selecting threshold class 275
applications summary of blocks 169
applications undo/redo change 241
applications updating conflicted 138
applications updating Contact Center view 139
applications updating in both views 140
applications updating in Local view 137, 140
applications zoom in 166
applications zoom out 167
area code and local exchange intrinsic 386
area code intrinsic 385
arranging blocks 163
assign agent in flow 181
ASSIGN command to assign contact data 429
ASSIGN MESSAGE 368
assign network skillset in flow 179
assign networked ACD in flow 183
assign paths based on caller feedback 185
assign skillset in flow 174
ASSIGN TO 299
assignment adding 245
assignment expressions 243
assignment expressions deleting 250
assignment locking example 148
automatic saving preferences 288
Avaya CallPilot® error 446
Avaya CallPilot® voice services 343
Avaya Media Server IVR integrated reporting 92
Avaya Mentor videos 17
AVERAGE SPEED ANSWER 404

B

block functions 145
block general information 147
block palette in flow editor 143
block summary viewing 169
blocks adding to application 144
blocks deleting 161
blocks disconnecting 163
blocks layout 163
blocks renaming 151
blocks using notes 165
both views updating applications 140
branch to script command 300
branch to section command 299
broadcast announcement command 348
broadcast announcements error 447
busy block creation 230
busy tone command 309

C

calculation operations 246
call age 375
call CDN 379

CALL DATA 375
call delays 211
CALL FORWARD 377
CALL FORWARD BUSY 378
CALL FORWARD DO NOT DISTURB 378
CALL FORWARD NO ANSWER 379
call instrinsics example 373
call is queued logic block example 77
CALL RATE 425
call routing command 316
call routing termination 229, 230, 232, 233
call treatment IVR command 340
call types SIP 438
call variable assignment command 299
call variables example 196
callback time intrinsic 393
caller feedback examples 295
caller information 185
caller ringback 207
calling line identification (CLID) 379
calls default routing error 446
calls on hold indicator 386
calls total number active 426
CCMA application accessibility 95
CCMA data refresh 292
CCMA default preferences 287
CCMA node creation 116
CCMA node deletion 117
CCMA properties 117
CCMA starting Variables tool 261
CCMS data refresh 292
CCMS node creation 109
CCMS node deletion 118
CCMS properties 118
CCMS reading preferences 287
CDN 379
CDN creation 254
cemoving applications 123
CHANGE PRIORITY IN AGENT 305
CHANGE PRIORITY IN NACD 331
CHANGE PRIORITY IN NETWORK SKILLSET 328
CHANGE PRIORITY IN SKILLSET 306
changes in application undo/redo 241
changing application name 121
changing applications 47
changing call variable command 321
changing names of blocks and nodes 151
changing properties of application 116
changing variable properties 259, 267
checking applications 125
checking for errors 122
checking variable references 266
choices transition example 60
choosing threshold classes 275
classes threshold creating 273
CLID 379
client accessibility 95
closed days logic block example 58
closed hours logic block example 68
closed hours output block example 71
closing applications 123
code generation preferences 289
coding event handling in block 203
COLLECT DIGITS 342
collecting digits in SIP 216
commands adding to scripts 239
commands for instant messages 368
comments in script applications 124
condition testing command 322
conditional assignments for input, output, and queue blocks 152
conditional commands 300
conditional expression example 61, 70
conditional expressions 243
conditional settings for flow 172
conditional transitions 156
CONFERENCED 381
configuration thresholds 273
configuration Web services and DIW 277
configuring CCMA and CCMS example 52
configuring details for blocks in applications 147
configuring master script 113
configuring network preferences 287
configuring resources 21
conflicted applications update 138
conflicted view items 133
connect several pages of application 237, 238
connecting blocks with conditions 156
connecting blocks with multiple conditions 159
connecting blocks with no conditions 156
connecting nodes example 59
CONSULTED 381
contact age 375
contact age intrinsic 429
contact agent assigned intrinsic 391
contact center and AML integrated reporting 94
contact center and IVR integrated reporting 91
Contact Center and Local views updating variables 140
Contact Center perspective 291
contact center resources 21
Contact Center start 19
contact center view 98

