



Enghouse
Interactive

Contact Center as a Service

Designer Help - Tenant

Contents

1: Intro	8
What's new	8
Differences between the legacy and new Designer	8
Migrate to the new Designer	9
Definitions	9
Legal disclaimer	10
Support	11
2: Overview	12
Log in	12
Password expired message	13
Additional login messages	14
Log in using Unified Admin	15
App selector menu	16
Select the language	17
Reseller view	18
Select the tenant to manage	18
Return to the Tenants page	18
Get started page	19
Errors page	21
Save current errors to a text file	21
Clear current errors	21
Navigation pane	22
List view	23
Call flow	24
Predefined events	25
Flowchart of events	27
How to use audio files	29
How to use speech	30
How to use video	31
Naming format for video files	31
How to record calls	32
How to use variables and constants	33
How to use variables	33
How to use the \$empty\$ constant	34
Log out	35
Additional logout messages	35
Session expiry	35
3: Workspaces	37
Workspace editor overview	37
Workspace tab bar	38

Script tabs panel	40
Script blocks panel	41
Flow editor	41
Properties panel	44
Manage workspace templates	46
Select a template	47
Delete a custom-tenant template	47
Open a workspace	48
Import a workspace	49
Manage workspaces	51
Save a copy of the workspace	51
Save a workspace as a tenant template	52
Revert a workspace	53
Reload the last published workspace	54
Delete a workspace	55
Publish workspaces	56
Location to publish to	56
Workspace versions	56
Publish a workspace	57
Manage tables	58
Upload a new table	59
Upload an updated table	60
Download a table	60
Publish a table	61
Rename a table	62
Delete a table	62
Manage workspace files	63
Upload a new file	64
Upload an updated file	65
Download a file	65
Publish a file	66
Rename a file	67
Delete a file	67
Manage version history	68
Open a snapshot	69
Edit a version's description	70
Republish a version	71
4: Scripts	73
Manage scripts	73
Add a new script	75
Add a system script	75
Duplicate a script	76
Rename a custom script	76
Save a script	77
Publish a script	77

- Validate a script 78
- Delete a custom script78
- How to iterate through scripts 79
- Example scripts 80
 - Ivr script 81
 - Agentless IVR script 85
 - IVR Timeout script 87
 - Call processing scripts88
 - Elsbeth integration scripts104
 - Music On Hold script106
 - Music On Hold In Queue script 107
 - In Queue Timeout script 110
 - Pickup Timeout script 110
 - Voicemail processing scripts 110
 - Web services XML examples 120

- 5: Blocks 122
 - Block availability per CallXML script123
 - Abort 128
 - Add Skill 128
 - Agent Selection Rule 129
 - Agentless Campaign Outcome 130
 - Answer 131
 - Assign 132
 - Bridge Call 133
 - Call Maturity/Priority 134
 - Caller Message 135
 - Caller Name 135
 - Caller URL 136
 - Check Date 136
 - Configure the Check Date block 136
 - Check Time 139
 - Configure the Check Time block 139
 - Clear Digits 140
 - Clear Digits Events 140
 - Clear User Data 141
 - Condition 141
 - Define Grammar 142
 - Routing 143
 - Backward compatibility 144
 - Edit Email 145
 - Get Agent Statistics 147
 - Get Digits 150
 - Get Message 152
 - Get Phone Information 153
 - Information fields 153

Get Queue Statistics	155
Get User Data	157
GoTo	158
GoTo Voicemail	158
Hangup	159
IVR Recording	160
Examples	161
IVR Transfer Recording	163
Leave Voicemail	164
Examples	166
Log	167
Logging Control	168
Mask Digits	169
MD5 Encryption	169
Menu	170
Configure the Menu block	171
Optional Parameters	173
Real-Time Speech Analytics	173
CCaaS UI CustomRing parameter	173
Percent Allocated	175
Configure the Percent Allocated block	175
Play Audio	176
Play Date	177
Play Greeting	179
Play Money	180
Play Number	181
Play Text	182
Play Video	184
Implementation	184
Playback Audio	185
Queue Name	186
Read SIP Header	187
Record Audio	188
Record Video	190
Record Voicemail	191
Recorder	193
Rules table	194
Remove Skill	195
ReQueue	195
Routing Mode	196
Routing Properties	197
Run	197
Schedule Callback	198
Script Entry Point	200
Send Callback	201
Send Email	203
Send Message	205

- Send User Data206
- Set Defer Type206
- Set Event207
- Set Language208
- Set SIP Header209
- Speech Enabled Menu210
 - Configure the Speech Enabled Menu block212
- Speech Play and Collect214
 - Configure the Speech Play and Collect block216
- Stop Video217
- Supervised Transfer218
 - Configure the Supervised Transfer block220
- Switch221
 - Configure the Switch block221
- Table223
- Table Lookup223
- Terminate224
- Transfer by Refer225
 - Configure the Transfer by Refer block226
- Transfer Call227
- UnQueue229
- VXML (VoiceXML) Transfer Call230
- Wait for Events232
- Wait232
- Web Services233

- 6: Objects234**
 - New model for messaging calls234
 - Common media API235
 - CallbackCall object239
 - EmailCall object240
 - OutgoingEmailCall object240
 - VoicemailCall object240
 - Comparison with CallData object245
 - CallData object250
 - Properties from SIP headers252
 - Playback-related properties254
 - Determining the Calldata.HideAniFromAgent value255
 - CCLEvent object256
 - CosmoCall object256
 - CosmoCall.CallbackData object257
 - CosmoCall.TtsInfo object258
 - Line object258
 - NewLine object259
 - Session object259

7: CallXML	260
General elements	261
assignvar	261
clear var	261
cleardigits	261
getDigits var	261
goto value	261
run value	262
sendEvent value	262
Call control elements	262
answer	262
call value	262
conference targetSessions	262
hangup	263
playAudioformat	263
playDateformat	263
playMoneyformat	264
playNumber	264
recordAudioformat	264
text format	265
waitForConferenceEnd	265
Test and debug elements	265
CallXML high level elements	265
block label	266
menu label	266
inputAudio label	266
inputDigitslabel	266
Additional Designer elements	267
addSkillvalue	267
arrivalTimeBiasvalue	267
callerMessagevalue	267
callerNamevalue	267
callSetNamevalue	267
callerURLvalue	267
conditionalexpr	268
optionalParametervar	268
removeSkillvalue	268
sendEmail server	268
unQueue and reQueue	269

1: Intro

CCaaS Designer is a stand-alone application that allows you to write scripts to manage call-flows, using a graphical interface to create the paths that a call can take through your call center. CCaaS Designer loads workspace and script data from a database and allows you to publish this as XML files to locations configured on a per-tenant basis.

This Help is for script designers who write scripts to manage incoming call flows to their contact center and to manage the events that occur in outbound dialing campaigns.

In the tenant help, the CCaaS UI for Agents and Supervisors is called CCaaS UI.

What's new

[Log in using Unified Admin](#) — Unified Admin is a portal application for tenant administrators that presents the administration applications you are allowed to access.

[Log out](#) — if you logged in using Unified Admin, when you click **Logout** in CCaaS Designer, the **App Selector** page appears.

[Get Message](#) and [Send Message](#) — added blocks for interactive chat response.

Differences between the legacy and new Designer

You now have to log in to use the Designer.

Workspaces are created and published in the context of a tenant.

Workspaces, scripts and tables are now stored in and loaded from a database.

XML files for workspaces, scripts and tables are only created when publishing either the whole workspace or an individual script and table.

XML files for workspaces, scripts and tables are only loaded by Designer when importing an existing workspace.

Tables can no longer be edited in Designer. A table must be created and edited outside of Designer in a suitable editor that can save the table as a tab-delimited text file. This table file must then be uploaded into Designer. All the tables for a workspace are listed on a dedicated page, where they can be uploaded, downloaded, published and renamed.

Designer creates workspace folders on publish.

Case branch properties are now edited in the Properties panel, instead of in the dialog.

To edit the scripts on which a block is validated, instead of editing the *Library.xml* file in the installation folder of Designer, you must edit the database table directly. Contact Customer Services for details on how to do this.

Workspaces cannot be opened from a URL. This is because workspaces must be loaded in from a folder location in order for all scripts (including ancillary scripts) and tables to be loaded.

See [Migrate to the new Designer](#).

Migrate to the new Designer

To migrate to the new Designer:

1. Prepare the workspace folder for import into Designer. Remove backup copies of script files and empty script files.
2. Import the workspace. See [Import a workspace](#).
3. Publish the workspace to the tenant's IVR root for applications. See [Publish a workspace](#).

Definitions

Term	Definition
ancillary script	An imported script XML file that is not directly referenced by an imported <i>workspace.xml</i> file but is still required to be part of the workspace. For example, PrP-generated files and files that are destinations for GoTo blocks.
block	Action with properties and behavior that can be applied to a call in a script. Blocks are either: <ul style="list-style-type: none">• Custom — created by the user with properties and behavior that can be edited as required• System — installed with the CCaaS Designer application that has predefined properties and behavior, with only the property values editable by the user.
call parameter	A call is associated with these parameters: TQoS, skills required, queue destination, caller name, URL, and any number of optional parameters.
CCaaS UI	CCaaS UI application for agents and supervisors. Previously called TouchPoint.
event	See Predefined events for a list of events that affect the routing of calls in CCaaS.
group	A collection of agents working in the same area of expertise in the contact center. The agents in a group receive calls from the same queues.
ICS	Internet Communication Service — provides chat capability.
maturity	TQoS less the amount of time the call has already been in the queue.
MCS	Message Communication Service — provides email capability.
multichat	Enables simultaneous handling of multiple chat sessions, up to a configured number between one and nine sessions.
priority	Independent of TQoS, you can assign priorities to calls arriving at your contact center

Term	Definition
	through scripts. The priorities are in the range from 0 through 99, with the default being 49, the lowest priority 0, and the highest priority 99. Within a queue, call priority overrides TQoS and call maturity.
queue	A mechanism to manage a collection of incoming phone, email, Internet, and video calls that require common handling.
release code	Codes used by agents to explain why they are making themselves unavailable to accept calls.
routing	The movement of a call from the ACD to a queue and then to an agent with the skills to handle the call.
script	A sequence of blocks that define a call script. Scripts are used to respond to predefined events that may be triggered in CCaaS.
skill	A skill is an attribute of an individual agent, who is defined by the administrator in the agent administration procedure. Skill requirements are also attributes of individual calls, that may be assigned to the call in the construction of the call request by the web or IVR application.
snapshot	Published version of a workspace, including its scripts, tables, and ancillary files, that is stored in the database. Up to 10 such versions can be kept for each workspace.
team	An administrative grouping of agents. Teams do not affect how calls are routed to agents.
timeout	If a call remains unanswered for longer than this time period, an event is triggered.
TQoS	Total Quality of Service — the amount of time in seconds in which a call should be answered and can be used to specify service levels for types of customers.
workspace	A set of scripts that relate to a single location that can be associated with an application entity in CCaaS Admin.
wrap up code	Code that represents the result of a call.
VCS	Voice over IP (VoIp) Connection Service — CCaaS service that supports voice call.
VCS only	This statement is for users who have access to the VCS or are responsible for configuring and managing CCaaS connection servers. The statement excludes applicability for web chat and email message handling and include voice call.

Legal disclaimer

This document is governed by the terms of the software license agreement and applicable contract (including addendums) entered into with Enghouse.

Support

To submit comments or questions about the information in this guide, please open a case with Enghouse Support.

2: Overview

This section contains the following topics:

- [Log in](#)
- [Log in using Unified Admin](#)
- [Select the language](#)
- [Reseller view](#)
- [Get started page](#)
- [Errors page](#)
- [Navigation pane](#)
- [List view](#)
- [Call flow](#)
- [Predefined events](#)
- [Flowchart of events](#)
- [How to use audio files](#)
- [How to use speech](#)
- [How to use video](#)
- [How to record calls](#)
- [How to use variables and constants](#)
- [Log out](#)

Log in

See also [Log in using Unified Admin](#).


The log in process is slightly different depending on the URL you use for CCaaS Designer and the identity provider configured by your administrator.

Note

- Reseller administrators can log on to multiple tenants.
- Admin users can open several tabs in the same browser with one login.
- If you log in again after your CCaaS application session becomes invalid, and while your ADFS or EIS provider's session is still valid, you do not need to type your password.

1. In your browser, in the **Address** bar, type the URL supplied by your supervisor or administrator, and then click **Go** or press **Enter**. The **Welcome to CCaaS Designer** page briefly appears.
If you are already logged in, the main **CCaaS Designer** page will appear.

Note

If you previously [logged in using Unified Admin](#), the  **App Selector** button appears at the top left of the main **CCaaS Designer** page.

2. If you are not already logged in:
 - If you are using the general application URL, for example, <https://ccsp1.pj16.loc/Designer>, the **Welcome to CCaaS** page will appear. Type your **Username** in User Principal Name (UPN) format, for example, someone@domainname.com, and click **Continue**.
 - If you are using the tenant's direct login URL, for example, <https://ccsp1.pj16.loc/portal/t1/Designer>, you do not see this page. Instead, go straight to step 3.
3. The next step depends on the identity provider configured by your administrator:
 - For Auth Server, if it is not already present, type your **Username** in User Principal Name (UPN) format, for example, someone@domainname.com. Type your **Password** and click **Log in**. The main **CCaaS Designer** page appears. If necessary, see [Password expired message](#) below.

Alternatively:

 - If you are using the general application URL and your system is set up with an identity provider, click **Login with your identity provider**.
 - To change the username, click **Change user** to go to the **Welcome to CCaaS** page.
 - For ADFS, you are redirected to the ADFS login page to authenticate. Please consult your local procedures for how to proceed from this point.
 - For Enghouse Identity Server (EIS), you are redirected to the EIS login page. Type your **Password** and click **Log in**. The main **CCaaS Designer** page appears.
4. If necessary, see [Additional login messages](#) below.

Next, see [Reseller view](#) or [Get started page](#).

Password expired message

For Auth Server, if your password has expired or your administrator has set your account to require a password change at the next login, the next time you enter your password on the **Welcome to CCaaS** page, the **Password expired** notification appears.

1. In the notification, click **Change password**. The **Change password** dialog appears.
2. In **Old password**, type your current password.
3. In **New password**, type your new password. Do not use the double quote (") character or spaces.
4. In **Confirm password**, type your new password again.
5. Click **OK**.

Additional login messages

The following messages may appear when you try to log in:

- *Failed to get authorization for Designer use.* Your user profile is not permitted to use the CCaaS Designer application. Only landlord, reseller, and tenant administrators are permitted to use the CCaaS Designer application.
- *You are not authorized to view this page.* This can occur because:
 - You are logged in to an external identity provider with a different username than the one you entered on the **Welcome to CCaaS** page to log in to the CCaaS client application.
 - You entered a username on the **Welcome to CCaaS** page and were redirected to an external identity provider, where you log on with a different username.
- *Auth Server is not available or there is a misconfiguration.* You must contact your administrator. You will not be able to log in to CCaaS Designer until the issue is resolved.

Log in using Unified Admin

See also [Log in](#).

Unified Admin is a portal application for tenant administrators that presents the administration applications you are allowed to access.

Note

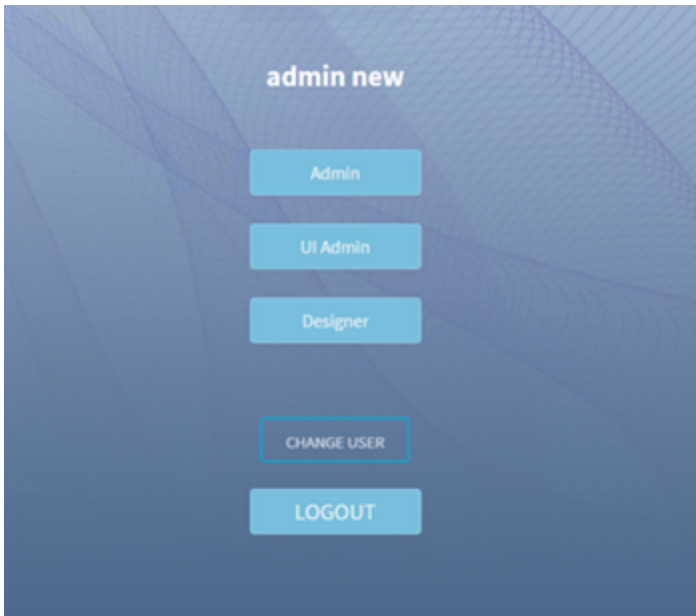
If you are allowed to access only one administration application, you go straight into that application without seeing the **App Selector** page.

1. If you are not already logged in:
 - If you are using the general application URL, for example, *https://ccsp1.pj16.loc/AuthServer/Pages/AppSelector.aspx*, the **Welcome to CCaaS** page will appear. Type your **Username** in User Principal Name (UPN) format, for example, *someone@domainname.com*, and click **Continue**.
 - If you are using the tenant's direct login URL, for example, *https://ccsp1.pj16.loc/portal/t1/apps*, you do not see this page. Instead, go straight to step 2.
2. The next step depends on the identity provider configured by your administrator:
 - For Auth Server, if it is not already present, type your **Username** in User Principal Name (UPN) format, for example, *someone@domainname.com*. Type your **Password** and click **Log in**. The **App Selector** page displays buttons for the administration applications you are allowed to access.

If necessary, see [Password expired message](#) or [Additional login messages](#).

Alternatively:


 - If you are using the general application URL and your system is set up with an identity provider, click **Login with your identity provider**.
 - To change the username, click **Change user** to go to the **Welcome to CCaaS** page.
 - For ADFS, you are redirected to the ADFS login page to authenticate. Please consult your local procedures for how to proceed from this point.
 - For Enhouse Identity Server (EIS), you are redirected to the EIS login page. Type your **Password** and click **Log in**. The **App Selector** page displays buttons for the administration applications you are allowed to access.

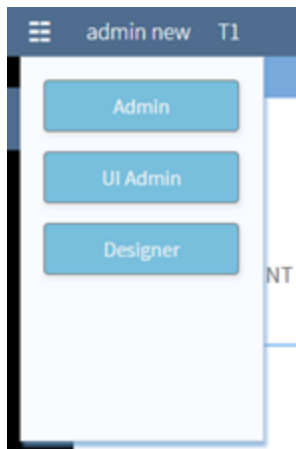


3. Click the **Designer** button. The main **CCaaS Designer** page appears.

Next, see [Get started page](#).

App selector menu

To open the other administration applications you are allowed to access, click  **App selector** at the top left of the main **CCaaS Designer** page, and then click the appropriate button.



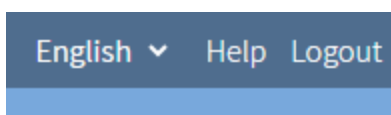
Each application opens in its own browser window. You can open several **CCaaS Designer** tabs in the same browser.

Select the language

Note

- By default, CCaaS Designer uses the language set in your browser. If the browser language is not available in CCaaS Designer, it uses the language configured by your administrator. If the configured language is not available in CCaaS Designer, it uses English.
- After you open CCaaS Designer, you can select to use one of the available languages. CCaaS Designer then automatically uses your selected language.

1. At the top right of the main CCaaS Designer page, click the language menu.



2. Select the language to use.

Reseller view

Select the tenant to manage

[Return to the Tenants page](#)

A reseller administrator is a tenant-wide administrator that performs tenant administrator tasks for the tenants that are assigned to them.

Caution

If no tenants are assigned to a reseller administrator, they can manage all tenants.

After logging in, reseller administrators see the **Tenants** page.



For navigation information, see [List view](#).

Select the tenant to manage

On the **Tenants** page, select the tenant to manage.

Next, see [Get started page](#).

Return to the Tenants page

To return to the **Tenants** page, click .

Get started page

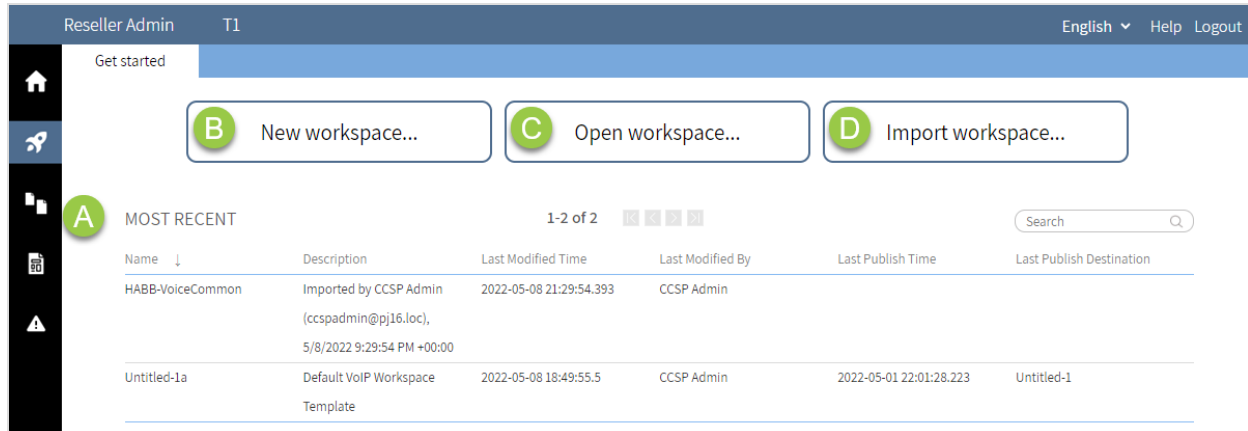
Note

- Each tenant can have multiple workspaces.
- Each tenant can have two or more workspaces based on the same template.
- On the **Get started** page, the **Most recent** list displays the workspaces that have been opened in the Workspace editor. If a snapshot has been opened, the owning workspace is listed, not the snapshot.

After selecting the tenant to work with, reseller administrators see the **Get started** page with that tenant's workspace information.

After logging in, tenant administrators see the **Get started** page with their workspace information.

Reseller view



The screenshot shows the 'Reseller Admin' interface for tenant 'T1'. The page title is 'Get started'. At the top right, there are links for 'English', 'Help', and 'Logout'. Below the title, there are three main action buttons: 'New workspace...' (labeled B), 'Open workspace...' (labeled C), and 'Import workspace...' (labeled D). Below these buttons is a section titled 'MOST RECENT' (labeled A) showing a table of workspace information. The table has columns for Name, Description, Last Modified Time, Last Modified By, Last Publish Time, and Last Publish Destination. There are two rows of data in the table. A search bar is located to the right of the table header.

Name ↓	Description	Last Modified Time	Last Modified By	Last Publish Time	Last Publish Destination
HABB-VoiceCommon	Imported by CCSP Admin (ccspadmin@pj16.loc), 5/8/2022 9:29:54 PM +00:00	2022-05-08 21:29:54.393	CCSP Admin		
Untitled-1a	Default VoIP Workspace Template	2022-05-08 18:49:55.5	CCSP Admin	2022-05-01 22:01:28.223	Untitled-1

Tenant view

Name	Description	Last Modified Time	Last Modified By	Last Publish Time	Last Publish Destination
HABB-VoiceCommon	Imported by CCSP Admin (ccspadmin@pj16.loc), 5/8/2022 9:29:54 PM +00:00	2022-05-08 21:29:54.393	CCSP Admin		
Untitled-1a	Default VoIP Workspace Template	2022-05-08 18:49:55.5	CCSP Admin	2022-05-01 22:01:28.223	Untitled-1

To open a recently edited workspace, in the **Most recent** list (A), click the workspace. For navigation information, see [List view](#). The **Workspace editor** appears. Next, see [Workspace editor overview](#).

To create a new workspace, click **New workspace** (B). The **Workspace templates** page containing a list of the available templates to choose from appears. Next, see [Manage workspace templates](#).

To open an existing workspace, click **Open workspace** (C). The **Workspaces** page containing a list of the available workspaces to choose from appears. Next, see [Open a workspace](#).

To import an existing workspace as part of the process of migrating a tenant from the legacy to the new Designer, click **Import workspace** (D). The **Import Workspace** dialog appears. Next, see [Import a workspace](#).


Errors page

Save current errors to a text file

Clear current errors

Note

- When an [Import](#) or [Validate](#) action causes an error, a dialog appears with a **See details on the Error Page** link or a **Details** button that redirects the user to the **Errors** page.
- Error List data is kept until the user session ends, unless the user manually clears the list.
- When a new action causes an error, previous action errors are automatically cleared.

1. On the navigation pane, click  **Errors**. The **Errors** list appears. For navigation information, see [List view](#).




Time	Action	Type	Workspace	Item	Description
11.12.2023. 12:59:42	Import	Invalid Xml file	DPSEU	Chat_Limit.xml	Unexpected end of file has occurred. The following elements are not closed: table. Line 43, position 1.
11.12.2023. 12:59:42	Import	Invalid Xml file	DPSEU	Holiday_WWn_LEP.xml	Unexpected end of file has occurred. The following elements are not closed: table. Line 7, position 1.
11.12.2023. 12:59:42	Import	Invalid Xml file	DPSEU	OpenHours_Bf_LAN.xml	Name cannot begin with the '"' character. Hexadecimal value 0x3C. Line 3, position 1.
11.12.2023. 12:59:42	Import	Invalid Table format	DPSEU	Chat_Limit.xml	Table is missing header element.
11.12.2023. 12:59:42	Import	Invalid Table format	DPSEU	Holiday_WWn_LEP.xml	Table is not in valid format.

The possible Import or Validate action error types are:

- Block name too long — the block display name exceeds 256 characters.
- Duplicate name — during import of a workspace, more than 1 script with the same name has been found.
- Invalid Block usage — a block exists on a script that is not valid for that script type.
- Invalid Table format — a table *.xml* file has invalid or missing elements.
- Invalid Xml file — the file is syntactically invalid or has incorrect or missing elements.

Save current errors to a text file











1. On the **Errors** page, click  **Download**.
2. The text file appears in your Downloads folder.

Clear current errors

1. On the **Errors** page, click  **Delete**.









Navigation pane

Use the navigation pane as follows.

Icon	Description	Visible to Reseller	Visible to Tenant
	Home — click to view the Tenants page and select the tenant to manage.	Yes	No
	Tenant Home — click to view the Get started page .	Yes	Yes
	Workspaces — click to manage workspaces .	Yes	Yes
	Workspace Templates — click to manage workspace templates .	Yes	Yes
	Errors — click to view the Errors page .	Yes	Yes
Visible after you open a workspace			
	Workspace Editor — click to view the workspace editor .	Yes	Yes
	Scripts — click to manage scripts .	Yes	Yes
	Tables — click to manage tables .	Yes	Yes
	Workspace Files — click to manage workspace files .	Yes	Yes
	Version History — click to manage version history .	Yes	Yes

List view

Use the list view as follows.

To	Action
Add a block or table	Click  .
Delete an item	Select the item's check box and then click  .
Delete multiple items	Select multiple check boxes and then click  .
Search for items	Type the criteria in the search box.
Sort	Click the column heading. The  indicates the sort column and sort direction.
Open an item	Click the item in the list to open it.
View counters	The number of visible items and the total items appear at the top of the page, for example, and 1-4 of 4 . The number of: <ul style="list-style-type: none">• Filtered items appears in front of the total items, for example, 1-2 of 2.• Selected items appears in brackets at the top of the check box column, for example, (1).
View the first, previous, next, or last page	Click the appropriate page navigation icon:     .

Call flow

CCaaS Designer is a highly flexible, graphical tool for designing and deploying call flows (scripts) in contact centers. Each script handles a specific call event.

In CCaaS, call flow is determined by:

- Events that depend on the media type of the call. By default, media types include voice, internet (chat) and message (email). For example, only VoIP calls enter the incoming call event and run the IVR script.
- The CCaaS Call Management Service response.

Using CCaaS Designer, you can create an IVR script that puts VoIP calls into a queue and gives the call skills based on caller input, for example, "Press 1 for Sales, press 2 for Support".

In CCaaS Designer, you draw the call flow using a simple connect-the-blocks approach with a graphical user interface. Each command block represents one logical step of the call flow, such as playing an announcement, collecting touch tone digits, or routing a call to a specific queue.

CCaaS Designer uses a superset of CallXML functions that enables web, email and VoIP calls to interact with the XML scripts to gather caller information and choices for call routing and caller identification. The CCaaS Designer CallXML functions are presented as a set of graphical tools called blocks.

The XML scripts gather the input for these functions and pass them to the XML Interpreter for processing. This enables CCaaS service providers, and their tenants, to quickly edit scripts without the need for VBScript programmers or CCaaS system administrator intervention.

Predefined events

In every CCaaS call lifecycle, there are several predefined events. You determine which events trigger a script and what task the script performs, depending on the media.

Note

CCaaS UI was previously called TouchPoint.

The following table describes the events supported by the different media.

Event/script	Description	Voice	Chat	Email	Voicemail	Callback
IVR	An incoming call. When a VoIP call arrives at the VCS.	Yes	No	No	No	No
IVR Timeout	IVR system timeout, triggered when a caller has been in the IVR system for more than 30 minutes.	Yes	No	No	No	No
Call Answered	Occurs when an agent answers an incoming voice call.	Yes	No	No	No	No
Call Arrival	When a call arrives at the contact center. This event is triggered after IVR, if that occurred, but before being submitted to the Call Management Service.	Yes	Yes	Yes	Yes	Yes
Call Aged	An email or voicemail has been in the deferred folder for longer than the configured Ageing Timeout Interval. Ageing starts from when the agent defers the call, not from call arrival. By default, an agent cannot defer a call after it has aged once. You can override this default in the Ageing script.	No	No	Yes	No	No
Call Pre Sent	After an agent initiates, replies to, or forwards an email by clicking Send but before the email is actually sent from the system to the recipient.	No	No	Yes*	No	No
Music On Hold In Queue	The time before a call is answered, while a VoIP caller is in queue. A Music on Hold While in Queue script might play <i>.wav</i> files while a VoIP caller is waiting.	Yes	No	No	No	No
Music On Hold	The time while a caller is on hold after an agent put them on hold.	Yes	No	No	No	No
In Queue Timeout	A call remains unanswered for longer than its allowed timeout period. An In Queue Timeout script might issue an alert to a supervisor or it might delete some skills from the call so that more agents are capable of answering the call.	Yes	Yes	Yes	Yes	Yes

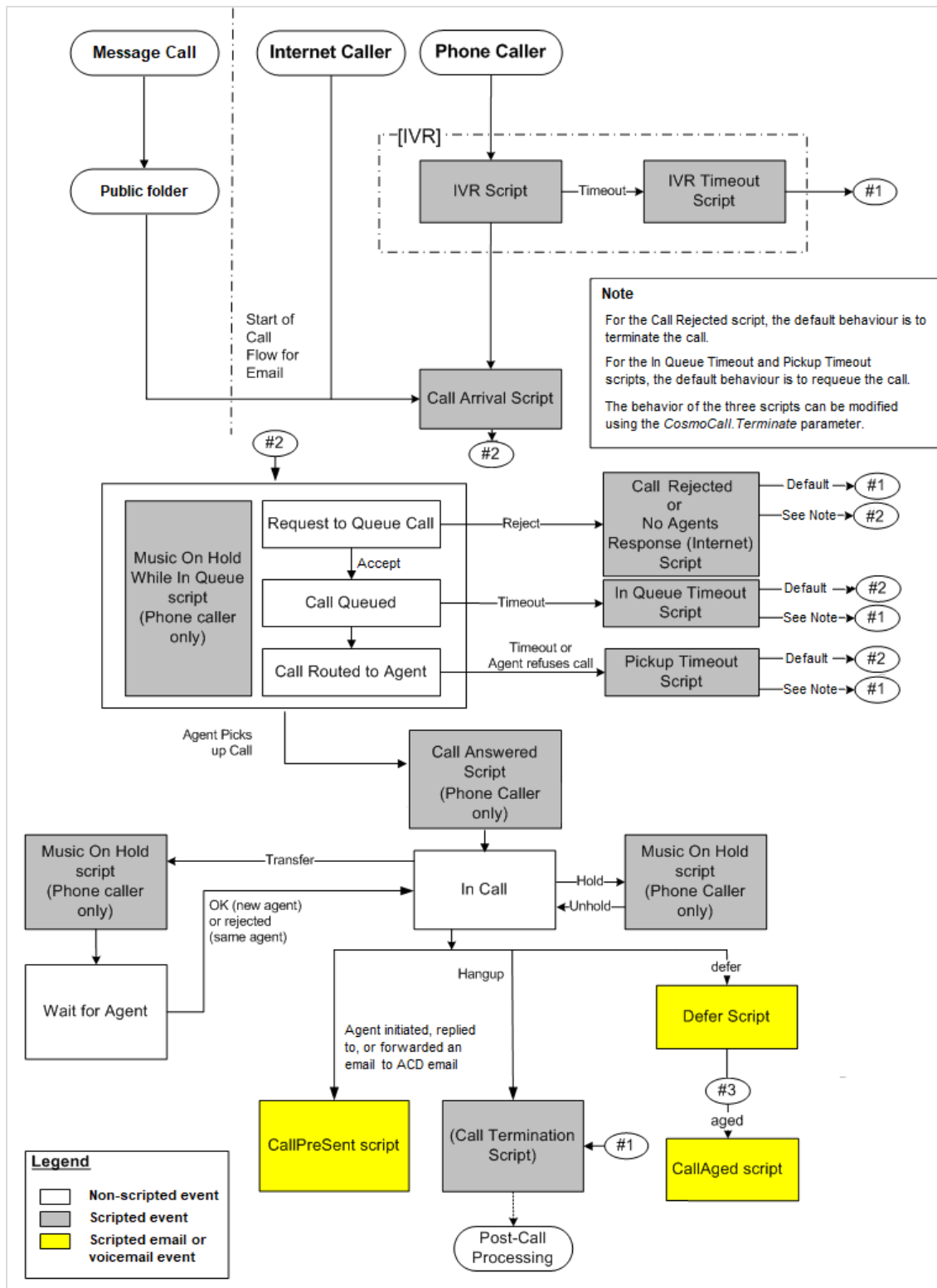
Event/script	Description	Voice	Chat	Email	Voicemail	Callback
Call Rejected	When a call is rejected for any reason, such as when there are no agents.	Yes	Yes	Yes	Yes	Yes
Pickup Timeout	When the ACD rings CCaaS UI but there is no response or the agent does not pickup the call.	Yes	Yes	Yes	Yes	Yes
Call Termination	The end of a call.	Yes	Yes	Yes	Yes	Yes
Outgoing Call	An outgoing call.	Yes	No	No	No	No
A2A Call Rejected	When a call being delivered to another agent is rejected.	Yes	No	No	No	No
Call Deferred	Agent is in call and defers the email or voicemail message before the completion of call termination.	No	No	Yes	No	No
Playback Audio Call Abandoned **	Plays back audio to a customer at the end of an abandoned call.	Yes	No	No	No	No
Playback Audio Rebound Queue **	Plays back audio to a customer after a rebound call is queued.	Yes	No	No	No	No
Playback Audio Rebound Rejected **	Plays back audio to a customer when a rebound call is rejected.	Yes	No	No	No	No
Post Call	When the caller is still connected after all the other participants of the call hang up, this event is triggered to post process the call, for example, to give a customer the chance to rate or answer a survey about the call.	Yes	No	No	No	No
* Only for outgoing email						
** Available only in the Elsbeth integration.						

The **UnQueue** and **Requeue** CCaaS Designer blocks allow scripts to remove a call from a queue and re-queue a call, if desired.

For examples of scripts that handle most of these events, see [Example scripts](#).

Flowchart of events

The following diagram shows the flow of the CCaaS call events. For a description of the CCaaS events, see [Predefined events](#).



How to use audio files

CCaaS has the ability to play audio (.wav) files as part of the call flow (scripts). CCaaS provides default music and voice files for your use. Alternatively, you can supply your own announcements that customers will hear when calling into your contact center and the message and/or music customers will hear while on hold.

All announcements and music must be standard .wav files saved with the following settings:

- Preferred quality — call quality
- Format — PCM
- Attributes — 8,000 kHz, 16 Bit, Mono

Note

- The path specifying a .wav file must be 80 characters or less in length. This limitation includes the file name.
- The URL can include the http or https protocol.
- For best audio quality, we recommend that you have these recordings made in a professional studio.

Audio files can be uploaded via Provisioning Portal, if it's installed and licensed on the platform.

How to use speech

CCaaS Designer provides blocks that you can use to create speech-enabled Interactive Voice Response (IVR) scripts, if this feature has been enabled for your contact center. See *external media* in the *CCSP Tenant Administrator Guide*.

These blocks support:

- Text-to-Speech (TTS) — an application that synthesizes a spoken sound version of text.
- Automatic Speech Recognition (ASR) — a technology that enables users of IVR systems to speak entries instead of entering information through DTMF.

The speech-enabling blocks are:

- [Set Language](#)
- [Speech Play and Collect](#)
- [Speech Enabled Menu](#)
- [Define Grammar](#)
- [Play text](#)

Note

Your system administrator should tell you which languages are available for speech enabled scripts.

To use ASR, you must first develop a grammar. A grammar specifies the words and patterns of words to be listened for by a speech recognizer. CCaaS speech servers are provided by third parties, so ask your system administrator to provide the information for creating a grammar.

You develop speech-enabled scripts in the same way as other scripts. Use the blocks listed above and the other blocks in the [Script blocks panel](#) to create your scripts. Test your speech-enabled scripts using both DTMF and speech interactions.

How to use video

Note

IVR scripts only affect calls that have a voice component.

CCaaS Designer provides blocks that you can use to process video calls for Interactive Voice and Video Response (IVVR).

These blocks are:

- [Play Video](#)
- [Stop Video](#)

Naming format for video files

File names for video files used in the Play Video block must have the extension *.avi*.

File name formats for file storage are in one of two forms:

- CIF — *<filename>cif<bitrate>_Kbps<framerate>_fps.avi*
For example, *greeting_cif_256Kbps_30fps.avi*
- QCIF — *<filename>_qcif<bitrate>_Kbps<framerate>_fps.avi*

File names used with the Play Video block are in the format *<filename>.avi*. In the CIF example, the file name used in the Video File property is *greeting.avi*.

How to record calls

The [Recorder](#) block determines how calls are recorded.

Note

Email callback requests are not recorded.

You can control whether calls are recorded when they are transferred from an agent to an external number or conferenced with an external number.

Two flags control the behavior of recordings when the call is transferred to or conferenced with an external number. These flags and values are:

- RecordTransfer
 - True — call recording continues on transfer (default)
 - False — call recording stops on transfer
- RecordConference
 - True — call recording continues on conference (default)
 - False — call recording stops on conference

To set the flags to True or False, in a [Run](#) block in the `OutgoingCall` script, call one of the following functions:

- `setTransferRecordingEx(RecordTransfer [Boolean],RecordConference [Boolean])` — sets each flag individually.
- `setTransferRecording([Boolean])` — sets both flags to the same setting. This function is maintained for backward compatibility.

For example:

- If you call `setTransferRecordingEx(false,true)` from a Run block in the `OutgoingCall` script, only conferenced calls are recorded.
- If you call `setTransferRecording(false)` from a Run block in the `OutgoingCall` script, transferred and conferenced calls are not recorded.

To confirm that recording has stopped, view the number of files in `Replay`.

How to use variables and constants

How to use variables

Caution

A variable must be either a built-in system variable or a user-defined variable previously created using the Assign block. If you specify a variable that does not exist, the variable is ignored and evaluates to an empty string in the results.

Variables may be used, instead of or in conjunction with literal strings, in all properties of the CCaaS Designer blocks. To indicate a variable name, enclose the name within dollar signs (\$). For example, if you have a variable called **GreetingFName** that holds the name of the greeting announcement, you could use it by setting the [Play Audio](#) block's Value property to:

```
$GreetingFName$
```

You can use literals and variables together in the same CCaaS Designer property. The interpreter expands all of the variables specified and concatenates them together with the literals before executing the block. For example, if you have two variables called **FName** and **LName** that hold the first and last names of the caller and you want to use them in the **Subject** property of the Send Email block, set the **Subject** property to:

```
Voicemail call from $FName$ $LName$
```

Any embedded spaces or quotes (") are included in the resulting string. So, in the above example, if **FName** contained the string "John" and **LName** contained the string "Doe", the resulting string is "Voicemail call from John Doe".

To include a literal dollar sign (\$) in a string, precede the dollar sign with another dollar sign. For example, if you want to assign a variable a string that contains a dollar sign and you have a variable called **Cost** that contains the string "4.00", specify it like this:

```
The cost for that is $$$Cost
```

The resulting string is "The cost for that is \$4.00".

Note

Be careful when using variables in the **Expression** property of the Condition block. The way that the interpreter expands variables into an intermediate string before using them can cause the interpreter to incorrectly evaluate an expression. This is the only property in CCaaS Designer in which you must use quotes (") to denote a string. Furthermore, if a variable is to be evaluated as a string, enclose the variable in quotes.

Examples

If you have a variable called **CallerEntry** and you want to check if it is equal to the number **1234**, the expression property is:

```
$CallerEntry$ = 1234
```

The resulting string is '1234 = 1234'.

If you have the same situation as above, but you want to check if **CallerEntry** is equal to the string **"1234"**, then the expression property is:

```
"$CallerEntry$" = "1234"
```

The resulting string is `'"1234" = "1234"'`. Both the variable and the literal 1234 are enclosed in quotes. Using quotes forces both sides of the equation to be the same data type, which is string.

If you have two variables called **FName** and **LName** and you wanted to check to see if the caller's name is equal string **"John Doe"**, the expression is:

```
"$FName$ $LName$" = "John Doe"
```

The resulting string is `'"John Doe" = "John Doe"'`. The entire left-hand side of the expression is enclosed in quotes, not each variable. If you specified the expression as, `"$FName$" "$LName$" = "John Doe"`, the resulting string would be `'"John" "Doe" = "John Doe"'` and an error would occur in the evaluation of that expression.

How to use the `$empty$` constant

For the CallPreSent script, you can use the `$empty$` constant in the Edit Email block to set a property to empty.

For example, in the Edit Email block, if you put `$empty$` in the **CC** field, any value previously there is removed and it is set to an empty value.

Log out

Note

If you are using an external identity provider such as ADFS or EIS, when you click **Log out of your CCaaS session**, you are also logged out of your external identity provider.

1. At the top right of the main CCaaS Designer page, click **Logout**.
2. In the **Logout** dialog, click **OK**.
3. Do one of the following:
 - If you logged in directly, the **Logout** page appears:
 - If you want to return to CCaaS Designer, click **Log back in**.
 - If you have finished working with CCaaS, click **Log out of your CCaaS session**. The **Confirm logout** dialog appears. Click **Yes**. The following message appears: *You are now logged out of your session*.
 - If you logged in using Unified Admin, the **App Selector** page appears:
 - If you want to return to CCaaS Designer or use another administration application that you are allowed to access, click the appropriate button.
 - If you want to change the username, click **Change user** to go to the **Welcome to CCaaS** page.
 - If you have finished working with CCaaS, click **Logout**. The **Confirm end session** dialog appears. Click **Yes**. The following message appears: *You are now logged out of your session*.

Additional logout messages

When you click **Log out of your CCaaS session**, the following message may appear:

- *Logout was not completed successfully due to the Authentication Server being unavailable. Please contact your system administrator.*
1. Tell your system administrator about the issue and wait for it to be resolved.
 2. When it is resolved, click **OK** to close the message dialog.
 3. Click **Log out of your CCaaS session** again.

The application will return to the **Welcome to CCaaS** page. Depending on the problem resolution, this may take slightly longer than normal.

Session expiry

Your CCaaS application session will become invalid when:

- The login session time exceeds the maximum CCaaS login session lifetime specified by your administrator.
- You log out of your CCaaS session using another application.
- You clear all cookies in the browser.

When your CCaaS application session becomes invalid, you must log in again. See [Log in](#).

If you are logged in with an external identity provider (ADFS or EIS) and that provider's login session lifetime exceeds the CCaaS login session lifetime:

- If you log in again using the general application URL, for example, *https://ccsp1.pj16.loc/Designer*, you need to enter only your username on the **Welcome to CCaaS** page, and are then automatically authenticated against your external identity provider.
- If you log in again using the tenant's direct login URL, for example, *https://ccsp1.pj16.loc/portal/t1/Designer*, you are automatically authenticated against your external identity provider and do not need to enter your username.

3: Workspaces

This section contains the following topics:

- [Workspace editor overview](#)
- [Manage workspace templates](#)
- [Open a workspace](#)
- [Import a workspace](#)
- [Manage workspaces](#)
- [Publish workspaces](#)
- [Manage tables](#)
- [Manage workspace files](#)
- [Manage version history](#)

See also [Get started page](#)

Workspace editor overview

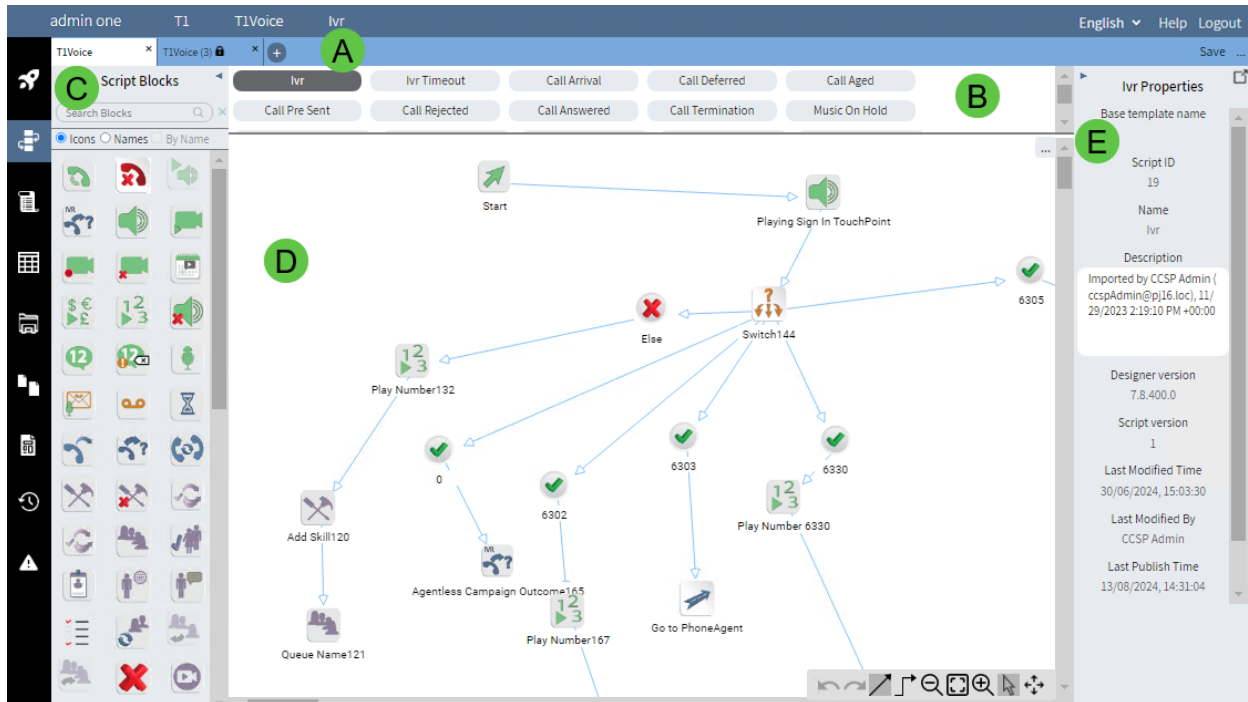
A — [Workspace tab bar](#)

B — [Script tabs panel](#)

C — [Script blocks panel](#)


D — [Flow editor](#)

E — [Properties panel](#)



Workspace tab bar

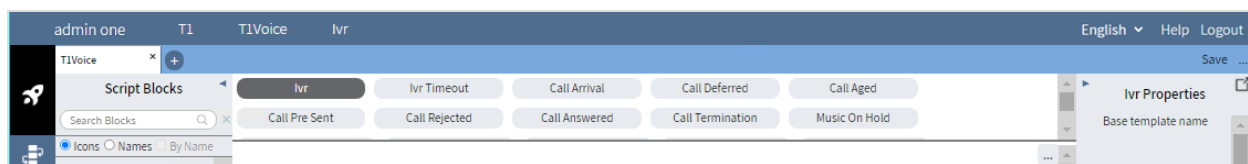
The workspace tab bar contains a tab for each opened workspace and snapshot.

To open a new workspace click . The **New workspace** tab displays the most recent workspace list and the buttons to create, open, or import a workspace that appear on the **Get started** page. For details, see [Get started](#).

To open a snapshot of the open workspace, see [Manage version history](#).


To close a workspace or snapshot tab, click the X on the tab. If the **New workspace** tab is the last tab left open, you cannot close it because the Workspace editor must have at least one tab.

Workspace tabs



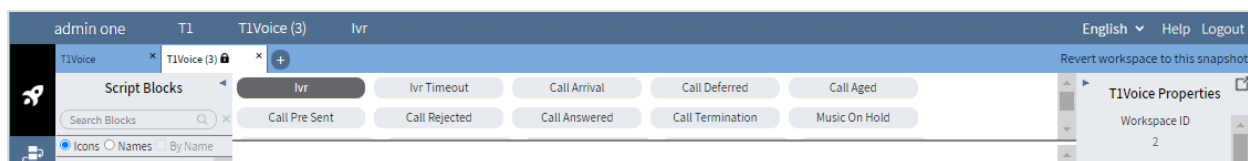
To save the currently selected workspace, click **Save**.

Workspace menu

Use the workspace menu  as follows:

- **Rename** — to rename the currently selected workspace. See [Rename a workspace](#).
- **Save as** — to save a copy of the currently selected workspace under a new name. The new workspace will be loaded in place of the current workspace. See [Save a workspace with a new name](#).
- **Save as Platform Template** (landlord only) — to save the currently selected workspace as type Custom-platform. See [Save a workspace as a Platform Template](#).
- **Save as Tenant Template** — to save the currently selected workspace as type Custom-tenant. See [Save a workspace as a Tenant Template](#).
- **Publish** — to publish the currently selected workspace. See [Publish a workspace](#).
- **Delete** — to delete the currently selected workspace. See [Delete a workspace](#).
- **Reload Last Published** — to reload the currently selected workspace from the last published location. See [Reload last published workspace](#).

Snapshot tabs



Each opened snapshot appears in its own tab, with the version number of the workspace in brackets and a lock icon. You cannot change the scripts, tables, or files in a snapshot. The script menu does not appear in the editor.

To revert the owning workspace of a snapshot to the snapshot's data, click **Revert workspace to this snapshot**. See [Revert a workspace to a snapshot](#).

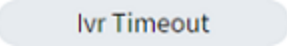
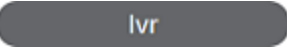
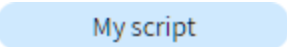
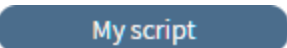
Script tabs panel

The Scripts tabs panel contains buttons for all the available scripts in the selected workspace, each of which corresponds to an event within the CCaaS system. See [Flowchart of events](#) and [Predefined events](#).

You can add buttons for scripts or for items not necessarily related to events, but that can be called from scripts. See [Add a new script](#).

To select a script to edit, click on the button with the script name. You can expand the Script tabs panel to see all the available scripts by dragging the bar down. You can also use the scroll bar to see the scripts.

The script buttons are color-coded as follows:

-  — indicates a system script.
-  — indicates a system script that is currently selected.
-  — indicates a custom script added by a user.
-  — indicates a custom script that is currently selected.


When you edit a script, an asterisk appears beside its name and an asterisk appears beside the name of the workspace in the Workspace tab bar.

When you save the script, both asterisks disappear.

Script menu

Note

The script menu only appears on workspace tabs. It is not available on snapshot tabs.

Use the script menu  as follows:

- **Add script** — to add a new script. See [Add a new script](#).
- **Add system script** — to add a system script. See [Add a system script](#).
- **Duplicate** — to duplicate the currently selected script.
- **Save** — to save the currently selected script.
- **Publish** — to publish the currently selected script. See [Publish a script](#).
- **Validate** — to validate the currently selected script. See [Validate a script](#).
- **Delete** — to delete the currently selected script. The Delete option is disabled for system scripts.

Script blocks panel

The Script blocks panel contains all of the available script elements from which you construct scripts. For a complete list of the blocks, their attributes, and uses, see [Blocks](#).


You can view the blocks by:

- **Icons** — the icons are arranged in the same order as the legacy CCaaS Designer blocks palette. When you point to a block, its name appears in a tooltip.
- **Names** — the icons with their names are arranged in the same order as the legacy CCaaS Designer blocks palette.
- **By name** — the icons with their names are sorted alphabetically. This option is not available when viewing only icons.

To see all the blocks, scroll the panel.

To search for a block, start typing in the search box located at the top of the panel. As soon as you begin typing, only blocks that contain those characters appear.

When you open the workspace to a particular script, blocks that are invalid for that script are unavailable in the Script blocks panel.

Custom blocks appear as icons with a question mark . After importing custom blocks, you must reload the current workspace to refresh the blocks grid and show the custom blocks at the top of the list.

To hide the Script blocks panel, click the left arrow. To reopen the Script blocks panel, click the right arrow.

Flow editor

Note

You cannot edit the flow on snapshot tabs.

The Flow editor is the area in which you lay out the script's call flow. You can scroll the Flow editor vertically and horizontally. If you double-click on the Flow editor background, the mouse cursor goes into grab mode.

When you create a new script, it contains only a Start block. You cannot delete or rename a Start block.

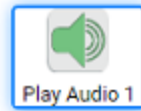
To add a block to the Flow editor, drag it from the Script blocks panel to the Flow editor. As blocks are added to the workspace, they are assigned unique names by appending unique numbers to the display name of the block. For example, if you add a Play Audio block to a script that already has 315 blocks in the workspace, it will appear as PlayAudio316.

You can change the name of any block through the Properties panel or by editing the label in the Flow editor.

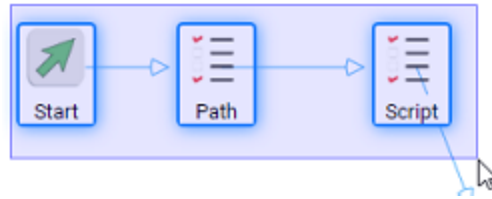
Note

Block names must be unique within each script and must not exceed 256 characters in length.

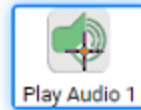
To select a block, click it. A blue border appears around it.



To select multiple blocks, click and drag a selection box around them. Alternatively, use the **Ctrl** or **Shift** keys to add individual blocks to your selection and to deselect blocks.



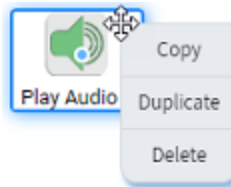
To connect two blocks, hover the mouse over the center of the icon so the cursor changes to a cross hair. Click and drag from the center of the first block to the second. A directed line with an arrowhead appears showing the connection and the direction of the call flow.



To move blocks, hover the mouse over the icon so the cursor changes to a move sign. Click the block and drag it to the new location. If you select multiple blocks, they move as a unit.



To copy blocks with their properties between scripts, use the right-click shortcut menu.



Block shortcut menu

When you right-click a block, the shortcut menu for the block appears. All blocks have the following actions:

- **Copy**
- **Duplicate**
- **Delete**

Some blocks have additional actions. For details about the individual blocks, see the [Blocks](#) section.

Flow editor shortcut menu










When you right-click the Flow editor background, the shortcut menu appears with the following actions:

- **Undo**
- **Redo**
- **Paste**

The actions are individually disabled if there is currently nothing to undo, redo, or paste.

Toolbar

Use the toolbar icons as follows.

Icon	Description
	Undo the last block action. For limitations, see below.
	Redo the last block action. For limitations, see below.
	Use straight connectors
	Use elbow connectors
	Zoom out script display
	Zoom script display to fit
	Zoom in script display
	Selection mode
	Editor drag mode

Undo or Redo will reverse or perform the following actions:

- Block creation (including paste of copied blocks)
- Block deletion
- Block move
- Block rename
- Block property change
- Connection creation
- Connection deletion

Properties panel

The Properties panel shows you information about the currently selected workspace, script, or block and lets you enter and modify information for it.

Note

You can use the Tab key to move between the fields on the Properties panel. Use Tab to move forward and Shift + Tab to move backward.

To hide the Properties panel, click the right arrow. To view the Properties panel, click the left arrow.

To undock the properties, click . To dock the properties to the right side again, click  or .

Note

The Undo action will reverse changes to a block property but it will not reverse changes to a workspace or script property.

Workspace properties

To see the properties of a workspace, click its tab in the Workspace tab bar. The properties are:

- **Workspace ID** — new unsaved workspaces have a negative ID.
- **Name** — the name of the workspace.
- **Description** — the description of the workspace. This information can be changed in the workspace editor.
- **Version** — the version number of the workspace. This number is incremented each time you save the workspace or an individual script.
- **Designer version** — the version of Designer that the workspace was saved in.
- **Last modified time** — this is currently the UTC time at which the workspace was last saved in the database. The time is updated when you save the whole workspace or an individual script.
- **Last modified by** — the name of the person who last saved the workspace.
- **Last publish time** — this is currently the UTC time at which the workspace was last published in the tenant's IVR root path, configured in Admin. The time is updated when you publish the whole workspace but not when you publish an individual script, table, or workspace file.
- **Last publish destination** — the name of the folder within the tenant's IVR root path in which the XML files were last saved.

Snapshot properties

To see the properties of a snapshot, click its tab in the Workspace tab bar. The properties are:

- **Workspace ID**
- **Workspace snapshot ID**
- **Workspace snapshot version** — the version number of this snapshot.
- **Workspace snapshot description** — the description of this snapshot. This information can be edited on the Version History page. See [Edit a snapshot's description](#).

- **Name** — the name of the workspace.
- **Description** — the description of the workspace.
- **Version** — the version number of the workspace.
- **Designer version** — the version of Designer that the workspace was saved in.
- **Last modified time** — this is currently the UTC time at which the workspace was saved for this snapshot.
- **Last modified by** — the name of the person who saved the workspace at this point.
- **Last publish time** — this is currently the UTC time at which this version of the workspace was published in the tenant's IVR root path, configured in Admin.
- **Last publish destination** — the name of the folder within the tenant's IVR root path in which the XML files for this version of the workspace were last saved.

Script properties

To see the properties of a script, click its tab in the Script tabs panel or click on the background of the Flow editor. The properties are:

- **Script ID** — new unsaved scripts have a negative ID.
- **Name**
- **Description** — this information can be edited.
- **Designer version** — the version of Designer that the script was saved in.
- **Script version** — the version of the script. This number is incremented each time you save the workspace or an individual script.
- **Last modified time** — the time is updated when you save the whole workspace or the individual script.
- **Last modified by**
- **Last publish time** — the time is updated when you publish the whole workspace or an individual script.
- **Published filename** — the name of the xml file.

Block properties

To see the properties of a block, click the block in the Flow editor. Note that if you select multiple blocks in the Flow editor, the Properties panel is blank.

In the **Comments** property, you can enter a description of how the block is used in your script.

For details about the properties of the individual blocks, see the [Blocks](#) section.


Manage workspace templates

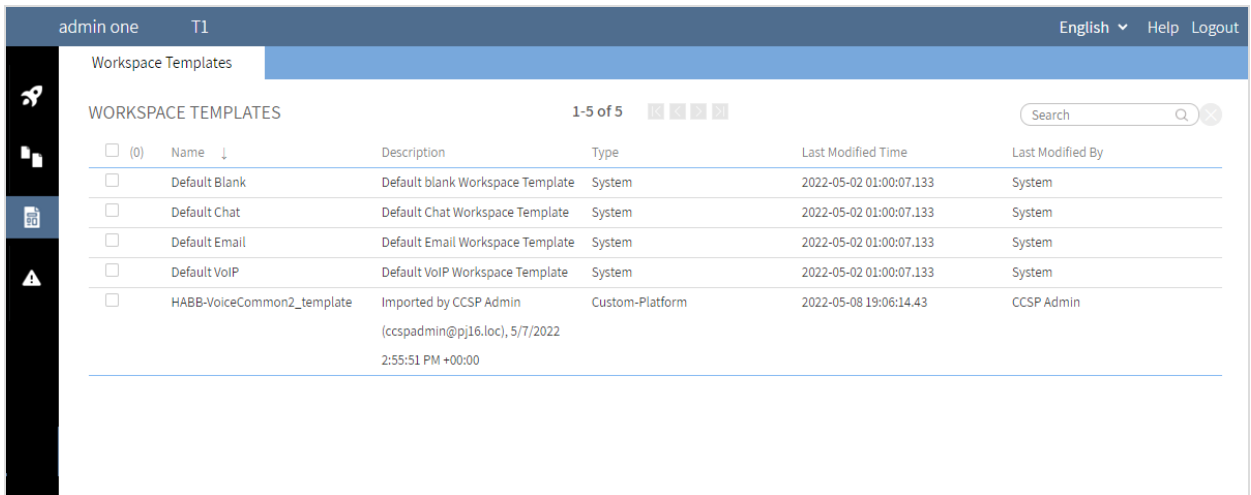
Select a template

Delete a custom-tenant template

Note

The system templates show only the scripts that are appropriate for the media type selected. For example, the Default Chat template only shows 5 scripts as these are the only ones applicable to the Chat media type.

1. On the navigation pane, click  **Workspace Templates**. The **Workspace templates** list appears. For navigation information, see [List view](#).
2. To see the custom templates at the top of the list, click the **Type** column heading.



<input type="checkbox"/> (0)	Name ↓	Description	Type	Last Modified Time	Last Modified By
<input type="checkbox"/>	Default Blank	Default blank Workspace Template	System	2022-05-02 01:00:07.133	System
<input type="checkbox"/>	Default Chat	Default Chat Workspace Template	System	2022-05-02 01:00:07.133	System
<input type="checkbox"/>	Default Email	Default Email Workspace Template	System	2022-05-02 01:00:07.133	System
<input type="checkbox"/>	Default VoIP	Default VoIP Workspace Template	System	2022-05-02 01:00:07.133	System
<input type="checkbox"/>	HABB-VoiceCommon2_template	Imported by CCSP Admin (ccspadmin@pj16.loc), 5/7/2022 2:55:51 PM +00:00	Custom-Platform	2022-05-08 19:06:14.43	CCSP Admin

The columns are:


- **Name**
- **Description**
- **Type** — system, custom-platform, or custom-tenant.
- **Last modified time** — this is currently the UTC time at which the workspace template was saved in the database.
- **Last modified by**

Select a template

To select a template, click on the row.

The Workspace editor opens with a new workspace based on the chosen template. The default workspace name is Untitled-1. Any further new templates created in the session are named Untitled-2, Untitled-3, and so on.


Delete a custom-tenant template

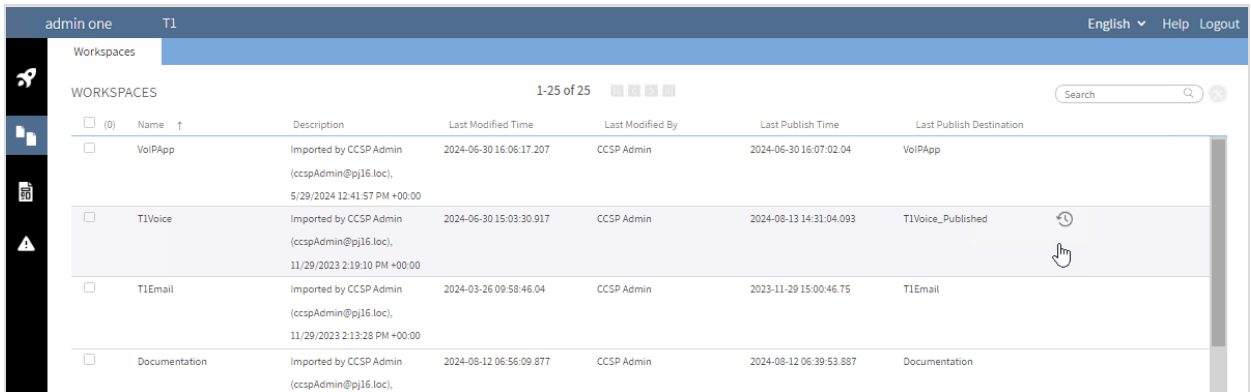
1. In the **Workspace templates** list, select the check box in front of the custom-tenant template to delete.
2. Click .
3. Confirm the deletion.


Open a workspace

Note


- You can open a recently used workspace directly from the [Get started](#) page.
- You can also open the **Workspaces** page by clicking **Open workspace** on the [Get started](#) page.

1. On the navigation pane, click  **Workspaces**. The **Workspaces** list appears, with the workspaces available in the database for the selected tenant. For navigation information, see [List view](#).



	Name	Description	Last Modified Time	Last Modified By	Last Publish Time	Last Publish Destination
<input type="checkbox"/>	VoIPApp	Imported by CCSP Admin (ccspAdmin@pj16.loc), 5/29/2024 12:41:57 PM +00:00	2024-06-30 16:06:17.207	CCSP Admin	2024-06-30 16:07:02.04	VoIPApp
<input type="checkbox"/>	TIVoice	Imported by CCSP Admin (ccspAdmin@pj16.loc), 11/29/2023 2:19:10 PM +00:00	2024-06-30 15:03:30.917	CCSP Admin	2024-08-13 14:31:04.093	TIVoice_Published 
<input type="checkbox"/>	TIEmail	Imported by CCSP Admin (ccspAdmin@pj16.loc), 11/29/2023 2:13:28 PM +00:00	2024-03-26 09:58:46.04	CCSP Admin	2023-11-29 15:00:46.75	TIEmail
<input type="checkbox"/>	Documentation	Imported by CCSP Admin (ccspAdmin@pj16.loc),	2024-08-12 06:56:09.877	CCSP Admin	2024-08-12 06:39:53.887	Documentation

The columns are:

- **Name** — the name of the workspace.
 - **Description** — the description of the workspace. This information can be changed in the workspace editor.
 - **Last modified time** — this is currently the UTC time at which the workspace was last saved in the database. The time is updated when you save the whole workspace or an individual script.
 - **Last modified by** — the name of the person who last saved the workspace.
 - **Last publish time** — this is currently the UTC time at which the workspace was last published in the tenant's IVR root path, configured in CCaaS Admin. The time is updated when you publish the whole workspace but not when you publish an individual script, table, or workspace file.
 - **Last publish destination** — the name of the folder within the tenant's IVR root path in which the XML files were last saved.
2. To resort the workspaces list, click a column heading. Each time you sort by a column heading, you update the list of workspaces.
 3. Do one of the following:
 - Click the workspace to load. Next see [Workspace editor overview](#).
 - Hover your mouse over the workspace and click  **Version history**. See [Manage version history](#).



Import a workspace

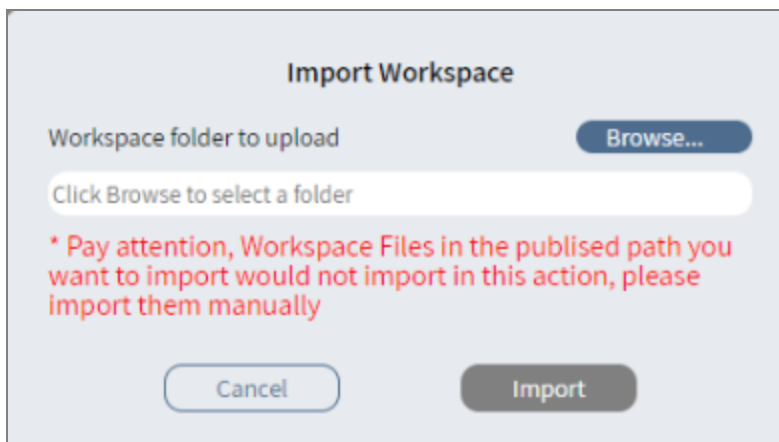
Note

- Import of a workspace should only be done as part of the process of migrating a tenant from the legacy to the new Designer.
- Two workspaces cannot exist for the same tenant with the same name. If you try to import a workspace that you have already imported, you are prompted for a new name for the workspace.
- Before importing a workspace, you must remove any backup copies of scripts and empty script files from the workspace folder. Script files that share the same Display element name cause the import to fail with a *Duplicated script name* error if the *workspace.xml* file uses this name for a script.
- After importing a workspace with existing VOIP scripts, you must manually add the Post Call script, if required. See [Post Call script](#).

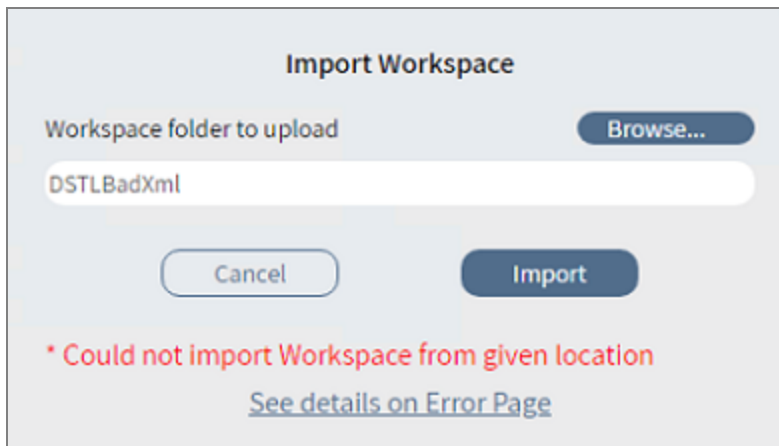
Caution

Workspace files in the publish location are not automatically imported. You must manually import them.

1. On the navigation pane, click  **Tenant Home** or in the Workspace editor, click . The **Get started** page appears.
2. Click **Import workspace**.
3. In the **Import Workspace** dialog, click **Browse**.
4. Select the file system folder to upload and click **Upload**.
5. Confirm that you want to upload all the files to the site.
6. In the **Import Workspace** dialog, click **Import**.



7. If the import fails, click **See details on the Error Page**. See [Errors page](#).



Manage workspaces

Save a copy of the workspace

Save a workspace as a tenant template

See [Publish workspaces](#).

Revert a workspace.

Reload the last published workspace


Delete a workspace

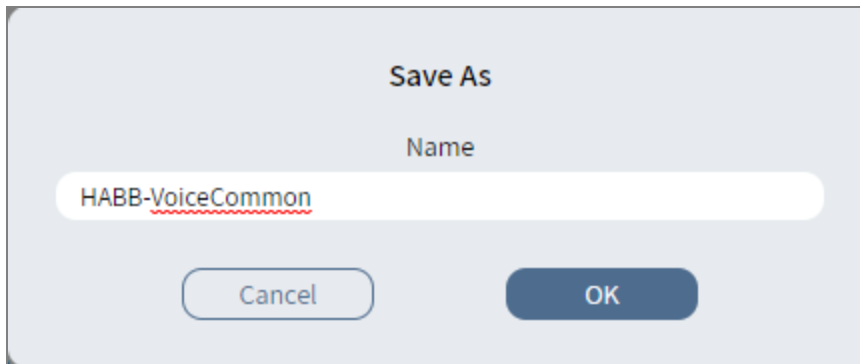
Note

If two users, A and B are editing the same workspace, whoever saves the workspace first prevents the other user from saving their changes under the same name.

For example: user A edits and saves the IVR script and then user B attempts to save their copy of the IVR script with their changes. In this case, user B receives an error that the workspace has been modified by someone else. User B then has to use the Save as option to save a copy of the workspace. They can then load the original version again and reconcile their changes with the person who made the updates.

Save a copy of the workspace


1. In the Workspace editor, in the workspace tab bar, select the workspace to duplicate.
2. Click  and then click **Save As**. The **Save As** dialog appears.

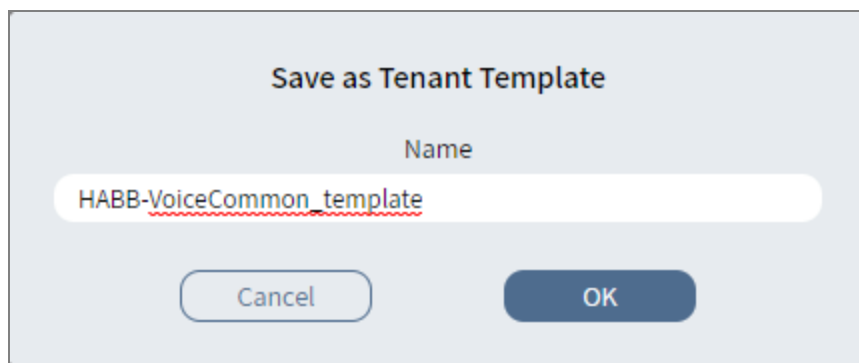


3. Enter the duplicate workspace name.
4. Click **OK**.
5. The message *Workspace successfully saved* appears. Click **OK**.

The duplicated workspace is loaded in place of the current workspace.

Save a workspace as a tenant template

1. In the Workspace editor, in the workspace tab bar, select the workspace to save as a tenant template.
2. Click  and then click **Save As Tenant Template**. The **Save As Tenant Template** dialog appears.



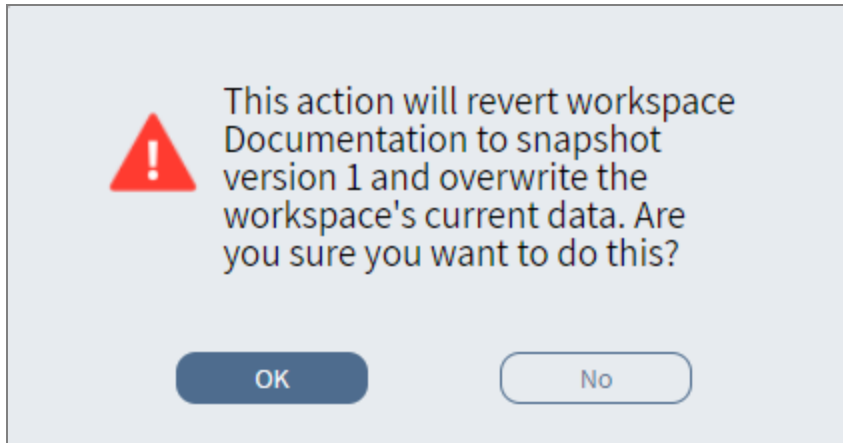
3. Enter the new template name.
4. Click **OK**.
5. The message *Workspace template successfully saved* appears. Click **OK**.

The new template of type Custom-tenant appears in the Workspace templates list.

Revert a workspace

You can revert a workspace to a previous version (snapshot).

1. In the Workspace editor, in the workspace tab bar, select the snapshot to revert to. See [Open a snapshot](#).
2. Click **Revert workspace to this snapshot**. A warning that *this action will overwrite the workspace's current data* appears.



3. Click **OK**.
4. The message *The workspace has been successfully reverted to snapshot version [number]* appears. Click **OK**.


Reload the last published workspace

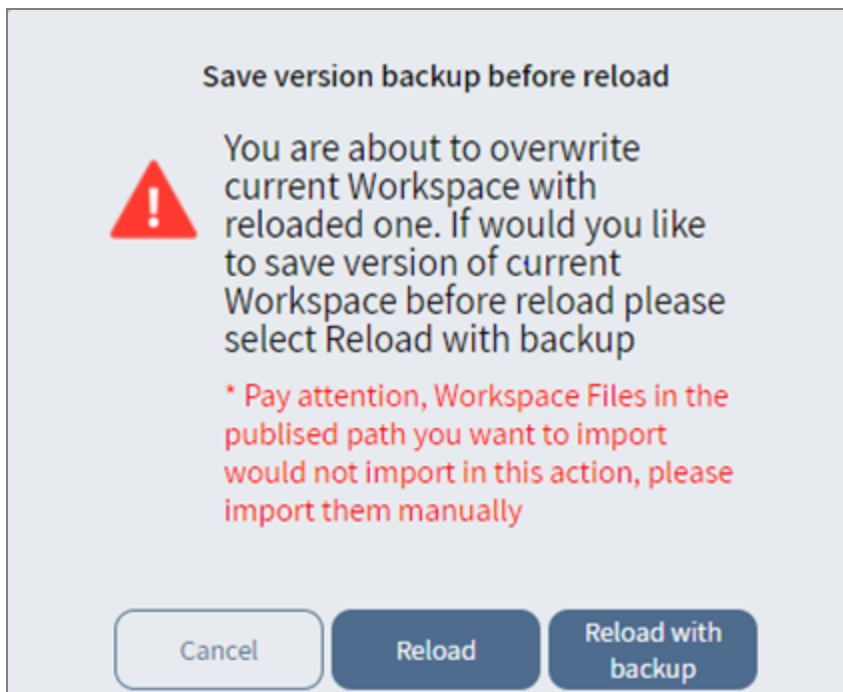
Note

- You can reload (import) the last published version of the current workspace from the network. The imported version overwrites the current data in the workspace.
- To avoid losing any unpublished changes in the workspace, you can backup this version before importing the workspace from the network.
- If the reload fails, click **See details on the Error Page**. See [Errors page](#).

Caution

Workspace files are not automatically imported when you reload the last published version of the current workspace from the network. You must manually import them.

1. In the Workspace editor, in the workspace tab bar, select the workspace to reload.
2. Click  and then click **Reload Last Published**. The **Reload Last Published** dialog appears.




3. Do one of the following:
 - a. To reload the current workspace from the network, click **Reload**.
 - b. To save a backup version of the current data in the workspace before reloading the workspace from the network, click **Reload with backup**.

4. The message *Workspace successfully reloaded* appears. Click **OK**.
5. If you clicked **Reload with backup**, you can go to the Version History page to view the new backup version of the workspace.



Delete a workspace

Note

When you delete a workspace, only the database is updated. Any published files remain.

1. In the Workspace editor, in the workspace tab bar, select the workspace to delete.
2. Click  and then click **Delete**.
3. Confirm the deletion.

You can also delete a workspace from the Workspaces list.

1. On the navigation pane, click  **Workspaces**.
2. In the **Workspaces** list, select the check box in front of the workspace to delete.
3. Click .
4. Confirm the deletion.

Publish workspaces

[Location to publish to](#)

[Workspace versions](#)

[Publish a workspace](#)

Location to publish to

CCaaS Designer publishes workspaces to a predefined location for the tenant. This location is defined by the IVR root path that is configured for the tenant in CCaaS Admin.

The IVR root path must be a valid network or local filesystem location that is accessible to the user under which the CCaaS Designer App Pool runs. The IVR root path must be unique for each tenant but multiple tenant folders can use the same parent folder. For example, tenants T1, T2 and T3 can have IVR root paths defined as below:

```
\\CCSP1\TenantIVRs\Published\T1
```

```
\\CCSP1\TenantIVRs\Published\T2
```

```
\\CCSP1\TenantIVRs\Published\T3
```

When Designer publishes workspaces, scripts and tables, if the leaf folder does not exist, it is created with permissions for landlord users and tenant administrator for that tenant. However, the parent folder must exist prior to the publish operation being attempted, otherwise the publish will fail. So in the above examples, `\\CCSP1\TenantIVRs\Published` must exist beforehand.

Note

Workspaces and scripts are always published from the database. If you are editing a script and select either the workspace or script Publish option, the current script or workspace is saved before publish. If the workspace is new and unsaved, you are prompted to provide a name before saving.


Workspace versions

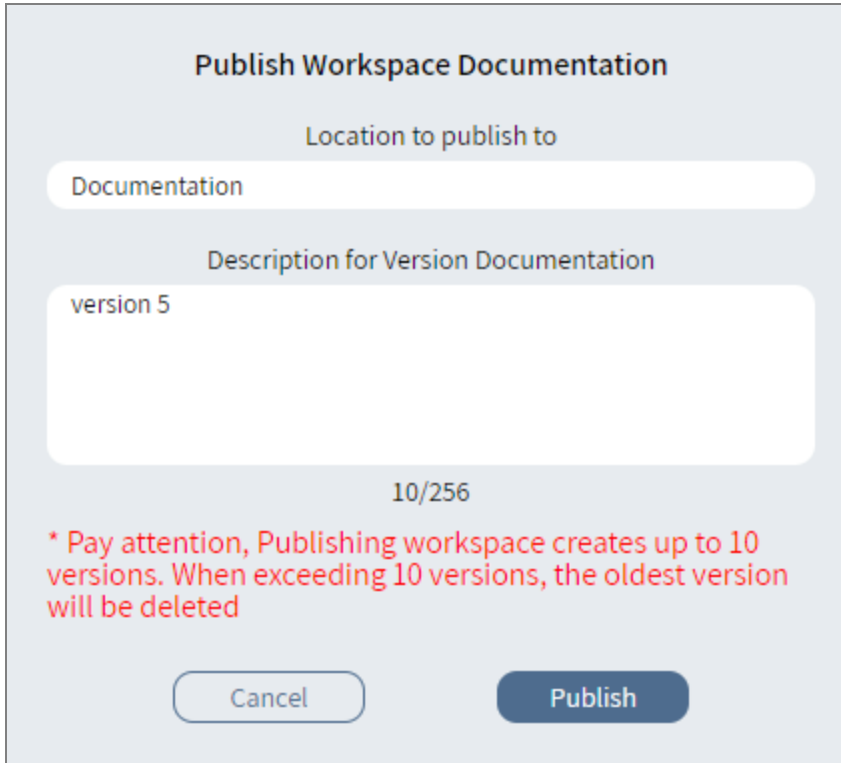
CCaaS Designer automatically creates a version (snapshot) of each published/republished workspace. The whole workspace (including scripts, tables, and files) is saved in the database and presented on the Version History page. See [Manage version history](#).

CCaaS Designer stores a maximum of 10 versions in the database and then replaces the oldest existing version with the new one.

Publishing an individual script, table, or file does not create a backup version of it. You must publish a whole workspace to create a backup version of its scripts, tables, and files.

Publish a workspace

1. In the Workspace editor, in the workspace tab bar, select the workspace to publish.
2. Click  and then click **Publish**. The **Publish Workspace [name]** dialog appears.



The screenshot shows a dialog box titled "Publish Workspace Documentation". It contains two input fields: "Location to publish to" with the text "Documentation" and "Description for Version Documentation" with the text "version 5". Below the description field is a character count "10/256". A red warning message states: "* Pay attention, Publishing workspace creates up to 10 versions. When exceeding 10 versions, the oldest version will be deleted". At the bottom are "Cancel" and "Publish" buttons.

3. Enter the **Location to publish to**.
4. Enter a **Description** for this version of the workspace.
5. Click **Publish**.
6. The message *Workspace successfully published* appears. Click **OK**.

Manage tables

[Upload a new table](#)

[Upload an updated table](#)


[Download a table](#)

[Publish a table](#)


[Rename a table](#)

[Delete a table](#)

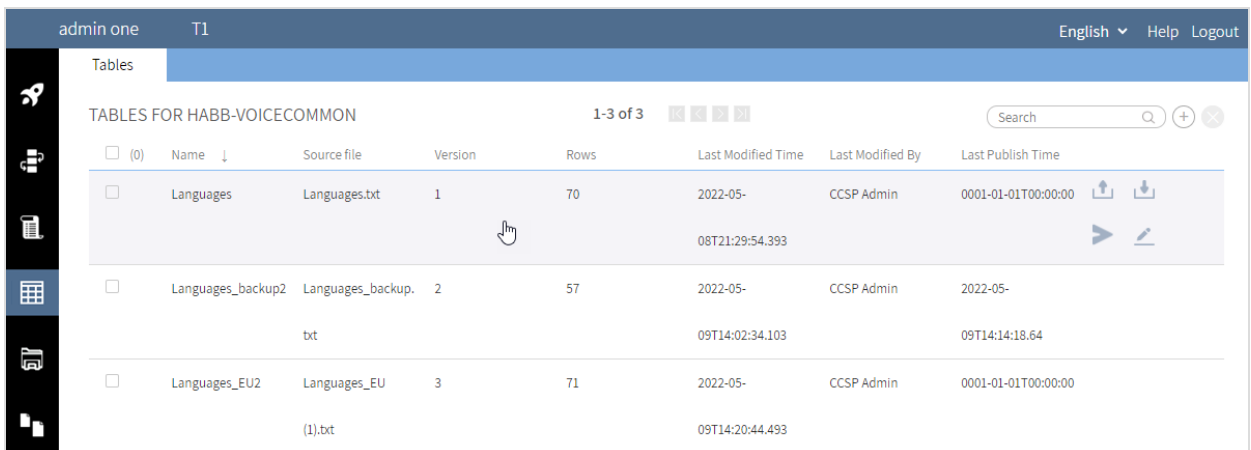
Note

- The  **Tables** icon is only visible in the navigation pane after you open a workspace. You manage the tables for that workspace on the **Tables** page.
- To change a table's contents, you must download it from the database as a text file, edit the text file, and then upload it to the database again. You then need to republish it.
- You select these tables in the Table Name property of a Table or Table Lookup block.
- When going to the **Tables** page from a snapshot tab, the **Tables** page is locked and the only available action is Download.





1. Open the workspace that contains the tables. See [Open a workspace](#).

2. On the navigation pane, click  **Tables**. The **Tables for [workspace]** list appears. For navigation information, see [List view](#).

3. To see the action buttons for a particular table, hover your mouse over that row.



The screenshot shows the 'Tables' management page in a web application. The page title is 'Tables' and the workspace is 'HABB-VOICECOMMON'. The table list has the following columns: Name, Source file, Version, Rows, Last Modified Time, Last Modified By, and Last Publish Time. The first row is 'Languages' with 70 rows, last modified on 2022-05-08T21:29:54.393 by CCSP Admin. The second row is 'Languages_backup2' with 57 rows, last modified on 2022-05-09T14:02:34.103 by CCSP Admin. The third row is 'Languages_EU2' with 71 rows, last modified on 2022-05-09T14:20:44.493 by CCSP Admin. The interface includes a search bar, a table icon in the navigation pane, and a mouse cursor hovering over the first row.

	Name ↓	Source file	Version	Rows	Last Modified Time	Last Modified By	Last Publish Time	
<input type="checkbox"/>	Languages	Languages.txt	1	70	2022-05-08T21:29:54.393	CCSP Admin	0001-01-01T00:00:00	 
<input type="checkbox"/>	Languages_backup2	Languages_backup.txt	2	57	2022-05-09T14:02:34.103	CCSP Admin	2022-05-09T14:14:18.64	 
<input type="checkbox"/>	Languages_EU2	Languages_EU(1).txt	3	71	2022-05-09T14:20:44.493	CCSP Admin	0001-01-01T00:00:00	

The columns are:


- **Name**
- **Source file**
- **Version** — the version of the source file.
- **Rows**
- **Last modified time** — this is currently the UTC time at which the table was saved in the database.
- **Last modified by**
- **Last publish time** — this is currently the UTC time at which the table was published in the tenant's IVR root path, configured in CCaaS Admin.

Upload a new table

Note

- The table file you upload to the database can be an *.xml* file or a tab-delimited text file.
- Do not give columns in your tables names with embedded blanks, for example, "Phone number". Column names must start with an alphabetic character. Subsequent characters in a column name must contain alpha/or numeric characters.
- If you need to leave a table cell empty, enter a non-printing alphanumeric character, for example, a space, in the cell. Empty cells may cause the table to be corrupted.


1. On the **Tables** page, click  **Add**. The **Upload Table** dialog appears.

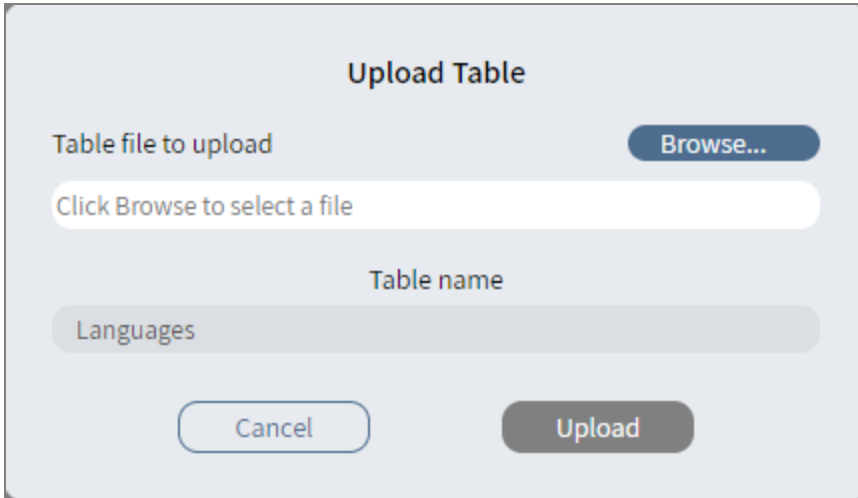


The screenshot shows a dialog box titled "Upload Table". It has two input fields. The first is labeled "Table file to upload" and contains the text "Key.txt". To the right of this field is a "Browse..." button. The second input field is labeled "Table name" and contains the text "Categories". At the bottom of the dialog, there are two buttons: "Cancel" and "Upload".

2. Click **Browse** and select the text file to upload.
3. Enter the table name.
4. Click **Upload**.
5. The message *Table successfully uploaded* appears. Click **OK**.