Contact Center view adding applications 134
Contact Center view changing applications 135
Contact Center view copying from 114
Contact Center view deleting applications 135
Contact Center view showing or hiding 108
Contact Center view updating applications 139
Contact Center views and Local views updating data 140
contact customer id 397
CONTACT DATA 429
contact date assigned intrinsic 392
contact in queue intrinsic 419
Contact Management Framework 435, 436
contact name of route point intrinsic 433
contact priority intrinsic 393
CONTACT RATE 425
contact removal from skillset command 316
contact route point intrinsic 431
contact router configuration 113
contact routes 29
contact routing command 370
contact skillset assigned intrinsic 394
contact subtype intrinsic 396
contact termination command 308
contact time presented to agent intrinsic 395
contact time to callback intrinsic 393
contact type intrinsic 431, 435
contacts average speed to answer intrinsic 404
contacts number enter in 10 minutes 425
contacts number enter system 425
contacts number intrinsic 403
contacts removing from network skillset command 330
contacts total number active 426
contacts unassign command 315
CONTACTSUBTYPE 396
CONTAINS 307
context-sensitive help 293
controlled busy command 309
controlled overflow command 310
controlled route call command 316
controlled routing URI command 370
CONVERT 308
converting scripts to flows 124
copying block example 81
copying blocks 160
copying data from Contact Center view to Local view 114
copying variable properties 268
creating application manager data 254
creating applications 43
creating assignments 152
creating CCMA node 116
creating CCMS node 109
creating flow application 110
creating hard copy 127
creating pathways for applications 156, 159
creating script application 110
creating threshold classes 273
creating variable example 55
creating variables in CCMA 261
creating variables with multiple values 264
creating variables with one value 262
custom block about 145
custom block creation 203
custom information from contact intrinsic 429
custom option in SIP 221
CUSTOMERID 397
CUSTOMFIELD 397

D

data adding to Local view 136
data copying 114
data creating a variable with multiple values 264
data creating a variable with one value 262
data exchange 353
data for applications creation 254
data for applications deletion 257
data for applications rename 256
data for applications view properties 256
data from contact intrinsic 429
data of caller 375
data refresh 292
data refreshing in Orchestration Designer 269
data removing from Local view 138
data requests command 356
data restrictions 251
data sending command 354
data set removing a value 260
data updating in both views 140
data updating in Local view 137, 140
Database Integration Wizard 277
DATE 421
date of assigned intrinsic 392
DAY OF MONTH 422
DAY OF WEEK 421
deactivating applications 121
decision-making example 172
default applications 25
default routing error 446
default transitions 156
default wait preferences 289
delay call processing 211
deleting application manager data 257

deleting application, application manager data or variables from Local view 138
deleting applications 123
deleting applications from Contact Center view 135
deleting blocks 161
deleting CCMA nodes 117
deleting CCMS nodes 118
deleting contact from NACD command 334
deleting contacts from network skillset command 330
deleting expressions 250
deleting flow pages 164
deleting multiple objects 162
deleting nodes 161
deleting transitions 161, 162
deleting variables 269
deliver calls to IVR command 340
demonstration for creating applications 51
descriptions of blocks 145
desktop folder exporting 119
destination shortcut about 145
destination shortcut creation 238
details for blocks 147
determine network call 385
determine out of service skillsets 414
determining agent assigned to contact 391
determining agents logged off intrinsic 409
determining call is queued 387
determining call is transferred 388
determining call on hold 386
determining callback time for contact 393
determining contact subtype 396
determining custom field content 397
determining date of contact 392
determining incoming number of call 388
determining priority assigned to contact 393
determining skillset assigned to contact 394
determining time contact assigned to agent 395
DIALED DN 382
digit collection command 342
DIRECT CALL 382
directing calls based on DTMF 223
DISCONNECT 308
disconnect block creation 229
disconnect example 66
disconnect time intrinsic 429
disconnecting transitions 163
display thresholds 274
displaying flow applications 236
displaying problems view 129
displaying synchronization view 132
DIW configuration with Web services 277
DN route call to 233
DNIS 383
DNIS creation 254
documenting applications 126
downloading Orchestration Designer 21
DTMF input directions 223