Upload an updated table


1. In the **Tables** list, hover your mouse over the table to upload and click  **Upload**.

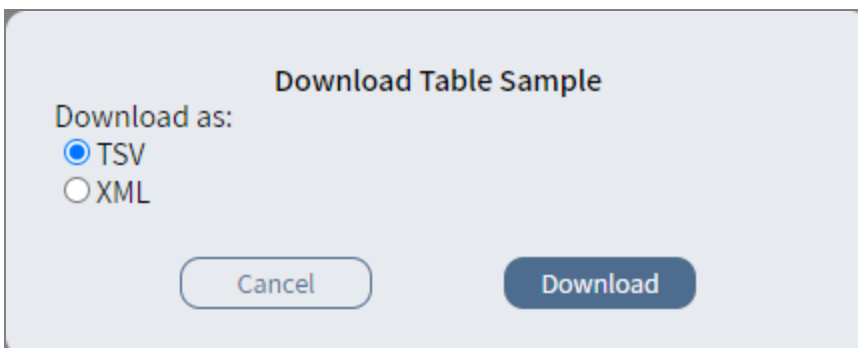


The screenshot shows a dialog box titled "Upload Table". It contains a "Table file to upload" label with a "Browse..." button to its right. Below this is a text input field with the placeholder text "Click Browse to select a file". Underneath is a "Table name" label with a text input field containing the word "Languages". At the bottom of the dialog are two buttons: "Cancel" and "Upload".

2. Click **Browse** and select the text file to upload.
3. You cannot change the table name.
4. Click **Upload**.
5. The message *Table successfully uploaded* appears. Click **OK**.

Download a table

1. In the **Tables** list, hover your mouse over the table to download and click  **Download**. The **Download Table [name]** dialog appears.





The screenshot shows a dialog box titled "Download Table Sample". It contains a "Download as:" label followed by two radio button options: "TSV" (which is selected) and "XML". At the bottom of the dialog are two buttons: "Cancel" and "Download".

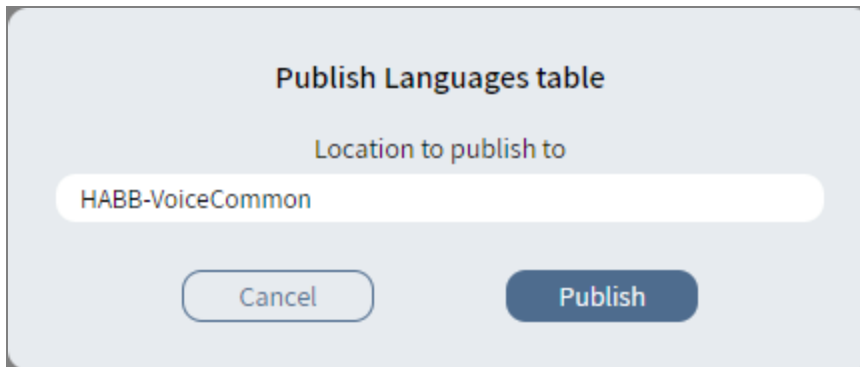
2. Select whether to download the file as a tab-delimited text file or an .xml file.
3. Click **Download**. The text file appears in your **Downloads** folder.

Publish a table

Note


- If an error message appears saying that the IVR root path cannot be found, or CCaaS Admin cannot be contacted, click  **Publish** again. If there is still a problem, see the configuration information in [Publish a workspace](#).
- Publishing an individual table does not create a backup version of it. You must publish a whole workspace to create a backup version of its scripts, tables, and files. For more information, see [Publish a workspace](#).

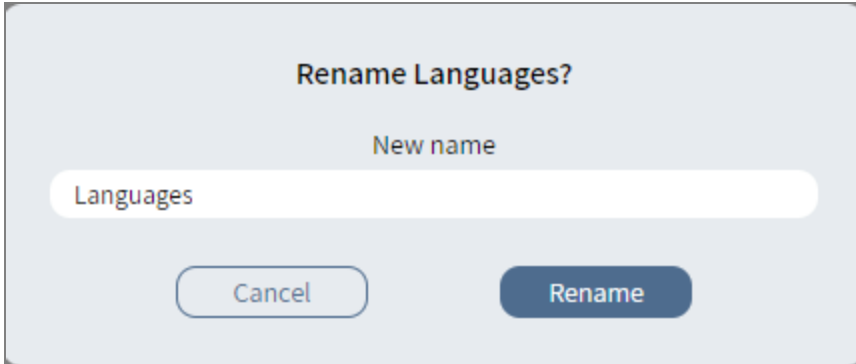
1. In the **Tables** list, hover your mouse over the table to publish and click  **Publish**. The **Publish [name] Table** dialog appears.



2. Enter the **Location to publish to**. You can use the default workspace location or change the name to publish to a different workspace folder.
3. Click **Publish**.
4. The message *Table successfully published* appears. Click **OK**.

Rename a table


1. In the **Tables** list, hover your mouse over the table to rename and click  **Rename**. The **Rename [name]** ? dialog appears.



The image shows a dialog box titled "Rename Languages?". It has a light gray background and rounded corners. At the top, the title "Rename Languages?" is centered. Below the title, the text "New name" is centered. Underneath, there is a white text input field with the word "Languages" entered. At the bottom of the dialog, there are two buttons: a light gray "Cancel" button on the left and a dark blue "Rename" button on the right.

2. Enter the new name.
3. Click **Rename**.

Delete a table

1. In the **Tables** list, select the check box in front of the table to delete.
2. Click  **Delete**.
3. Confirm the deletion.

Manage workspace files

[Upload a new file](#)

[Upload an updated file](#)


[Download a file](#)

[Publish a file](#)


[Rename a file](#)

[Delete a file](#)



Note

- The  **Workspace Files** icon is only visible in the navigation pane after you open a workspace. You manage the files for that workspace on the **Workspace Files** page.
- You can upload *.xml*, *.txt*, and *.wav* files to the workspace.
- You select these files in your workflow blocks.
- Table *.xml* or *.txt* files must be uploaded on the **Tables** page. See [Manage tables](#).
- When going to the **Workspace Files** page from a snapshot tab, the **Workspace Files** page is locked and the only available action is Download.

1. Open the workspace that contains the files. See [Open a workspace](#).

2. On the navigation pane, click  **Workspace Files**. The **Workspace Files** list appears. For navigation information, see [List view](#).



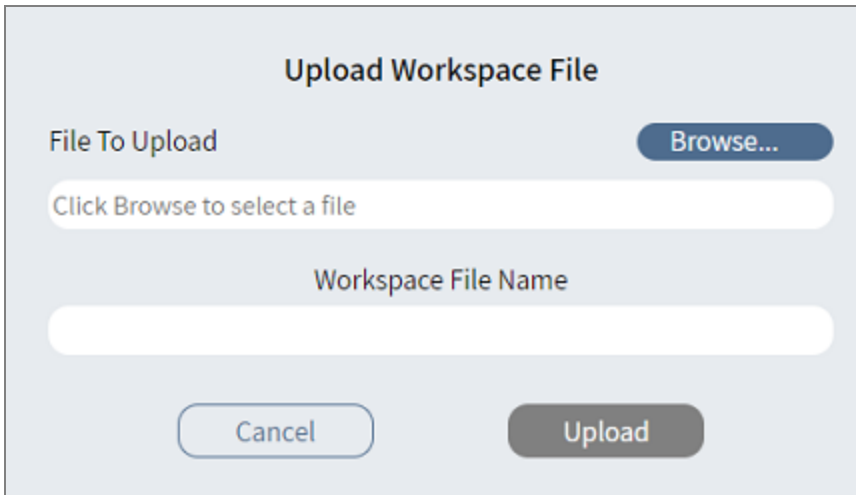
<input type="checkbox"/>	(0) Name ↓	Source file	Size	Last Modified Time	Last Modified By	Last Publish Time	
<input type="checkbox"/>	December	January.txt	0 KB	2023-12- 14T13:15:21.027+02:00	CCSP Admin	0001-01-01T00:00:00	   
<input type="checkbox"/>	February	February.txt	1 KB	2023-12- 14T13:17:32.223+02:00	CCSP Admin	0001-01-01T00:00:00	
<input type="checkbox"/>	January	January.txt	0 KB	2023-12- 14T12:54:32.977+02:00	CCSP Admin	0001-01-01T00:00:00	
<input type="checkbox"/>	Months	Months.xml	0 KB	2023-12- 14T12:55:29.713+02:00	CCSP Admin	0001-01-01T00:00:00	

The columns are:

- **Name**
- **Source file**
- **Size**
- **Last modified time** — this is currently the UTC time at which the file was saved in the database.
- **Last modified by**
- **Last publish time** — this is currently the UTC time at which the file was published in the tenant's IVR root path, configured in CCaaS Admin.

Upload a new file

1. On the **Workspace Files** page, click  **Add**. The **Upload Workspace File** dialog appears.

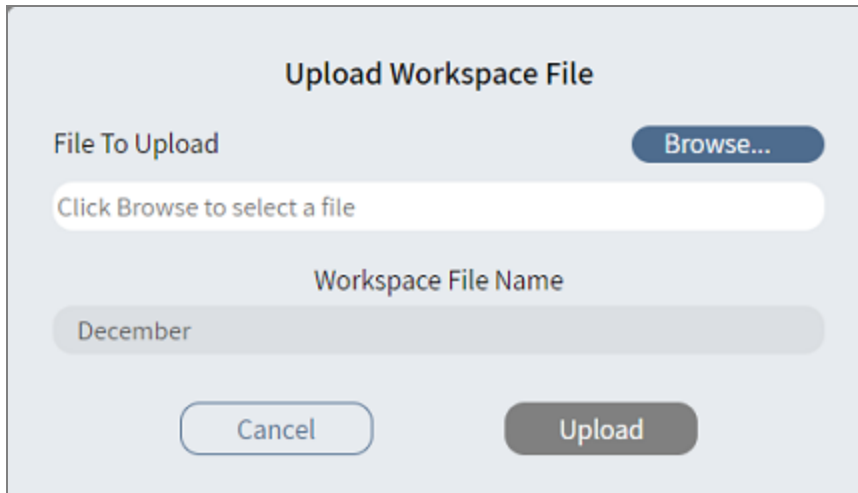


The screenshot shows a dialog box titled "Upload Workspace File". It contains two input fields: "File To Upload" with a "Browse..." button to its right, and "Workspace File Name". At the bottom, there are "Cancel" and "Upload" buttons.

2. Click **Browse** and select the file to upload.
3. Enter the workspace file name.
4. Click **Upload**.
5. The message *Workspace file successfully uploaded* appears. Click **OK**.

Upload an updated file


1. In the **Workspace Files** list, hover your mouse over the file to upload and click  **Upload**.

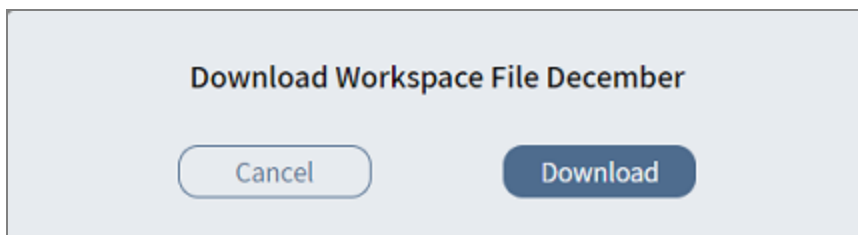


The screenshot shows a dialog box titled "Upload Workspace File". It contains two input fields: "File To Upload" with a "Browse..." button to its right, and "Workspace File Name" with the text "December" entered. At the bottom, there are two buttons: "Cancel" and "Upload".

2. Click **Browse** and select the text file to upload.
3. You cannot change the workspace file name.
4. Click **Upload**.
5. The message *Workspace file successfully uploaded* appears. Click **OK**.

Download a file

1. In the **Workspace Files** list, hover your mouse over the file to download and click  **Download**. The **Download Workspace File [name]** dialog appears.





The screenshot shows a dialog box titled "Download Workspace File December". It contains two buttons: "Cancel" and "Download".

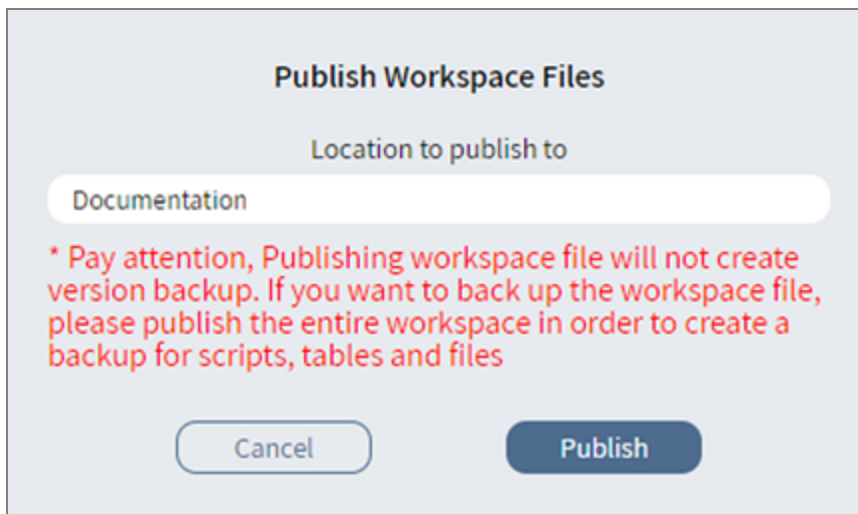
2. Click **Download**. The workspace file appears in your **Downloads** folder.

Publish a file

Note


- If an error message appears saying that the IVR root path cannot be found, or CCaaS Admin cannot be contacted, click  **Publish** again. If there is still a problem, see the configuration information in [Publish a workspace](#).
- Publishing an individual file does not create a backup version of it. You must publish a whole workspace to create a backup version of its scripts, tables, and files. For more information, see [Publish a workspace](#).

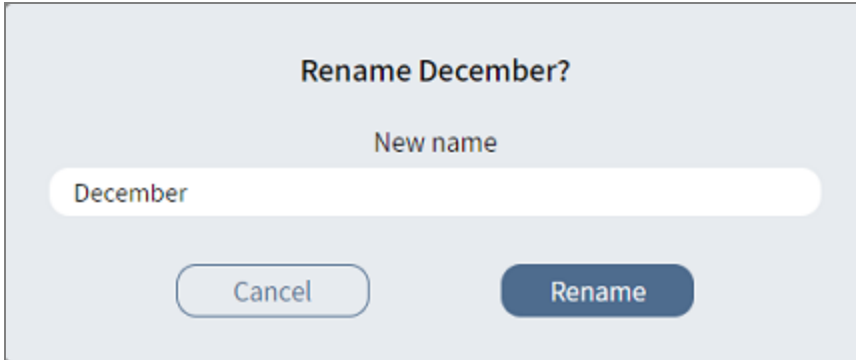
1. In the **Workspace Files** list, hover your mouse over the file to publish and click  **Publish**. The **Publish Workspace Files** dialog appears.



2. Enter the **Location to publish to**. You can use the default workspace location or change the name to publish to a different workspace folder.
3. Click **Publish**.
4. The message *Workspace file successfully published* appears. Click **OK**.

Rename a file


1. In the **Workspace Files** list, hover your mouse over the file to rename and click  **Rename**. The **Rename [name] ?** dialog appears.



The image shows a dialog box titled "Rename December?". It has a text input field labeled "New name" containing the word "December". Below the input field are two buttons: "Cancel" and "Rename".

2. Enter the new name.
3. Click **Rename**.

Delete a file

1. In the **Workspace Files** list, select the check box in front of the workspace file to delete.
2. Click  **Delete**.
3. Confirm the deletion.


Manage version history

[Open a snapshot](#)




[Edit a version's description](#)

[Republish a version](#)

Note

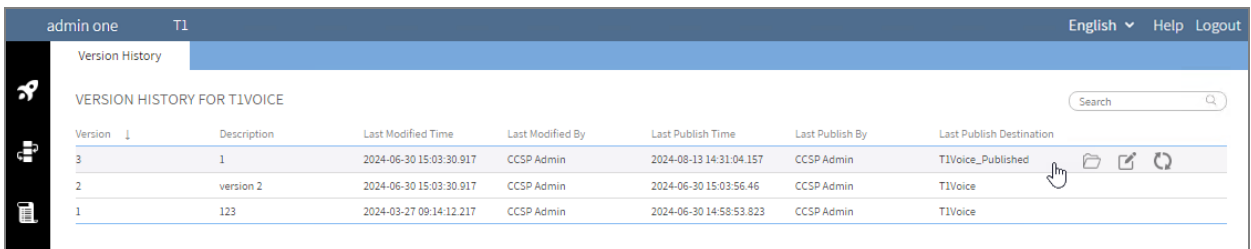
- CCaaS Designer automatically creates versions (snapshots) of each published/republished workspace. CCaaS Designer stores a maximum of 10 versions in the database and then replaces the oldest existing version with the new one.
- Publishing an individual script, table, or file does not create a backup version of it. You must publish a whole workspace to create a backup version of its scripts, tables, and files.
- The  **Version History** icon is only visible in the navigation pane after you open a workspace. You manage the versions for that workspace on the **Version History** page. If the workspace has not yet been published, the Version History list is empty.

1. Do one of the following:

- On the navigation pane, click  **Workspaces**. In the **Workspaces** list, hover your mouse over the owning workspace and click  **Version history**. See [Open a workspace](#).
- In the Workspace editor, in the workspace tab bar, select the owning workspace tab or snapshot tab. On the navigation pane, click  **Version History**.

The **Version History** list appears.

2. To see the action buttons for a particular version, hover your mouse over that row.




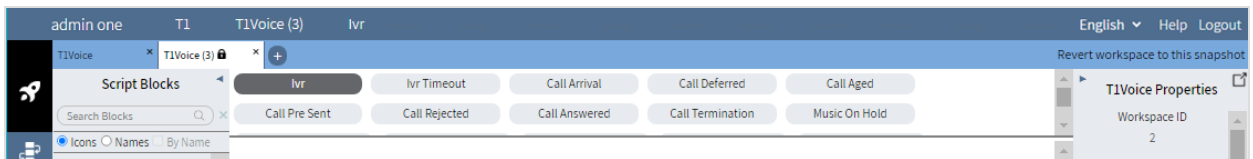
Version	Description	Last Modified Time	Last Modified By	Last Publish Time	Last Publish By	Last Publish Destination
3	1	2024-06-30 15:03:30.917	CCSP Admin	2024-08-13 14:31:04.157	CCSP Admin	T1Voice_Published
2	version 2	2024-06-30 15:03:30.917	CCSP Admin	2024-06-30 15:03:56.46	CCSP Admin	T1Voice
1	123	2024-03-27 09:14:12.217	CCSP Admin	2024-06-30 14:58:53.823	CCSP Admin	T1Voice

The columns are:

- **Version** — the version number of this snapshot.
 - **Description** — the description of this snapshot.
 - **Last modified time** — this is currently the UTC time at which the workspace was saved for this snapshot.
 - **Last modified by** — the name of the person who saved the workspace at this point.
 - **Last publish time** — this is currently the UTC time at which this version of the workspace was published in the tenant's IVR root path, configured in CCaaS Admin.
 - **Last published by** — the name of the person who published the workspace and created this snapshot.
 - **Last publish destination** — the name of the folder within the tenant's IVR root path in which the XML files for this version of the workspace were last saved.
3. To resort the list, click a column heading. Each time you sort by a column heading, you update the list of versions.


Open a snapshot

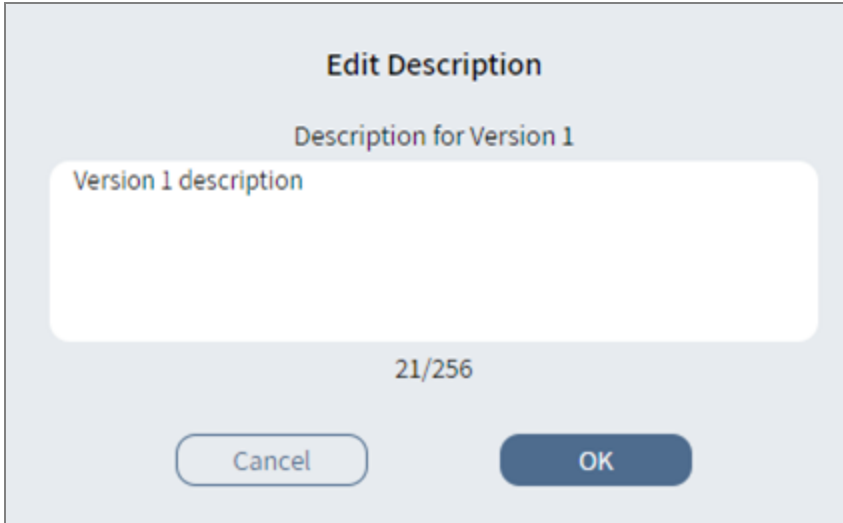
1. On the **Version History** page, hover your mouse over the version to open and click  **Open snapshot**.
2. The snapshot opens in the Workspace editor, with the version number of the workspace in brackets and a lock icon. See [Snapshot tabs](#) and [Revert a workspace to a snapshot](#).



Edit a version's description

You can edit the description you gave a workspace version when you published it.

1. On the **Version History** page, hover your mouse over the version to edit and click  **Edit description**. The **Edit Description** dialog appears.



Edit Description

Description for Version 1

Version 1 description

21/256


Cancel OK

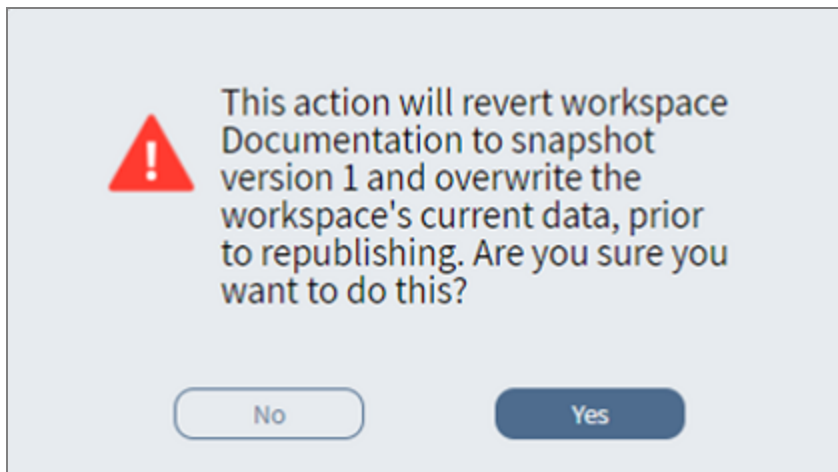
2. Edit the version description.
3. Click **OK**.

Republish a version

Note

- Republishing a version overwrites the workspace's current data with the data of the selected version.
- CCaaS Designer automatically creates a new version (snapshot) of the republished workspace.

1. On the **Version History** page, hover your mouse over the version to republish and click  **Republish**. A warning that *this action will revert workspace [parent name] to snapshot version [number] and overwrite the workspace's current data, prior to republishing* appears.



2. Click **Yes**. The **Publish Workspace [name]** dialog appears.

Publish Workspace Documentation

Location to publish to

Documentation

Description for Version Documentation

Version 1 description

21/256

* Pay attention, Publishing workspace creates up to 10 versions. When exceeding 10 versions, the oldest version will be deleted

3. Enter the **Location to publish to**.
4. Enter a **Description** for the republished version of the workspace.
5. Click **Publish**.
6. The message *Workspace successfully published* appears. Click **OK**.

4: Scripts

This section contains the following topics and subsection:

- [Manage scripts](#)
- [How to iterate through scripts](#)
- [Example scripts](#)

See also:

- [Script tabs panel](#) in the [Workspace editor overview](#).
- [Block availability per CallXML script](#)

Manage scripts

[Add a new script](#)

[Add a system script](#)

[Duplicate a script](#)

[Rename a custom script](#)


[Save a script](#)


[Publish a script](#)

[Validate a script](#)

[Delete a custom script](#)

Note

- The  **Scripts** icon is only visible in the navigation pane after you open a workspace. You view the scripts for that workspace on the **Scripts** page.
- You manage scripts through the [Workspace editor](#).
- Script names cannot contain the following characters: \ /:*?"<>|

1. Open the workspace that contains the scripts. See [Open a workspace](#).
2. On the navigation pane, click  **Scripts**. The **Scripts for [workspace]** list appears. For navigation information, see [List view](#).

admin one T1 English Help Logout

Scripts

SCRIPTS FOR HABB-VOICECOMMON 1-25 of 25 Search

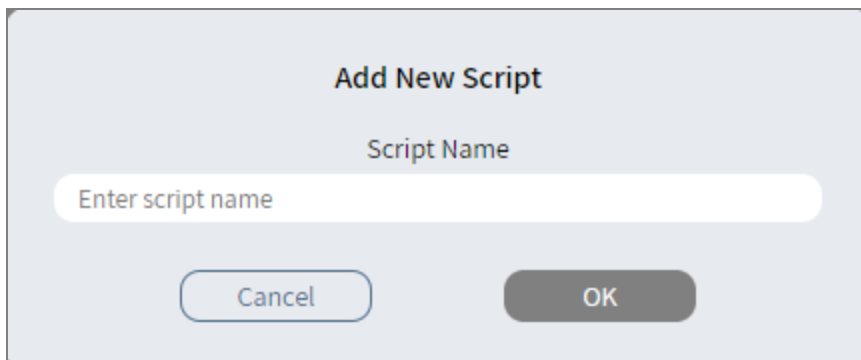
Name ↓	Published filename	Version	Last Modified Time	Last Modified By	Last Publish Time
A2A Call Rejected	A2ACallRejected.xml	1	2022-05-08T21:29:54.393	CCSP Admin	
Call Aged	CallAged.xml	1	2022-05-08T21:29:54.393	CCSP Admin	
Call Answered	CallAnswered.xml	1	2022-05-08T21:29:54.393	CCSP Admin	
Call Arrival	CallArrival.xml	1	2022-05-08T21:29:54.393	CCSP Admin	
Call Deferred	CallDeferred.xml	1	2022-05-08T21:29:54.393	CCSP Admin	
Call Pre Sent	CallPreSent.xml	1	2022-05-08T21:29:54.393	CCSP Admin	
Call Rejected	CallRejected.xml	1	2022-05-08T21:29:54.393	CCSP Admin	
Call Termination	CallTermination.xml	1	2022-05-08T21:29:54.393	CCSP Admin	
CB	CB.xml	1	2022-05-08T21:29:54.393	CCSP Admin	

The columns are:

- **Name**
- **Published filename**
- **Version** — the version of the script.
- **Last modified time** — this is currently the UTC time at which the script was saved in the database.
- **Last modified by**
- **Last publish time** — this is currently the UTC time at which the script was published in the tenant's IVR root path, configured in CCaaS Admin.

Add a new script

1. In the Workspace editor, click  and then click **Add Script**. The **Add New Script** dialog appears.

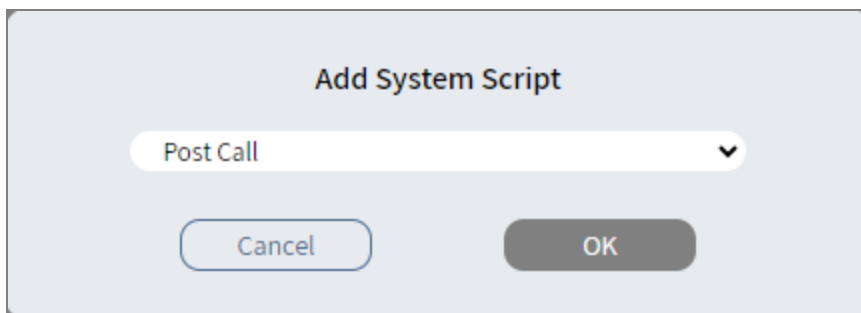


2. Enter the script name. You cannot use the name of a system script.
3. Click **OK**.

The script is added to the workspace and selected.

Add a system script


1. In the Workspace editor, click  and then click **Add System Script**. The **Add System Script** dialog appears.

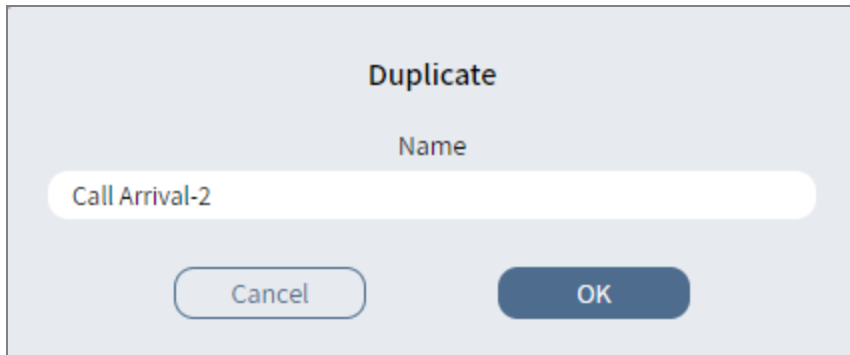


2. Select the script.
3. Click **OK**.

The script is added to the workspace and selected.

Duplicate a script

1. In the Workspace editor, in the Script tabs, select the script to duplicate.
2. Click  and then click **Duplicate**. The **Duplicate** dialog appears.




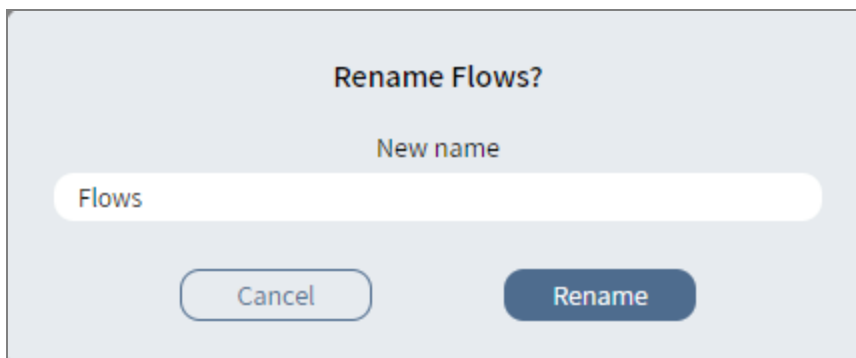
3. Enter the duplicate script name.
4. Click **OK**

The duplicate script is added to the workspace and selected.

Rename a custom script

You can rename only custom scripts.

1. In the Workspace editor, in the Scripts list, select the script to rename.
2. Click  and then click **Rename**. The **Rename [name]?** dialog appears.




3. Enter the new script name.
4. Click **Rename**.

The renamed script is added to the workspace and selected.

Save a script


When you edit a script, an asterisk appears beside its name and an asterisk appears beside the name of the workspace in the Workspace tab bar. When you save the script, both asterisks disappear.

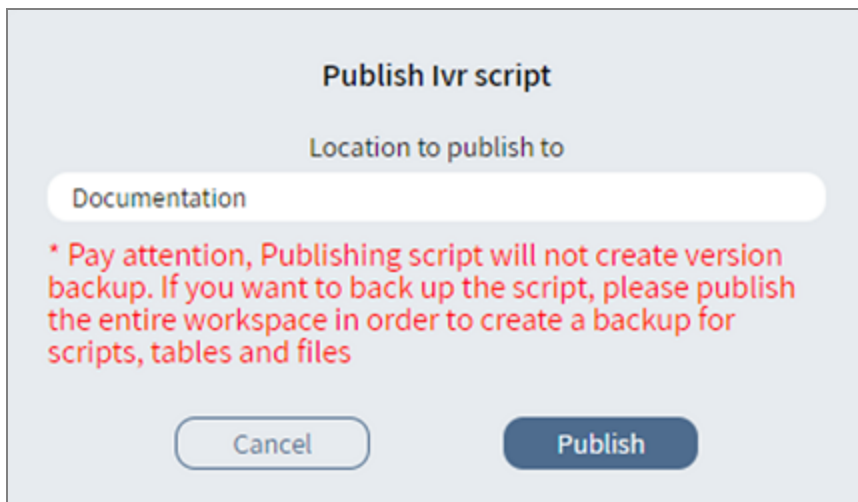
1. In the Workspace editor, in the Script tabs panel, select the script to save.
2. Click  and then click **Save**.
3. The message *Script successfully saved* appears. Click **OK**.

Publish a script

Note

- If an error message appears saying that the IVR root path cannot be found, or CCaaS Admin cannot be contacted, click **Publish** again. If there is still a problem, see the configuration information in [Publish a workspace](#).
- Publishing an individual script does not create a backup version of it. You must publish a whole workspace to create a backup version of its scripts, tables, and files. For more information, see [Publish a workspace](#).

1. In the Workspace editor, in the Script tabs panel, select the script to publish.
2. Click  and then click **Publish**. The **Publish [name] script** dialog appears.



Publish IvR script

Location to publish to

Documentation


* Pay attention, Publishing script will not create version backup. If you want to back up the script, please publish the entire workspace in order to create a backup for scripts, tables and files

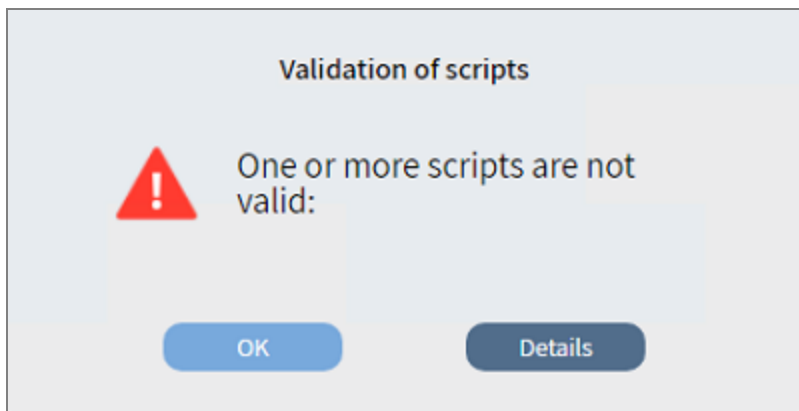
Cancel Publish

3. Enter the **Location to publish to**. You can use the default workspace location or change the name to publish to a different workspace folder.
4. Click **Publish**.
5. The message *Script successfully published* appears. Click **OK**.

Validate a script

CCaaS Designer provides script validation to check that each individual call event script contains only CCaaS Designer blocks that are valid for that particular call event. Script validation is performed automatically when you click **Save**.


1. In the Workspace editor, in the Script tabs panel, select the script to validate.
2. Click  and then click **Validate**.
3. If the script is valid, the message *Script is valid* appears.
4. If the script is not valid, the message *Script is not valid* appears. Click **Details** to view more information on the [Errors page](#).



5. Click **OK**.

Delete a custom script

You can delete only custom scripts.

1. In the Workspace editor, in the Script tabs panel, select the script to delete.
2. Click  and then click **Delete**.
3. Confirm the deletion.

How to iterate through scripts

To iterate through a script to set or extract variable information, where the variable data are in the form `dataItem1, dataItem2, dataItem3, ..., dataItemnn`, you need to use a workaround.

Part of the problem with iteration is that the `$` token, which is used to extract variable information, can only operate on a single parameter at one time, whereas the desired information would need to be nested in the iteration loop, in the form `$varName$counter$$`. Hardcoding the variable name is not a viable solution for this problem.

The following workaround takes advantage of the `ReadDictionary` function of XML.

In this example, data was obtained from a web server in the following format:

```
GetFrequentFlierInformation.Response.n:PersonName.n:LastActivity.1.n:ActivityText;value
=IBOO

GetFrequentFlierInformation.Response.n:PersonName.n:LastActivity.2.n:ActivityText;value
=AFR 1621 TLV C

GetFrequentFlierInformation.Response.n:PersonName.n:LastActivity.3.n:ActivityText;value
=AFR 007 JFK CDG C

GetFrequentFlierInformation.Response.n:PersonName.n:LastActivity.4.n:ActivityText;value
=AFR 006 CDG JFK C

GetFrequentFlierInformation.Response.n:PersonName.n:LastActivity.5.n:ActivityText;value
=AFR 1621 TLV CDG C

GetFrequentFlierInformation.Response.n:PersonName.n:LastActivity.n:ActivityText;value=J
HIUSDFR
```

For which the following code works:

```
For nCount = 1 to 6
    strN = $nCount$ & "."
    If nCount==6 then strN = ""
        LOG $ReadDictionary
("GetFrequentFlierInformation.Response.n:PersonName.n:LastActivity." & ReadDictionary
("strN") & "n:ActivityText")$
Next nCount
```

The following is an example of what could be used if you want to work with different bases:

```
BASE = "GetFrequentFlierInformation.Response.n:PersonName.n:LastActivity."
For nCount = 1 to 6
    strN = $nCount$ & "."
    If nCount==6 then strN = ""
        LOG $ReadDictionary(ReadDictionary("BASE") & ReadDictionary
("strN") & "n:ActivityText")$
Next nCount
```

Example scripts

In the example scripts:

- The workspace file contains all of the information about the CCaaS Designer workspace and its collection of XML scripts.
- The `..wavfiles` directory contains all of the recorded `.wav` files used in the scripts.
- The `DBLookup.asp` script looks up a customer ID in a database.

This subsection contains the following example scripts:

- [Ivr script](#)
- [Agentless Ivr script](#)
- [Ivr Timeout script](#)
- [Call Arrival script](#)
- [Call Deferred script](#)
- [Call Aged script](#)
- [Call Rejected script](#)
- [Call Answered script](#)
- [Call Termination script](#)
- [Call Pre Sent script](#)
- [Post Call script](#)
- [Elsbeth integration scripts](#)
- [Music On Hold script](#)
- [Music On Hold In Queue script](#)
- [In Queue Timeout script](#)
- [Pickup Timeout script](#)
- [Leave Voicemail script](#)
- [Retrieve Voicemail script](#)
- [Voicemail processing prompts](#)
- [Web services XML examples](#)

Ivr script

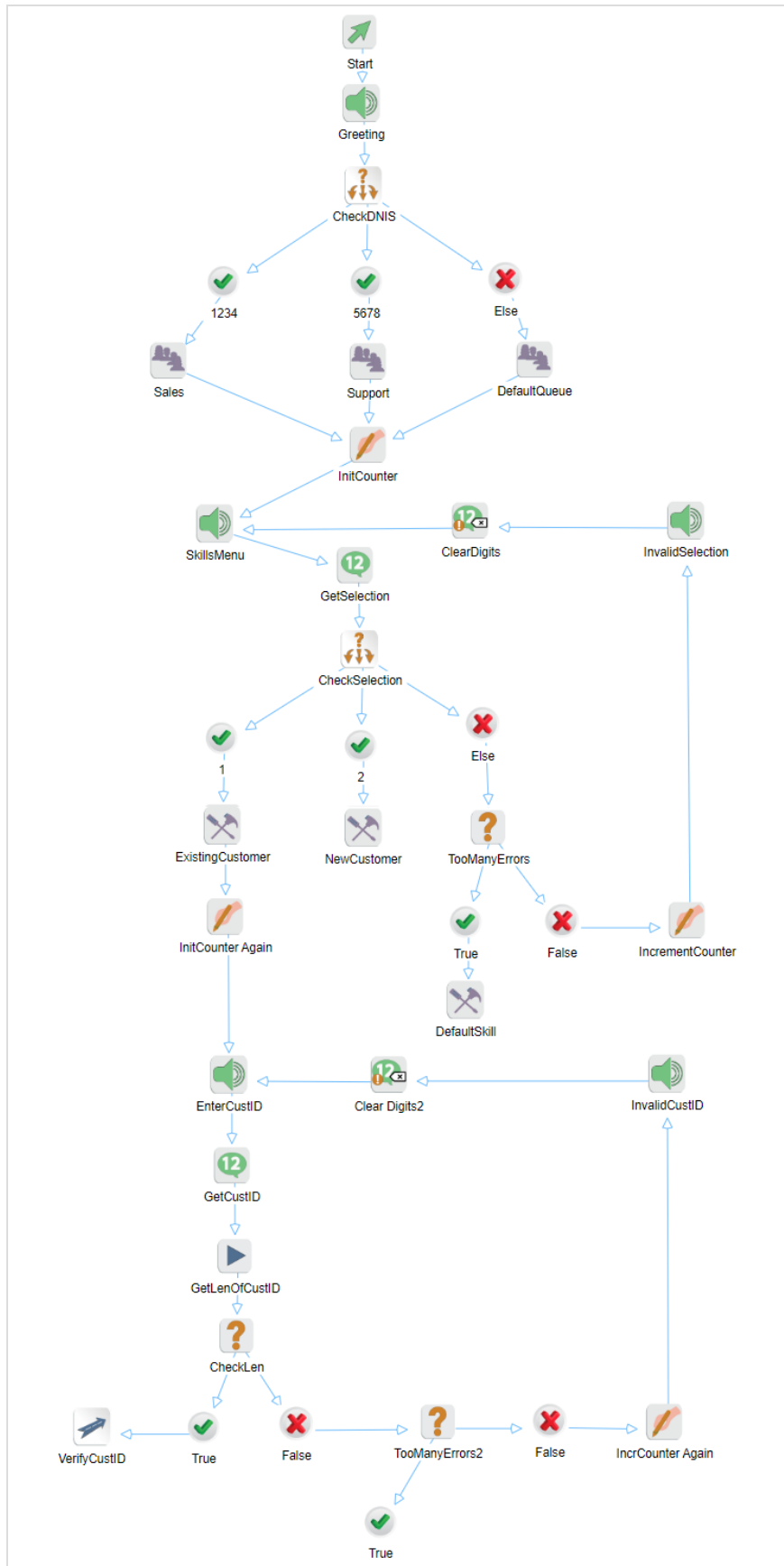
This script is interpreted when the **Incoming Call Event** is fired for an incoming VCS call.

It executes as follows:

1. Play a greeting announcement.
2. Assign a queue depending on the DNIS of the inbound call.
3. Offer a menu for the caller to select if they are a new or existing customer and sets the skill accordingly.
4. If the caller is an existing customer, it prompts them to enter their customer ID, validating it against a database.

Caution

Your IVR script should assign a queue for every voice call.



The blocks function as follows.

Block	Name	Description
Play Audio	Greeting	Plays the greeting announcement to the caller. It clears the DTMF buffer prior to playing. It also allows the caller to interrupt the announcement with any DTMF entry.
Switch	CheckDNIS	Checks the DNIS of the inbound caller to see if it is equal to either 1234 or 5678 or neither, and selects the appropriate path.
Queue Name	Sales	Sets the queue for the current call to Sales , if the DNIS is equal to 1234.
	Support	Sets the queue for the current call to Support , if the DNIS is equal to 5678.
	DefaultQueue	Sets the queue for the current call to DefaultQueue if the DNIS is anything else.
Assign	InitCounter	Initializes a counter called Counter to be used in keeping track of invalid entries made by the caller.
Play Audio	SkillsMenu	Plays the skills menu announcement. It allows the caller to interrupt the announcement with any DTMF entry.
Get Digits	GetSelection	Waits for a one digit DTMF entry from the caller and saves the entry in the variable called Selection . It waits up to 5 seconds for this digit before timing out.
Switch	CheckSelection	Checks the selection entered by the caller and selects the appropriate path.
Add Skill	ExistingCustomer	Adds the skill of ExistingCustomerSkill to the call if the caller selects 1 from the Skills Menu.
	NewCustomer	Adds the skill of NewCustomerSkill to the call if the caller selects 2 from the Skills menu.
Condition	TooManyErrors	If the caller selects anything else from the Skills menu, then this checks to see if they exceeded the maximum number of allowable invalid entries, which is three (3).
Add Skill	DefaultSkill	This assigns the DefaultSkill to the call if too many errors were made by the caller.
Assign	IncrementCounter	Increments the counter by one.
Play Audio	InvalidSelection	Plays the invalid selection announcement to the caller. It does not allow the caller to interrupt this announcement.