E

e-mail message destination intrinsic 402
e—mail message received intrinsic 398
edit toolbar in flow editor 143
editing applications 47, 116
editing expressions 246
editor flow 100
editor preferences 289
editor script 100
enabling applications 120
end execution command 301
END VOICE SESSION 343
endless loop error 445
error checking applications 122, 125
errors from validation 451
erros in application location 130
Event Browser command 320
EVENT HANDLER 319
event handling in flows 203
example activating application 90
example call intrinsics 373
example conditional expression 61, 70
example connecting nodes 59
example copying block 81
example creating logic block 58, 68, 72, 77
example creating logical transition 60
example creating message output block 80, 82
example creating output block 71
example creating queue to skillset 74
example creating treatment block 76
example creating variable 55
example creating wait block 84
example disconnect block 66
example exiting calls 85
example of adjusting position of nodes 236
example of anchor node 235
example of anonymous node 236
example of call variables to play messages 196
example of configure CCMA and CCMS 52
example of decisions 172
example of different languages 226
example of give music command 359
example of host block to receive data 153, 200
example of host block to send data 153, 200

example of locked assignments 148
example of logic block 148, 172
example of playing messages 226
example of playing multiple messages 196
example of queue to skillset 175
example of skillset queue 175
example playing music 76, 359
example recorded announcements 64
example reviewing inconsistencies 86
example running application 90
example skillset intrinsics 372
example synchronizing views 86
example time intrinsics 373
example traffic intrinsics 373
examples for caller feedback 295
EXECUTE 299
EXECUTE SCRIPT 300
existing applications and variables 35
existing applications open 115
exit block about 145
exit node creation 238
EXPECTED WAIT TIME 405
exporting applications 119
expressions adding 245
expressions deleting 250
expressions for conditions and assignments 243
external flow linking 235
external ID for CMF 436

F

feedback for callers 295
field custom intrinsic 397
file printing variables to 270
final treatments 229
final treatments busy 230
final treatments overflow 232
final treatments route call 233
finding errors in applications 125, 130
finish block about 145
fit zoom to 166
flow application moving 168
flow applications adding blocks 144
flow applications anchor node 235
flow applications anonymous node 236
flow applications busy block 230
flow applications conversion from scripts 124
flow applications custom block 203
flow applications destination shortcut 238
flow applications disconnect block 229
flow applications exit 238
flow applications input block 185
flow applications IVR block 212
flow applications IVR custom block 221
flow applications IVR play prompts and collect digits block 216
flow applications IVR play prompts block 214
flow applications IVR send IM block 219
flow applications logic block 172
flow applications menu block 223
flow applications overflow block 232
flow applications play annocucements block 192
flow applications play announcements block 190
flow applications play music block 206
flow applications play prompts block 194
flow applications queue to agent block 181
flow applications queue to network ACD block 183
flow applications queue to network skillset block 179
flow applications queue to skillset block 174
flow applications reference block 204
flow applications request information block 199
flow applications ringback treatment block 207
flow applications route contact block 233
flow applications send information block 197
flow applications silent treatment block 209
flow applications source shortcut 237
flow applications wait block 211
flow editor operations 143
flow editor view 100
flows adding new pages 164
flows deleting pages 164
flows renaming pages 165
flows resizing 167
forward call 377
forward call for no answer 379
forward call if busy 378
forward call to do not disturb 378
FROMADDRESS 398
functions of blocks 145