Block	Name	Description
Clear Digits	ClearDigits	If the caller enters any DTMF digits while the Invalid Selection announcement is being played, the caller would not be interrupted, but the VCS would store those DTMF digits in its buffer. So the buffer should be cleared out before continuing. Not doing this causes confusion to the caller.
Assign	InitCounterAgain	If this is an existing customer, the system is now going to prompt them to enter their customer ID. So initialize the counter again.
Play Audio	EnterCustID	Plays an announcement instructing the caller to enter their customer ID number.
Get Digits	GetCustID	Gets up to (4) four digits from the caller. Waiting for a maximum of ten (10) seconds for all digits to be entered or a maximum of three (3) seconds between digits. It stores the resulting entry in the variable CustID.
Run	GetLenOfCustID	Gets the length of the CustID entry and stores it in a variable called CustIDLen . It uses the VBScript function Len() to do this.
Condition	CheckLen	Checks to see if the number of digits entered by the caller is equal to four.
GoTo	VerifyCustID	Runs the <i>DBLookup.asp</i> script that looks up the entered Customer ID in the database to see if it is valid.
Condition	TooManyErrors2	If the customer ID is invalid, then this checks to see if they exceeded the maximum number of allowable invalid entries, which is three (3).
Assign	IncrCounterAgain	Increments the counter by one.
Play Audio	InvalidCustID	Plays the invalid customer ID announcement. It does not allow the caller to interrupt this announcement.
Clear Digits	ClearDigits2	If the caller enters any DTMF digits while the Invalid Selection announcement is being played, the caller would not be interrupted, but the VCS would store those DTMF digits in its buffer. So the buffer should be cleared out before continuing. Not doing this causes confusion to the caller.

Agentless IVR script

This script determines the call outcome.



The blocks function as follows.

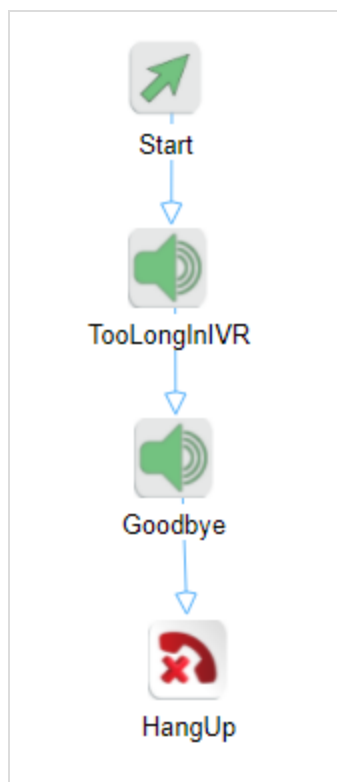
Block	Name	Description
Play Audio	PlayWelcome	Plays the greeting announcement to the caller. It clears the DTMF buffer prior to playing. It also allows the caller to interrupt the announcement with any DTMF entry.

Block	Name	Description
Menu	Menu	Caller hears a menu and is required to choose if he was satisfied and the call can be terminated (Else case) or he wants to continue to a CCaaS agent (press 1).
Agentless Campaign Outcome	MoveToAgent	Routes the call to a CCaaS agent, based on the existing routing properties.
	Terminate	Ends the call in IVR.

IVR Timeout script

This script is interpreted when the **IVR Timeout Event** is fired. It executes as follows:

1. Play an announcement indicating that the time allotted for an IVR call has expired.
2. Disconnect the caller.



The blocks function as follows.

Block	Name	Description
Play Audio	TooLongInIVR	Plays an announcement that explains that they have spent too much time in IVR.
Play Audio	Goodbye	Plays the goodbye announcement.
Hangup	HangUp	Disconnects the caller.

Call processing scripts

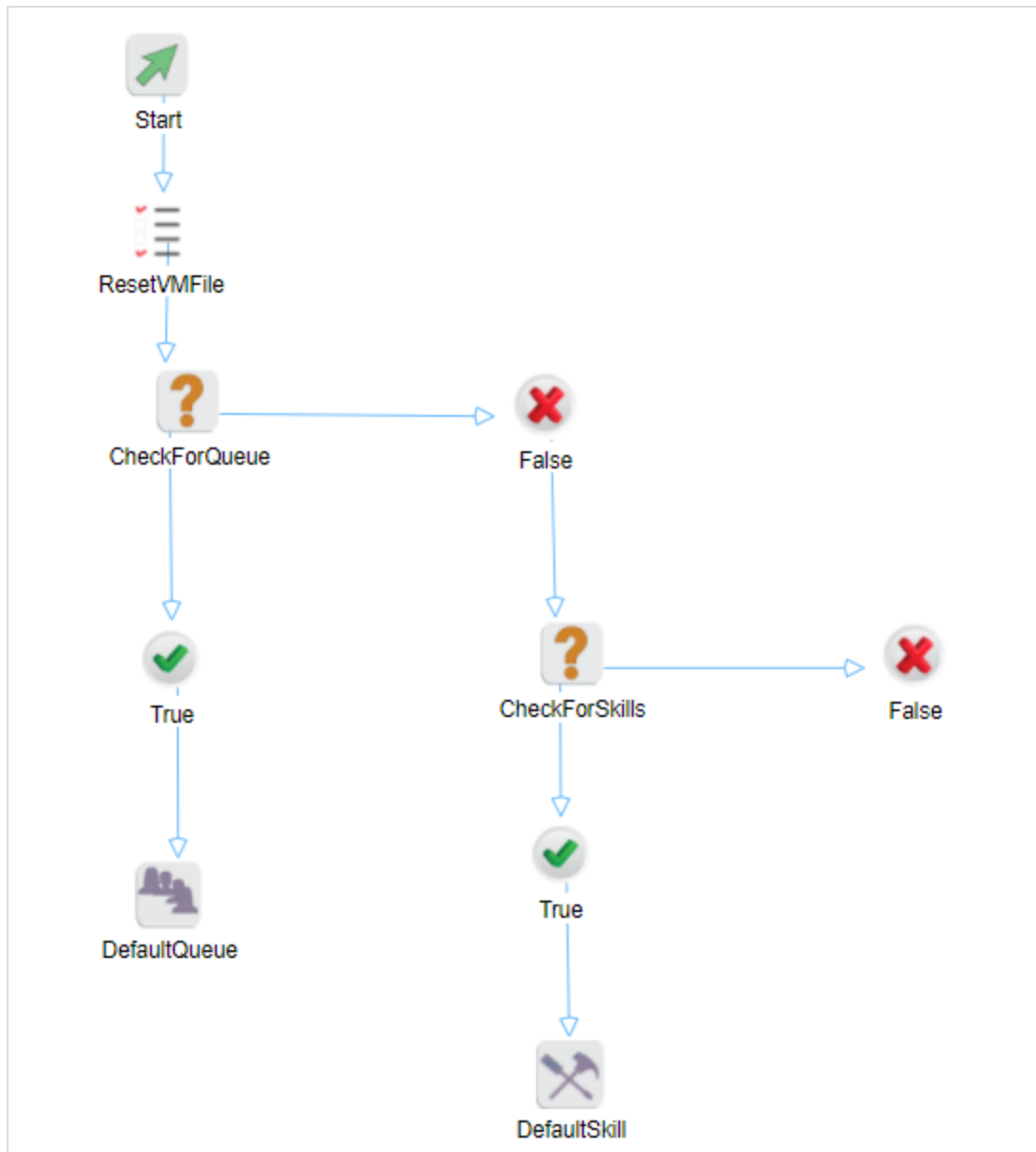
This subsection contains the following scripts:

- [Call Arrival script](#)
- [Call Deferred script](#)
- [Call Aged script](#)
- [Call Rejected script](#)
- [Call Answered script](#)
- [Call Termination script](#)
- [Call Pre Sent script](#)
- [Post Call script](#)

Call Arrival script

This script is interpreted when the **Call Arrival Event** is fired.

It executes as follows: if the current call has not been assigned a queue and/or at least one skill, then assign the DefaultQueue and/or DefaultSkill to it.



The blocks function as follows.

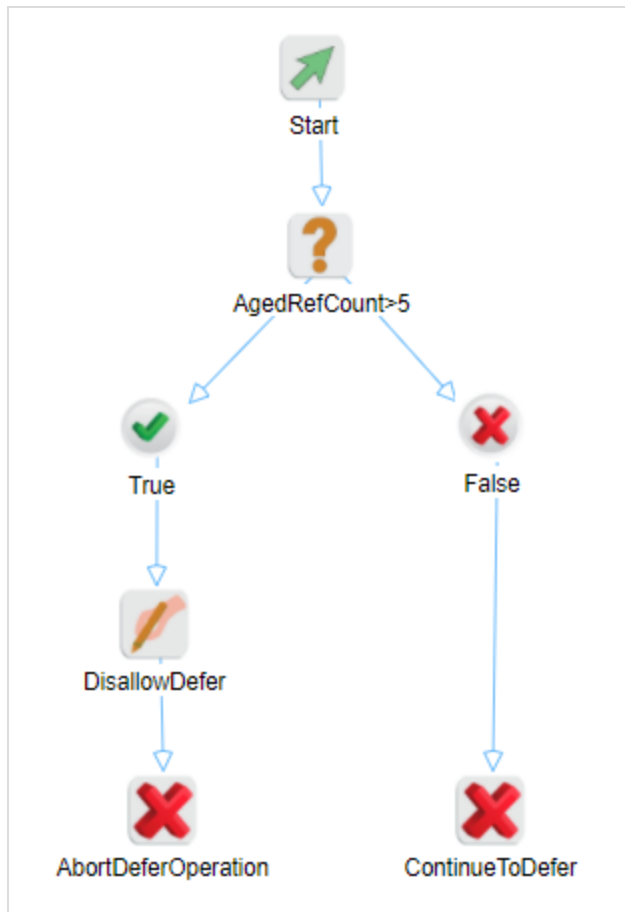
Block	Name	Description
Optional Para-	ResetVMFile	Resets the optional parameter called VMFile . This parameter is

Block	Name	Description
meters		used to store the name of the voicemail file when it is recorded. The call termination script uses this optional parameter to determine if the caller has recorded a voicemail.
Condition	CheckForQueue	Checks to see if a queue has been set for this call.
Queue Name	DefaultQueue	If no queue has been set yet, then set it to DefaultQueue .
Condition	CheckForSkills	Checks to see if a skill has been added to this call.
Add Skill	DefaultSkill	If no skill has been added yet, then add the DefaultSkill .

Call Deferred script

This script can only be run on email or voicemail messages.

This script is run while the agent is in call and before the completion of call termination. It defers the message.



The blocks in the above example function as follows.

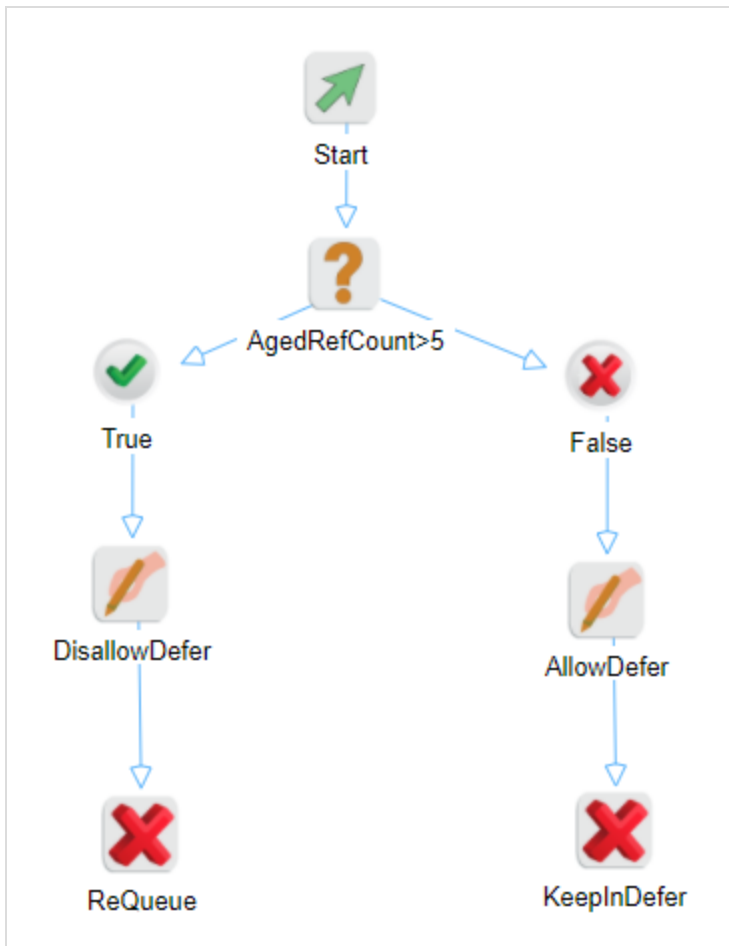
Block	Name	Description
Condition	AgedRefCount>5	Check if Ageing RefCount is bigger than 5. To check, set Expression = \$EmailCall.AgeingInfo.AgedRefCount\$ > 5 Each time the item is aged, the ref count increases by one.
Assign	DisallowDefer	If true, don't allow this item to be deferred again. To disallow defer, set \$EmailCall.AgeingInfo.IsDeferAvailable\$=False
Terminate	AbortDeferOption	If disallow defer, abort the defer operation and requeue the call. To requeue, set Terminate = No.
	ContinueToDefer	If false, put this item into the deferred folder. To continue to defer, set Terminate = Yes.

Call Aged script

This script can only be run on email or voicemail messages.

It is called when an email or voicemail has been in the deferred folder for longer than the configured Ageing Timeout Interval. Ageing starts from when the agent defers the call, not from call arrival.

By default, an agent cannot defer a call after it has aged once. You can use this script to override the default as follows.



The blocks in the above example function as follows.

Block	Name	Description
Condition	AgedRefCount > 5	Check if Ageing RefCount is bigger than 5. To check, set Expression = \$EmailCall.AgeingInfo.AgedRefCount\$ > 5 Each time the item is aged, the ref count increases by one.
Assign	DisallowDefer	If true, don't allow this item to be deferred again. To disallow defer, set \$EmailCall.AgeingInfo.IsDeferAvailable\$=False

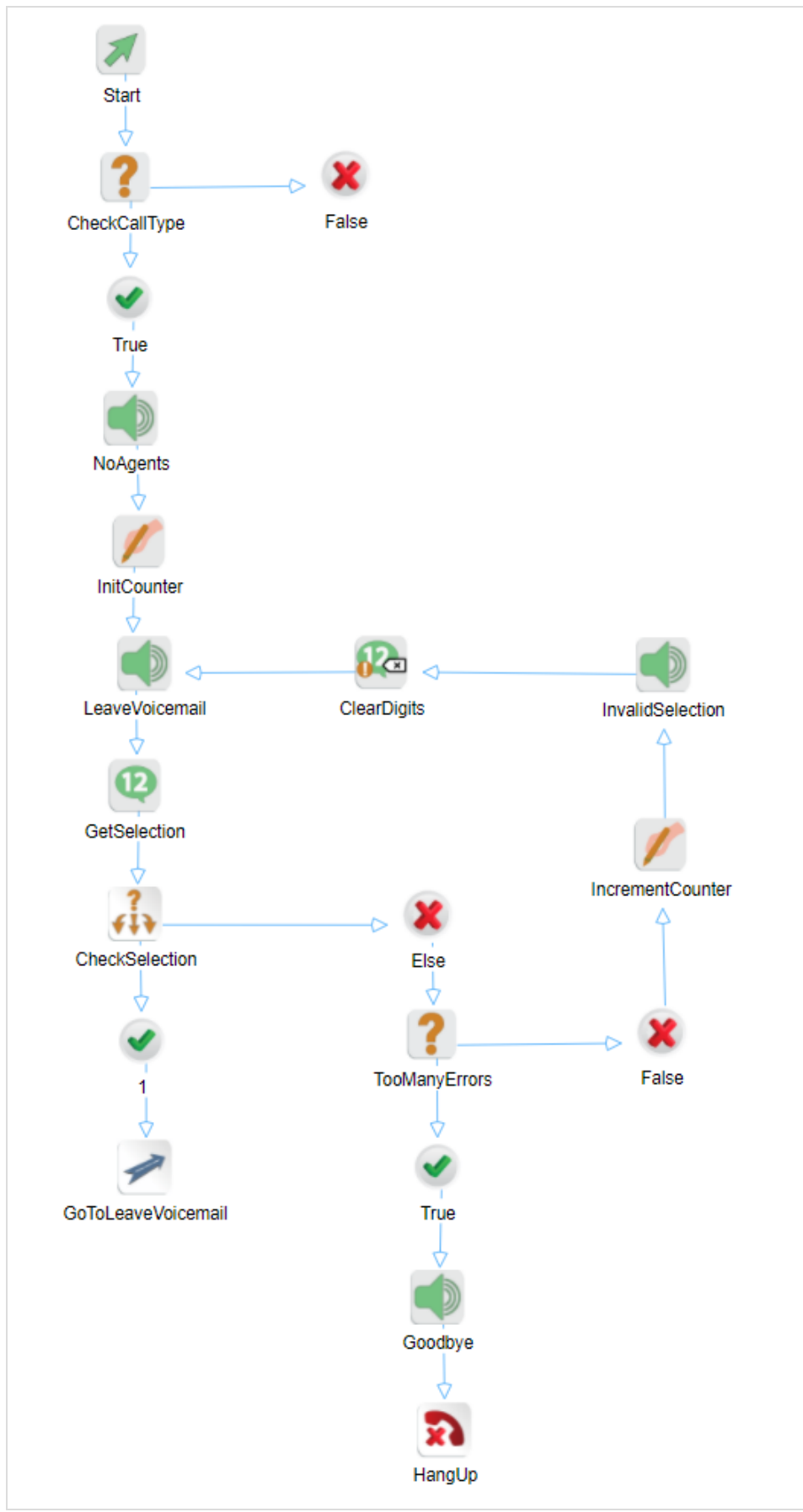
Block	Name	Description
	AllowDefer	If false, allow this item to be deferred again. To allow defer, set \$EmailCall.AgeingInfo.IsDeferAvailable\$=True
Terminate	ReQueue	If disallow defer, continue the process and requeue the call. To requeue, set Terminate = No.
	KeepInDefer	If allow defer, keep this item in the deferred folder. To keep in the deferred folder (terminating the call cancels the ageing operation), set Terminate = Yes.

Call Rejected script

This script is interpreted when the **Call Rejected Event** is fired.

It executes as follows:

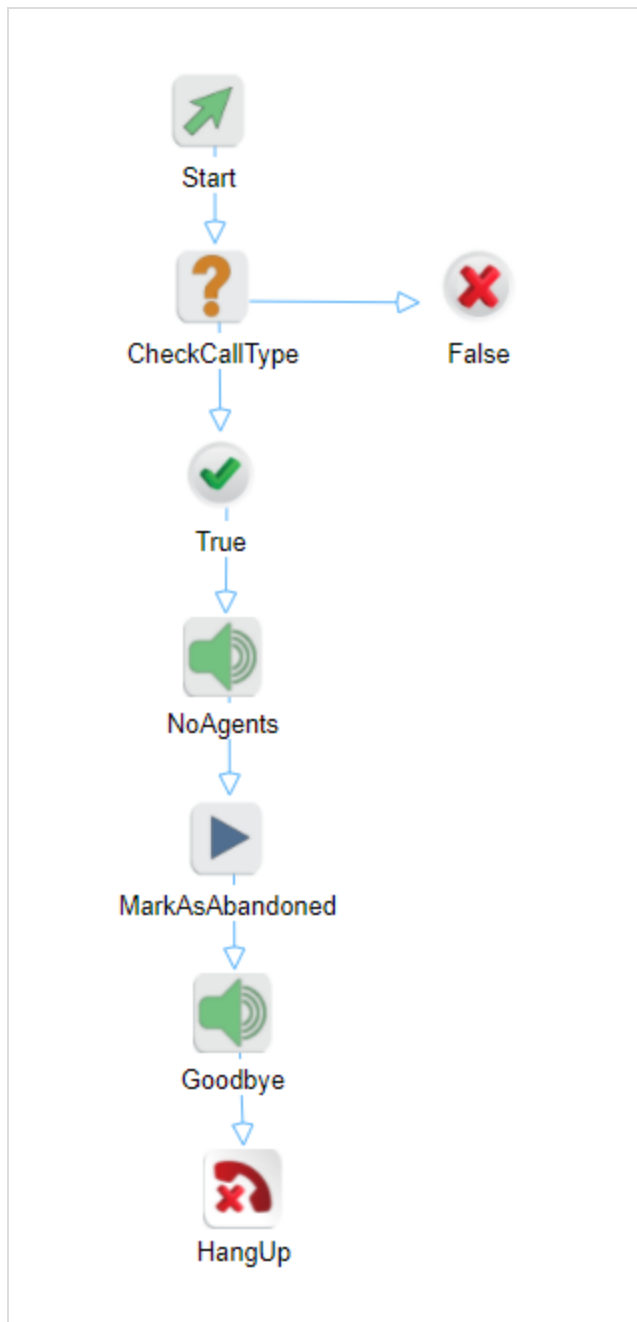
1. Play an announcement that no agents are available.
2. Offer a menu asking the caller if they want to leave a voicemail.
3. If caller selects to leave a message, record the message.



The blocks function as follows.

Block	Name	Description
Condition	CheckCallType	Checks the call type of the current call. Since all this script does is to leave a voicemail, it only makes sense to do this for VCS calls. The CCLEvent.CallType property is set to 0 for chat calls, 2 for email calls and 3 for VCS calls.
Play Audio	NoAgents	Plays the no agents announcement. It clears the DTMF buffer before playing and allows the caller to enter any DTMF digit to interrupt the announcement.
Assign	InitCounter	Initializes a counter called Counter to one (1).
Play Audio	LeaveVoicemail	Plays the <i>Do you want to leave a voicemail ...</i> announcement. It allows the caller to interrupt the announcement with any DTMF entry.
Check Digits	GetSelection	Gets a one digit DTMF entry from the caller and saves the entry in the variable called Selection . It waits up to five (5) seconds for this digit before timing out.
Switch	CheckSelection	Checks the selection entered by the caller and selects the appropriate path.
GoTo	GoToLeaveVoicemail	If the caller enters a 1 , then jump to the <i>LeaveVoiceMail.xml</i> script.
Condition	TooManyErrors	If the caller selects anything else from the menu, this checks to see if they exceeded the maximum number of allowable invalid entries, which is three (3).
Assign	IncrementCounter	Increments the counter by one.
Play Audio	InvalidSelection	Plays the invalid selection announcement to the caller. It does not allow the caller to interrupt this announcement.
Clear Digits	ClearDigits	If the caller enters any DTMF digits while the Invalid Selection announcement is being played, the caller is not interrupted, but the VCS would store those DTMF digits in its buffer. So the buffer should be cleared out before continuing. Not doing this causes confusion to the caller.

Alternatively, you can configure the rejected script to set every abandoned VoIP call as abandoned in queue.



The blocks function as follows.

Block	Name	Description
Condition	CheckCallType	Checks the call type of the current call.
Play Audio	NoAgents	Plays the no agents announcement.
Run	MarkAsAbandoned	Executes cosmocall.MarkAsAbandoned(true), which sets

Block	Name	Description
		<p>abandoned calls in the rejected script to be reported as abandoned.</p> <p>Note</p> <p>If <code>cosmocal.MarkAsAbandoned(false)</code>, or the Run block is missing, sets abandoned calls in the rejected script to be reported as rejected.</p>

Call Answered script

This script is interpreted when the **Call Answered Event** is fired and allows system handling of an arriving call when an agent answers the call. The default script is empty.

Note

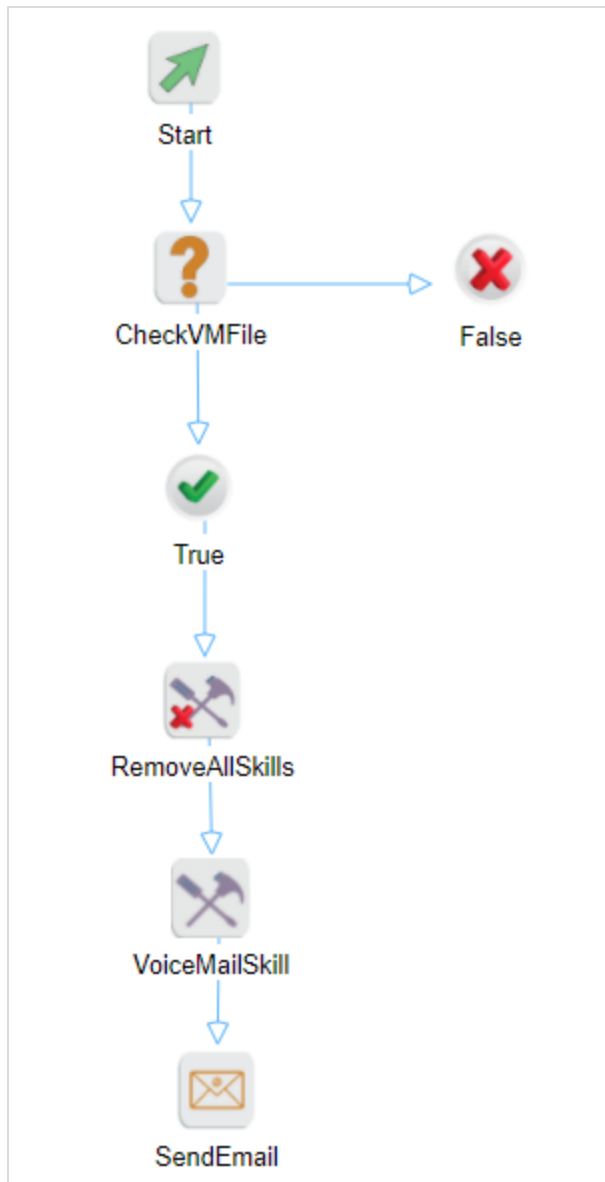
- You cannot use the [Recorder](#) block in this script.
- *CallAnswered.xml* is not supported in preview campaigns.

Call Termination script

This script is interpreted when the **Call Termination Event** is fired.

It executes as follows:

1. Check to see if a voicemail has been recorded.
2. If one has, then strip all skills from the call and assign the **VoiceMailSkill**.
3. Send the voicemail using email.



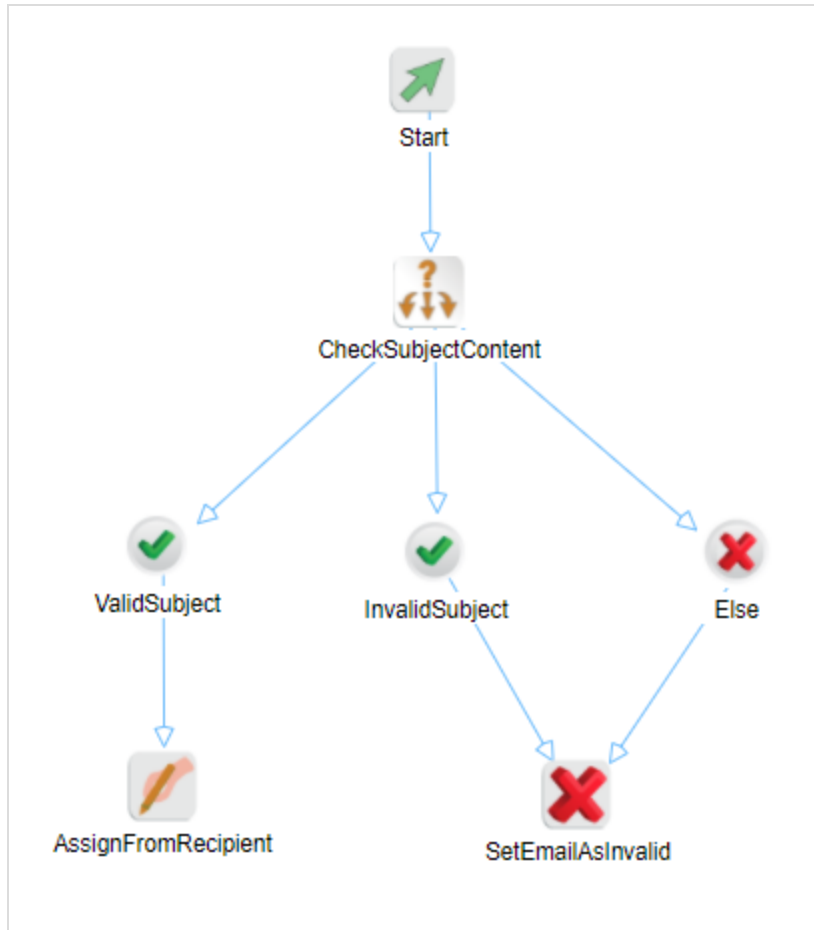
The blocks function as follows.

Block	Name	Description
Condition	CheckVMFile	Checks to see if the optional parameter VMFile is set to anything but NULL (""). If it is not NULL, then the caller has recorded a voicemail that must be sent via email before terminating.
Remove Skill	RemoveAllSkills	Removes all skills that have been assigned to this call so far.
Add Skill	VoicemailSkill	Adds the VoiceMailSkill to the call.
Send Email	SendEmail	Sends the voicemail as an attachment to an email.

Call Pre Sent script

This script can only be run on an outgoing email.

It is called after an agent initiates, replies to, or forwards an email by clicking Send but before the email is actually sent from the system to the recipient.



The blocks function as follows.

Block	Name	Description
Switch	CheckSubjectContent	Switch expression: \$outgoingEmailcall.outgoingEmailData.Subject\$ <ul style="list-style-type: none">• Subject is valid: use the Assign block to set the From recipient value• Subject is invalid: use the Terminate block to suppress the email

Block	Name	Description
Assign	AssignFromRecipient	Set the From recipient to: my@enghouse.com
Terminate	SetEmailAsInvalid	Yes: the call will be terminated and the item will be moved to the Sent Items folder with the Invalid status. No: (Default) continue the send flow

Post Call script

This script is interpreted when the caller is still connected after all the other participants of the call hang up. It is triggered to post process the call, for example, to give a customer the chance to rate or answer a survey about the call.

For an incoming call, this script is triggered if the call was answered, and the customer is still connected after the other participants hang up.

This also applies:

- If the call was transferred to another agent and the customer is still connected after the transferred agent hangs up.
- If the call was transferred to an external party (for example, a manager) and the customer (first external party) is still connected after the second external party hangs up.
- If the call was conferenced and the customer (first external party) is still connected after all other participants of the call hang up.

For an outgoing call, this script is triggered if the customer is still connected after the other participants hang up. The call is always answered because it is initiated by the agent.

For the list of blocks that can be used in this script, see [Block availability per CallXML script](#).

Caution

After an unsuccessful transfer by agent or another network issue, a caller could end up in the Rejected script instead of the Post Call script.

To handle this, in the Call Rejected script add a Condition block with the expression:

`calldataAnsweringAgentId <> 0` and a GoTo block that redirects the caller to the Post Call script.

Note

After importing a workspace with existing VOIP scripts, you must manually add the Post Call script:

1. [Add a new script](#) named Post Call.
2. Save the script. This process converts the new script to a regular Post Call system script.
3. Add the permitted blocks and publish the workspace.

Elsbeth integration scripts

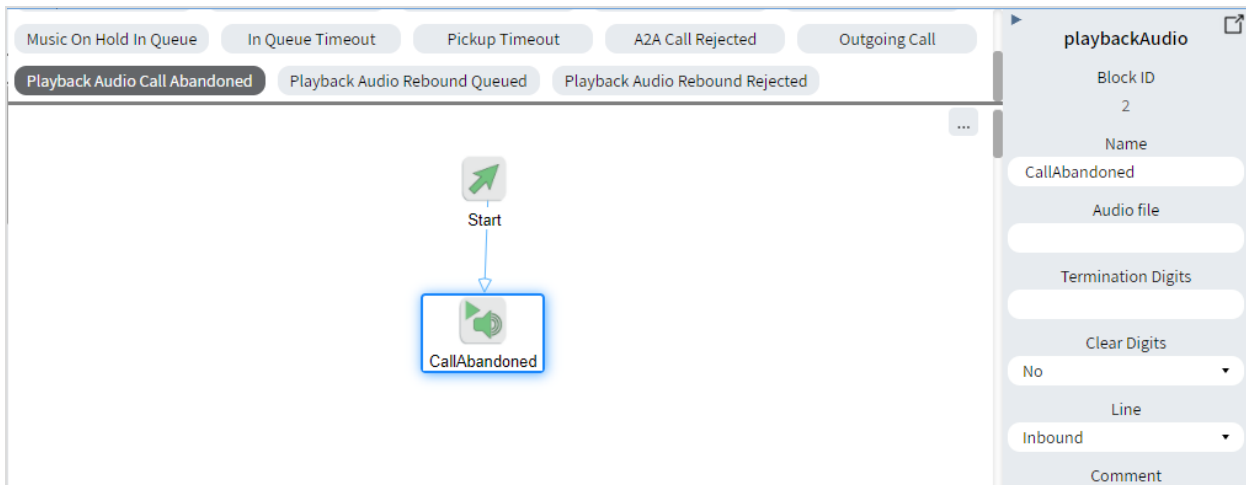
As part of the integration between the CCaaS and Elsbeth products, an audio file needs to be played back to a customer:

- At the end of an abandoned call — PlaybackAudioCallAbandoned script
- After a rebound call is queued — PlaybackAudioReboundQueued script
- If a rebound call is rejected — PlaybackAudioReboundRejected script

These scripts are system-initiated.

As a prerequisite for playback audio to work, you must do the following:

1. Select the appropriate tab, depending on your Playback Audio scenario.
2. Add a new [Playback Audio](#) block. Leave the Audio file property blank. CCaaS takes the URL of the file to play from the campaign parameters configured in CCaaS Admin.
3. Connect it with the start block and save the script.



Playback Audio Call Abandoned script

This script is system-initiated and plays back audio to a customer at the end of an abandoned call.

The audio file in the **Filename** property must match the URL configured in the **Audio File URL In Call Abandoned** parameter in the CCaaS Admin Campaign properties.

Playback Audio Rebound Queued script

This script is system-initiated and plays back audio to a customer after a rebound call is queued.

The audio file in the **Filename** property must match the URL configured in the **Audio File URL Rebound in queue timeout reject** parameter in the CCaaS Admin Campaign properties.

Playback Audio Rebound Rejected script

This script is system-initiated and plays back audio to a customer when a rebound call is rejected.

The audio file in the **Filename** property must match the URL configured in the **Audio File URL Rebound no queueing reject** parameter in the CCaaS Admin Campaign properties.

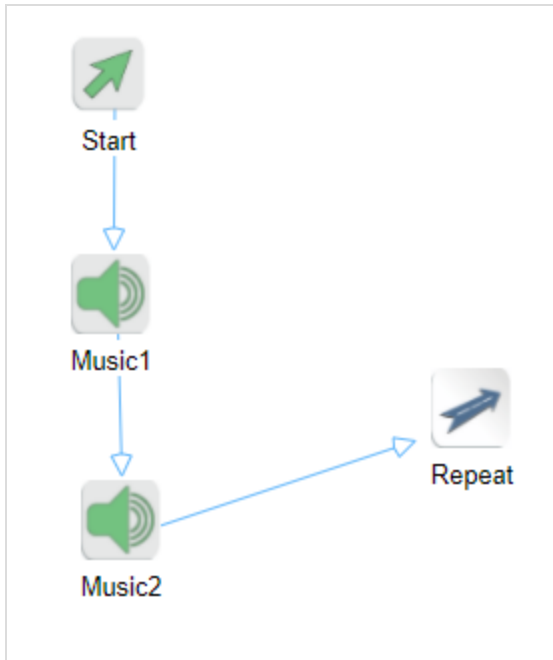
Music On Hold script

This script is interpreted when the **Music On Hold Event** is fired.

It executes as follows: Play selected music (.wav file).

Note

To have the music loop, you must program this behavior within the script.



The blocks function as follows.

Block	Name	Description
Play Audio	Music1	Plays a music announcement. It does not allow the caller to interrupt the announcement.
Play Audio	Music2	Plays another music announcement. It does not allow the caller to interrupt the announcement.
GoTo	Repeat	Set the GoTo property to Music1 to loop around again.

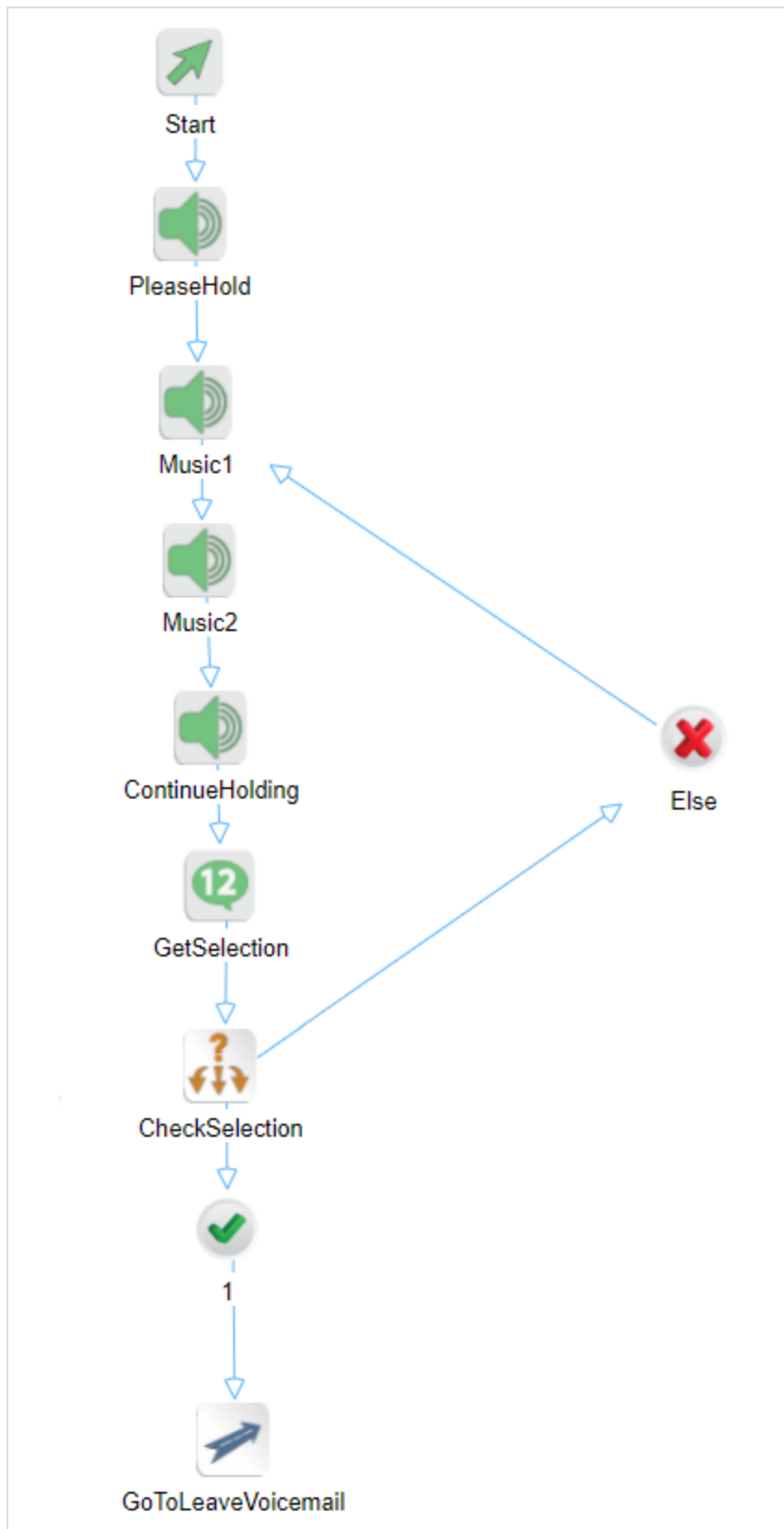
Music On Hold In Queue script

This script is interpreted when the **Music On Hold In Queue Event** is fired. It executes as follows:

1. Play please hold announcement
2. Play selected music (.wav file).
3. Offer a menu asking the caller if they want to leave a voicemail or continue holding.
4. If caller selects to leave a voicemail, record the voicemail, otherwise loop back and start playing music again.

Note

To have the music loop, you must program this behavior within the script.



The blocks function as follows.

Block	Name	Description
Play Audio	PleaseHold	Plays the please hold announcement.
Play Audio	Music1	Plays a music announcement. It does not allow the caller to interrupt the announcement.
Play Audio	Music2	Plays another music announcement. It does not allow the caller to interrupt the announcement.
Play Audio	ContinueHolding	Plays an announcement asking the caller if they would like to continue holding or leave a voicemail.
Get Digits	GetSelection	Gets a one digit DTMF entry from the caller and saves the entry in the Selection variable. It waits up to five (5) seconds for this digit before timing out.
Switch	CheckSelection	Checks the selection entered by the caller and selects the appropriate path.
GoTo	GoToLeaveVoicemail	If the caller enters 1 , then jump to the <i>LeaveVoicemail.xml</i> script.

In Queue Timeout script

This script is interpreted when the **In Queue Timeout Event** is fired.

By default, this script is empty. For a list of the blocks you can add to this script, see [Block availability per CallXML script](#).

Pickup Timeout script

This script is interpreted when the **Pickup Timeout Event** is fired.

By default, this script is empty. For a list of the blocks you can add to this script, see [Block availability per CallXML script](#).

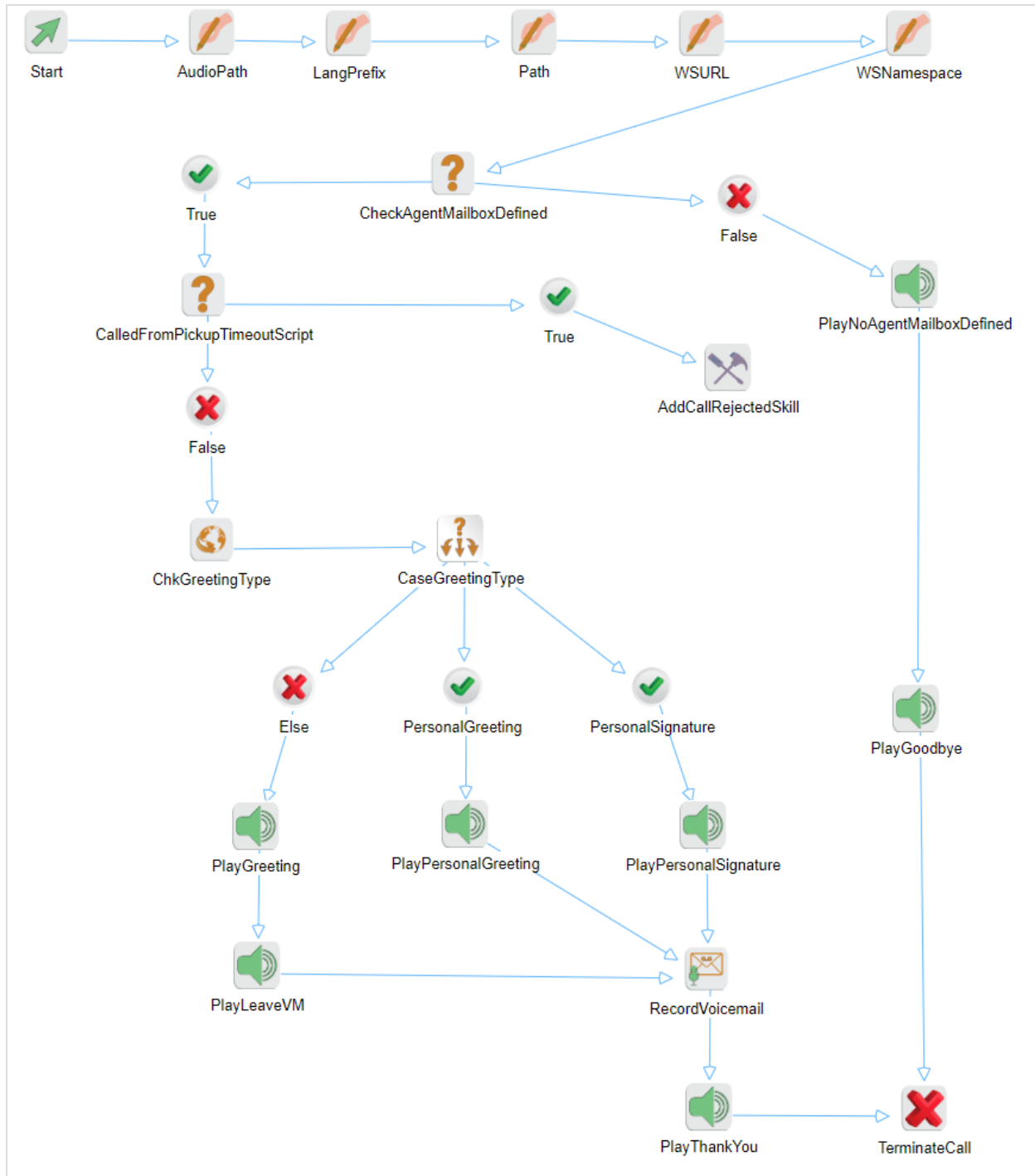
Voicemail processing scripts

This subsection contains the following scripts and topic:

- [Leave Voicemail script](#)
- [Retrieve Voicemail script](#)
- [Voicemail processing prompts](#)

Leave Voicemail script

This script guides a caller into leaving a voicemail message for a called party.