G

generating documentation for applications 126
GET RESPONSE 357
getting started 19
GIVE BUSY 309
GIVE CONTROLLED BROADCAST ANNOUNCEMENT 348
GIVE CONTROLLED BROADCAST ANNOUNCEMENT error 447
Give controlled broadcast command 343
GIVE IVR 340
GIVE IVR for SIP 362
GIVE IVR troubleshooting 446

GIVE MUSIC 309
GIVE MUSIC example 359
GIVE OVERFLOW 310
GIVE RAN 311
GIVE RINGBACK 311
GIVE SILENCE 312
GIVE-IVR treatments 94
AML and contact center integrated reporting 94
global preferences 289
group command 301
guidelines for caller feedback 295

H

halt activity command 302
hard copies of applications 127
HDX application references 197, 199
help viewing 293
hiding Contact Center view 108
hiding line numbers in scripts 240
hiding Local view 108
hold indicator 386
hold message output block example 82
holiday conditional expression example 61, 70
host block about 145
host block example 153, 200
host command 356
Host Data Exchange 353

I

icons for block types 145
icons inactive applications 35
icons invalid applications 35
icons valid applications 35
iconsactive applications 35
IDLE AGENT 407
IDLE AGENT COUNT 407
IF THEN ELSE END IF 300
IF THEN END IF 300
importing applications 119
importing Web services 277
inactive application 35
incoming contacts answering speed intrinsic 404
incoming contacts number intrinsic 403
incoming number intrinsic 388
input block creation 185
input block overview 145
input blocks conditional assignments 152
instant messages 368
instant messages append command 369
instant messages assign command 368
instant messages send command 369
Instant messaging in SIP 219
integrated reporting AML and contact center 94
integrated reporting Avaya Media Server IVR 92
integrated reporting information 91
integrated reporting IVR and contact center 91
integrating Web services 277
internal ID for CMF 435
INTERNATIONAL CALL 383
interrupible IVR command 340
intrinsics %AGENT%AGENT 391
intrinsics %DATE%DATE 392
intrinsics %INTEGER%WAITTIME 393
intrinsics %PRIORITY%PRIORITY 393
intrinsics %SKILLSET%SKILLSET 394
intrinsics %TIME%TIME 395
intrinsics about 371
intrinsics AGE OF CALL 375
intrinsics AGE OF CONTACT 429
intrinsics ANSWERED COUNT 403
intrinsics AVERAGE SPEED ANSWER 404
intrinsics CALL DATA 375
intrinsics call example 373
intrinsics CALL FORWARD 377
intrinsics CALL FORWARD BUSY 378
intrinsics CALL FORWARD DO NOT DISTURB 378
intrinsics CALL FORWARD NO ANSWER 379
intrinsics CALL RATE 425
intrinsics CDN 379
intrinsics CLID 379
intrinsics CONFERENCED 381
intrinsics CONSULTED 381
intrinsics CONTACT DATA 429
intrinsics Contact Management Framework 435, 436
intrinsics CONTACT RATE 425
intrinsics Contact Type 431, 435
intrinsics CONTACTSUBTYPE 396
intrinsics CUSTOMERID 397
intrinsics CUSTOMFIELD 397
intrinsics DATE 421
intrinsics DAY OF MONTH 422
intrinsics DAY OF WEEK 421
intrinsics DIALED DN 382
intrinsics DIRECT CALL 382
intrinsics DNIS 383
intrinsics EXPECTED WAIT TIME 405
intrinsics FROMADDRESS 398
intrinsics IDLE AGENT 407
intrinsics IDLE AGENT COUNT 407
intrinsics INTERNATIONAL CALL 383