Caution

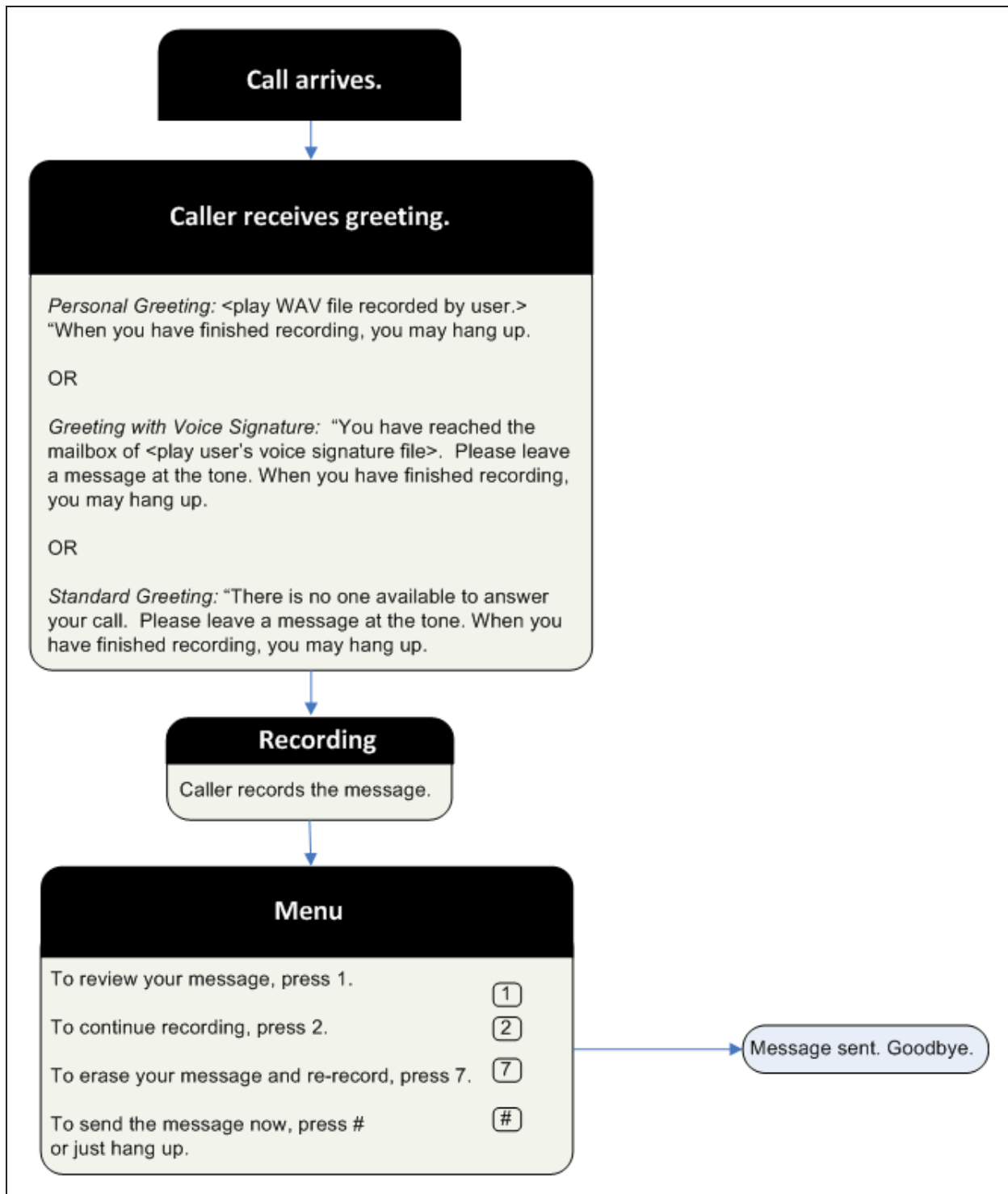
We recommend that you do not modify this script.

The blocks function as follows.

Block	Name	Description
Assign	AudioPath	Assign variable AudioPath, value=../Voicemail/. This is an audio file path local to the VCS system. Default location is <i>C:/Program Files/Cosmocom/Server Components/Scripts/AllWaves/Voicemail</i> .
	LangPrefix	Assign variable =LangPrefix. The language prefix used before playing any audio file. If no prefix is specified, the language is English.
	Path	Assign variable =\${AudioPath}\${LangPrefix}. The complete path to the audio file, including any language prefix specified.
	WSURL	Assign variable =WSURL, value=\${session.VoicemailWSURL}VoicemailWebService.aspx. This is the Voicemail web service URL.
	WSNamespace	Assign variable =WSNamespace, value=http://CosmoCom.com/WebServices/VoicemailWS. This is the voicemail web service namespace.
Condition	ChkAgentMailboxDefined	Conditional expression=\${calldata.DIDAgentVoiceMailEnabled}. Check that the user mailbox is enabled. If the user does not have a voicemail box, the user cannot receive voicemail. If TRUE, go to CalledFromPickupTimeoutScript. If FALSE, go to PlayNoAgentMailBoxDefined.
	CalledFromPickupTimeoutScript	Conditional expression=\${CallData.Option(ScriptID)}. If called from pickup timeout script, call should be rejected. Therefore, stop executing any logic in the leave voicemail script. If TRUE, go to AddCallRejectedSkill. If FALSE, go to ChkGreetingType.
Web Services	ChkGreetingType	Get the type of greeting to play.
Switch	CaseGreetingType	Check the type of greeting to play. Valid greeting types are: <ul style="list-style-type: none"> StandardGreeting — greeting designed for all users in an organization. PersonalGreeting — greeting recorded by a user for their own use. PersonalSignature Greeting — greeting designed for all

Block	Name	Description
		<p>users in an organization but with the insertion of a user's voice signature.</p> <p>Else branch: Go to PlayGreeting to play the standard greeting.</p> <p>PersonalGreeting branch: Go to PlayPersonalGreeting.</p> <p>PersonalSignaturebranch: Go to PlayPersonalSignature.</p>
Play Audio	PlayGreeting	Play standard greeting.
	PlayPersonalGreeting	Play personal greeting.
	PlayPersonalSignature	Play personal signature greeting.
	PlayLeaveVM	Play Please leave voicemail message.
Record Voicemail	RecordVoicemail	Record caller's voicemail message
Add Skill	AddCallRejectedSkill	Reject the call. The CallRejectedSkill skill must exist in your tenant.
Play Audio	PlayNoAgentMailBoxDefined	Play the agent mailbox not defined message.
	PlayThankYou	Play thank you and goodbye message.
	PlayGoodbye	Play the good bye announcement.
Terminate	TerminateCall	Terminate the call

This script implements the following caller message flow.



For the voice prompts used in this script, see [Voicemail processing prompts](#).

Retrieve Voicemail script

Note

Do not confuse this message retrieval with agents retrieving ACD voicemail.

Caution

We recommend that you do not modify this script.

This script helps users retrieve personal voicemail, replay the messages, and manage their voice mailboxes by storing or deleting messages.

This scripts also allows users to:

- Create or change a Personal Identification Number (PIN).
- Create a personal greeting or add their voicemail signature to a standardized greeting. Alternatively, the system can play a standard greeting.

This script plays the time the message was received. The tenant's time zone is encoded in the script.

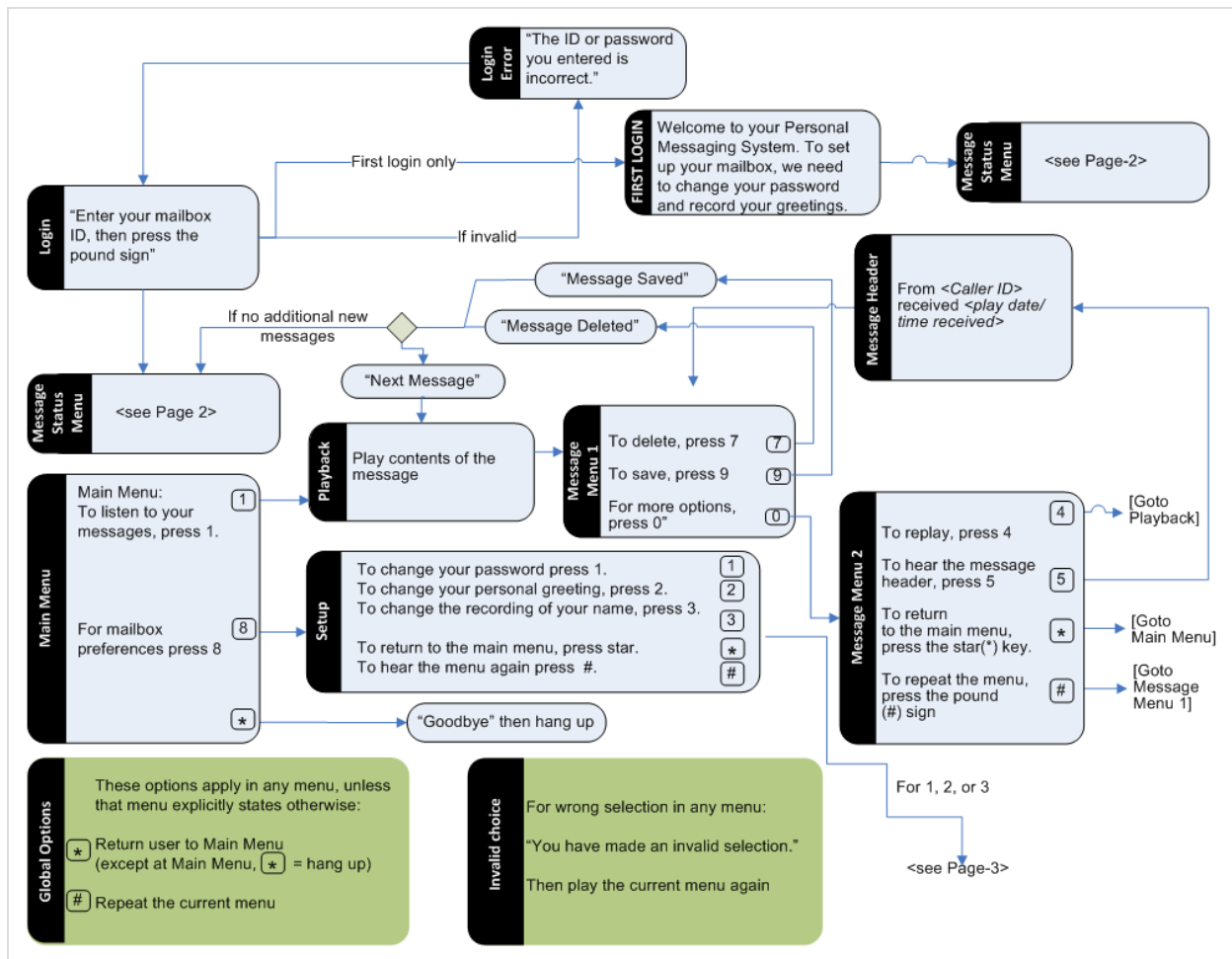
Note

Ask your sales engineer to configure your time zone in this script.

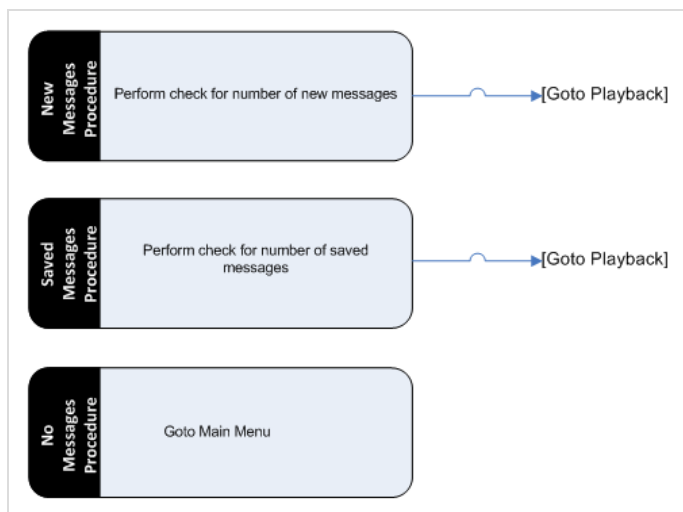
It requires the user to have DID and a voice mailbox.

This script implements the following user message retrieval flow.

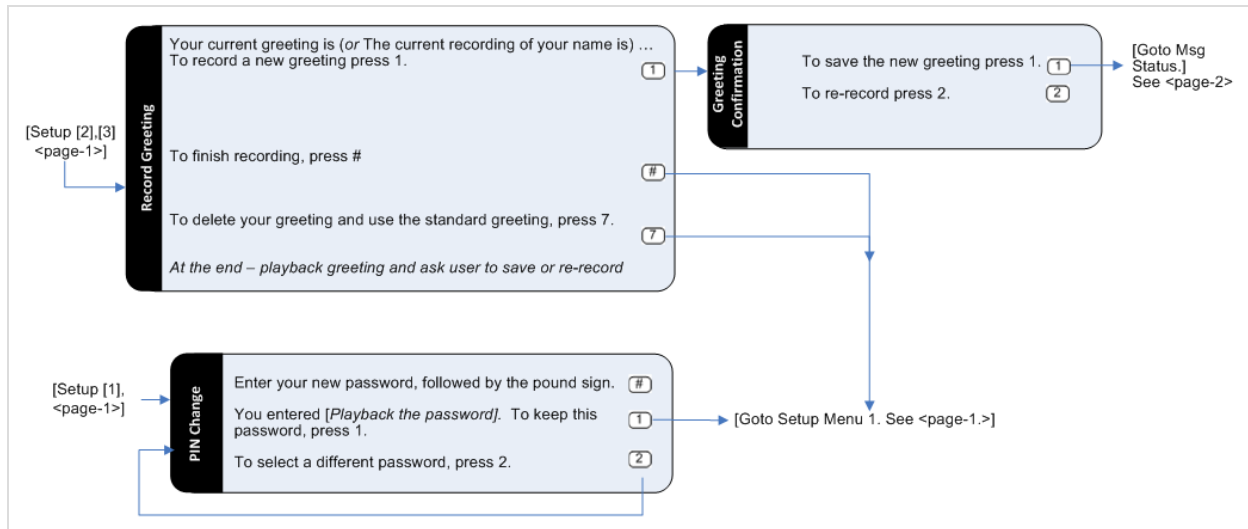
Flow of Message Retrieval (Page 1)



Flow of Message Retrieval (Page 2)



Flow of Message Retrieval (Page 3)



For the voice prompts used in this script, see [Voicemail processing prompts](#).

Voicemail processing prompts

This table lists the voicemail prompts used in the voicemail processing scripts. The Prompt column provides the English language text spoken in the prompts.

Filename	Prompt
<i>And.wav</i>	... and ...
<i>Arrived.wav</i>	Message arrived ...
<i>DaysAgo.wav</i>	... days ago.
<i>EnterMailboxID.wav</i>	Please enter your mailbox ID then press pound
<i>EnterMailboxPIN.wav</i>	Please enter your mailbox password then press pound
<i>From.wav</i>	... from...
<i>GetVoicemailFail.wav</i>	The request was unsuccessful. Please wait for few seconds.
<i>Goodbye.wav</i>	Goodbye.
<i>HoursAgo.wav</i>	... hours ago.
<i>InvalidMailboxIdOrPIN.wav</i>	The mailbox ID or password you entered is incorrect.
<i>InvalidSelection.wav</i>	You have made an invalid selection.
<i>LeaveVM.wav</i>	There is no one available to answer your call. Please leave a message at the tone. When you have finished recording, press pound or just hang up.
<i>MainMenu.wav</i>	To listen to your messages press one.
<i>Message.wav</i>	... unheard message.
<i>MessageDeleted.wav</i>	Message deleted.
<i>MessageMenu1.wav</i>	To delete, press seven. To save, press nine. For more options, press zero
<i>MessageMenu2.wav</i>	To replay, press four. To hear the message header, press five. To return to the main menu press the star. To repeat this menu, press pound .
<i>Messages.wav</i>	... unheard messages.
<i>MessageSaved.wav</i>	Message saved.
<i>MinutesAgo.wav</i>	... minutes ago.
<i>New.wav</i>	.. unheard and ...

Filename	Prompt
<i>NextMessage.wav</i>	Next Message.
<i>NoAgentMailboxDefined.wav</i>	The person you are calling does not have a mailbox defined.
<i>NoMessages.wav</i>	You have no messages.
<i>OldMessage.wav</i>	.. saved message.
<i>OldMessages.wav</i>	...saved messages.
<i>On.wav</i>	... on ...
<i>OneDayAgo.wav</i>	... yesterday.
<i>OneHourAgo.wav</i>	... one hour ago.
<i>OneMinuteAgo.wav</i>	... one minute ago.
<i>TechnicalProblem.wav</i>	We are currently experiencing technical difficulties. Our apologies for the inconvenience. Please try again later.
<i>ThankYou.wav</i>	Thank you for leaving your message, Good bye.
<i>ThankYouForCalling.wav</i>	Thank you for calling.
<i>YouHave.wav</i>	You have ...

Web services XML examples

This script requests client information.

XML request example

```
<?xml version="1.0" ?>
  <SOAP-ENV:Envelope xmlns:q0="http://wsdl.use.abc.com"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SOAP-ENV:Body>
      <q0:GetClientInformation xmlns:q0="http://wsdl.use.abc.com">
        <Request>
          <q0:RequestWrapper>
            <q0:PIN_Number>123456789</q0:PIN_Number>
            <q0:Surname>Doell</q0:Surname>
            <q0:GivenName>John</q0:GivenName>
          </q0:RequestWrapper>
        </Request>
      </q0:GetClientInformation>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

XML response example

This script provides the requested client information.

```
<?xml version="1.0" ?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Header>
      <ns1:ReplyHeader soapenv:mustUnderstand="0"
        xmlns:ns1="http://wsdl.use.abc.com/XHeader">
        <ns1:From xsi:type="xsd:string">Anonymous</ns1:From>
        <ns1:QueryDateAndTime xsi:type="xsd:string" />
        <ns1:MessageId xsi:type="xsd:string">unknown</ns1:MessageId>
        <ns1:HeaderVersion xsi:type="xsd:string" />
      </ns1:ReplyHeader>
    </soapenv:Header>
```

```

<soapenv:Body>
<GetClientInformation xmlns:n="http://wsdl.use.abc.com">
<Response xsi:type="n:Response">
<n:PersonNameResponse>
  <n:PersonInfo>
    <n:PIN_Number>123456789</n:PIN_Number>
    <n:BirthDate>19730731</n:BirthDate>
    <n:Contract>
      <n:ContractNumber>1111</n:ContractNumber>
      <n:SignatureLocation>New York</n:SignatureLocation>
      <n:SignatureDate>20040220</n:SignatureDate>
      <n:SignatureTime>124010</n:SignatureTime>
    </n:Contract>
    <n:Contract>
      <n:ContractNumber>2222</n:ContractNumber>
      <n:SignatureLocation>Boston</n:SignatureLocation>
      <n:SignatureDate>20040425</n:SignatureDate>
      <n:SignatureTime>123015</n:SignatureTime>
    </n:Contract>
    <n:Contract>
      <n:ContractNumber>3333</n:ContractNumber>
      <n:SignatureLocation>Washington DC</n:SignatureLocation>
      <n:SignatureDate>20050330</n:SignatureDate>
      <n:SignatureTime>151215</n:SignatureTime>
    </n:Contract>
  </n:PersonInfo>
</n:PersonNameResponse>
</Response>
</GetClientInformation>
</soapenv:Body>
</soapenv:Envelope>

```

5: Blocks

Note

Block names must be unique within each script and must not exceed 256 characters in length.

This section contains the following topics:

- [Block availability per CallXML script](#)

See also [Script blocks panel](#) in the Workspace editor overview.

Block availability per CallXML script

This table indicates which CCaaS Designer blocks are available in particular CallXML scripts.

CCaaS Designer blocks are listed in the first column and the CallXML scripts in the subsequent column headings. The script names also correspond to CCaaS events.

Block	Ivr	Ivr Timeout	Call Answered	Call Arrival	Call Aged	Call Pre Sent	Music On Hold In Queue	Music On Hold	In Queue Timeout	Call Rejected	Pickup Timeout	Call Termination	Outgoing Call	A2A Call Rejected	Call Deferred	Playback Audio Call Abandoned	Playback Audio Rebound Queued	Playback Audio Rebound Rejected	Post Call
Abort	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Add Skill	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No
Agent Selection Rule	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No
Agentless Campaign Outcome	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Answer	Yes*	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Assign	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Bridge Call	Yes	Yes	No	No	No	No	No	No	No	Yes	No	No	No	No	No	No	No	No	Yes
Call Maturity/Priority	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No
Caller Message	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No	Yes	No	No	No	Yes
Caller Name	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No	Yes	No	No	No	Yes
Caller URL	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No	Yes	No	No	No	Yes
Check Date	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Check Time	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes

Block	Ivr	Ivr Timeout	Call Answered	Call Arrival	Call Aged	Call Pre Sent	Music On Hold In Queue	Music On Hold	In Queue Timeout	Call Rejected	Pickup Timeout	Call Termination	Outgoing Call	A2A Call Rejected	Call Deferred	Playback Audio Call Abandoned	Playback Audio Rebound Queued	Playback Audio Rebound Rejected	Post Call
Clear Digits	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Clear Digits Events	Yes	Yes	Yes	No	No	No	Yes	No	No	Yes	No	No	No	No	No	No	No	No	Yes
Clear User Data	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Condition	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Define Grammar	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Edit Email	No	No	No	No	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No
Get Agent Statistics	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Get Digits	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Get Phone Information	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
Get Queue Statistics	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Get Message	No	No	No	Yes	No	No	No	No	Yes	Yes	No	No	No	No	No	No	No	No	No
Get User Data	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
GoTo	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
GoTo Voicemail	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Hangup	Yes	Yes	No	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	No	No	No	Yes
IVR Recording	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes
IVR Transfer Recording	Yes	Yes	No	No	No	No	No	No	No	Yes	No	No	No	No	No	No	No	No	Yes
Leave Voicemail	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes

Block	Ivr	Ivr Timeout	Call Answered	Call Arrival	Call Aged	Call Pre Sent	Music On Hold In Queue	Music On Hold	In Queue Timeout	Call Rejected	Pickup Timeout	Call Termination	Outgoing Call	A2A Call Rejected	Call Deferred	Playback Audio Call Abandoned	Playback Audio Rebound Queued	Playback Audio Rebound Rejected	Post Call
Log	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Logging Control	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Mask Digits	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
MD5 Encryption	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Menu	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Optional Parameters	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes
Percent Allocated	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Play Audio	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Play Date	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Play Greeting	No	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes
Play Money	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Play Number	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Play Text	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Play Video	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Playback Audio**	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	No
Queue Name	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	Yes	No	No	No	No	No	No
Read SIP Header	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Record Audio	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Record Video	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Record Voicemail	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes

Block	Ivr	Ivr Timeout	Call Answered	Call Arrival	Call Aged	Call Pre Sent	Music On Hold In Queue	Music On Hold	In Queue Timeout	Call Rejected	Pickup Timeout	Call Termination	Outgoing Call	A2A Call Rejected	Call Deferred	Playback Audio Call Abandoned	Playback Audio Rebound Queued	Playback Audio Rebound Rejected	Post Call
(deprecated)																			
Recorder	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No	No	Yes
Remove Skill	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No
ReQueue	No	No	No	No	No	No	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No
Routing Mode	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No
Routing Properties***	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No
Run	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Schedule Callback	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Script Entry Point	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Send Callback (deprecated)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Send Email	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Send Message	No	No	No	Yes	No	No	No	No	Yes	Yes	No	No	No	No	No	No	No	No	No
Send User Data	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Set Defer Type	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No	No
Set Event	Yes	Yes	Yes	No	No	No	Yes	No	No	Yes	No	No	No	No	No	No	No	No	Yes
Set Language	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Set SIP Header	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No	No	No	No
Speech Enabled Menu	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Speech Play and	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes

Block	Ivr	Ivr Timeout	Call Answered	Call Arrival	Call Aged	Call Pre Sent	Music On Hold In Queue	Music On Hold	In Queue Timeout	Call Rejected	Pickup Timeout	Call Termination	Outgoing Call	A2A Call Rejected	Call Deferred	Playback Audio Call Abandoned	Playback Audio Rebound Queued	Playback Audio Rebound Rejected	Post Call
Collect																			
Stop Video	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Supervised Transfer	Yes	Yes	No	No	No	No	No	No	No	Yes	No	No	No	No	No	No	No	No	Yes
Switch	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Table	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Table Lookup	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Terminate	Yes	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	No	No	Yes	No	No	No	No
Transfer Call	Yes	Yes	No	No	No	No	No	No	No	Yes	No	No	No	No	No	No	No	No	Yes
Transfer by Refer	Yes	Yes	No	No	No	No	No	No	No	Yes	No	No	No	No	No	No	No	No	Yes
UnQueue	No	No	No	No	Yes	No	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No
VXML Transfer Call	Yes	Yes	No	No	No	No	No	No	No	Yes	No	No	No	No	No	No	No	No	Yes
Wait	Yes	Yes	No	No	No	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	No	Yes
Wait for Events	Yes	Yes	Yes	No	No	No	Yes	No	No	Yes	No	No	No	No	No	No	No	No	Yes
Web Services	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes

* Available only when **[script_answer_supervision] = 1** in Registry.

** Available only in the Elsbeth integration.

*** Not necessarily available.

Abort



This block aborts playback of asynchronous mode *.wav* files, that are currently active for either the inbound or outbound line.

Properties	Values	Description
Line	Inbound	The function is applied to the original (incoming) call.
	Outbound	The function is applied to the outbound call (to where the call is transferred).

This block is used to end a [Play Audio](#), [Play Date](#), [Play Money](#), or [Play Number](#) block that has been set to Async Mode. It has the same function as termination digits except the script determines when to end the *.wav* file instead of the caller.

Add Skill



This block assigns a skill to a call.

Properties	Values	Description
Skill Name	String	<p>The name of the skill to be added to the current call.</p> <p>This must be a valid skill for this tenant for the skill to be added, so spelling is critical but skill names are not case sensitive.</p> <p>You can select the skill name from a drop-down list of available skills in the tenant. This list is refreshed every time you select the block. If a new skill is added while you are editing the block, to see the added skill, select another block or select the Flow editor background and then reselect the Add Skill block.</p> <p>If a skill is removed from the tenant and the value is currently being used as the skill name, the value remains as the selected value, until you manually change it.</p> <p>You are not restricted to the values provided in the drop down list, so you can use an expression as the selected value.</p>

Agent Selection Rule



This block routes call traffic based on skill weighting, if it is implemented in your contact center.

For more information on assigning skills to agents, see the *CCSP Tenant Administrator Guide*.

Properties	Values	Description
Rule	Skill Neutral	Route the call to the agent with the longest time in stage.
	Least Skilled	Route the call to the available agent with the highest number assigned to this skill.
	Most Skilled	Route the call to the available agent with the lowest number assigned to this skill.

In the skill definition, the skill should be defined either to enable weighting or not. Upon enabling weighting, the administrator may then set the weights as needed.

Note

If no weight is specified for a particular agent, the weight is assumed to be 0.

The call object contains another field **agent selection rule** that can be set to **most skilled**, **least skilled** or **skill neutral**.

Skill weight is considered by the Call Matcher (CMS) only if the CMS is doing agent selection (call comes in, agent must be chosen to answer the call). It is not relevant to call selection (calls waiting, agent becomes available, choose call).

CMS first considers the pool of agents for which all required skills are met. If none of the skills are enabled for weighting, or if the skill neutral flag is set on the call object, CMS routes the call to the agent with the longest time in stage.

If one of the skills is configured for weighting, then the CMS selects the agent with the lowest number (for the most skilled flag in call object) or the agent with the highest number (for the least skilled flag in call object). If there is more than one agent with the same number, then the secondary criteria is load, and the third criteria is longest time in stage.

If more than one of the skills is configured for weighting, then the CMS sums the agents' individual skill weights (the sum is only for skills which are present in the call requirements) to create a skill score, and then proceeds to choose either the most skilled (lowest score) or least skilled (highest score) agent.

Agentless Campaign Outcome



This block determines the action to take when the IVR Agentless script completes.

Properties	Values	Description
Action	Terminate Call	End the call in IVR.
	Route To Agent	Route the call to a CCaaS agent, based on the existing routing properties.

For an example, see [Agentless IVR script](#).

Answer



VCS only.

This block answers a call.

Properties	Values	Description
Action	Answer Call	Answer the incoming call.
	Continue Ringing	Force the incoming call to remain in the ringing state.
	Reject Call	Cause the incoming call to be disconnected.
Comment	Optional text	Description of what this block does in your script.

By default, the VCS answers all calls before firing the Incoming Call Event (the event that runs the IVR script). This behavior can be changed through a registry setting on the VCS, (**script_answer_supervision**) to make the event fire before the call is answered, thus allowing a script to control answer supervision. This setting should be changed by the system administrator.

Setting this block to **Answer Call** connects the call, allowing the script to play audio prompts and receive DTMF. Setting this block to **Continue Ringing** allows the call to continue ringing on the caller's phone until an agent answers the call. Then the call is answered by the VCS.

Setting this block to **Reject Call** allows the call to be hung up without charging the caller or service provider for the call. The SIP code for rejection is configured in CCaaS Admin, in the VCS parameters.

Caution

Your system administrator must create the `../VCSdebug/script_answer_supervision` registry key and set it to **1**.

Assign



This block assigns a static variable to use in this scriptable event. This variable is only available for the duration of the current event. A variable set in one event is *not* available in other events for the same call. These variables are not available to the applications. Also see [How to use variables](#).

To use the value of a variable set in an Assign block, add `$` before and after the variable name. For example, to use a variable named **Counter** in another block, type `$Counter$`.

Note

If you want to use variables that persist and are available to the agent applications, see [Optional parameters](#).

Properties	Values	Description
Variable Name	String	Name of the variable. To use hyphenated names, enclose the name in quotes, for example, "My-VCS"
Variable Value	String (Boolean number)	Value of the variable.

Direct Inward Dial extensions in scripts

To work with Direct Inward Dial (DID) extensions in scripts:

1. Use one Assign block to assign the DID number to the **calldata.directextension** variable.
2. Use a second Assign block to assign a boolean value to the **calldata.directextensioncall** variable. A value of **0** indicates an ACD call. A value of **1** indicates a DID call.

Bridge Call



VCS only.

This block sets the bridging method for calls. See [Transfer Call](#) and [Supervised Transfer](#) for more information.

Typically this block is only used after a Transfer Call or Supervised Transfer block which has the **Bridge Method** set to **Do not Bridge**.

Properties	Values	Description
Bridge Method	Bridge calls	Bridges the call control and media stream so the individual calls become a single call. If either party disconnects, the call terminates.
	Bridge media only	Parties can communicate with each other, but if one party disconnects, call does not terminate.
	Don't bridge	Use only for indexing backward compatibility. When used, the following warning message is written in the CCaaS Designer logs "WARNING: bridge call not supported for Don't Bridge option in Bridge block" . The calls are not bridged when the Transfer Call (or Supervised Transfer) and Bridge blocks are used and both blocks have the Don't Bridge value.
	Bridge media & DTMF	Parties can communicate with and forward DTMF tones to each other, but if one party disconnects, call does not terminate.

Call Maturity/Priority



This block modifies the arrival time bias for the current call. The lower the value, the higher you move the call in the queue.

Usually, the system computes call maturity. The Call Maturity/Priority block enables you to set your own value for call maturity.

Note

The system interprets a call maturity of **0** literally. A call maturity value of zero implies that the default TQoS of a queue is used as the call maturity.

You can separately assign **Call Priority** in your scripts. Priority overrides maturity. Calls with a higher priority are handled before calls with a lower priority, regardless of the maturity of calls. For calls with the same priority, maturity determines the order in which calls are handled.

Properties	Values	Description
Call Maturity	Integer	Number of seconds to use for the time bias. Note This <i>replaces</i> the current value. It is not added to it.
Call Priority	Integer	A number between 0 and 99, where 99 is the highest priority. The default is 49.

For more information on the following topics, see the *CCSP Tenant Administrator Guide*:

- TQoS — Target Quality of Service
- Call Priority
- Routing of calls

Caller Message



This block displays a message to the agent about the caller.

Properties	Values	Description
callerMessage	String	Can display any message about the caller. Caution <ul style="list-style-type: none">This message must not exceed 30K characters.This message overwrites any previously configured Caller Message unless set to <code>\$(CallData.CallerMessage\$ New Message, for example.</code>

Caller Name



This block set the caller's name.

Properties	Values	Description
Caller Name	String	The name of the caller. This is displayed to the agent when the call is answered. For a phone call, this can be populated by a database lookup, or from the Caller ID, if available.

Note

Outbound dialing campaigns only. If this string is set to `$(callData.CampaignName$`, then the name of the campaign appears is in the **Caller Name** field in the agent applications.

Caller URL



This block sets the caller's URL.

Properties	Values	Description
URL	String	This can be any URL related to the call or caller. For a Web call or email it is typically the website the caller was viewing when they requested CCaaS.

Check Date



This block routes calls based on the date criteria you select.

Properties	Values	Description
Time Zone	Local	The current local date for the connection server is used.
	GMT	The current GMT date is used.

Configure the Check Date block

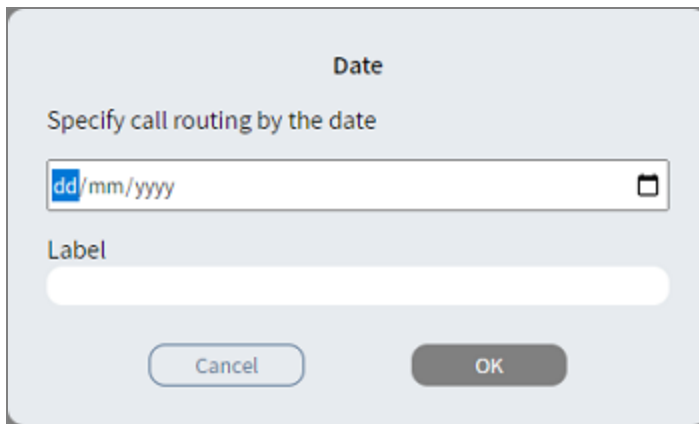
1. Right-click the block and click **Add case**.
2. Click one of the following options and follow the corresponding instructions:
 - [Date](#)
 - [Day of week](#)
 - [Holiday](#)

3. Connect the **Case** branch  to the appropriate block in the script.

4. Connect the **Else** branch  to the appropriate block in the script.

Configure date

1. In the **Date** dialog, select a single date or a block of dates.



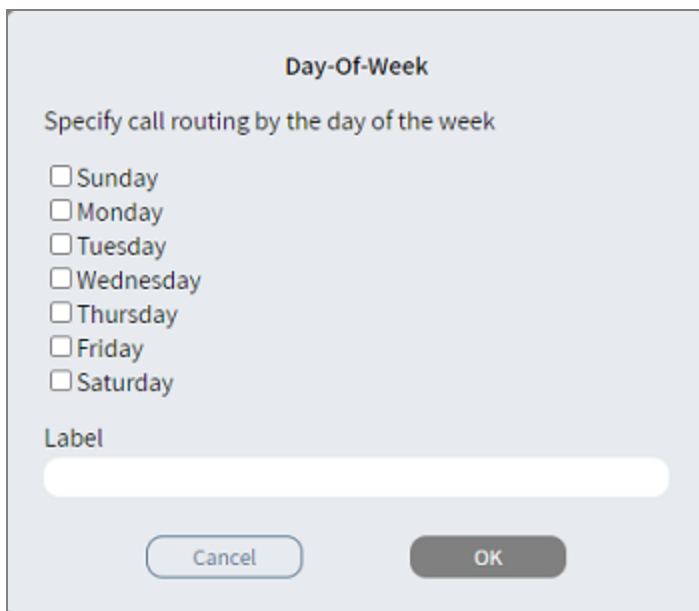
The screenshot shows a dialog box titled "Date". Below the title is the instruction "Specify call routing by the date". There is a text input field containing the placeholder "dd/mm/yyyy" and a calendar icon to its right. Below the input field is a "Label" text input field. At the bottom of the dialog are two buttons: "Cancel" and "OK".

2. Optionally, in **Label**, type an alternative description.

3. Click **OK**. The **Case** branch  appears.


Configure day of week

1. In the **Day-Of-Week** dialog, select one or more days.



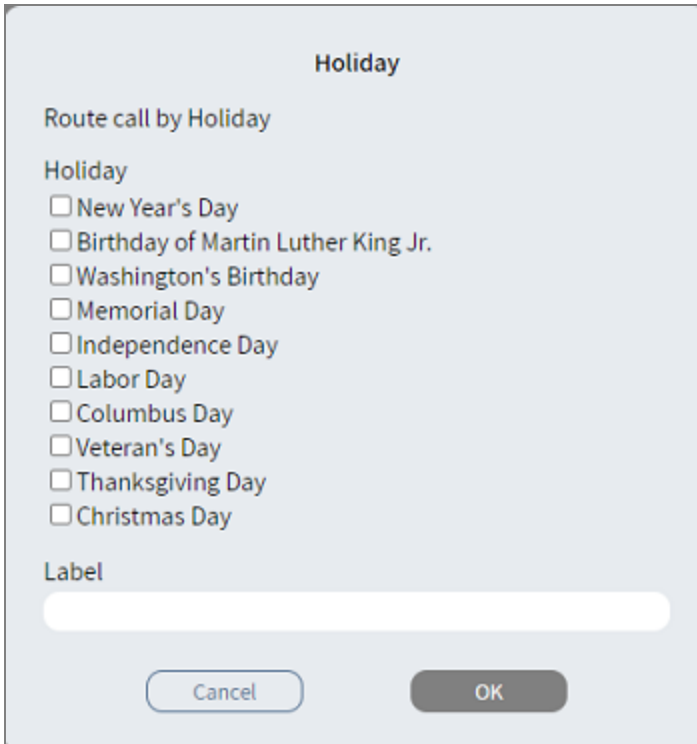
The screenshot shows a dialog box titled "Day-Of-Week". Below the title is the instruction "Specify call routing by the day of the week". There is a list of days of the week, each with an unchecked checkbox: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. Below the list is a "Label" text input field. At the bottom of the dialog are two buttons: "Cancel" and "OK".

2. Optionally, in **Label**, type an alternative description.


3. Click **OK**. The **Case** branch  appears.

Configure holiday

1. In the **Holiday** dialog, select one or more holidays.



2. Optionally, in **Label**, type an alternative description.

3. Click **OK**. The **Case** branch  appears.

Note

- The information for this file is contained in the file *Holiday.xml*. The default location of this file is:
`<installation directory>/CosmoCom/Server Components/Designer`
- This file can be edited with a text editor and should be updated on a yearly basis, or as often as needed.

Check Time

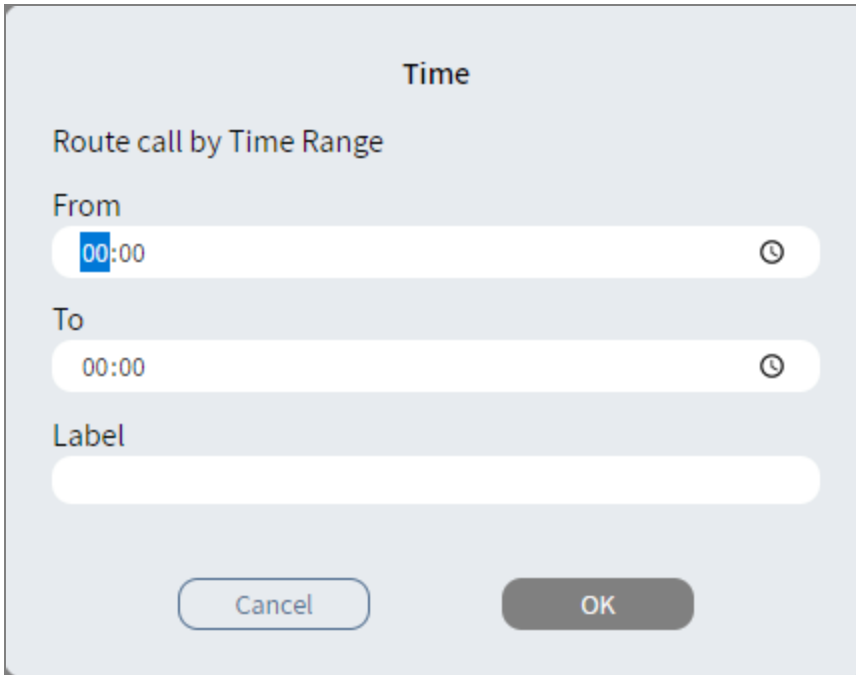


This block routes calls by the time range that you select.


Properties	Values	Description
Time Zone	Local	The current local time for the connection server is used.
	GMT	The current GMT time is used.

Configure the Check Time block


1. Right-click the block and click **Add case**. The **Time** dialog appears.



The screenshot shows a dialog box titled "Time" with the subtitle "Route call by Time Range". It features three input fields: "From" (set to 00:00), "To" (set to 00:00), and "Label" (empty). The "From" and "To" fields have a clock icon on the right. At the bottom, there are "Cancel" and "OK" buttons.

2. Enter a time range.
3. Optionally, in **Label**, type an alternative description.
4. Click **OK**. The **Case** branch  appears.
5. Connect the **Case** branch to the appropriate block in the script.

6. Repeat steps 1-5 to add the branches you need.

7. Connect the **Else** branch  to the appropriate block in the script.

Clear Digits



This block clears the digit buffer before the script processes the next command. This block is typically used to clear wrong digits entered from the buffer before prompting the caller to reenter the digits.

For example, "The Customer ID you entered is not valid, please try again".

Then clear the digits and return to the GetDigits block.

Properties	Values	Description
Line	Inbound	The function is applied to the original (incoming) call.
	Outbound	The function is applied to the outbound call (to where the call is transferred).

Clear Digits Events



This block clears the digit buffer before the script processes to the next command. This block is typically used before you want to subscribe to the **On Digits** event, used by the [Set Event](#) block.

Clear User Data



This block clears the line you that you specify.

Properties	Values	Description
Line	String	Inbound or outbound line which you want to clear. Clears the user data buffer on the selected line. Any data received up to this point that has not been processed by the Get User Data block is discarded.

Condition



This block contains an expression which can be evaluated as True or False.

Properties	Values	Description
Expression	True or False	Causes branching in the script so must be able to be evaluated as True or False . This property is data-type sensitive. For string to string comparison, text should be surrounded by quotes on both sides. For example, "\$CallData.callset Name\$" <> "" returns True if the Queue has not been configured.

Note

Be careful when using variables in the **Expression** property of the Condition block. The way that the interpreter expands variables into an intermediate string before using them can cause the interpreter to incorrectly evaluate an expression. This is the only property in CCaaS Designer in which you must use quotes (") to denote a string. Furthermore, if a variable is to be evaluated as a string, enclose the variable in quotes.

Define Grammar

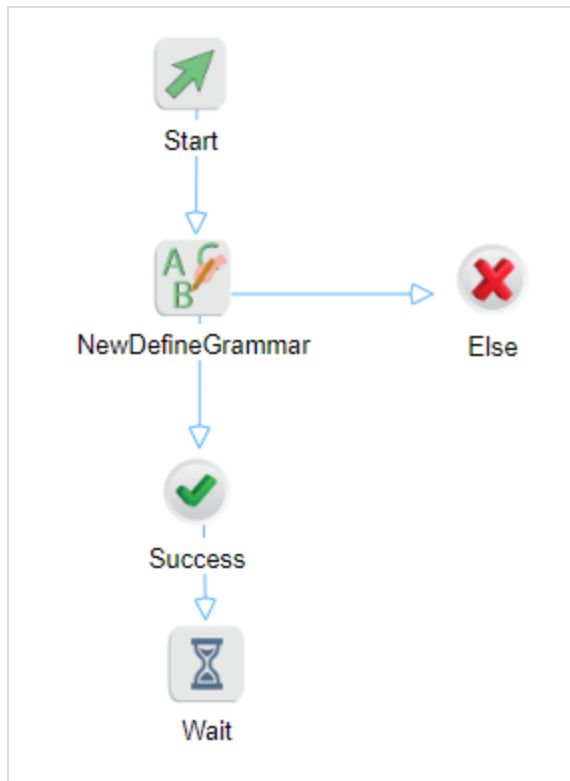


This block defines the Automatic Speech Recognition (ASR) grammar used in the application.

Properties	Values	Description
Grammar	Text	A grammar can be specified in the following ways: <ul style="list-style-type: none">• Refer to a built-in grammar.• Specify it by using XML.• Refer to the URI of a grammar stored on a web server.
Content-Id	Text	ID or name for the grammar by which it will be referenced, for example, <code>request1@form-level.store</code> .
Comment	Text	User description of the block in the script.

Routing

To include routing, add Success and Else cases, as show in the following example.

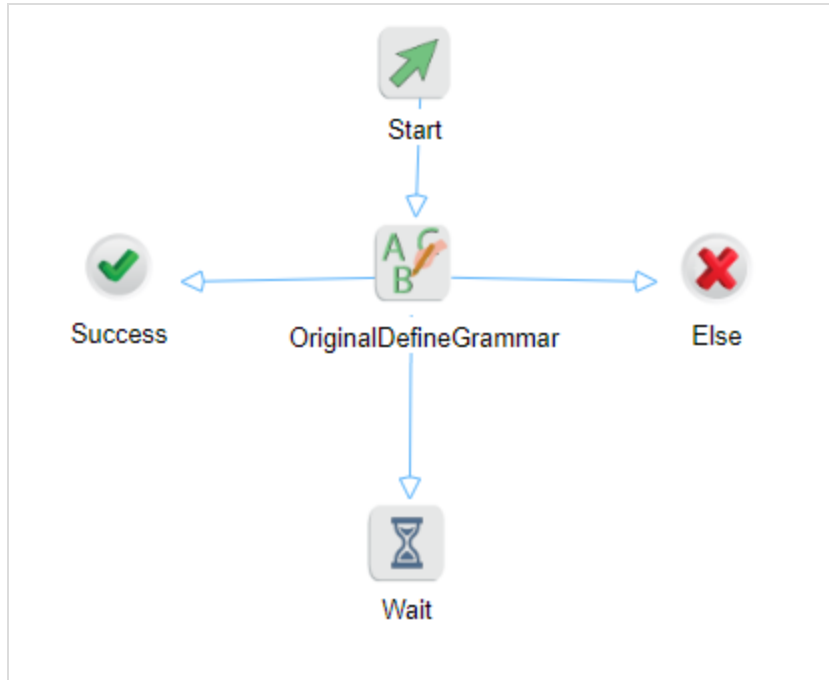


Note

In case of error (Else case), there is an additional delay of 5 seconds until the MRCP server responds to the request.

Backward compatibility

To maintain backward compatibility in existing scripts, this block treats the next block it is directly connected to as the Success case, as show in the following example.



Caution

If you delete the connection between the original DefineGrammar and Wait blocks, you must reconnect it through the Success case.

Edit Email



This block allows manipulation of the email fields.

Properties	Values	Description
Rule	Reply To	If set, the reply to the email is sent to this address.
	From	Email originator's email address.
	To	Primary recipient's email address.
	CC	Carbon copy - secondary recipient's email address.
	BCC	Blind carbon copy - private secondary recipient's email address.
	Subject	Email's title.
	Importance	Email's priority level: <ul style="list-style-type: none">• Low — low priority• Normal — normal priority• High — high priority• NoChange — leave as is

Note

- A validation process runs on the input of the email address values. If validation fails, the email is moved to the Sent Items folder with the Invalid state.
- Validation is minimal and doesn't cover the full RFC 5322.

All recipient email address fields can have multiple values separated by a semi-colon (;).

Email addresses can be configured to display as follows.

Syntax	Example
Only email address	Jane.Green@enghouse.com
Name and email address	Jane < Jane.Green@enghouse.com >
Name with quotation marks ("").	"Jane" < Jane.Green@enghouse.com >

Syntax	Example
Usually used when the display name includes a space.	
Email address with left and right angle brackets (<>)	<Jane.Green@enghouse.com>
Multiple email addresses - combination of any of the above formats	Jane.Green@enghouse.com; Simon.Brown@enghouse.com;Julie.Black@enghouse.com

To set the Reply To, To, CC, and BCC fields to empty, use `$empty$`. See [How to use the \\$empty\\$ constant](#).

Get Agent Statistics



This block allows polling of the Realtime Reports (RTR) server to make decisions based upon current agent activity.

Properties	Values	Description
URL to RTRWeb	Web address	<p>The location of the RTR server as a URL. For example, <code>http://ais.yourcompany.com/rtrdll/</code></p> <p>This can be specified as an IP address.</p> <p>Caution</p> <p>This should be only the path to the <i>RTRWeb.dll</i> file, and should not include the <i>RTRWeb.dll</i> filename.</p>
RTR Timeout	Seconds	<p>Maximum time to wait for RTR to respond before continuing on to the next block. Default is 10 seconds.</p> <p>Note</p> <p>This property is a new feature of this block. When you apply the corresponding hot fix to existing scripts, the default value does not appear but is still applied.</p>
Group Name	String	<p>You can select the group name from a drop-down list of available groups in the tenant. This list is refreshed every time you select the block. If a new group is added while you are editing the block, to see the added group, select another block or select the Flow editor background and then reselect the Get Agent Statistics block.</p> <p>If a group is removed from the tenant and the value is currently being used as the group name, the value remains as the selected value, until you manually change it. You are not restricted to the values provided in the drop down list, so you can use an expression as the selected value.</p>

Properties	Values	Description
Agent state	Initialized	Get statistics on all agents who are currently initialized.
	All	Get statistics on all agents who are logged on.
	Available	Get statistics on all agents who are logged on and waiting for calls.
	Available In Call	Get statistics on all agents who are in one or more calls but may have capacity for additional calls.
	Released	Get statistics on all agents who are unavailable to take calls.
	Released In Call	Get statistics on all agents who are in one or more calls but not available for additional calls.
Agent in call state	All	Get statistics on all agents who are in all the following states.
	In Call	Get statistics on all agents who are in the In all state.
	On Hold	Get statistics on all agents who are in the On Hold state.
	Preview	Get statistics on all agents who are in the Preview state.
	Ringing	Get statistics on all agents who are in the Ringing state.
	Wrap Up	Get statistics on all agents who are in the Wrapup state.
Call Type		Caution For valid combinations, see the following table.
	All	For cumulative stats for all call types.
	CallBack	For callback calls only.
	Internet	For chat calls.
	Message	For email or voicemail messages.
	Multiple	For multichat agents.
	Phone	For voice calls.
Result Variable	String	Variable name in which to store a returned value for further processing. Returns -1 for timeout or failure. See How to use variables .

The valid combinations of Agent state, Agent in call state, and Call type are as follows.

Agent state	Agent in call state	Call type
Initialized	N/A	N/A
Available	N/A (All)	N/A (All)
	Ringling	N/A (All)
Available In Call	In Call	CallBack, Internet, Message, Multiple, Phone
	On Hold	CallBack, Internet, Message, Multiple, Phone
	Preview	Phone
	Ringling	Multiple
	Wrap Up	CallBack, Internet, Message, Multiple, Phone
Released	N/A (All)	N/A (All)
Released In Call	In Call	CallBack, Internet, Message, Multiple, Phone
	On Hold	CallBack, Internet, Message, Multiple, Phone
	Preview	Phone
	Wrap Up	CallBack, Internet, Message, Multiple, Phone

Get Digits



VCS only.

This block allows the script to capture DTMF digits (tones) entered by the caller.

Properties	Values	Description
Max Time	Seconds (integer)	Maximum amount of time to wait for the entire string of digits. If Max Time is left blank, it is calculated by multiplying the Max Silence (default is 3 seconds) by the Max Digits.
Max Silence	Seconds (integer)	Maximum amount of time (silence) between digits. If left blank the default is 3 seconds per digit. So if the number could be 3 or 4 digits, the Max Digits would be set to 4, and if 4 digits are entered, the getDigits block ends. If 1, 2 or 3 digits are entered, the block waits 3 seconds then ends.
Termination Digits	0-9 # and *	Enter digit(s), # or *. These value(s) can be used by the caller to terminate this command. For example, to allow any touch tone to be a termination digit, set this to 1234567890#*
Include Termination	Yes or No	Include termination digit in the resulting string. For example, the termination digit is # and caller enters 1234#. If this is Yes, the result is 1234#. If this is No, the result is 1234.
Max Digits	Integer	Maximum number of DTMF digits allowed.
Result Variable	String	Name of a variable into which the digit string is to be passed. Must start with an alphabetic character and not exceed 255 characters in length. See How to use variables .
Clear Digits	Yes or No	Clear the digit buffer of any digits entered prior to this command. This should be set to No by default.

Properties	Values	Description
Line	Inbound	The function is applied to the original (incoming) call.
	Outbound	The function is applied to the outbound call (to where the call is transferred).

Note

If the caller has entered 10 digits and there are 5 GetDigits blocks, each with **Max Digits** set to **2**, each GetDigits block takes 2 digits so there will be 10, then 8 then 6, then 4 then 2 and after the 5th GetDigits block there will be no digits left in the buffer, but only if none of the blocks have **Clear Digits** set to **Yes**.

Get Message



This block pauses the script execution while waiting for caller input and the subsequent block is executed when either a message is received or the timeout elapses without any response from the caller.

Properties	Values	Description
Timeout	Integer	Number of seconds, in the range 1-600, to wait for a response from the caller. The default is 300. Note: <ul style="list-style-type: none">• If the value is 0 or greater than 600, then 600 is set.• If the value is negative, the block is skipped and the next block in the script is executed.
Return Value	String	Name of a variable that contains the caller's input. If the timeout is reached without or before the caller input, an empty value is set. Note: <ul style="list-style-type: none">• The value must not be an empty string.• If the value is empty, the block is skipped and the next block in the script is executed.

Get Phone Information



This block allows an agent to view information about the caller displayed in the **Chat** window of their agent application.

Caution

This block must be used in conjunction with the TARGUSinfo phone-based data service. To be able to use this block, you need an account (Service ID) with TARGUSinfo, as well as a username and a password.

Properties	Values	Description
URL to WebGW	Web address	Currently, the default value for this is already entered in this field, and should not be changed.
Service Id	Number	This is the account ID or Service ID assigned to your account by TARGUSinfo.
User Name	String	The user name for this account.
Password	String	The password for this account.
Phone Number	String	This number must be available to this block, either as the result of caller input, or, for example, by getting line.ANI information from the Line object. Note This service is only available for US and Canadian calls. This field only accepts 10-digit phone numbers without spaces or characters other than digits.
Result Variable	String	Variable name in which to store returned value(s) for further processing. See How to use variables .

Information fields

The following information fields are available in the Result Variable.

Field	Description
.Error	0 if the call was successful.
.Building	If the building has a name, such as the Empire State Building, that information may appear here.
.Address	The street address of the caller
.City	The caller's city
.State	The caller's state
.Zip	The caller's 5-digit ZIP code
.Zip4	The caller's 4-digit ZIP code extension
.NameType	<p>This is either B for a business or C for a home consumer number.</p> <p>For B, a business, the following values appear:</p> <ul style="list-style-type: none"> • .BusinessName — the name of this business • .BusinessSuffix — this field may show such information as "Inc.", "LTD", or "LLP". <p>For C, a home consumer, the following values appear:</p> <ul style="list-style-type: none"> • .LastName — the caller's last name • .FirstName — the caller's first name • .MiddleInitial — the caller's middle initial
.TimeZone	The caller's time zone

Get Queue Statistics



This block allows polling of the Realtime Reports (RTR) server to make decisions based upon current queue activity.

Properties	Values	Description
URL to RTRWeb	Web address	The location of the RTR server as a URL. For example, <code>http://ais.yourcompany.com/rtrdll/</code> This can be specified as an IP address. Caution This should be only the path to the <i>RTRWeb.dll</i> file, and should not include the <i>RTRWeb.dll</i> filename.
RTR	Seconds	Maximum time to wait for RTR to respond before continuing on to the next block. Default is 10 seconds. Note This property is a new feature of this block. When you apply the corresponding hot fix to existing scripts, the default value does not appear but is still applied.
Call Type	All	For cumulative statistics for all call types listed below.
	CallBack	For callback calls only
	Internet	For chat calls
	Message	For email or voicemail messages
	Phone	For voice calls
Statistic	Calls Waiting	Number of calls waiting in this queue
	Calls Being Handled	Number of calls connected to agents
	Longest Wait Time	Longest wait time of current calls in this queue
	Average Wait Time	Average waiting time in this queue
Queue Name	String	To find this name, in CCaaS Admin, use the toolbar to turn on Detail view and from the Action menu, select Customize and enable the First column in detail view.

Properties	Values	Description
Result Variable	String	Variable name in which to store a returned value for further processing. Returns -1 for timeout or failure. See How to use variables .

Get User Data



This block waits for data to arrive at the specified line during the time out interval and receives the data if it arrives within that interval.

Properties	Values	Description
Line	String	Inbound or outbound line from which you want to get the data.
Termination Digits	0-9 # and *	Enter a digit, #, or * that terminates this command.
Timeout	Integer	Optional. If greater than 0, the number of seconds to wait for information.
	-1	Infinite.
	0	Zero timeout (do not block).
	No value	Zero timeout (do not block).
Result Variable	String	Variable name to store value returned from this command. The returned value is the body of the received INFO message.

The GetUserData block parses a SIP INFO message to retrieve its content values.

Build the SIP INFO message, as follows:

- The content type is `application/ccu-integration`.
- Format the content this way:

```
<ParameterName1>=<ParameterValue1> [Return]
<ParameterName2>=<ParameterValue2> [Return]
.
< ParameterNamen>=<ParameterValuen> [Return]
```

Retrieved values are recorded in the Result Variable of the block, as follows:

```
<ReturnVariableName>.<ParameterName1>
<ReturnVariableName>.<ParameterName2>
.
<ReturnVariableName>.<ParameterNamen>
```

GoTo



This block continues execution at a call block in this script or another script.

Properties	Values	Description
GoTo	Depends on where control is passed. See examples below.	
	Same script	#RecordAudio924 (#NameOfBlock)
	Different script (top)	http://ServerName/Directory/CallArrival.xml
	Different script (block)	http://ServerName/Directory/MusicOnHold.xml#PlayAudio865
	ASP page	http://ServerName/Directory/DBLookup.asp
Method	GET or POST	When the GoTo property is an ASP page, all CCaaS Designer variables are automatically sent to the page using either GET (query string) or POST (form). The Method property controls how they are sent. For example, the <i>DBLookup.asp</i> script uses GET (QueryString) to retrieve information from the XML script as a URL. Therefore the GoTo block must also use the GET method.

GoTo Voicemail



This block continues execution at the Leave Voicemail script or the Retrieve Voicemail script.

Properties	Values	Description
Voicemail script	Leave Voicemail or Retrieve Voicemail	Leave the current script and execute the selected script. The default is Leave Voicemail.

Hangup



VCS only.

This block terminates the call.

Properties	Values	Description
Line	Inbound	The <i>.wav</i> file is played to the original (incoming) caller.
	Outbound	The <i>.wav</i> file is played to the transferee (the party to whom the call is transferred).

IVR Recording



VCS only.

This block enables you to record parts of the IVR script.

The block is allowed only in the Ivrr and Post Call scripts.

Properties	Values	Description
Action	Start	Starts recording.
	Stop	Stops recording.

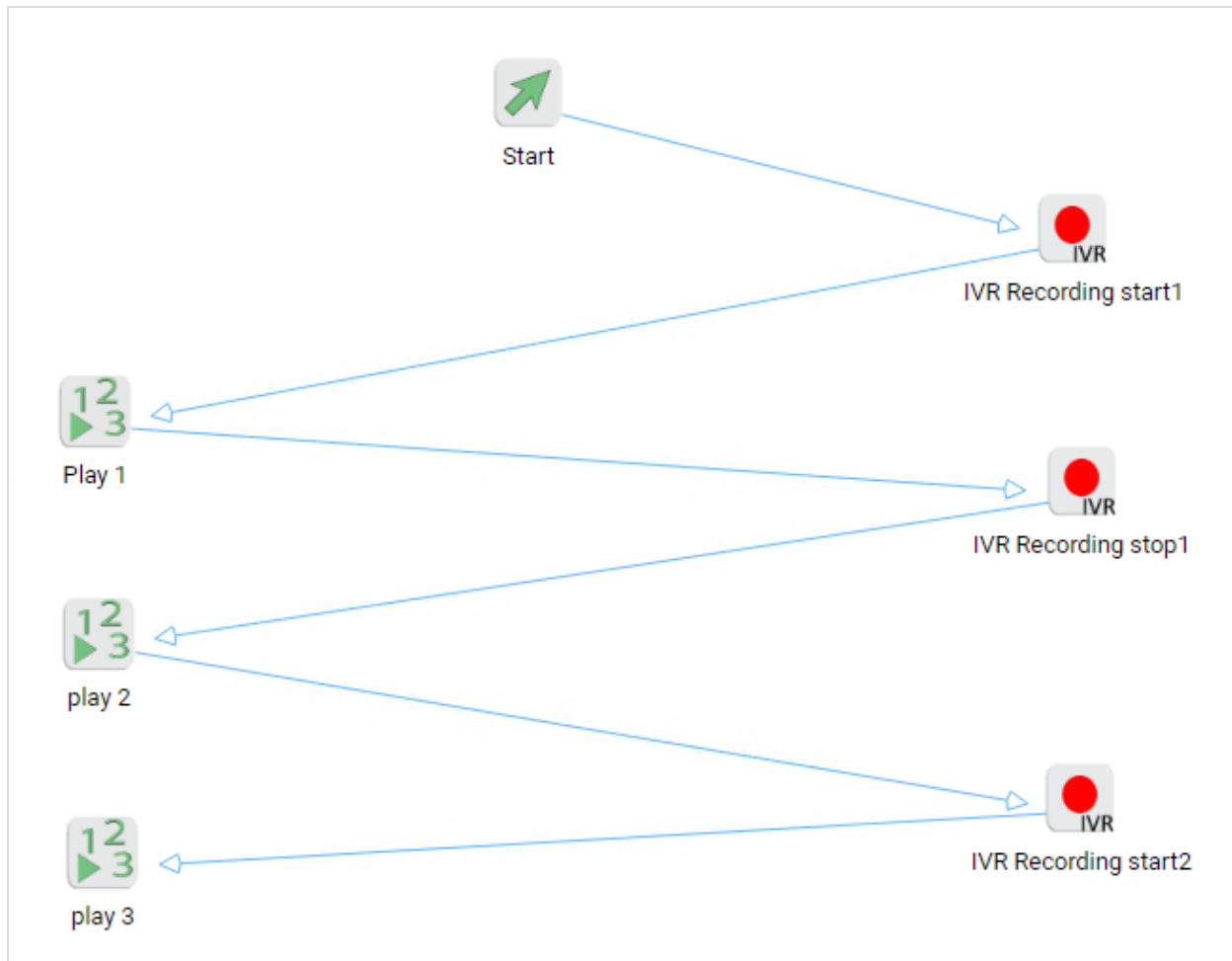
IVR recording always stops automatically at the end of the script. Consequently, a final IVR Recording block set to Stop is not required.

Caution

- The IVR recording includes everything a customer says while IVR recording is on.
- IVR recordings are precise to 1 whole second, which means that the recording of a block may start too soon or finish too early, unless you insert a Wait block set to 1 second before the start and stop IVR Recording blocks. See the second example below.
- Leave Voicemail and Record Voicemail blocks are not recorded. Additionally, they stop IVR recording. If you want IVR recording to continue after these blocks, you must add a start IVR Recording block.

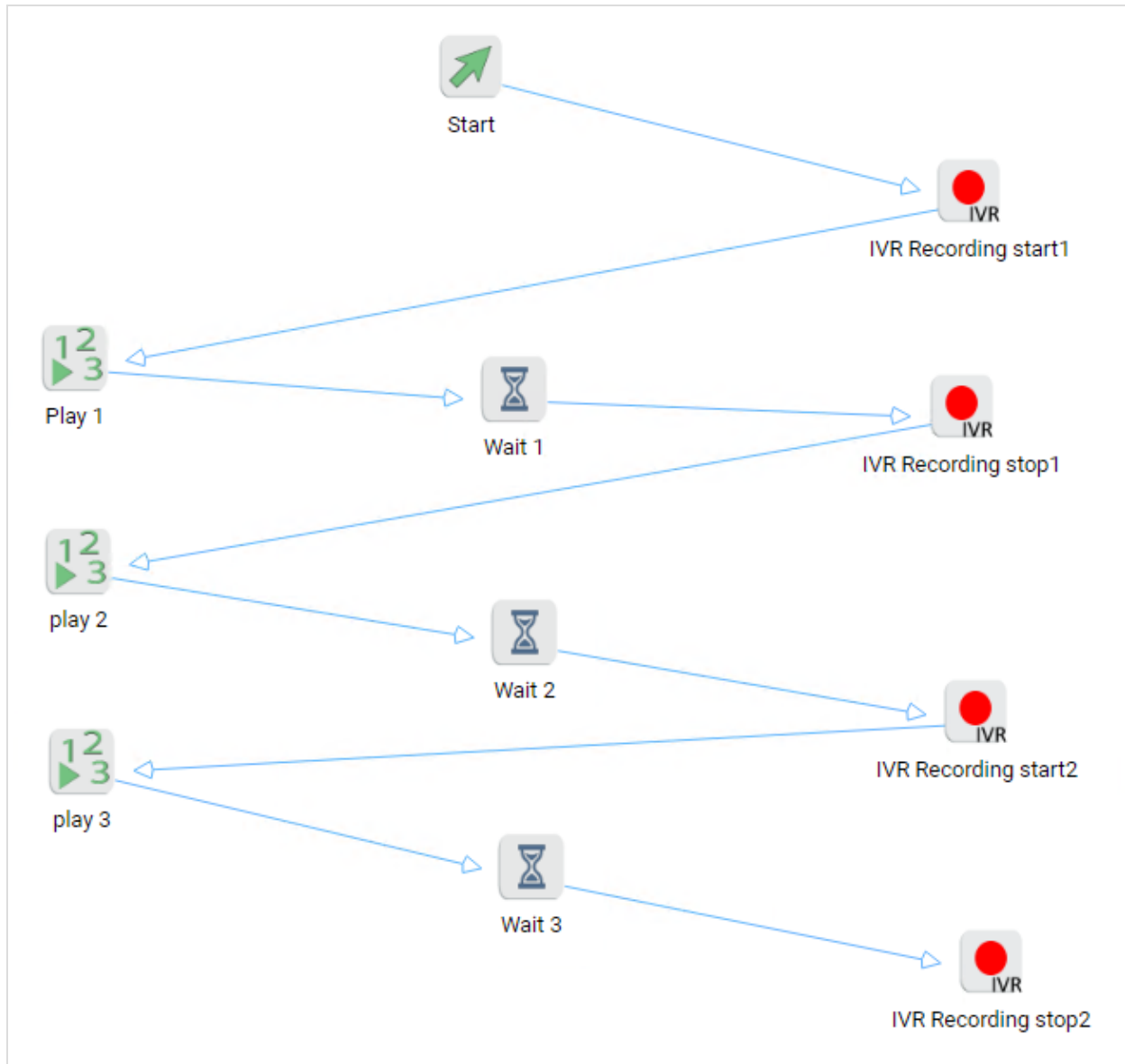
Examples

In the following example, the script records the Play 1 and Play 3 blocks, but not the Play 2 block.



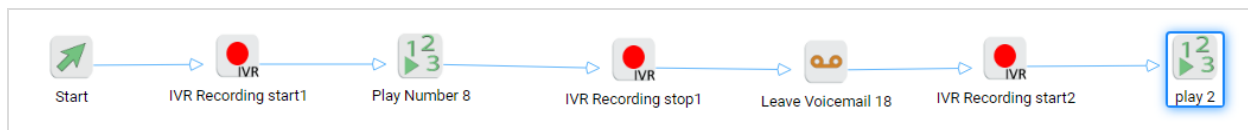
IVR recordings are precise to 1 whole second, which means that the recording of a block may start too soon or finish too early.

To handle this limitation, insert a Wait block set to 1 second before the start and stop IVR Recording blocks.



Leave Voicemail and Record Voicemail blocks are not recorded and stop the IVR recording.

If you want IVR recording to continue after these blocks, you must add a start IVR Recording block.



IVR Transfer Recording



This block records bridged calls to an external number.

It must be used after the Supervised Transfer, Transfer Call, or Bridge Call blocks.

Leave Voicemail



This block records voicemail. It replaces the deprecated [Record Voicemail](#) block.

Properties	Values	Description
Max Time	Seconds (integer)	Maximum length of recording. If left blank, the default is 300 seconds.
Termination Digits	0-9 # and *	Enter digit(s), # or *. These value(s) can be used by the caller to terminate this command. For example, to allow any touch tone to be a termination digit, set this to 1234567890#*
Clear Digits	Yes or No	Clear the digit buffer of any digits entered prior to recording. The default is No.
Beep Type	None	Type of beep to play at start of recording time, or none. If left blank, the default is 0 (None).
	Short, Normal, or Long	The filenames that correspond to the various beep lengths are: <ul style="list-style-type: none">• <i>beep_short.wav</i>• <i>beep.wav</i> (default)• <i>beep_long.wav</i>
Line	Inbound	The function is applied to the outbound call (to where the call is transferred).
	Outbound	The function is applied to the Email address of the sender.
Store to DB	Yes or No	Determines if stored in recording database. Note If you select Yes, you must also set the mandatory From and Recipient email addresses otherwise the voicemail is not stored in the database.
Store Voicemail	Folder or full file path	The folder is the valid directory path that holds the voicemail for review before sending. The folder can be empty, in which case the voicemail is not saved to a file. If a folder is specified but not the filename, the filename is assigned automatically by the system.

Properties	Values	Description
		<p>All recorded files are in MP4 format.</p> <p>Note Only the MP4 file is saved, not the call information file.</p>
ACD Voicemail	String	<p>Name of the tenant's voicemail folder. If left blank, the tenant's default voicemail folder is used (default).</p> <p>Interaction will ring at an agent as an ACD voicemail.</p> <p>If set to none and Recipient or CC properties match the rules for ACD voicemail, ACD voicemail is created with SMTP services only.</p> <p>If left blank or has a value other than none, ACD voicemail is created without SMTP services.</p> <p>Note See Examples below.</p>
Segregation Group Id	Integer	<p>Group ID used to filter voicemail items in Supervisor Message View.</p> <p>Left blank by default.</p>
From	String	<p>When sending ACD voicemail over SMTP or when sending voicemail as an email with an attachment, specify the email address of the sender.</p>
Recipients	String	<p>Email addresses of recipients to which the voicemail will be sent over SMTP.</p> <p>Multiple email addresses can be delimited by a semi-colon (;).</p> <p>If a recipient email address starts with voicemail, an ACD voicemail interaction is created and rings at an agent as a voicemail.</p> <p>If a recipient email address matches one of the tenant's email folders, an ACD email interaction is created and rings at an agent as an incoming email.</p> <p>Left blank by default.</p> <p>Note See Examples below.</p>
CC	String	<p>Email addresses of recipients that will receive a carbon copy of the voicemail being sent over SMTP.</p> <p>For additional rules, see the Recipients property.</p> <p>Note See Examples below.</p>
Subject	String	<p>When sending voicemail as an email with an attachment, specify the email subject.</p>
Body	String	<p>When sending voicemail as an email with an attachment, specify the email body text.</p>

Examples

The following table provides some examples of values for the **ACD Voicemail** and **Recipients/CC** properties and the expected behavior.

In this example, the email addresses are:

- *folder@tenant.loc* — the email address of a tenant's CCaaS folder defined in CCaaS Admin
- *voicemailFolder@tenant.loc* — the email address of a tenant's folder for incoming voicemail only
- *someone@home.loc* — the email address of a recipient external to CCaaS

The expected behavior types shown in the table are:

1. ACD voicemail is created without using SMTP services.
2. ACD voicemail is created using SMTP services and rings at an agent as an ACD voicemail interaction.
3. ACD email and a recording attachment are created using SMTP services and ring at an agent as an ACD email interaction.
4. Regular email and a recording attachment are created and sent to an external recipient using SMTP services.

Note

As you can see in the table, ACD Voicemail is disabled when:

- ACD Voicemail = **none** AND Recipient is empty AND CC is empty
- ACD Voicemail = **none** AND Recipient/CC fields specify the email address of a tenant folder or the email address of a recipient external to CCaaS

ACD Voicemail parameter	Recipient/CC parameter	Expected behavior
Left blank or set to voicemail		1
Left blank or set to voicemail	folder@tenant.loc	1, 3
Left blank or set to voicemail	voicemailFolder@tenant.loc	1, 2
Left blank or set to voicemail	someone@home.loc	1, 4
Left blank or set to voicemail	folder@tenant.loc; voicemailFolder@tenant.loc; someone@home.loc	1, 2, 3, 4
Set to none	folder@tenant.loc	3
Set to none	voicemailFolder@tenant.loc	2
Set to none	someone@home.loc	4
Set to none	folder@tenant.loc; voicemailFolder@tenant.loc; someone@home.loc	2, 3, 4

Log



This block writes a message to the event log file on the connection server. This log file usually resides on each connection server in the `%TEMP%/CosmoDesigner` directory. The actual location depends on the setting for the logging directory registry key.

Ask your CCaaS service provider for guidelines in using Log blocks in CCaaS Designer scripts.

Properties	Values	Description
Message	String	<p>Use for debugging purposes only.</p> <p>Messages starting with the prefix:</p> <ul style="list-style-type: none">• "ERROR:", "WARNING:", or "INFO:" are printed in log files and also automatically placed in the Windows Event Log.• "DEBUG:" are printed only in log files.• "TRACE:" are printed only in log files, provided that trace is enabled in CCaaS Admin. <p>Messages without one of the above prefixes are written in log files, regardless of the level set in CCaaS Admin (unless the setting is None).</p> <p>The log files are:</p> <ul style="list-style-type: none">• CCaaS Designer's dedicated log-per-call file (must be enabled in CCaaS Admin)• Syslog log file (must be enabled in CCaaS Admin) <p>Note</p> <p>Prefixes must be uppercase.</p>

Logging Control



This block turns logging on or off for the duration of the current event.

At the end of the CallTermination event, CCaaS emails the entire log file as an attachment to the designated Recipient.

Note

- If you want to turn logging on or off for the whole call, you must put a Logging Control block in every event script.
- If script logging is globally enabled and you disable logging via a Logging Control block for a certain event script, CCaaS still emails the logs of the other event scripts to the Recipient. The global script logging settings determine the verbosity and behavior of these logs.
- In the event script, logging follows the global script logging settings until CCaaS parses the Logging Control block.

Properties	Values	Description
Value	On or Off	Enable or disable logging.
From	Email address of the sender.	Must be in the form of <i>mailbox@domainname</i> Ask your CCaaS service provider for an email address to use for this purpose.
Recipient	Email address to which to send the log file.	Must be in the form of <i>mailbox@domainname</i> This can be an email address outside of the CCaaS system domain.

This block is useful for testing if a new script ran or if a new branch of an IVR script ran and the result of each block in that script after the Logging Control block.

Mask Digits



VCS only.

This block masks the DTMF digits (tones) entered by the caller so the caller's PIN digits are not entered in the VCS log.

Properties	Values	Description
Mask digits	Yes or No	Enable or disable masking of DTMF digits in IVR. The default is No.

MD5 Encryption



This block encrypts any string variable using MD5 hashing. This is useful for taking sensitive data collected via IVR and passing it to an external application.

Properties	Values	Description
URL	Web address	Web services URL, for example, <i>http://www.abc.com/soap/GetClientInformation.asmx</i>
String to Encrypt	String	The string that is to be encrypted. Note This operation is a one-way hash function, and it is not possible to de-encrypt the result variable. This can only be used to verify the integrity of information passed to the system, as the function returns the same result given the same input.
Timeout	String	Number of seconds to wait for response from the web service before timing out. If the web service does not respond within the specified timeout, the operation is aborted and a log entry is created in the CCaaS Designer log as follows: No response was received from <i>http://webserver/webservices/test.asmx</i> . Operation aborted.
Result Variable	String	A variable name where responses are being prefixed. See XML response example , for example, myVar. The result is in MD5 Format4 encryption.

Menu



VCS only.

This block specifies menu behavior.

Properties	Values	Description
Menu Prompt	Filename	The sound file to play for menu options. Enter a URL address to the file you wish to play.
Invalid Prompt	Filename	The sound file to play for an invalid choice. Enter a URL address to the file you wish to play.
No Entry Prompt	Filename	The sound file to play if the caller does not enter anything within Max Time. Enter a URL address to the file you wish to play.
Termination Digits	0-9 # and *	Enter digit(s), # or * . These value(s) can be used by the caller to terminate this command. For example, to allow any touch tone to be a termination digit, set this to 1234567890#*
Clear Digits	Yes or No	Clear the digit buffer of any digits entered before playing the audio file. The default is No.
Max Time	Integer	Maximum amount of time in seconds to wait for the entire string of digits. The default is 3. Note Typically 3 is too low a number to work reliably and this value is set to 10 or 3 x number of digits to be entered.
Max Tries	Integer	Maximum number of times the caller can respond to a prompt. The default is 1.
Result Variable	String	Name of a variable into which the digit string is to be passed. Must start with an alphabetic character and not exceed 255 characters in length. There is no default value for this, so leaving this field blank causes this block not to function properly. See How to use variables .

Properties	Values	Description
Line	Inbound	The next block in the script is applied to the original (incoming) call.
	Outbound	The next block in the script is applied to the outbound call (to where the call is transferred).


Configure the Menu block

1. Right-click the block and click **Add case**. The **Menu** dialog appears.


2. From the **Menu** list, select one or more menu items.

Note

This item is a key press option for the caller. If you select more than one digit, the case accepts any one of your selected digits if it matches caller input.

3. Optionally, in **Label**, type an alternative description.
4. Click **OK**. The **Case** branch  appears.
5. Connect the **Case** branch to the appropriate block in the script.

6. Repeat steps 1-5 to add the branches you need.

7. Connect the **Else** branch  to the appropriate block in the script.

Optional Parameters



Optional parameter values are available for the life of the call. These values are also available to the agent for use in third party integrations. Two examples of possible optional parameters are **Account_ID** and **IncidentNumber**.

For example, you can set an optional parameter in IVR named **PathToWavFiles** and use it in every [Play Audio](#) block as follows: `$CallData.Option("PathToWavFiles") $Greeting.wav` to use a set of previously created scripts without needing to edit every PlayAudio block, or to set the language of the *.wav* files.

Note

By default, optional parameters are not stored in the historical database because they would consume a very large amount of storage. If you want to store optional parameters in the historical database, estimate the additional storage requirements and request Enghouse Technical Support's assistance.

Properties	Values	Description
Parameter Name	String	Name of the parameter. Optional parameter names are case sensitive.
Parameter Value	String (Boolean number)	Value of the parameter.

Real-Time Speech Analytics

To enable the output of recording files for post call analysis using the batch-mode integration with Real-Time Speech Analytics (VoiceCoach), include an optional parameter called **EVCCconfiguration** for each call that you want processed. This parameter contains the name of the configuration to use.

The actual value does not matter to CCaaS, its existence means the DTR will process the recording and output the relevant files for Real-Time Speech Analytics. The value of this parameter is included in the resulting *.dat* files.

CCaaS UI CustomRing parameter

Note

- CCaaS UI was previously called TouchPoint.
- If an agent selects auto-answer then CCaaS does not play the ring tone. This setting is only for CCaaS UI, not IP phones or alternative devices.

A unique optional parameter is available for CCaaS UI: **CustomRing**.

Use this optional parameter to send a customized ring file location (URL) when a call is delivered to CCaaS UI. This ring file can be used to announce any desired message to an agent before the call is picked up.

CCaaS UI looks for a **CustomRing** optional parameter. This parameter should contain the URL for the desired ring file.

If this parameter is present, CCaaS UI ignores ring preferences and uses the ring file specified in the **CustomRing** optional parameter.

Percent Allocated

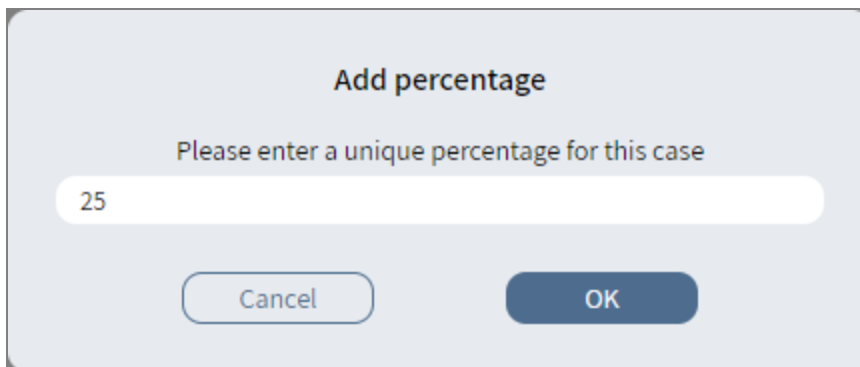


This block sets the percentage point (based, for example, on the number of calls handled) at which calls are sent along different paths.

All percent case blocks must add up to 100%.

Configure the Percent Allocated block

1. Right-click the block and click **Add Case**.
2. In the **Add percentage** dialog, enter the percentage and click **OK**.



Add percentage

Please enter a unique percentage for this case

3. Repeat steps 1-2 to add the percentages you need.

Play Audio



VCS only.

This block plays an audio (.wav) file to the caller.

Note

PlayAudio only works in IVR, IVRTimeOut, CallRejected, MusicOnHold, and MusicOnHoldWhileInQueue events.

Properties	Values	Description
Audio file	Filename	Enter a URL address to the file you wish to play.
Termination Digits	0-9 # and *	Enter digit(s), # or *. These value(s) can be used by the caller to terminate this command. For example, to allow any touch tone to be a termination digit, set this to 1234567890#*
Clear Digits	Yes or No	Clear the digit buffer of any digits entered before playing the audio file. This should be set to No unless digits from a previous Get Digits block need to be cleared. Since callers often become familiar with your IVR and enter the digits for 2 or 3 prompts ahead, if you clear the digits you delete the entries for the next one (or more) Get Digits block(s).
Line	Inbound	The .wav file is played to the original (incoming) caller.
	Outbound	The .wav file is played to the transferee (the party to whom the call is transferred).
Async Mode	Yes or No	Async Mode allows the script to do other tasks, for example, to verify a customers ID number from a database, while a .wav file plays. The default is No. Caution Asynchronous playback should only be used for one .wav file at a time. Attempting to use Async Mode for multiple simultaneous .wav files can produce unpredictable results.

Play Date



VCS only.

This block plays a specific date to the caller.

Properties	Values	Description
Date	Digits only, no text.	<p>The date to play, as specified in the Date Format property. Must be in the same format as specified in the Date Format property.</p> <p>Typically, the value of a variable set earlier in the script is entered into this field, for example, <code>\$TransactionDate\$</code>.</p> <p>Note</p> <p>Entering <code>\$Now\$</code> does not work since it gives the date and time and the date format does not match any of the date formats for this block. To use Now, change the value to match the chosen date format.</p> <p>For example, if Date Format = yymmdd, this is a 6-digit number, such as 010130</p>
Termination Digits	0-9 # and *	<p>Enter digit(s), # or *. These value(s) can be used by the caller to terminate this command.</p> <p>For example, to allow any touch tone to be a termination digit, set this to 1234567890*#</p>
Clear Digits	Yes or No	Clear the digit buffer of any digits entered prior playing the date. This should be set to No by default.
Date Format	yyyyymmddhhnn yyyyymmdd yymmddhhnn yymmdd mmddhhnn mmddhhnnss hhnnss hhnn	<p>Format in which the output date is spoken, where:</p> <ul style="list-style-type: none"> • yyyy = four-digit year • yy = two-digit year • mm = two-digit month • dd = two-digit day • hh = two-digit hour • nn = two-digit minute • ss = two-digit seconds <p>Note</p> <p>Default year: If yy and yyyy are omitted, the current year is played.</p>
Line	Inbound	The .wav file is played to the original (incoming) caller.

Properties	Values	Description
	Outbound	The <i>.wav</i> file is played to the transferee (the party to whom the call is transferred).
Async Mode	Yes or No	<p>Allow asynchronous playback of a <i>.wav</i> file. The default is No.</p> <p>Caution</p> <p>Asynchronous playback should only be used for one <i>.wav</i> file at a time. Attempting to use Async Mode for multiple simultaneous <i>.wav</i> files can produce unpredictable results.</p>

Note

The Play Date, Play Money, and Play Number blocks use prerecorded *.wav* files to read the date, dollar amount or number to the caller. These files are provided in English only but the service provider may have recordings in other languages created by outside companies.

Play Greeting



This block plays a message to the caller and to the agent.

Properties	Values	Description
Greeting	Text	Path to <i>.wav</i> file to play or text to speech. If Agent Greeting is blank, both the agent and customer hear this message.
Agent Greeting	Text	Optional path to <i>.wav</i> file to play or text to speech presented only to the agent before answering the call, for example, <i>Incoming call from Sales</i> . Note <ul style="list-style-type: none">• To use Agent Greeting, you must also have a caller Greeting.• To avoid either party hearing silence, we recommend that the Greeting and Agent Greeting files are the same duration.
Async Mode	Yes or No	Async Mode allows the script to do other tasks, for example, to verify a customer's ID number from a database, while a <i>.wav</i> file plays. The default is No. Note <ul style="list-style-type: none">• If you use Async Mode, to ensure that CCaaS plays the complete messages to the relevant parties, we recommended that you follow this block with an Agent Greeting block in Sync Mode that plays media files that last a few milliseconds. Otherwise, CCaaS plays neither message because it executes the next block and connects the call.• If you don't use Async Mode, both messages will finish before the script moves to the next block.
Comment	Text	Optional text you can enter to describe what this block does in your script.

Play Money



VCS only.

This block plays a specific dollar or dollar-and-cents amount to the caller.

Properties	Values	Description
Dollar	ddd or ddd.cc	A monetary amount to play. In dollars only or dollars-and-cents format.
Termination Digits	0-9 # and *	Enter digit(s), # or * . This value(s) can be used by the caller to terminate this command. For example, to allow any touch tone to be a termination digit, set this to 1234567890#*
Clear Digits	Yes or No	Clear the digit buffer of any digits entered prior to this command. This should be set to no by default.
Line	Inbound	The <i>.wav</i> file is played to the original (incoming) caller.
	Outbound	The <i>.wav</i> file is played to the transferee (the party to whom the call is transferred).
Async Mode	Yes or No	Allow asynchronous playback of a <i>.wav</i> file. The default is No. Caution Asynchronous playback should only be used for one <i>.wav</i> file at a time. Attempting to use Async Mode for multiple simultaneous <i>.wav</i> files can produce unpredictable results.

Note

The [Play Date](#), [Play Money](#), and [Play Number](#) blocks use prerecorded wav files to "read" the date, dollar amount or number to the caller. These files are provided in English only but the service provider may have recordings in other languages created by outside companies.

Play Number



VCS only.

This block plays a specific number to the caller.

Properties	Values	Description
Number	Number	The number to play.
Termination Digits	0-9 # and *	Enter digit(s), # or *. These value(s) can be used by the caller to terminate this command. For example, to allow any touch tone to be a termination digit, set this to 1234567890#*
Clear Digits	Yes or No	Clear the digit buffer of any digits entered prior to playing the number. This should be set to no by default.
Format	Number	Play digits as cardinal number (123 = "one hundred and twenty-three")
	Ordinal	Play digits as ordinal number (123 = "one hundred and twenty-third")
	Digits	Play digits individually (123 = "one, two, three"). This format is ideal for playing back a number entered by the caller. For example, if the number 51389 was entered, the first Play Audio block can play "the customer ID you entered was" and then a Play Number block can play "five, one, three, eight, nine".
Line	Inbound	The <i>.wav</i> file is played to the original (incoming) caller.
	Outbound	The <i>.wav</i> file is played to the transferee (the party to whom the call is transferred).
Async Mode	Yes or No	Allow asynchronous playback of a <i>.wav</i> file. The default is No. Caution Asynchronous playback should only be used for one <i>.wav</i> file at a time. Attempting to use Async Mode for multiple simultaneous <i>.wav</i> files can produce unpredictable results.

Note

The [Play Date](#), [Play Money](#), and Play Number blocks use prerecorded *.wav* files to read the date, dollar amount or number to the caller. These files are provided in English only but the service provider may have recordings in other languages created by outside companies.

Play Text



This block plays text to the caller.

Properties	Values	Description
Text to play	Text or Speech Synthesis Mark Up Language (SSML). Text containing optional variables.	Text containing optional variables. A variable has a format similar to the one in this example: "The number < phone > \$line.DNIS\$ is unavailable." For more information on SSML, see http://www.w3.org/TR/ssml-sayas/ .
Speech Rate	Slow, Medium (default) or Fast	-
Volume	Soft, Medium (default) or Loud	-
Termination Digits	0-9 * and #	Numeric digit(s), # or * . The caller can use these value(s) to indicate the end of their response. When a termination digit is pressed, playback stops, and the script executes the next block.
Clear Digits	Yes or No	Clear the digit buffer of any digits entered before playing the text file. The default is No. This should be set to No unless digits from a previous Get Digits block need to be cleared. Since callers often become familiar with your IVR and enter the digits for 2 or 3 prompts ahead, if you clear the digits, you delete the entries for the next one (or more) Get Digits block(s).
Line	Inbound	The text file is played to the original (incoming) caller. The default is Inbound.
	Outbound	The text file is played to the transferee (the party to whom the call is transferred).

Properties	Values	Description
Async Mode	Yes or No	Async Mode allows the script to do other tasks, for example, to verify a customer's ID number from a database, while a text file plays. The default is No. Caution Asynchronous playback should only be used for one text file at a time. Attempting to use Async Mode for multiple simultaneous text files can produce unpredictable results.
Comment	Text	Optional description of what this block does in the script.

Play Video



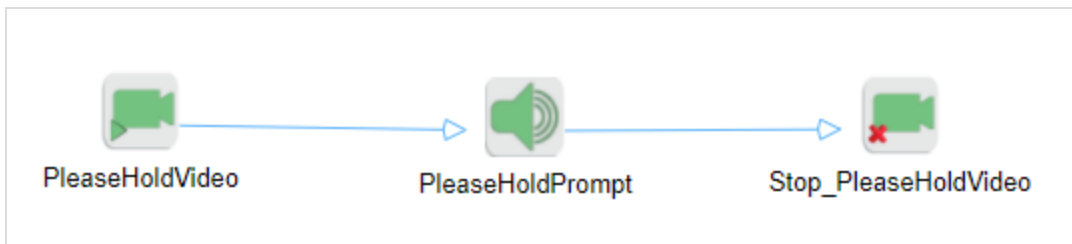
This block plays a video file to the caller.

Properties	Values	Description
Video File	Filename	This must be a name for a local video file, with or without a path.
Termination Digits	0-9 # and *	N/A
Clear Digits	Yes or No	N/A
Async Mode	Yes or No	N/A (Yes by default)

Implementation

This block delivers *only* a video file to the recipient. In addition, the block delivers control to the next block in the script when it has finished executing. This is intentional. To implement this feature properly, the following blocks must follow this block in the script:

- A [Wait](#) or [Play Audio](#) block — this block must have the same duration as that of the Play Video block. This is to ensure synchronization of the playback of the Play Video block.
- A [Stop Video](#) block — this block must be inserted after the Wait or Play Audio block to ensure that all the blocks end at the same time.



Playback Audio



Elsbeth integration only.

This block plays an audio (.wav) file to the caller.

Properties	Values	Description
Audio file	Filename	Enter a URL address to the file you want to play. Note For the Elsbeth integration scripts , leave blank. CCaaS takes the URL of the file to play from the campaign parameters configured in CCaaS Admin.
Termination Digits	0-9 # and *	Enter digit(s), # or *. These value(s) can be used by the caller to terminate this command. For example, to allow any touch tone to be a termination digit, set this to 1234567890#*
Clear Digits	No or Yes	Clear the digit buffer of any digits entered before playing the audio file. This should be set to No unless digits from a previous Get Digits block need to be cleared. Since callers often become familiar with your IVR and enter the digits for 2 or 3 prompts ahead, if you clear the digits you delete the entries for the next one (or more) Get Digits block(s).
Line	Inbound	The .wav file is played to the original (incoming) caller.
	Outbound	The .wav file is played to the transferee (the party to whom the call is transferred).
	Bi-Directional	The .wav file is played to the original caller and the transferee.

Queue Name



This block directs the call to a specific queue (callSetName).

Properties	Values	Description
Queue name	String	<p>The name of the queue to which the current call is to be directed.</p> <p>Note</p> <p>Queue names should not be longer than 32 characters.</p> <p>This must be a valid queue name for the tenant, so spelling is critical but queue names are not case sensitive.</p> <p>You can select the queue name from a drop-down list of available queues in the tenant. This list is refreshed every time you select the block. If a new queue is added while you are editing the block, to see the added queue, select another block or select the Flow editor background and then reselect the Queue block.</p> <p>If a queue is removed from the tenant and the value is currently being used as the queue name, the value remains as the selected value, until you manually change it. You are not restricted to the values provided in the drop down list, so you can use an expression as the selected value.</p>

Read SIP Header



This block reads SIP headers inside the INVITE message of incoming calls.

The block is allowed only in the Ivr script.

Properties	Values	Description
Header Name	String	The name of the SIP header. Note <ul style="list-style-type: none">• The name must start with an uppercase letter as in the SIP message and be exclusive, for example From, To, Via.• Do not include a colon (:).
Header Value	String	The name of the variable to return the SIP header.

Record Audio

VCS only



This block captures a voice message left by the caller.

Properties	Values	Description
File Name	Filename	<p>The name of the file that holds the recording. This can be either a fully-qualified filename, for example, <i>C:\MyFiles\foo.wav</i>, or just a filename, for example, <i>foo.wav</i>.</p> <p>Use a unique or temporarily unique name so the script has time to send the email with the wav file attached before another wav file is recorded with the same name, for example: <code>message\$line.lineID\$.wav</code> creates a wav file named <i>message23.wav</i> for line 23 on a VCS and is not overwritten until another caller on line 23 of that VCS records a <i>.wav</i> file. See How to use variables.</p> <p>If it is a fully-qualified name, the file is not automatically deleted.</p> <p>If it is not a fully-qualified filename, then the file is placed in the directory pointed to by the CosmoDesignerTempDir registry key and is automatically deleted at the end of the call.</p> <p>All recorded files are in <i>.wav</i> format, 8KHz – 16 bit mono.</p> <p>If the filename specified does not contain the <i>.wav</i> extension, this extension is automatically appended to the filename.</p>
Max Time	Seconds (integer)	Maximum length of recording. If left blank the default is 300 seconds.
Max Silence	Seconds (integer)	Maximum amount of time (silence) indicating the end of the recording. If left blank the default is 3 seconds.
Termination Digits	0-9 # and *	<p>Enter digit(s), # or *. These value(s) can be used by the caller to terminate this command.</p> <p>For example, to allow any touch tone to be a termination digit, set this to <code>1234567890#*</code></p>
Clear Digits	Yes or No	Clear the digit buffer of any digits entered prior to recording. This should be set to No by default.
Beep Type	None	<p>Type of beep to play at start of recording time, or none.</p> <p>If left blank, the default is 0 (None)</p>

Properties	Values	Description
	Short, Normal, or Long	The filenames that correspond to the various beep lengths are: <ul style="list-style-type: none"> • <i>beep_short.wav</i> • <i>beep.wav</i> • <i>beep_long.wav</i>
File Size	String	The name of the variable that holds the resulting file size (in bytes) of the recording. This can be used to determine if this <i>.wav</i> file should be sent or not. For example, set File Size to VMSize and then compare VMSize to 40000 (2.5 seconds). If recording is < 40000 (2.5 seconds) then do send voicemail (email with <i>.wav</i> file attachment). Note The audio format for recording is 8k – 16 bit, so a one-second recording is equal to 16000. See How to use variables .
Line	Inbound	The function is applied to the original (incoming) call.
	Outbound	The function is applied to the outbound call (to where the call is transferred).

Record Video



This block captures a video message left by the caller.

Properties	Values	Description
File Name	Filename	The name of the file that holds the recording. This can be either a fully-qualified filename, for example, <i>C:\MyFiles\foo.avi</i> , or just a filename, for example, <i>foo.avi</i> . If the specified filename does not contain the <i>.avi</i> extension, this extension is automatically appended to the filename.
Max Time	Seconds	Maximum length of recording. The default is 30.
Termination Digits	0-9 # and *	This value(s) can be used by the caller to terminate this command. For example, to allow any touch tone to be a termination digit, set this to <i>1234567890*#</i>
Clear Digits	Yes or No	Clear the digit buffer of any digits entered prior to recording. The default is No.

Record Voicemail

Caution

This block is deprecated. Instead, use the [Leave Voicemail](#) block.



This block records voicemail.

Properties	Values	Description
Max Time	Seconds (integer)	See the corresponding property in the Leave Voicemail block.
Termination Digits	0-9 # and *	See the corresponding property in the Leave Voicemail block.
Clear Digits	Yes or No	See the corresponding property in the Leave Voicemail block.
Beep Type	None, Short, Normal, or Long	See the corresponding property in the Leave Voicemail block.
Line	Inbound or Outbound	See the corresponding property in the Leave Voicemail block.
From	String	This property is deprecated.
Recipient	String	This property is deprecated. In the Leave Voicemail block, this property is mapped to ACD Voicemail Folder Name or External Mailbox Address. Note <ul style="list-style-type: none">• If the value (email address) starts with voicemail, it is mapped to ACD Voicemail Folder Name. Otherwise, it is mapped to External Mailbox Address.• Recipient has a mapping precedence over CC\BCC. The CC\BCC properties are ignored.
CC	String	This property is deprecated. In the Leave Voicemail block, this property is mapped to ACD Voicemail Folder Name or External Mailbox Address. Note

Properties	Values	Description
		<ul style="list-style-type: none"> If the value (email address) starts with voicemail, it is mapped to ACD Voicemail Folder Name. Otherwise, it is mapped to External Mailbox Address. CC has a mapping precedence over BCC. The BCC property is ignored.
BCC	String	<p>This property is deprecated.</p> <p>In the Leave Voicemail block, this property is mapped to ACD Voicemail Folder Name or External Mailbox Address.</p> <p>Note</p> <p>If the value (email address) starts with voicemail, it is mapped to ACD Voicemail Folder Name. Otherwise, it is mapped to External Mailbox Address.</p>
Subject	String	This property is deprecated.
Body	String	This property is deprecated.
Store to DB	Yes or No	See the corresponding property in the Leave Voicemail block.
Store Voicemail	Folder or full file path	See the corresponding property in the Leave Voicemail block.

Recorder



This block determines how calls are recorded.

For more information on recording, see [CallData](#) object.

Note

- You cannot use this block in the [CallAnswered](#) script.
- For email calls, agents can only control the screen recording. They cannot control the automatic email recording by the server.
- CCaaS UI was previously called TouchPoint.

Properties	Field type	Values	Description
Logging	String	On	The entire call is recorded.
		Off	The call is not recorded unless the agent initiates the recording.
Screen Recording	String	On	When a call is recorded, the screens are also captured.
		Off	When a call is recorded, the screens are not recorded.
Agent Control (recording)	String	Allow	The Record button on CCaaS UI is enabled when an agent is in call. The agent can start recording by clicking this button. See the Rules table .
		Disallow	The Record button on CCaaS UI is disabled. The agent cannot start a recording by clicking this button. See the Rules table .
Logging Indicator	String	Show	The Record button on CCaaS UI displays an indication if a call is being recorded.
		Hide	The Record button on CCaaS UI does not indicate if a call is being recorded.

When an agent receives a call, he or she may or may not be able to start recording, depending upon how the values are set. See [Rules table](#).

Note

If the Record button is added to the CCaaS UI menu bar, the agent is only able to start recording if Logging is set to Off, Agent Control is set to Allow and Logging Indicator is set to Show. If Logging is set to On, the agent cannot stop the recording.

Rules table

The logging rules are as follows.

Logging	Agent control	Logging indicator	Effect
Off	Disallow	Hide	A call is not recorded from the beginning. The Record button is not shown. Agents cannot control recording. Recording is off completely.
Off	Disallow	Show	A call is not recorded from the beginning. The Record button is not shown. Agents cannot control recording.
Off	Allow	Hide	A call is not recorded from the beginning. The Record button is shown. Agents can control recording by clicking the Record button.
Off	Allow	Show	A call is not recorded from the beginning. The Record button is shown. Agents can control recording by clicking the Record button.
On	Disallow	Hide	A call is recorded from the beginning. The Record button is not shown. Agents cannot control recording.
On	Disallow	Show	A call is recorded from the beginning. The Record button is shown. Agents cannot control recording.
On	Allow	Hide	A call is recorded from the beginning. The Record button is not shown. Third-party applications control recording.
On	Allow	Show	A call is recorded from the beginning. The Record button is shown. Agents cannot control recording. Clicking the Record button has no effect.

Remove Skill



This block removes a skill from a call.