intrinsics LOC 384
intrinsics LOGGED AGENT COUNT 408
intrinsics LOGGED OUT AGENT 409
intrinsics LONGEST IDLE AGENT 410
intrinsics MONTH OF YEAR 423
intrinsics MOST LOGGED AGENTS 411
intrinsics NETWORK CALL 385
intrinsics NEXTYEAR 399
intrinsics NPA 385
intrinsics NPANXX 386
intrinsics NXX 386
intrinsics OLDEST 413
intrinsics ON HOLD 386
intrinsics OUT OF SERVICE 414
intrinsics POSITION IN QUEUE 415
intrinsics PRIORITY IN NETWORK QUEUE 418
intrinsics PRIORITY IN QUEUE 416
intrinsics QUEUED 387
intrinsics QUEUED COUNT 419
intrinsics QUEUETYPE 400
intrinsics return value 371
intrinsics ROUTE NUMBER 388
intrinsics ROUTE POINT 431
intrinsics SIP 435
intrinsics skillset example 372
intrinsics time example 373
intrinsics TIME OF DAY 423
intrinsics TOADDRESS 402
intrinsics TOTAL ACTIVE CALLS 426
intrinsics TOTAL ACTIVE CONTACTS 426
intrinsics traffic example 373
intrinsics TRANSFERRED 388
intrinsics URI 433
introduction to Orchestration Designer 15
items adding to application 144
items on menu 101
items on toolbar 106
IVR and contact center integrated reporting 91
IVR block about 145
IVR block creation 212
IVR call data storage 375
IVR command 340
IVR custom block creation 221
IVR in AML environment 94
IVR integrated reporting on Avaya Media Server 92
IVR play prompts and collect digits block creation 216
IVR play prompts block creation 214
IVR send IM block creation 219

K

keys for contact data intrinsics 429

L

language messages example 196
languages for menu example 226
languages for SIP 437
latest version of Orchestration Designer 21
layout applications 163
leaving application 238
line number particular in script 240
line numbers in scripts 240
link to external flow 235
linking applications 237, 238
LOC 384
Local and Contact Center views updating data 140
Local and Contact Center views updating variables 140
local exchange and area code intrinsic 386
local exchange intrinsic 386
local machine accessibility 95
local view 99
Local view adding applications 136
Local view adding data 136
Local view adding variables 136
Local view copying to 114
Local view importing 119
Local view object synchronization 132
Local view removing applications 138
Local view removing data 138
Local view removing variables 138
Local view renaming applications 121
Local view showing or hiding 108
Local view updating applications 137, 140
Local view updating data 137, 140
Local view updating variables 137, 140
locating errors in application 130
locating errors in applications 125
locking assignments example 148
LOG 320
LOGGED AGENT COUNT 408
logged on most intrinsic 411
LOGGED OUT AGENT 409
logic block about 145
logic block creation 172
logic block example 58, 68, 72, 77, 148, 172
logic transition example 60
logical operators 246
LONGEST IDLE AGENT 410
longest idle agent command 314
longest idle to agent command 313
loop error 445

M

magnifying 166
management timers 97
manager data for applications adding to OD 257
manager data for applications creating 254
manager data for applications deleting 257
manager data for applications rename 256
manager data for applications viewing properties 256
master application 29
master script 29
master script configuration 113
MATCHES 312
mathematical operators 246
mathematical order of operations 246
media management 30
Media Server and IVR integrated reporting 92
Media Server RANs 190
menu block about 145
menu block creation 223
menu items 101
menu prompt example 226
message output block example 80, 82
messages append command 369
messages assign command 368
messages send command 369
messaging in SIP 219
minimizing script maintenance 297
modes of operation 95
modifying applications 47
month day intrinsic 422
MONTH OF YEAR 423
MOST LOGGED AGENT 411
most logged agents command 314
moving applications from previous releases 33
moving flow application 168
multimedia contact subtype intrinsic 396
multimedia contact termination command 308
multimedia routing command 316
multiple conditions command 300
multiple message example 196
multiple objects deleting 162
multiple results command 322
multiple-value variables creating 264
music on hold playing 206
music playing command 309
music playing example 76