Properties	Values	Description
Skill Name	String	The name of the skill to be removed from the current call. If you want to remove all skills enter the word all in this field.

ReQueue



VCS only.

This block resubmits the call to the system by resubmitting the call request to the Call Management Service.

Note

UnQueue and ReQueue should only be used for the Music on Hold While In Queue event. See the Note for [Unqueue](#).

See also [unQueue](#) and [reQueue](#).

Routing Mode



This block determines how each call in queue is routed when *all* the agents that meet the call parameters are Logged off, Unavailable, or Busy.

All the suitable agents must be in the same state, for example, all the suitable agents must be Logged off. This block is ignored if the agents are in a combination of the above states.

Properties	Values	Description
Logged Off	Default, Queue or Reject	Controls how the call is routed when all the suitable agents are logged off: <ul style="list-style-type: none">• Default — reject or queue the call as configured in the registry setting.• Queue — queue the call until an agent becomes available.• Reject — reject the call immediately.
Unavailable	Default, Queue or Reject	Controls how the call is routed when all the suitable agents are in an unavailable state. For a description of the values, see the Logged Off property.
Busy	Default, Queue or Reject	Controls how the call is routed when all the suitable agents are in a busy state. For a description of the values, see the Logged Off property

Note

- The default values are set in the `AcceptOnlyIfThereAreActiveAgents` and `RejectIfBusy` registry values in `HKLM\SYSTEM\CurrentControlSet\Services\CCUCallManagementServer\Parameters`.
- The value `Route` is obsolete.

Routing Properties



This block is not currently enabled except for a workaround for DID extension. For example, the workaround adds a skill, for example, `ext: 2155` to the call, the agent must have his/her DID set to 2155 and must be servicing the queue of the call and have all other skills assigned to the call.

Run



This block runs a VBScript subroutine or function and receives a return value from it.

The following functions are restricted and cannot be accessed with the Run block:

- CreateObject
- Eval
- Execute
- Function
- GetObject
- GetRef
- InputBox
- Kill
- LoadPicture
- MsgBox
- RGB
- SetLocale
- Stop
- Sub

Property	Values	Description
Return Value	Variable name (string)	Name of Return Value. See How to use variables .
Routine		Name of function or subroutine to execute.

Schedule Callback



This block schedules a callback request. It replaces the deprecated [Send callback](#) block.

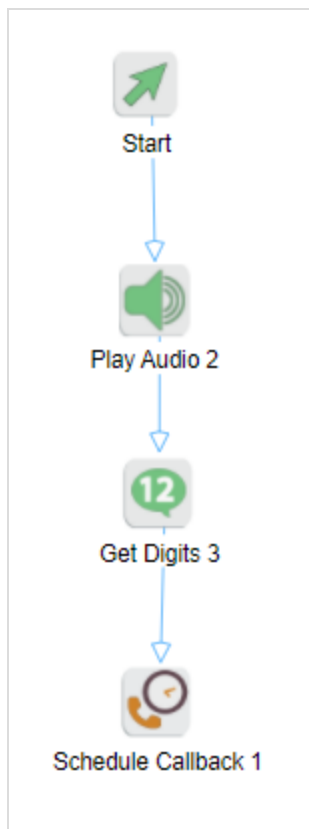
Properties	Values	Description
Callback Destination	String	The phone number to which the callback should be made. Generally would be <code>\$line.ani\$</code> but can be from a Getdigits block. Note The maximum outgoing call number or SIP URI is 370 characters.
ACD Callback Folder	String	Name of the tenant's callback folder. If left blank, the tenant's default callback folder is used (default). Note This property is for future use when more than one callback folder per tenant is available.
Destination Agent Id	Integer	Enable routing the callback to a specific agent, using Agent ID. Left blank by default (don't route the callback to a specific agent). Note If a personal callback is scheduled without the Segregation Group Id, the Destination Agent Id is removed, an error is logged, and the callback is treated as a non-personal callback and routed to the next suitable agent.
Immediate Dialing	Yes or No	Dial out immediately when the call is answered. The default is No.
Segregation Group Id	Integer	Group ID used to filter voicemail items in Supervisor Message View. Left blank by default. Note Personal callbacks scheduled without the Segregation Group Id are treated as non-personal callbacks.
Callback Time	String	The time or day to call back in the format: Month/Day/Year Hour:Minutes AM/ PM, for example, 1/18/2018 6:05 PM. For systems outside the United States, the date is usually Day/Month/Year and the time format is in accordance. To test, configure a CallerMessage block to <code>\$Now\$</code> and get the current date and time. This information should be obtained from the caller, from a Menu

Properties	Values	Description
		block or from a Getdigits block. The default is \$Now\$

Note

When rescheduling a new callback from callback calls (callback media) and outgoing callback calls (VoIP media), the RescheduleCounter value of the newly scheduled callback is greater by 1 than the value in the underlying call. When scheduling a callback from any other call and media type, the RescheduleCounter value is set to 0 for the scheduled callback.

The following example shows how to use this block.



Script Entry Point



This block defines an entry point to a script that is not the start point. You can pass a single parameter to the script through the **Variable Name** property of this block.

Properties	Values	Description
Variable Name	String	The parameter to pass to the script. This field is Required and the block will abort the script if one is not entered. This variable does not have to be used in the script in any form.

Note

- You can pass a string that is parsed by your script through the Variable Name parameter. That way you can pass multiple parameters to your script.
- Make sure that the Script Entry Point and the script in which you insert the Script Entry Point are consistent with the entry point established in CCaaS Admin.
- To be able to use the parameter defined through the Variable Name property, the Parameter Accepted option of the Entry Points dialog of CCaaS Admin must be enabled. If the option is not enabled, the Variable Name property is not available.

For more information, see *entry point* in the *CCSP Tenant Administrator Guide*.

Send Callback

Caution

This block is deprecated. Instead, use the [Schedule Callback](#) block.

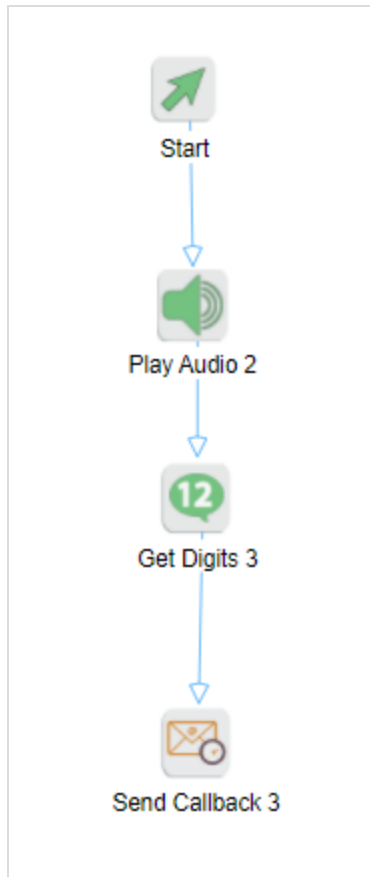


This block sends a Callback request to an agent.

Properties	Values	Description
From	String	Email address of the sender. This is typically the service account for the contact center. This can be the actual email address of the caller.
Recipient	String	Email address of recipient(s). Multiple addresses separated by a semicolon (;). These generally are prepopulated, but can be elicited through a Menu.
CC	String	Email address of carbon copy recipient(s). Multiple addresses separated by a semicolon (;) These generally are prepopulated, but can be elicited through a Menu.
BCC	String	Email address of blind carbon copy recipient(s). Multiple addresses separated by a semi colon (;). These generally are prepopulated, but can be elicited through a Menu.
Subject	String	Subject of the email message. This should be prepopulated, but a choice can be given through the use of a Menu.
Body	String	Since this a request for a callback, this should be prepopulated with some generic text.
Attachment(s)	String	For use in third-party integrations.
Callback Time	String	The time or day to call back. This is based on the local time of the connection Server: ICS, MCS or VCS and so must match the format, typically Month/Day/Year Hour:Minutes AM/ PM, for example, 1/18/2003 6:05 PM. For systems outside the United States, the date is usually Day/ Month/Year and the time format is in accordance. To test, configure a Caller Message block to <code>\$(Now)</code> and get the current date and time. This information should be obtained from the caller, from a Menu block or from a Get Digits block.

Properties	Values	Description
Callback Number	String	The phone number to which the callback should be made. Generally would be <code>\$line.ani\$</code> but can be from a Getdigits block. Note The maximum outgoing call number or SIP URI is 370 characters.

The following example shows how to use this block to elicit a callback number and time from an after-hours caller.



Send Email



This block sends email to a valid recipient.

Properties	Values	Description
From	String	Email address of the sender.
Recipient	String	Email address of recipient(s). Multiple addresses separated by a semicolon (;).
CC	String	Email address of carbon copy recipient(s). Multiple addresses separated by a semicolon (;).
BCC	String	Email address of blind carbon copy recipient(s). Multiple addresses separated by a semicolon (;).
Subject	String	Subject of the email message.
Body Format	Html or Plain text	Determines the format of the body of the email message. The default value is Html.
Body	String	<p>Body of the email message, in valid HTML format, unless the Body Format value is plain text.</p> <p>To embed variables into this property, surround the variable's name with percent (%) characters. To display a percent character in the body, use two percent characters together (%%).</p> <p>For example, if you have a variable called average, use a string like this:</p> <pre>The class average for the last test was %average% %%.</pre> <p>The email body will then contain something like:</p> <pre>The class average for the last test was 85%.</pre> <p>See How to use variables.</p>
Attachment (s)	String	<p>Filename(s) of attachment(s) to include. Multiple attachments separated by a semicolon (;).</p> <p>The default directory is the same as that for the Record block, so for Voicemail the name of the file must match the name set in the Record block. If a specified path is used in the Record block, then the same path must be used in the attachment field of the SendEmail block.</p>
Send On Error	Yes or No	If set to Yes the email is sent even if an attachment fails. If set to No the email is not sent if an attachment fails.
Result	String	Enter a variable name to hold the return result of the send email. The value is

Properties	Values	Description
		<ul style="list-style-type: none"> • 0 if successful • 1 if an attachment failed to be loaded • 2 if the send email failed • 3 for an unknown error
Minimum File Size	Number	<p>If set, the email is not sent unless the attachment file size is equal or greater than this value. This value is in bytes.</p> <p>Note</p> <p>For CCaaS wav files, 1 second equals 16000 bytes ex. Setting this field to 40000 would cause emails with attached <i>.wav</i> files that are 2.5 seconds or less not to be sent.</p>

Send Message



This block displays a message on the caller side.

Properties	Values	Description
From	String	<p>Address of the sender. The default is System.</p> <p>Note:</p> <ul style="list-style-type: none">• This value must not contain the following characters: dollar (\$), percent (%), less than (<), more than (>), open bracket ([), closed bracket (]). The system removes these symbols from this property.• It can contain all other symbols including UNICODE symbols.• It can be an empty string.
Message Text	String	<p>The message text.</p> <p>Note:</p> <ul style="list-style-type: none">• The value must not be an empty string.• If the value is empty, the block is skipped and the next block in the script is executed. <p>To embed variables into this property, surround the variable's name with percent (%) characters. To display a percent character in the body, use two percent characters together (%%).</p> <p>For example, if you have a variable called average, use a string like this: The class average for the last test was %average% %%. The message will then contain something like: <i>The class average for the last test was 85%.</i></p> <p>See How to use variables.</p>

Send User Data



This block sends a SIP INFO message.

Properties	Values	Description
Format	String or blank	The format string defines the content type of the INFO message. The default is blank. If the format is blank, then VCS inserts <code>application/ccu-integration</code> as the content type of the INFO message.
Line	String	Inbound /outbound line on which you want to send the INFO
User Data	String	The User Data is the body of the INFO message.

Set Defer Type



This block determines whether the deferred item is personal or shared.

Properties	Values	Description
DeferType	Personal or Shared	Determines whether the deferred item is personal or shared. Personal items can only be seen by the deferring agent and his supervisor. Shared items can be seen by all members of the agent's group. The default value is the defer type chosen by the agent when deferring the item. This value can be overridden by the Defer script.

Set Event



This block subscribes to one of these events:

- On Connected
- On Disconnected
- On Digits

This block must be used with the [Wait for Event](#) block.

Properties	Values	Description
Event	On Connected	When the call is first established.
	On Disconnected	When an outbound call disconnects. (Inbound calls have the Call Termination event associated with them.)
	On Digits	When a specific digit or digit string is entered by the caller. Uses Data property to specify digits.
Data	Digit or digit string	This can only be used if On Digits is specified, to check for the entry of a specific digit or digit string.
Line	Inbound	The function is applied to the original (incoming) call.
	Outbound	The function is applied to the outbound call (to where the call is transferred).
Goto	Works the same as the GoTo block	What to do (where to go to) when this event fires. See GoTo .

Use the Set Event and Wait for Event blocks as follows:

1. Use a Set Event block with Event set to On Disconnected.
2. Use a Supervised Transfer block to make an outbound call.
3. Bridge the outbound call with the original call.
4. Use a Wait for Event block to offer new actions to the original caller only after the second call disconnects.

Set Language



This block is only available if Integrated Speech IVR is enabled.

This block sets the language of prompts for Text to Speech (TTS) synthesis and grammar for Automated Speech Recognition (ASR). See the [Speech Play and Collect](#) block for a list of the available grammars.

This block can be used anywhere within a script. Therefore, it is possible to change the language of a script while it is running.

Properties	Values	Description
IVR Language	String	Type a command string provided to you by your system administrator. For example, MRCP=<language_code1> The default code for <language_code1> is en-US for American English.
Comment	Text	Optional description of this block in your script.

Set SIP Header



This block adds a custom SIP header to the INVITE message of an outgoing call or the outgoing leg part of the call.

The custom SIP header starts with the prefix **X-**

If the Header Name value already starts with **X-**, it is not changed. If it starts with **x-**, it is changed to uppercase (for example **x-abc** becomes **X-abc**).

If the Header Name value does not start with either of the above, it is prefixed with **X-** (for example **abc** becomes **X-abc**).

To use, add this block to the Outgoing Call script or to a custom script that is pointed to from the Outgoing call script by a [GoTo](#) block. More than one block can be used in the same script. For each block used, CCaaS Designer adds the appropriate header.

Properties	Values	Description
Header Name	String	The name of the added custom SIP header. Note <ul style="list-style-type: none">The name must contain only alphanumeric characters and hyphens (-). The maximum length is 30 characters (including the prefix).It must not be an empty string or just the prefix.
Header Value	String	The value of the added custom SIP header. Note <ul style="list-style-type: none">The value must contain only alphanumeric characters and the following characters: space, comma, hyphen (-), underscore (_), semi-colon (;), plus sign (+), double-quotes ("), and forward slash (/). The maximum length is 200 characters.It must not be an empty string.

Speech Enabled Menu



This block is only available if Integrated Speech IVR is enabled.

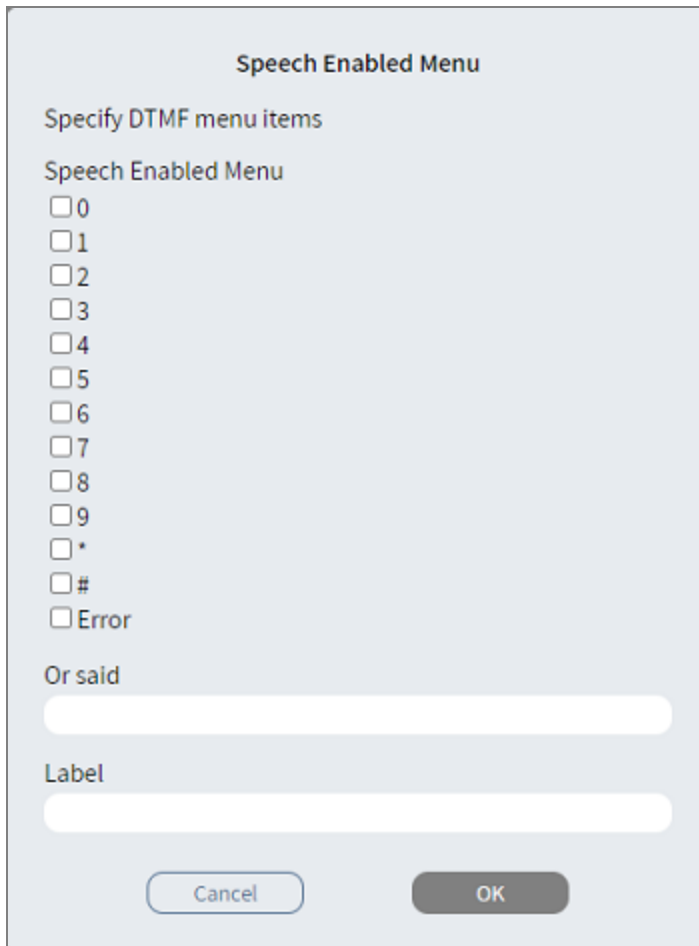
This block is similar to the Menu block with the addition of Text to Speech (TTS) and Automated Speech Recognition (ASR) speech capability. It is also an extension of the [Speech Play and Collect](#) block.

Properties	Values	Description
Menu Prompt	String	The sound file to play for menu options. Enter a <i>.wav</i> file, plain text, or Speech Synthesis Markup Language (SSML) string. For a <i>.wav</i> file, do one of the following: <ul style="list-style-type: none">• Enter a complete URL address to the file you wish to play.• If your system administrator created a shortcut to the <i>.wav</i> files, enter <code>/ [file_name .wav]</code>.
Invalid Prompt	String	The sound file to play for an invalid choice. See the Menu Prompt description.
No Entry Prompt	String	The sound file to play if the caller does not enter anything within Max Time (for speech response). See the Menu Prompt description.
Stop Prompt On	Start of Input, Recognition Complete, or Don't Stop	This property determines if caller barge-in is enabled or disabled: <ul style="list-style-type: none">• Start of Input — the prompt is stopped as soon as the user starts to speak or presses a key on the phone keypad.• Recognition Complete — the prompt is stopped only after recognition is achieved according to the configured grammar. This is also known as Hot Word mode.• Don't Stop — no barge in during prompt.
Confidence Level (optional)	Percent	This is a confidence threshold parameter. The speech recognizer computes a confidence level. If the level is below the threshold you set, the recognizer returns no-match as the recognition result. If the level is at or above the threshold, then recognition is successful. The default is 50.
Speech Rate	Slow, Medium, or Fast	The rate a TTS message is read.

Properties	Values	Description
Volume	Soft, Medium, or Loud	The volume level a TTS message is read.
Termination Digits	0-9 # and *	Enter digit(s), # or * . These value(s) can be used by the caller to terminate this command. When a termination digit is pressed, playback stops, and the script executes the next block.
Clear Digits	Yes or No	Clear the digit buffer of any digits entered before playing the audio file. The default is No.
Max Time	Seconds	Maximum waiting time. The length of time the script waits for caller response after the prompt completes. The default is 20 and the maximum permitted value is 30. This property applies only to Automatic Speech Recognition (ASR).
Max Silence	Seconds	The maximum length of time to wait for the caller's next response. The default is 5.
Max Tries	Integer	Maximum number of times the caller can respond to a prompt. The default is 1.
Result Variable	String	Name of a variable into which the digit string is to be passed. Must start with an alphabetic character and not exceed 255 characters in length. There is no default value for this, so leaving this field blank causes this block not to function properly. See How to use variables .
Input Mode	Speech and DTMF, Speech only, or DTMF only	If you select Speech and DTMF, and if the caller presses a key (DTMF), the system ignores anything the caller says after the key press. If you select Speech and DTMF, and if the caller speaks, the system ignores any key press after anything the caller says. The default is Speech and DTMF.
Close MRCP Session	Yes or No	Close the MRCP session after leaving the MRCP block.
Comment	Text	Optional description of this block in your script.

Configure the Speech Enabled Menu block

1. Right-click the block and click **Add case**. The **Speech Enabled Menu** dialog appears.





The screenshot shows a dialog box titled "Speech Enabled Menu". It contains a section "Specify DTMF menu items" with a sub-section "Speech Enabled Menu" listing items 0 through 9, *, #, and Error, each with an unchecked checkbox. Below this is a text input field labeled "Or said" and another labeled "Label". At the bottom are "Cancel" and "OK" buttons.

2. Perform one or both of these steps:
 - a. From the **Speech Enabled Menu** list, select one or more menu items.
 - b. In **Or said**, type the text for matching the caller's spoken response. You can type up to 15 separate words. Separate the words with semicolons (" ; ").

Note

- This is an alternative spoken response option for the caller.
- The case accepts any one of your selected digits or words if the caller response is matched in this case.

3. Optionally, in **Label**, type an alternative description.
4. Click **OK**. The **Case** branch  appears.
5. Connect the **Case** branch to the appropriate block in the script.
6. Repeat steps 1-5 to add the branches you need.
7. Connect the **Else** branch  to the appropriate block in the script.

Speech Play and Collect



This block is only available if IVR is enabled.

This block enables the playing of audio or Text-to-Speech (TTS) and collection of human speech or touchtone digits.

The block also enables caller barge-in, which occurs when a caller speaks during playback. See [Stop Prompt On](#) property below.

Note

The times used for speech recognition are dependent on each other and used in the following way:

- **Max Time** = total time required for the speech recognition process to complete
- **Max Silence** = maximum time waited for caller's responses, both for starting to speak and to determine end of speaking

To enable successful speech recognition process to complete, the Max Time must exceed the maximum expected speech input time of the caller plus two times the Max Silence time.

For example, a scenario with a maximum expected speech input time of 5 seconds to input a single number or name, and a maximum silence time of 2 seconds to wait for start and completion of input, requires the following settings, in seconds:

- **Max Time = 10**, which exceeds the minimum of 9, calculated as 5 second input plus 2 x silence time of 2 seconds
- **Max Silence = 2**

Properties	Values	Description
Text or Audio File	String	Enter a <i>.wav</i> file, plain text, or Speech Synthesis Markup Language (SSML) string. For a <i>.wav</i> file, do one of the following: <ul style="list-style-type: none">• Enter a complete URL address to the file you wish to play.• If your system administrator created a shortcut to the <i>.wav</i> files, enter <code>/[file_name.wav]</code>.
Stop Prompt On	Start of Input, Recognition Complete, or Don't Stop	This property determines if caller barge-in is enabled or disabled: <ul style="list-style-type: none">• Start of Input — the prompt is stopped as soon as the user starts to speak or presses a key on the phone keypad.• Recognition Complete — the prompt is stopped only after

Properties	Values	Description
		<p>recognition is achieved according to the configured grammar. This is also known as Hot Word mode.</p> <ul style="list-style-type: none"> Don't Stop — no barge in during prompt.
Grammar	String	<p>The grammar determines what to recognize in a caller's speech when barge-in is enabled, and a barge-in occurs.</p> <p>Specify your grammar(s) in one of the following ways:</p> <ul style="list-style-type: none"> Use a built-in grammar if a built-in grammar is installed in your system. For example, if you have built-in grammar boolean, enter <code>built-in:grammar/boolean</code>. The syntax is <code>built-in:grammar/[your_grammar]</code>. Enter complete URL(s) to your grammar(s). Type a list of comma-delimited words instead of a grammar. <p>If more than one grammar is used, separate the grammars with a semicolon (;). You can list up to ten grammars separated by ; or list the Content-ID.</p> <p>The speech is compared to the grammar. If recognition is successful, playback is stopped, the result variable is assigned, and the next block is executed.</p>
Confidence Level (optional)	Percent	<p>This is a confidence threshold parameter. The speech recognizer computes a confidence level. If the level is below the threshold you set, the recognizer returns no-match as the recognition result. If the level is at or above the threshold, then recognition is successful. The minimum is 50 and the maximum is 100. The default is 50.</p>
Speech Rate	Slow, Medium, or Fast	<p>The rate for reading a Text to Speech (TTS) message.</p>
Volume	Soft, Medium, or Loud	<p>The volume level for reading a TTS message.</p>
Termination Digits	0-9 # and *	<p>Enter digit(s), # or * . These value(s) can be used by the caller to indicate the end of their response.</p> <p>When a termination digit is pressed, playback stops, and the script executes the next block.</p>
Max Time	Seconds	<p>Maximum permitted duration of the caller's spoken response, starting when the caller first begins speaking. Generally, there is no reason to change this from the default value. Maximum permitted setting is 30 seconds. The default is 20.</p> <p>This property applies only to Automatic Speech Recognition (ASR).</p>
Max Digits	Number	<p>The maximum number of caller response DTMF digits. When the limit</p>

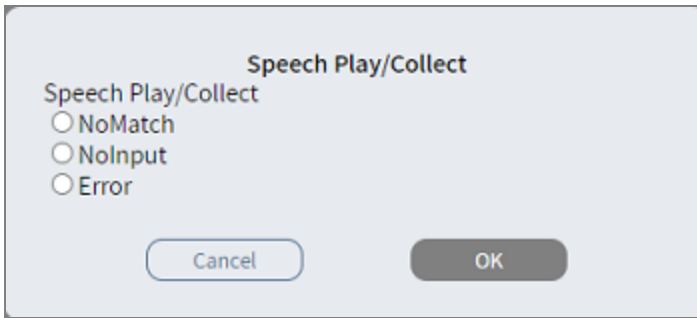
Properties	Values	Description
		is reached, the script proceeds to the next block. Note Typically 3 is too low a number to work reliably and this value is set to 10 or 3 x number of digits to be entered.
Max Silence	Seconds	The maximum length of time to wait for the caller's next response. The default is 5.
Clear Digits	Yes or No	Clear the digit buffer of any digits entered before playing the audio file. The default is No.
Result Variable	String	Name of a variable into which the digit string is to be passed. Must start with an alphabetic character and not exceed 255 characters in length. There is no default value for this, so leaving this field blank causes this block not to function properly. See How to use variables .
Line	Inbound	The next block in the script is applied to the original (incoming) call.
	Outbound	The next block in the script is applied to the outbound call (to where the call is transferred).
Input Mode	Speech and DTMF, Speech only, or DTMF only	If you select Speech and DTMF, and if the caller presses a key (DTMF), the system ignores anything the caller says after the key press. If you select Speech and DTMF, and if the caller speaks, the system ignores any key press after anything the caller says. The default is Speech and DTMF.
Comment	Text	Optional description of this block in your script.

Configure the Speech Play and Collect block


Caution

Do not rename the **Success** branch. Otherwise, when the external speech server returns success, CCaaS routes the script to the else case.

1. Right-click the block and click **Add case**. The **Speech Play/Collect** dialog appears.



2. From the **SpeechPlayCollect** list, select one of the following:
 - **NoMatch** — recognition did not succeed or caller pressed invalid DTMF.
 - **NoInput** — the caller did not respond.
 - **Error** — technical issues prevented speech recognition.

3. Click **OK**. The **Case** branch  appears.
4. Connect the **Case** branch to the appropriate block in the script.
5. Repeat steps 1-4 to add the branches you need.
6. Connect the **Success** branch to the appropriate block in the script.

7. Connect the **Else** branch  to the appropriate block in the script.

Stop Video



This block ensures that the [Play Video](#) block that it follows ends synchronously with any associated [Wait](#) or [Play Audio](#) block.

Properties	Values	Description
Line	Inbound	Default
	Outbound	N/A

Supervised Transfer



VCS only.

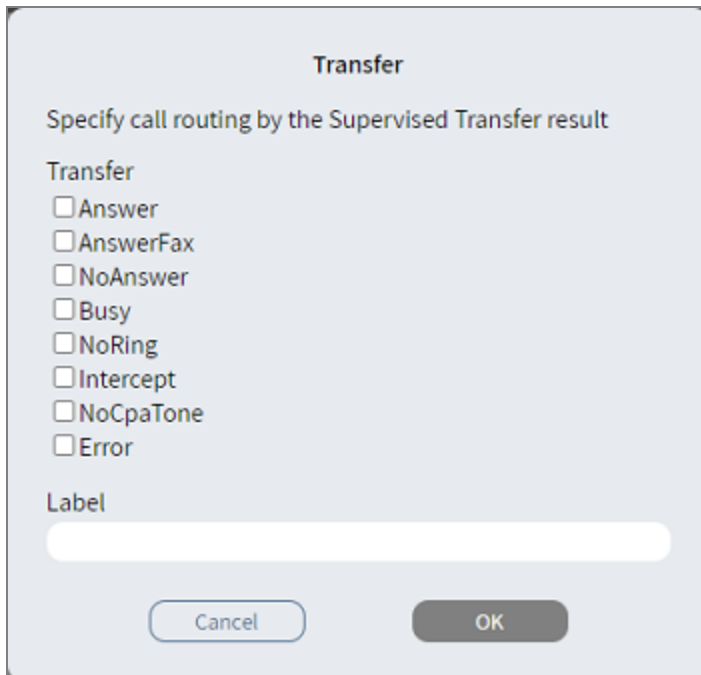
This block redirects a call to another phone number. The difference between Supervised Transfer and Transfer is that the former waits for the outgoing call to be answered or failed. It then allows the call to be re-routed, depending on the result.

Properties	Values	Description
Phone Number	0-9 * and #	<p>Any valid phone number to which to redirect the call, for example, +16315551212</p> <p>By using the + format, the number is location independent. This should be used unless you know all of your VCSs are in one location. If using a variable, add + to the value of the variable, for example, +\$NumberDialed\$.</p> <p>Note</p> <ul style="list-style-type: none">• The maximum outgoing call number or SIP URI is 370 characters.• This can also be an IP address, if you prefix the number with TA:, for example, TA:208.203.191.74.
Source ID	String	<p>Note</p> <p>This information is shown only if the Source display property is missing.</p> <p>When transferring a call, you can optionally set the source address (CallingPartyNumber) from the script you are using.</p> <p>If this is left blank, the ANI for the gateway is used, if that is not set, the ANI for the tenant is used.</p> <p>This can be either an H323-ID or an E164 number:</p> <ul style="list-style-type: none">• H323-ID takes the format NAME:ARNIE or NAME:\$anyvariable\$• E164 takes the format TEL:1234 or TEL:\$anyvalidnumberorvariable\$ <p>All possible values for this property should be contained in a single string, such as:</p> <pre>NAME:NJONES, NAME:FSCHMIDT, TEL:1234567</pre> <p>The source address can be specified as either an H323-ID value denoted by the NAME: prefix, or as an E164 address denoted by the TEL: prefix.</p> <p>You do not have to send both types of source address, but it is legal if</p>



Properties	Values	Description
		needed. However, if not blank, the source address must contain one or the other prefix format.
Source display	String	What to display as a source. This property is also known as CallerID or CallingPartyName.
Connect Timeout	Number	The number of seconds to wait for the calling party to answer. If this is exceeded, the result of No Answer is returned. If left blank, the phone service times-out the dial attempt.
Wave File	Filename	Performs an Asynchronous Play Audio while the transfer is in progress. The <i>.wav</i> file specified should be longer than the timeout value for the connection, or the caller hears silence after the <i>.wav</i> is played. Enter a URL address to the file you wish to play.
Bridge Method	Bridge Calls	Bridges the call control and media stream (the individual calls become a single call). If either party disconnects, the call terminates. Caution For TTS scripts, the transferred call bridges with TTS and the call does not transfer to a new agent. When the agent presses transfer or the transferred call is bridged (whichever happens first), the TTS call disconnects, the agent disconnects, and the transfer monitor moves to the main call (the incoming call). Note The media stream is no longer available in the IVR.
	Bridge Media Only	Parties can communicate with each other, but if one party disconnects, call does not terminate. Note The media stream is no longer available in the IVR.
	Do not Bridge	Do not bridge the calls. Note To retain call control in the IVR, set the Bridge Method to Do not Bridge. Add an Answer case to the Supervised Transfer block and connect this case to a Bridge Call block to finally connect the call. Add other cases to the Supervised Transfer block to perform normal IVR actions with the caller, such as play messages or transfer to other destinations.
	Bridge Media & DTMF	Parties can communicate with and forward DTMF tones to each other, but if one party disconnects, call does not terminate. Note The media stream is no longer available in the IVR.

Configure the Supervised Transfer block

1. Right-click the block and click **Add case**. The **Transfer** dialog appears.



The image shows a dialog box titled "Transfer". Inside the dialog, there is a section titled "Specify call routing by the Supervised Transfer result". Below this, there is a list of transfer options, each with an unchecked checkbox: Answer, AnswerFax, NoAnswer, Busy, NoRing, Intercept, NoCpaTone, and Error. Below the list is a text input field labeled "Label". At the bottom of the dialog are two buttons: "Cancel" and "OK".

2. From the **Transfer** list, select one or more items.
3. Optionally, in **Label**, type an alternative description.
4. Click **OK**. The **Case** branch  appears.
5. Connect the **Case** branch to the appropriate block in the script. For example, the next block could play "The line appears to be busy. If you want to try again in ten seconds, press one."
6. Connect the **Else** branch  to the appropriate block in the script.

Switch

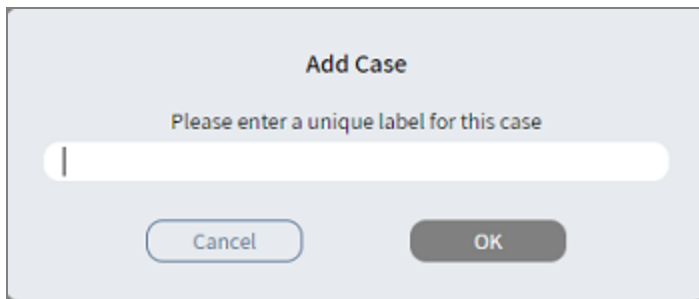


This block evaluates a case statement and branches according. For important information about variables in CCaaS Designer, see [How to use variables](#).

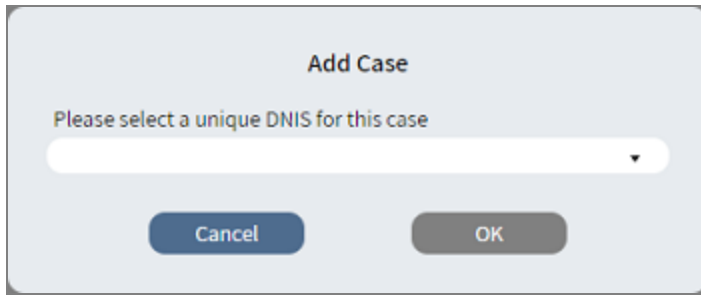
Properties	Values	Description
Expression	The left side of any valid expression equals. (VBScript)	<p>Causes branching in the script.</p> <p>The name of the case branch you add to the block is the value to check the expression is equal to.</p> <p>For example, <code>\$Line.DNIS\$</code> branches to 5551234 when the DNIS value for this call is 5551234.</p> <p>When the switch expression is <code>\$line.DNIS\$</code>, the Add case dialog presents a drop down list of the available DNIS values for the tenant. You are not restricted to using the DNIS values in the drop down list. For example, you can use an expression as the selected value.</p>



Configure the Switch block

1. Right-click the block and click **Add case**. The **Add Case** dialog appears.



If the switch expression is `$line.DNIS$`, the **Add Case** dialog appears with a drop down list of the available DNIS values for the tenant.



2. Type a unique description or select the DNIS value to use.
3. Click **OK**. The **Case** branch  appears.
4. Connect the **Case** branch to the appropriate block in the script.
5. Repeat steps 1-4 to add all the branches you need.
6. Connect the **Else** branch  to the appropriate block in the script.

Note

The following blocks are used for call routing:

- [Check Date](#)
- [Check Time](#)
- [Percent Allocated](#)

They all require Case statements to be evaluated to direct the call.

Table



Note

The Table block is obsolete because tables are now managed through the Tables page. See [Manage tables](#). However, they can still be added to a script and are still imported as part of a legacy script.

This block links to a table to use for quick lookups of information. This block and the [Table Lookup](#) block are best used for static lookups.

Properties	Values	Description
Table Name	Text	Select from the list of tables in the workspace. See Manage tables .

Although the block does not require linking to other blocks to be functional, it is best practice to link it to the [Table Lookup](#) block to aid understanding.

Table Lookup



This block must be linked to the [Table](#) block using the return value from the lookup, for example, Customer Name from CustID. CustID can be a variable populated in a [Get Digits](#) block.

You can configure the Table Lookup block to use the value of the variable CustID to lookup the matching Name for the row with a matching CustID column value.

Properties	Values	Description
Table Name	Text	Select from the list of tables in the workspace. See Manage tables .
Key Column Name	Text	Name of column to search for matching value.
Key Column Value	Text	Value to match in Key column
Lookup Col. Name	Text	Column name for return value.
Lookup Col. Value	Text	Variable to populate with Lookup Column Name for row matching Key Column Value.

Terminate



This block sets the condition for termination.

Properties	Values	Description
Terminate	Yes or No	<p>If Yes, the call terminates on completion of this script event and goes straight to the call terminate event.</p> <p>If No, the event ends normally and moves to the next event according to the predefined flowchart. The default is No for all scripts except Call Reject.</p> <p>Note</p> <ul style="list-style-type: none">• In the Call Reject event, by default this parameter is set to Yes to terminate the call. To resubmit a call to the queue, you can use a Terminate block set to No to overwrite this default.• In the Call Pre Sent event, if set to Yes, the call will be terminated and the item will be moved to the Sent Items folder with the Invalid status.

Transfer by Refer



VCS only.

This block transfers a SIP call without using IVR ports.

Properties	Values	Description
Phone Number	0-9 * and #	<p>Any valid phone number to which to redirect the call, for example, +16315551212</p> <p>The + format makes the number location independent. This should be used unless you know all of your VCSs are in the same location. If you are using a variable, add + to the value of the variable, for example, +\$NumberDialed\$.</p> <p>After the Answer case, no blocks are executed.</p> <p>If the phone number is not reachable (404 SIP error code), the NoAnswer case is executed.</p> <p>Note</p> <ul style="list-style-type: none">• The maximum outgoing call number or SIP URI is 370 characters.• This can also be an IP address. Use a prefix in the format sip:DNIS@IP_ADDRESS, for example, sip:6111@192.168.100.100, where 6111 is the DNIS.
Source ID	String	<p>Note</p> <p>This information is shown only if the Source display property is missing.</p> <p>When transferring a call, you can optionally set the source address (CallingPartyNumber) from the script you are using.</p> <p>If this is left blank, the ANI for the gateway is used, if that is not set, the ANI for the tenant is used.</p> <p>E.164 takes the format TEL:1234 or TEL:\$anyvalidnumberorvariable\$</p> <p>All possible values for this property should be contained in a single string, such as: NAME:NJONES, NAME:FSCHMIDT, TEL:1234567</p> <p>The source address can be specified as an E.164 address denoted by the TEL: prefix.</p>
Source Display	String	<p>What to display as a source. This property is also known as CallerID or CallingPartyName.</p>

Properties	Values	Description
Connect Timeout	Integer	<p>The number of seconds to wait for a connection. The default of 10 seconds is used when the value is:</p> <ul style="list-style-type: none"> • Left blank • 0 • Less than 0 • Greater than 300

Configure the Transfer by Refer block

1. Right-click the block and click **Add case**. The **Transfer** dialog appears.

Transfer



Specify call routing by the Supervised Transfer result

Transfer

- Answer
- AnswerFax
- NoAnswer
- Busy
- NoRing
- Intercept
- NoCpaTone
- Error

Label

Busy, NoRing

2. From the **Transfer** list, select one or more items.
3. Optionally, in **Label**, type an alternative description.
4. Click **OK**. The **Case** branch  appears.
5. Connect the **Case** branch to the appropriate block in the script.
6. Connect the **Else** branch  to the appropriate block in the script.

Transfer Call



VCS only.

This block redirects a call to another phone number.

Properties	Values	Description
Phone Number	0-9 * and #	<p>Any valid phone number to which to redirect the call, for example, +16315551212</p> <p>The + format makes the number location independent. This should be used unless you know all of your VCSs are in the same location. If you are using a variable, add + to the value of the variable, for example, +\$NumberDialed\$.</p> <p>Note</p> <ul style="list-style-type: none">• The maximum outgoing call number or SIP URI is 370 characters.• This can also be an IP address. For H.323 calls, prefix the number with <code>TA:</code>, for example, <code>TA:208.203.191.74</code>. For SIP calls, use a prefix in the format <code>sip:DNIS@IP_ADDRESS</code>, for example, <code>sip:6111@192.168.100.100</code>, where 6111 is the DNIS.
Transfer Type	Blind Transfer	<p>This transfer type causes the VCS to access a second line from the gateway and generate an outbound call to the phone number specified in the Phone Number property. It then bridges the original caller and the outbound call together.</p>