N

NACD changing priority command 331
NACD deleting contact command 334
NACD queue command 333
network application error 449
NETWORK CALL 385
network connection preferences 287
network queue priority intrinsic 418
network queue skillset command 329
network skillset assignment 179
network skillset priority command 328
network skillset removing contacts command 330
networked ACD queue block 183
NEXTYEAR 399
node connection example 59
node creation CCMA 116
node deletion for CCMA 117
node deletion for CCMS 118
nodes deleting 161
nodes renaming 151
notes in blocks 165
NPA (number plan area) 385
NPANXX 386
number of incoming contacts intrinsic 403
number prompts command 346
numbers selecting incoming 388
NXX 386

O

object synchronization 132
OLDEST 413
ON HOLD 386
on hold music in flows 206
one-value variables creating 262
online help 293
OPEN VOICE SESSION 343
opening existing applications 115
opening Orchestration Designer 22
operation modes 95
operations of flow editor 143
operations of problems view 129
operations of synchronization view 131
operators for editing 246
operators logical 246
operators mathematical 246
operators relational 246
Orchestration Designer 15
Orchestration Designer downloading 21
Orchestration Designer existing applications 35
Orchestration Designer opening 22
Orchestration Designer operations 95
Orchestration Designer refreshing 269
order of CCMS preferences 287

order of operations for calculations 246
OUT OF SERVICE 414
out of service logic block example 72
output block about 145
output block creation 194
output block example 71
output blocks conditional assignments 152
output message block example 80, 82
overflow block creation 232
overflow tone command 310

P

P-header 441–443
P-Intrinsics 441–443
pages adding to flow 164
pages deleting from flow 164
pages in flow editor 143
pages renaming flow 165
particular line number 240
passing changed values command 321
pasting content to blocks 160
pegging thresholds 274
permissions access 251
personal number route call to 233
perspective for Contact Center 291
phantom voice contact error 450
planning applications 37
play announcements block creation 190, 192
play music block creation 206
PLAY PROMPT 346
play prompts block creation 194
playing announcements for SIP command 362
playing collected digits 216
playing menu example 226
playing music command 309
playing music example 76, 359
playing prompts in SIP 214
POSITION IN QUEUE 415
positioning nodes example 236
preferences automatic saving 288
preferences global 289
preferences network connection 287
preferred language SIP 437
previous release applications 33
printing applications 127
printing synchronization view 142
printing variables 271
printing variables to file 270
priority assign from IVR command 340
priority changes in network skillset command 328
priority changing in agents command 305
priority changing in NACD command 331
priority changing in skillsets command 306
PRIORITY IN NETWORK QUEUE 418
PRIORITY IN QUEUE 416
priority values preferences 289
Private header 441
Private Header 441–443
problems view 101
problems view displaying 129
problems view operations 129
procedures for creating applications 43
programming custom flow components 203
prompt callers in flow 185
prompts 30
prompts in flow 194
properties changing for variables 267
properties of application changes 116
properties of application manager data 256
properties of CCMA 117
properties of CCMS 118
properties of variables copying 268
protecting applications 122

Q

question callers 185
queue block overview 145
queue blocks conditional assignments 152
queue position intrinsic 415
queue priority in network intrinsic 418
queue priority intrinsic 416
QUEUE TO AGENT 313
queue to agent block creation 181
QUEUE TO AGENT LONGEST IDLE AGENT 313
QUEUE TO NACD 333
queue to network ACD block creation 183
QUEUE TO NETWORK SKILLSET 329
queue to network skillset block creation 179
QUEUE TO SKILLSET 314
queue to skillset block creation 174
queue to skillset block example 74
queue to skillset example 175
QUEUE TO SKILLSET MOST LOGGED AGENTS 314
QUEUED 387
QUEUED COUNT 419
QUEUETYPE 400
QUIT 301

R

RAN playing 190, 192

reading CCMS order preferences 287
READVAR 321
rearrange flow applications 236
receiving data example 153, 200
recorded announcements command 311
recorded announcements example 64
recording messages to Event Browser command 320
redoing changes in applications 241
reference block about 145
reference block creation 204
reference other flows 204
refresh data 292
refreshing content in Orchestration Designer 269
related resources 17
Avaya Mentor videos 17
relational operators 246
remove contact from agent command 315
REMOVE FROM AGENT 315
REMOVE FROM NACD 334
REMOVE FROM NETWORK SKILLSET 330
REMOVE FROM SKILLSET 316
removing a value from a variable set 260
removing applications from Contact Center view 135
removing applications from Local view 138
removing data from the Local view 138
removing variables 261
removing variables from Local view 138
renaming application manager data 256
renaming applications 121
renaming blocks 151
renaming flow pages 165
renaming nodes 151
repeating application components 237, 238
report printing synchronization view 142
reporting AML and contact center 94
reporting IVR and contact center 91
reporting IVR Avaya Media Server 92
request information block creation 199
requests for data command 356
resetting perspective 291
resizing 167
resolving errors in applications 130
resource configuration in Contact Center 19
resources 21
responses from SEND REQUEST command 357
restrictions to access for data 251
results multiple command 322
return value of intrinsics 371
reviewing errors example 86
reviewing existing applications and variables 35
ringback tone command 311
ringback treatment block creation 207
ROUTE CALL 316
ROUTE CALL URI 370
route contact block creation 233
ROUTE NUMBER 388
ROUTE POINT 431
route point label intrinsic 433
routing default error 446
running application example 90

S

sample of application 51
SAVEVAR 321
saving application preferences 288
saving applications 122
Screen Pop 441
script applications commands 239
script applications conversion to flows 124
script applications line numbers 240
script editor preferences 289
script editor view 100
script execution command 300
script maintenance minimization 297
script test command 320
script variables in CCMA 261
scripts for routing 29
scripts reference from flows 204
SECTION 301
section creation command 301
section execution command 299
selecting threshold classes 275
SEND INFO 354
send information block creation 197
SEND MESSAGE 369
SEND REQUEST 356
sending data example 153, 200
sending IMs in SIP 219
service out intrinsic 414
session management 97
settings for preferences 287
sharing applications 119
showing Contact Center view 108
showing discrepancies 133
showing line numbers in scripts 240
showing Local view 108
showing problems view 129
showing synchronization view 132
shrinking view 167
silence command 312
silent treatment block creation 209
SIP call types 438

SIP header 441
SIP intrinsics 435
SIP preferred language 437
SIP-enabled custom options 221
SIP-enabled IVR command 362
SIP-enabled playing prompts 214
SIP-enabled prompt and collect digits 216
SIP-enabled RAN 190
SIP-enabled send IM 219
sizing flow diagrams 167
skillset age of contact intrinsic 413
skillset agent command 314
skillset assigned intrinsic 394
skillset assignment 174
skillset contacts in queue intrinsic 419
skillset idle agent intrinsic 407
skillset instrinsics example 372
skillset longest idle agent intrinsic 410
skillset most logged on intrinsic 411
skillset network queue command 329
skillset number logged intrinsic 408
skillset number of idle agents intrinsic 407
skillset out of service intrinsic 414
skillset queue block example 74
skillset queue position intrinsic 415
skillset queue priority intrinsic 416
skillset queueing example 175
skillset remove command 316
skillset remove contact from network command 330
skillset wait time intrinsic 405
skillsets change priority in network command 328
skillsets changing priority command 306
skillsets creation 254
sound file management 30
source shortcut about 145
source shortcut creation 237
speed of answering contacts intrinsic 404
starting 19
starting application example 90
starting applications 43
starting Orchestration Designer 22
statements conditional 300
stopping applications 121
string matches command 312
string variables command 307
strings assign command 368
strings from variables conversion command 308
subtype intrinsic 396
suggestions for caller feedback 295
summary of blocks 169
support 16
contact 16
suspend activitiy command 302
switch forwarded calls intrinsic 382
Switch RANs 192
switched transitions 159
synchronization view 99
synchronization view displaying 132
synchronization view operations 131
synchronization view printing report 142
synchronization view showing discrepancies 133
synchronizing objects from Local view 132
synchronizing views example 86

T

tabs in flow editor 143
terminate call example 66
terminate command 308
terminating applications 238
terminating call example 85
terminating call routing 229, 230, 232, 233
testing script command 320
text matches command 312
threshold classes assignment 275
threshold classes creating 273
threshold configuration 273
thresholds display and pegging 274
time assigned intrinsic 395
time date intrinsic 421
time day of month intrinsic 422
time day of week intrinsic 421
time delay example 84
time instrinsics example 373
time month intrinsic 423
TIME OF DAY 423
time of day intrinsic 423
timers 97
TOADDRESS 402
toolbar items 106
TOTAL ACTIVE CALLS 426
TOTAL ACTIVE CONTACTS 426
traffic call rate intrinsic 425
traffic contact rate intrinsic 425
traffic instrinsics example 373
traffic total active contacts intrinsic 426
traffic total active intrinsic 426
TRANSFERRED 388
transition example 60
transitions deleting 161, 162
transitions disconnecting 163
transitions with conditions 156
transitions with multiple conditions 159

transitions with no conditions 156
treatment block 207
treatment block about 145
treatment block example 76
treatment GIVE IVR error 446
treatment IVR command 340
treatments play music 206
treatments ringback 207
treatments silent 209
troubleshooting default routing 446
troubleshooting endless loops 445
troubleshooting GIVE CONTROLLED BROADCAST ANNOUNCEMENT 447
troubleshooting GIVE IVR treatment 446
troubleshooting network application execution 449
troubleshooting phantom voice contacts 450
troubleshooting validation error 445
troubleshooting validation errors 451
troubleshooting voice processing commands 447
tying blocks together example 235
type of queue intrinsic 400
types of variables changing 267
types SIP calls 438

U

unassign from agent command 315
undoing changes in applications 241
Uniform Resource Identify (URI) command 370
updating applications in both views 140
updating applications in Contact Center view 135, 139
updating applications in Local view 137, 140
updating conflicted application 138
updating data in both views 140
updating data in the Local view 137, 140
updating variables in both views 140
updating variables in Local view 137, 140
updating version of Orchestration Designer 21
upgrading applications 33
URI 433
user prompts playing 194
using notes in blocks 165

V

validation error 445
validation errors 451
validation preferences 289
value assignment command 299
values removing single from data set 260
variable access classes 251
variable call changing command 321
variable creation example 55
variable properties 259
variables adding characters to 305
variables adding to Local view 136
variables changing properties 267
variables checking for references 266
variables conversion to string command 308
variables copying properties 268
variables creating with multiple values 264
variables creating with one value 262
variables deleting 269
variables existing in Orchestration Designer 35
variables in CCMA 261
variables in Orchestration Designer 257
variables minimizing script maintenance 297
variables printing to file 270
variables printing to printer 271
variables refreshing list in Orchestration Designer 269
variables removing 261
variables removing from Local view 138
variables removing value from a set 260
variables string command 307
variables updating in both views 140
variables updating in Local view 137, 140
videos 17
Avaya Mentor 17
view application manager data 256
view displaying problems 129
viewing block summary 169
viewing CCMA properties 117
viewing CCMS properties 118
viewing context-sensitve help 293
viewing online help 293
views contact center 98
views flow editor 100
views local 99
views perspective 291
views problems 101
views script editor 100
views synchronization 99
voice call treatment IVR command 340
voice calls ringback tone command 311
voice calls silence command 312
voice contact error 450
voice contact termination command 308
voice contacts RAN command 311
voice ports command 343
voice processing error 447
voice prompts command 346
VXML treatment command 362

W

WAIT 302
wait block about 145
wait block creation 211
wait block example 84
wait preferences 289
wait time intrinsic 405
Web services activating 277
Web services importing to DIW 277
Web services integration 277
WHERE-EQUALS 322

Y

year month of intrinsic 423

Z

zoom in 166
zoom out 167
zoom to fit 166