Transfer Type	Outdial	<p>This transfer type dials the specified digits. No hook flash or other signalling is issued, so the script remains in control of the call and can interact with the original caller after the called party hangs up.</p>
Source ID	String	<p>Note</p> <p>This information is shown only if the Source display property is missing.</p> <p>When transferring a call, you can optionally set the source address (CallingPartyNumber) from the script you are using.</p> <p>If this is left blank, the ANI for the gateway is used, if that is not set, the ANI for the tenant is used.</p> <p>This can be either an H.323-ID or an E.164 number:</p> <ul style="list-style-type: none">• H.323-ID takes the format <code>NAME:ARNIE</code> or <code>NAME:\$anyvariable\$</code>• E.164 takes the format <code>TEL:1234</code> or <code>TEL:\$anyvalidnumberorvariable\$</code>

Properties	Values	Description
		<p>All possible values for this property should be contained in a single string, such as: <code>NAME:NJONES,NAME:FSCHMIDT,TEL:1234567</code></p> <p>The source address can be specified as either an H.323-ID value denoted by the <code>NAME:</code> prefix, or as an E.164 address denoted by the <code>TEL:</code> prefix</p> <p>You do not have to send both types of source address, but it is legal if needed. However, if not blank, the source address must contain one or the other prefix format.</p>
Source display	String	What to display as a source. This property is also known as CallerID or CallingPartyName.
Bridge Method	Bridge Calls	Bridges the call control and media stream (the individual calls become a single call). If either party disconnects, the call terminates.
	Bridge Media Only	Parties can communicate with each other, but if one party disconnects, call does not terminate.
	Do not Bridge	Do not bridge the calls, use the Bridge Call function, instead. See Bridge Call .
	Bridge Media & DTMF	Parties can communicate with and forward DTMF tones to each other, but if one party disconnects, call does not terminate.

UnQueue



VCS only.

This block immediately removes a call from queue.

An example of how this may be used is to unqueue the call in `MusicOnHoldWhileInQueue` and add a skill that causes the call to be rejected when requeued. The [CallRejected script](#) can send the call to voicemail and remove the skill just added.

Note

Unqueue and Requeue requests are sent to the Call Matcher only when all of the following criteria are met:

- The call is in the appropriate state (waiting for agent ring)
- The script is in `MusicOnHoldInQueue`, `InqueueTimeout`, or `PickupTimeout`
- The media is voicecall
- The Unqueue and Requeue blocks are paired in the script (for each Unqueue there's an appropriate Requeue in the script)

For any other situation, the VCS log indicates that a particular Unqueue or Requeue request was ignored due to the given reason, however, the script execution continues to the next block.

See also [unQueue](#) and [reQueue](#).

VXML (VoiceXML) Transfer Call



This block interfaces with any speech application platform for Speech Recognition and Text-to-Speech services. The platform must support SIP.

Properties	Values	Description
Server	String	SIP URI server information. For more information on the format of this property, see the article at: http://www.voip-info.org/wiki-SIP+uri .
VoiceXML URL	String	Location (URI) of VXML application to be used.
Transfer Type	Blind Transfer	This transfer type causes the VCS to access a second line from the gateway and generate an outbound call to the phone number specified in the Phone Number property. It then bridges the original caller and the outbound call together.
Source ID	String	<p>Note</p> <p>This information is only shown if the Source display property is missing.</p> <p>When transferring a call, you can optionally set the source address (CallingPartyNumber) from the script you are using. If this is left blank, the ANI for the gateway is used, if that is not set, the ANI for the tenant is used.</p> <p>This can be either an H323-ID or an E164 number:</p> <ul style="list-style-type: none">• H323-ID takes the format <code>NAME:ARNIE</code> or <code>NAME:\$anyvariable\$</code>• E164 takes the format <code>TEL:1234</code> or <code>TEL:\$anyvalidnumberorvariable\$</code> <p>All possible values for this property should be contained in a single string, such as: <code>NAME:NJONES, NAME:FSCHMIDT, TEL:123456 7</code></p> <p>The source address can be specified as either an H323-ID value denoted by the <code>NAME:</code> prefix, or as an E164 address denoted by the <code>TEL:</code> prefix. You do not have to send both types of source address, but sending both is legal. However, if not blank, the source address must contain one or the other prefix.</p>
Source Display	String	What to display as a source. This property is also known as CallerID or CallingPartyName.

Properties	Values	Description
Bridge Method	Bridge Calls	Bridges the call control and media stream (the individual calls become a single call). If either party disconnects, the call terminates.
	Bridge Media Only	Parties can communicate with each other, but if one party disconnects, call does not terminate.
	Do not Bridge	Do not bridge the calls, use the Bridge Call function, instead. See Bridge Call .
	Bridge Media & DTMF	Parties can communicate with and forward DTMF tones to each other, but if one party disconnects, call does not terminate.

Wait for Events



This block allows the script to continue processing after one of the events described in [Set Event](#) occurs.

Properties	Values	Description
Timeout	Milliseconds	The default is 0. This causes the block to wait indefinitely for the event to occur. If the event occurs before the timeout, the script automatically passes control to the Set Event block. If the event times out, control passes to the block connected to the Wait for Event block, and not to the Set Event block.

Note

In scenarios where a Wait for Events block follows a Bridge Call block, and is then followed by Play Audio blocks, we recommend that you add a Wait block with a duration of at least 1 second.

Wait



VCS only.

This block pauses execution of the script for the specified number of seconds, or until the caller enters the termination digit.

Properties	Values	Description
Termination Digits	0-9 * and #	Enter digit(s), # or * . These value(s) can be used by the caller to terminate this command. For example, to allow any touch tone to be a termination digit, set this to 1234567890#*
Value	Integer	Number of seconds to pause before continuing.

This block is typically used after a [Play Audio](#) block with Async Mode set to Yes to allow the script to wait a number of seconds until proceeding to the next block.

Web Services



This block incorporates information available as a web service into the self-service and call routing logic.

Properties	Values	Description
URL	Web address	Web services URL, for example, <code>http:// www.abc.com/soap/GetClientinformation.asmx</code>
Namespace	String	Web services namespace, <code>http:// wsdl.use.abc.com</code>
Namespace Prefix	String	A namespace prefix for the xml request to be sent to the web services, for example, <code>q0</code> .
Method	String	The web services method to be called, for example, <code>GetClientInformation</code> .
Parameters	String	The key/value parameter pairs for the web services method, for example, <code>PIN_Number, 123456789, Surname, Doell, GivenName, John</code>
Parameter Wrapper	String	The nested wrapper parameters, for example, <code><Request><q0:RequestWrapper>%%1</q0:RequestWrapper></Request></code> . Note %%1 is required for indicating where the required parameters are being substituted.
Envelope Prefix	String	The soap envelope prefix, for example, <code>SOAP-ENV</code> .
Timeout	String	Number of seconds to wait for response from web service before timing out. The default is 15. If the web service does not respond within the specified timeout, the operation is aborted and a log entry is created in the CCaaS Designer log as follows: <code>No response was received from:http://webserver/webservices/test.asmx. Operation aborted.</code>
Result Variable	String	A variable name where responses are being prefixed. See XML response example .

See [Web services XML examples](#).

6: Objects

This section describes the CCaaS scripting object model.

Note

The new model should only be used on scripts executed by the MCS. The new objects are not valid for VCS and ICS.

[New model for messaging calls](#) describes the scripting object model valid only for email, callbacks, and voicemail.

CCaaS 7.3.x continues to support the following objects for MCS, VCS, and ICS media and scripts:

- [CallData object](#) — for messaging data.
- [CCLEvent object](#) — for the call media type.
- [CosmoCall object](#) — for callbacks.
- [Line object](#) — for VoIP.
- [NewLine object](#) — associated with the outbound call from a call transfer.
- [Session object](#) — available to all call types.

New model for messaging calls

This subsection describes the new scripting object model valid only for email, callbacks, and voicemail.

Note

The new model should only be used on scripts executed by the MCS. The new objects are not valid for VCS and ICS.

The new model for messaging calls contains the following API, objects, and sub-objects:

- [Common media API](#) — for identifying the call, its type, and common routing and history properties. Has multiple sub-objects.
- [CallbackCall object](#) — for additional callback-specific information.
- [EmailCall object](#) — for additional email-specific information.
- [OutgoingEmailCall object](#) — for additional outgoing email-specific information.
- [VoicemailCall object](#) — for additional voicemail-specific information.
- Sub-objects:
 - [AgeingInfo object](#) — for additional aging information for email and voicemail.
 - [DeferInfo object](#) — for additional defer information for email and voicemail.
 - [EmailCall.EmailData object](#) — for additional information about the email data.
 - [EmailCall.LastAgentInfo object](#) — for additional information about the last agent to handle the email.
 - [EmailCall.RecordingControl object](#) — for additional information about the email recording.

- [OutgoingEmailCall.OutgoingEmailData object](#) — for additional information about the outgoing email data.
- [PullInfo object](#) — for additional pull information for email and voicemail.
- [VoicemailCall.LastAgentInfo object](#) — for additional information about the last agent to handle the voicemail.
- [VoiceMailCall.RecordingControl object](#) — for additional information about the voicemail recording.

To see which of the new model for MCS media objects are compatible with the [CallData object](#) properties, see [Comparison with CallData object](#).

Common media API

The common media API methods and properties are applicable to the [EmailCall object](#), [CallbackCall object](#), and the [VoicemailCall object](#).

Property name	Accessibility	Applies to	Description
AccountName	Read/Write	All	Allow to set or get associated account name.
CallDirection	Read	All	Call direction type: <ul style="list-style-type: none"> • 0 — Incoming • 1 — Outgoing • 4 — Callback Request
CallerInfo	Read	All	Provides access to the CallerInfo object.
CallId	Read	All	The call ID.
CallType	Read	All	Media type of call: <ul style="list-style-type: none"> • 2 — Email • 4 — Callback • 5 — Voicemail
LastError	Read	All	Gets the last known error code.
Message	Read/Write	All	The caller's message to display to the agent. Previously known as CallerMessage.
RoutingProperties	Read	ACD calls	Contains ACD-related routing properties.
TenantID	Read	All	The tenant ID.
Terminate	Read/Write	All	Set to true to terminate the call on script completion.
TtsInfo	Read	Email/Voicemail	Contains TTS-related information.

RoutingProperties

RoutingProperties contains ACD-related routing properties. It is applied on all MCS media (email, callbacks, and voicemail).

Property name	Accessibility	Description
AgentSelectionMode	Read/Write	Defines whether skill weighting is being used (when call arrives) and how: <ul style="list-style-type: none">• 0 = not used• 1 = least skilled• 2 = most skilled
AssignedPriority	Read/Write	Affects HardPriority.
CallRejectedCounter	Read	How many times the call was rejected.
DirectRouteAgentId	Read/Write	AgentID where this call should be routed to.
Maturity	Read/Write	The current arrival time bias. Arrival time bias is used to offset TQOS by adding bias to the arrival time.
OptionalParameters	Read	Returns Optional Parameters collection.
QueueName	Read/Write	The name of the queue for this call.
RoutingModes	Read	Provides a way to override the default setting for the routing mechanism in different modes.
Skills	Read	Provides access to the skills collection.

RoutingModes

RoutingProperties.RoutingMode provides a way to override the default setting for the routing mechanism in different modes:

- 0 = Default
- 1 = Route
- 2 = Queue
- 3 = Reject

Property name	Accessibility	Description
AgentsBusy	Read/Write	Controls routing when all applicable agents are busy.
AgentsLoggedOff	Read/Write	Controls routing when all applicable agents are logged off.
AgentsUnavailable	Read/Write	Controls routing when all applicable agents are unavailable.

Skills collection

RoutingProperties.Skills provides access to the skills collection.

Property name	Accessibility	Description
Count	Read	Size of collection.
Item(id)	Read	Gets skill name by index. This method is used internally for collection traversal.
SetItem(Id, name)	Write	Sets skill name at given index. This method is used internally for collection traversal.
Add	Write	By name (compressed - case insensitive).
Remove(name)	Write	Removes skill from collection by name.
RemoveByIndex (index)	Write	Removes skill from collection by index. This method is used internally.
Clear	Write	Clears container.
Contains	Read	Checks if given item (by name) exists.

CallerInfo object

The CallerInfo object has the following properties.

Property name	Accessibility	Description
Name	Read/Write	Caller's name.
ANI	Read	Caller's address.
DNIS	Read	Designated address used by caller.
IvrDigits	Read	Only for callbacks and voicemail.

ErrorInfo

The ErrorInfo property contains detailed error information.

Property name	Type	Description
ErrorCode	Read	Enumeration varies per script and media.
ErrorMessage	Read	The error message.

Property name	Type	Description
ErrorLevel	Read	Error level types: <ul style="list-style-type: none"> • 0 = Info • 1 = Warning • 2 = Error

Optional parameters collection

The Optional Parameters property provides access to the optional parameters collection.

Property name	Accessibility	Description
Count	Read	Size of collection.
Item(id)	Read	Gets optional parameter name by index. This method is used internally for collection traversal.
SetItem(Id, name)	Write	Sets optional parameter name at given index. This method is used internally for collection traversal.
Add	Write	By name (compressed – case insensitive).
Remove(name)	Write	Removes optional parameter from collection by name.
RemoveByIndex (index)	Write	Removes optional parameter from collection by index. This method is used internally.
Clear	Write	Clears container.
Contains	Read	Checks if given item (by name) exists.

ItemInfo object

The ItemInfo object is available to email, callback and voicemail call types and includes the following properties.

Property name	Type	Description
ItemId	Read	Returns item's unique identifier.
FolderInfo	Read	Provides access to the FolderInfo object.

FolderInfo object

The FolderInfo object is available to email, callback and voicemail call types and includes the following properties.

Property name	Type	Description
Name	Read	The folder's name.
Id	Read	The folder's unique identifier.
EmailAddress	Read	The folder's email address.

TtsInfo object

The TtsInfo object is available to transferrable media types (email and voicemail) and includes the following properties.

Property name	Type	Description
ApplicationId	Read	Returns associated EntryPoint application ID.
BlockName	Read	Returns associated EntryPoint block name.
Parameter	Read	Returns associated EntryPoint parameter name.
ScriptName	Read	Returns associated EntryPoint script name.

CallbackCall object

The CallbackCall object applies to callbacks and includes the following properties.

Property name	Type	Description
ItemInfo	Read	Returns callback's item information.
OutDialingAddress	Read	Returns designated callback address (usually phone number).
ImmediateDialing	Read	Returns immediate dialing flag. If set to true, when the callback arrives at the agent, the callback out-dialing starts automatically.
OriginatingAgentID	Read	Returns AgentID of the agent scheduling this callback.
AttachmentUrl	Read	Returns the caller's URL to display to the agent. Previously known as CallerURL.
RescheduleCount	Read	Returns the callback's current reschedule counter: <ul style="list-style-type: none"> Value=0 for initial callback Value>0 for continuous callback
MaxRescheduleCount	Read	Returns the callback's maximum allowed reschedule attempts.

EmailCall object

The EmailCall object applies to email and includes the following properties.

Property name	Type	Description
ItemInfo	Read	Returns email's item information.
DeferInfo	Read	Returns email's associated defer information.
PullInfo	Read	Returns email's associated pull information.
AgeingInfo	Read	Returns email's associated ageing information.

OutgoingEmailCall object

The OutgoingEmailCall object applies to an outgoing email, reply, or forward and includes the following properties.

Property name	Type	Description
OutgoingEmailData	Read	Reference to OutgoingEmailData interface.
OutgoingMsgType	Read	Email message type: 1 — REPLY 2 — FORWARD 3 — NEW (outgoing email)

VoicemailCall object

The VoicemailCall object applies to voicemail and includes the following properties.

Property name	Type	Description
ItemInfo	Read	Returns voicemail's item information.
DeferInfo	Read	Returns voicemail's associated defer information.
PullInfo	Read	Returns voicemail's associated pull information.
AgeingInfo	Read	Returns voicemail's associated aging information.

AgeingInfo object

The AgeingInfo object applies to email and voicemail and includes the following properties.

Property name	Type	Description
AgedRefCount	Read	Number of ageing cycles.
IsDeferAvailable	Read/Write	<p>True or false.</p> <p>Write is performed through the Assign block.</p> <p>For email, set Variable Name to <code>EmailCall.AgeingInfo.IsDeferAvailable</code> and Variable Value to <code>True</code>.</p> <p>For voicemail, set Variable Name to <code>VoiceMailCall.AgeingInfo.IsDeferAvailable</code> and Variable Value to <code>True</code>.</p>

DeferInfo object

The DeferInfo object applies to email and voicemail and includes the following properties.

Property name	Type	Description
DeferByAgentID	Read	Agent ID of agent that performed the defer action.
DeferByGroupID	Read	Group ID of agent that performed the defer action.
DeferType	Read/Write	<p>Defer type:</p> <ul style="list-style-type: none"> 0 — Personal 1 — Shared <p>Write is performed through the Set Defer Type block.</p>

EmailCall.EmailData object

The EmailCall.EmailData object applies to email and includes the following properties.

Property name	Type	Description
From	Read	The sender.
To	Read	<p>The recipients. May be a list of values, separated by comma or semi-colon (;).</p> <p>To find out the destination of the email, use: <code>\$EmailData.To\$</code></p>
Cc	Read	The carbon copy recipients. May be a list of values, separated by comma or semi-colon (;).
Subject	Read	The subject.
BodyPreview	Read	The first three lines of the body as text.
UrgencyIndicator	Read	Email importance setting:

Property name	Type	Description
		<ul style="list-style-type: none"> • 0 — Low • 1 — Normal • 2 — High • 3 — No-change
ReceivedTime	Read	The email arrival time.
SentTime	Read	The email sent time.
Body	Read	<p>The part of the body sent from the MS to the MCS to expose to scripts, especially the Call Arrival script.</p> <p>Use the CCaaS Admin to disable body fetching, set the body format, and, for text only, set the maximum size. For details, see <i>Adding a folder</i> in the <i>CCSP Tenant Administrator Guide</i>.</p>
KeepBody	Read	<p>Whether and when to delete Body:</p> <ul style="list-style-type: none"> • 0 (default value) — clear on route. Clear MCS memory upon successful call routing to MCS. • 1 — delete immediately. For example, if you only want to use Body in the Call Arrival script. • 2 — keep. For example, if you want to use Body in the Call Termination script, you need to keep it for the entire call lifecycle.

EmailCall.LastAgentInfo object

The EmailCall.LastAgentInfo object applies to email and includes the following properties.

Property name	Type	Description
AgentId	Read	The agent's ID.
AgentGlobalId	Read	The agent's Global User ID. This is the ID the agent received during a logged on session.
FirstName	Read	The agent's first name.
LastName	Read	The agent's last name.

EmailCall.RecordingControl object

The EmailCall.RecordingControl object applies to email and includes the following properties.

Property name	Type	Description
CallRecording	Read/Write	Controls whether the call will be recorded.
ScreenRecording	Read/Write	Controls whether the screen will be recorded.
AgentControl	Read/Write	Controls whether the agent will be able to stop the screen recording.
ShowIndicator	Read/Write	Controls whether the agent will see indication of the screen recording.

OutgoingEmailCall.OutgoingEmailData object

The OutgoingEmailCall.OutgoingEmailData object applies to an outgoing email, reply, or forward and includes the following properties.

Property name	Type	Description
From	Read/Write	Email originator's email address.
To	Read/Write	Primary recipient's email address.
CC	Read/Write	Carbon copy - secondary recipient's email address.
BCC	Read/Write	Blind carbon copy - private secondary recipient's email address.
Subject	Read/Write	Email's title.
UrgencyIndicator	Read/Write	Email's priority level.
ReplyTo	Read/Write	If set, the reply to the email is sent to this address.

PullInfo object

The PullInfo object applies to email and voicemail and includes the following properties.

Property name	Type	Description
PullType	Read	Type of the pull action. Possible values are: <ul style="list-style-type: none"> • 1 — pull from Deferred folder • 2 — pull Not-Yet-Queued from ACD folder • 3 — pull Queued from ACD folder • 4 — pull from Rejected folder • 5 — pull from Archived folder • 6 — pull from Sent-Items folder • 7 — pull from Outbox folder • 8 — pull from Drafts folder

Property name	Type	Description
IsRestoredFromArchiveMSR	Read	Flag indicating if the pulled item was restored from the Archive MSR
PulledByAgentID	Read	Agent ID of agent that performed the pull action.
PulledByGroupID	Read	Group ID of agent that performed the pull action.

VoiceMailCall.LastAgentInfo object

The VoiceMailCall.LastAgentInfo object applies to voicemail and includes the following properties.

Property name	Type	Description
AgentId	Read	The agent's ID.
AgentGlobalId	Read	The agent's Global User ID. This is the ID the agent received during a logged on session.
FirstName	Read	The agent's first name.
LastName	Read	The agent's last name.

VoiceMailCall.RecordingControl object

The VoiceMailCall.RecordingControl object applies to voicemail and includes the following properties.

Property name	Type	Description
CallRecording	Read/Write	Controls whether the call will be recorded.
ScreenRecording	Read/Write	Controls whether the screen will be recorded.
AgentControl	Read/Write	Controls whether the agent will be able to stop the screen recording. This property is used as a toggle to start and then stop recording. Each start-stop creates a new segment.
ShowIndicator	Read/Write	Controls whether the agent will see indication of the screen recording.

Comparison with CallData object

The following table lists the [CallData object](#) properties and shows which of the new model for MCS media objects they are compatible with.

CallData property name	Accessibility	Compatible with	Comments
AccountName	Read/Write	CallbackCall.AccountName EmailCall.AccountName VoicemailCall.AccountName	Extended to support multiple accounts per call.
AddSkill	Write	CallbackCall.RoutingProperties.Skills.Add EmailCall.RoutingProperties.Skills.Add EmailCall.RoutingProperties.Skills.Add	
AgentAvailableRoutingMode	Read/Write	N/A	Unsupported
AgentBusyRoutingMode	Read/Write	CallbackCall.RoutingProperties.RoutingMode EmailCall.RoutingProperties.RoutingMode EmailCall.RoutingProperties.RoutingMode	
AgentLoggedOffRoutingMode	Read/Write	CallbackCall.RoutingProperties.RoutingMode EmailCall.RoutingProperties.RoutingMode EmailCall.RoutingProperties.RoutingMode	
AgentUnavailableRoutingMode	Read/Write	CallbackCall.RoutingProperties.RoutingMode EmailCall.RoutingProperties.RoutingMode EmailCall.RoutingProperties.RoutingMode	
AnsweringAgentId	Read	EmailCall.LastAgentInfo.AgentId VoicemailCall.LastAgentInfo.AgentId	Email and voicemail calls only.
answeringagentfirstname	Read	EmailCall.LastAgentInfo.FirstName VoicemailCall.LastAgentInfo.FirstName	

CallData property name	Accessibility	Compatible with	Comments
answeringagentguid	Read	EmailCall.LastAgentInfo.AgentGlobalId VoicemailCall.LastAgentInfo.AgentGlobalId	
answeringagentlastname	Read	EmailCall.LastAgentInfo.LastName VoicemailCall.LastAgentInfo.LastName	
ApplicationId	Read	EmailCall.TtsInfo.ApplicationId VoicemailCall.TtsInfo.ApplicationId	
ArrivalTimeBias	Read/Write	CallbackCall.RoutingProperties.Maturity EmailCall.RoutingProperties.Maturity EmailCall.RoutingProperties.Maturity	
BlockName	Read	EmailCall.TtsInfo.BlockName VoicemailCall.TtsInfo.BlockName	
BlockParamName	Read	EmailCall.TtsInfo.BlockParamName VoicemailCall.TtsInfo.BlockParamName	
CallAssignedPriority	Read/Write	CallbackCall.RoutingProperties.AssignedPriority EmailCall.RoutingProperties.AssignedPriority EmailCall.RoutingProperties.AssignedPriority	
CallerMessage	Read/Write	CallbackCall.Message EmailCall.Message VoicemailCall.Message	
CallerName	Read/Write	CallbackCall.CallerInfo.Name EmailCall.CallerInfo.Name VoicemailCall.CallerInfo.Name	
CallerURL	Read/Write	Callback.AttachmentURL	

CallData property name	Accessibility	Compatible with	Comments
CallRejectedCounter	Read	CallbackCall.RoutingProperties.CallRejectedCounter EmailCall.RoutingProperties.CallRejectedCounter VoicemailCall.RoutingProperties.CallRejectedCounter	
CallSetName	Read/Write	CallbackCall.RoutingProperties.QueueName EmailCall.RoutingProperties.QueueName EmailCall.RoutingProperties.QueueName	
CampaignName	Read/Write	N/A	Unsupported
DirectRouteAgentId	Read/Write	CallbackCall.RoutingProperties.DirectRouteAgentId EmailCall.RoutingProperties.DirectRouteAgentId EmailCall.RoutingProperties.DirectRouteAgentId	
Disconnect_Code	Read/Write	N/A	Unsupported
EstimatedWaitTime	Read	N/A	Unsupported
HasSkill	Read	CallbackCall.RoutingProperties.Skills.Contains EmailCall.RoutingProperties.Skills.Contains EmailCall.RoutingProperties.Skills.Contains	
MediaType	Read	CallbackCall.CallType EmailCall.CallType VoicemailCall.CallType	
Option	Read/Write	CallbackCall.OptionalParameters.Item EmailCall.OptionalParameters.Item VoicemailCall.OptionalParameters.Item	
Options	Read	CallbackCall.OptionalParameters EmailCall.OptionalParameters VoicemailCall.OptionalParameters	

CallData property name	Accessibility	Compatible with	Comments
Priority	Read/Write	N/A	Unsupported
Recording	Read/Write	EmailCall.RecordingControl.CallRecording VoicemailCall.RecordingControl.CallRecording	
RemoveSkill	Write	CallbackCall.RoutingProperties.Skills.Remove EmailCall.RoutingProperties.Skills.Remove EmailCall.RoutingProperties.Skills.Remove	
RoutingPriority	Read/Write	CallbackCall.RoutingProperties.AgentSelectionMode EmailCall.RoutingProperties.AgentSelectionMode EmailCall.RoutingProperties.AgentSelectionMode	
ScreenRecording	Read/Write	EmailCall.RecordingControl.AgentControl VoicemailCall.RecordingControl.AgentControl	
ScriptName	Read	EmailCall.TtsInfo.ScriptName VoicemailCall.TtsInfo.ScriptName	
SelectiveRecording	Read/Write	EmailCall.RecordingControl.AgentControl VoicemailCall.RecordingControl.AgentControl	
ShowRecording	Read/Write	EmailCall.RecordingControl.ShowIndicator VoicemailCall.RecordingControl.ShowIndicator	
Skill	Read/Write	CallbackCall.RoutingProperties.Skills.Item EmailCall.RoutingProperties.Skills.Item EmailCall.RoutingProperties.Skills.Item	
Skills	Read	CallbackCall.RoutingProperties.Skills EmailCall.RoutingProperties.Skills EmailCall.RoutingProperties.Skills	

CallData property name	Accessibility	Compatible with	Comments
SkillsCount	Read	CallbackCall.RoutingProperties.Skills.Count EmailCall.RoutingProperties.Skills.Count EmailCall.RoutingProperties.Skills.Count	
TenantID	Read	CallbackCall.TenantID EmailCall.TenantID VoicemailCall.TenantID	
Wrapup_Code	Read	N/A	Unsupported
Wrapup_Data	Read	N/A	Unsupported

CallData object

For a comparison with the new model for MCS media, see [Comparison with CallData object](#).

This object is available to all call types.

Property name	Type	Description
CallData.AgentDeliveryInitiated	Boolean	Only valid in MusicOnHoldInQueue script for incoming calls. Indicates whether a current call is already in the process of being delivered to an agent.
CallData.AnsweringAgentFirstName	String	The agent's first name.
CallData.AnsweringAgentGuid	Integer	The agent's Global User ID. This is the ID the agent receives during a logged on session.
CallData.AnsweringAgentId	Integer	The agent's ID.
CallData.AnsweringAgentLastName	String	The agent's last name.
CallData.ArrivalTimeBias	Time	The current arrival time bias. Arrival time bias is used to offset TQOS by adding bias to the arrival time.
CallData.CallAssignedPriority	Integer	A priority in the range from 0 through 99 can be assigned to a call from a script. A call with priority 99 has the highest priority in its queue. A call with priority 0 has the lowest priority in its queue. The default is 49. This property can be assigned to calls through the Call Maturity/Priority and Assign CCaaS Designer blocks. Call priority takes precedence over call maturity within a particular queue.
CallData.CallDirection	Integer	The values are: <ul style="list-style-type: none">• 0 — Incoming• 1 — Outgoing• 2 — Internal• 3 — Agent to agent• 4 — Callback request - initial ACD callback call (exists on MCS servers only when the callback reschedule count == 0)• 5 — Callback reschedule - continuous ACD callback call (exists on MCS servers only when the callback reschedule count > 0)• 6 — Callback response - outgoing VoIP callback call (exists on VCS servers only when dialed out as the

Property name	Type	Description
		<p>result of a callback call)</p> <ul style="list-style-type: none"> • 7 — Predictive • 8 — Predictive preview • 9 — IVR dialing
CallData.CallerMessage	String	The caller's message to display to the agent.
CallData.CallerName	String	The caller's name to display to the agent.
CallData.CallerURL	String	The caller's URL to display to the agent.
CallData.calloutAgentName	String	The calling agent's user name.
CallData.CallRejectedCounter	Integer	How many times the call was rejected.
CallData.CallSetName	String	The name of the queue for this call.
CallData.CampaignName	String	For outbound dialing campaigns only. Can be used to display the name of the outbound dialing campaign to the agent.
CallData.DirectExtension	Integer	The direct extension that was called.
CallData.DirectExtensionCall	Boolean	True if the call is to a direct extension. False if it is not.
CallData.DirectRouteAgentID	Integer	The agent's ID. Allows calls to be directly routed to an agent via a script.
CallData.Disconnect_Code	Integer	The IVR disconnect reason code logged to the Call Data Records. If this is non-zero when the IVR completes, the Call Matcher does not queue the call.
CallData.EstimatedWaitTime	Integer	<p>Estimated time until agent pickup while call is in queue. Value is equal to the average wait for the queue minus the time waited so far. If the value is negative, zero is reported.</p> <p>The value is in seconds.</p> <p>The measurement interval for the waiting time is estimated for at least 5 minutes and at least 10 calls, as follows:</p> <p>If there are more than 10 calls during the last 5 min., the waiting time is calculated based on all the calls.</p> <p>If there are less than 10 calls during the last 5 min., the waiting time is calculated based on 10 calls in a time interval greater than the last 5 min.</p>
CallData.HasSkill("SkillName")	Boolean	Tests if the supplied skill is attached to the call.

Property name	Type	Description
Calldata.HideAniFromAgent	Boolean	Whether the ANI should be hidden from an agent in response to the network presentation indicator. See Determining the Calldata.HideAniFromAgent value .
Calldata.IsCallback	Boolean	True if call is callback type. False if it is not.
CallData.MediaType	Integer	The media type for the current call. Possible values are: <ul style="list-style-type: none"> • 0 — Chat call • 1 — Phone call • 2 — Email call • 3 — VOIP call • 4 — Callback call • 5 — Voicemail call
CallData.Option("Parameter")	String	Parameter is the name of a previously set optional parameter. This is used to access any optional parameters that are set using the Optional Parameters block.
CallData.PickUpFailurescounter	Integer	How many times Pickup Timeout occurred for one call.
CallData.RefusingAgentId	Integer	The ID of the agent refusing to accept the call.
CallData.RefusingAgentName	String	The logon name of the agent refusing to accept the call.
CallData.Skills	String	The skills to display to the agent.
CallData.SkillsCount	Integer	The current number of skills attached to call.
CallData.TenantID	Integer	The CCaaS tenant ID for the current call.
CallData.TtsCounter	Integer	How many times the call was transferred to system (all media).

Properties from SIP headers

Session Initiation Protocol (SIP) is a procedure, called a protocol, used in the setting up of telephone calls between two systems that serve a caller and the called party. The header provides information used in managing a call and does not carry content, such as voice signals.

Property name	Type	Description
CallData.line.SIPPhoneContext	String	Phone-context parameter from incoming call P-Asserted-Identity (PAISIP) header. The information extracted supplements the phone

Property name	Type	Description
		<p>URI (tel URI) in the SIP header. The tel URI may have purely local significance. The context can provide information about the remote end point of a call so that the tel URI can be properly interpreted.</p> <p>For example, if the tel URI is 640-1234, the context can be 001-631 where the country code of the United States is followed by a Long Island area code.</p>
CallData.line.SIPPrivacyList	String	<p>Values from incoming call SIP Privacy header.</p> <p>This data indicates whether the caller identifying information is to be revealed to the called party. The list is a string extracted from the INVITE message. For example, if the Privacy header is present in the INVITE message and contains "user" or "id," then the caller's identity is expected to be hidden from the called party.</p> <p>The treatment of this information depends on your system administrator's settings for privacy in your network. See <i>SIP privacy</i> parameters in the <i>CCSP Tenant Administrator Guide</i>.</p>
CallData.line.SIPPAISIP	String	Value from incoming call PAISIP header in SIP format.
CallData.line.SIPPAITEL	String	Value from incoming call PAITEL header. It is the PAI in phone URI format.
CallData.line.SIPPANILocationAreaCode	String	Value from incoming call P-Access-Network-Info (PANI) header. The PANI supplements the PAITEL or PAISIP with sending service provider location information. It is typically used for billing between service providers and between service providers and their subscribers.
CallData.line.SIPPANIOperatorCode	String	Value from incoming call PANI header. One of the components of the PANI header.
CallData.line.SIPPANIAccessType	String	Value from incoming call PANI header. One of the components of the PANI header.
CallData.line.SIPMediasList	String	<p>List of media selected in the Session Description Protocol (SDP) part of the SIP INVITE message.</p> <p>Note</p> <p>SDP describes multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation.</p>

Playback-related properties

The properties shown below are only available for Playback-enabled contact centers.

Note

- For information on how these values are set using the Playback block in CCaaS Designer, see [Playback](#).
- CCaaS UI was previously called TouchPoint.

Property name	Type	Description
CallData.Recording	Integer	Values can be 0 or 1. See the table below for an explanation of the effects.
CallData.SelectiveRecording	Integer	Values can be 0 or 1. See the table below for an explanation of the effects.
CallData.ShowRecording	Integer	Values can be 0 or 1. See the table below for an explanation of the effects.

The Playback implementation allows the addition of a recording button to CCaaS UI to allow the agent to selectively record a partial call if SelectiveRecording is turned on. If ShowRecording is turned on, this button shows the current status of the recording. This button is not shown when the agent is not in call unless the agent is a supervisor and he or she has selected to monitor a call. The state of this button depends on how the CallData parameters shown below are set for the call. The table below may help to clarify the choices that are available.

Recording	SelectiveRecording	ShowRecording	Description
0	0	0	Recording is not active for this call. Think of this scenario as if there is no recording installed at all on the agent application machines.
1	0	0	Record the entire call.
0	1	0	Do not record this call. The agent can start recording and stop recording at any time during the call.
0	0	1	Do not record this call, but show the status of the recording button. The status of the recording button should indicate that this call is not being recorded.

Recording	SelectiveRecording	ShowRecording	Description
1	1	0	Record the entire call and the agent can play around with the recording button, but it does not do anything. That is, the agent thinks that he or she can turn recording off, but in fact, the entire call is recorded no matter what the agent does with the recording button.
0	1	1	Do not record this call. The agent can start recording at any time during the call. The status of the recording button should show whether the call is currently being recorded or if recording is stopped.
1	0	1	The entire call is recorded. The agent cannot play around with the recording button. The recording button shows that this call is currently being recorded.
1	1	1	The entire call is recorded. The agent can play around with the recording button, but it does not do anything. The recording button shows that this call is currently being recorded.

Determining the Calldata.HideAniFromAgent value

The value of the HideAniFromAgent property can be verified by checking the `$calldata.HideAniFromAgent$` expression, which can be true or false. The value is defined by a combination of two conditions:

- The VCS DWORD **presentationIndicatorSupported** registry key
- The H323 presentation indicator value received from the gateway

The following table displays the value of this property depending upon these two conditions.

presentationIndicatorSupported	H323 presentation indicator	HideAniFromAgent
0 or absent	clear (no ANI restrict)	false
0 or absent	restricted (ANI restrict)	false
1 or greater	clear (no ANI restrict)	false
1 or greater	restricted (ANI restrict)	true

CCLEvent object

This object is available to all call types.

Property name	Type	Description
CCLEvent.CallType	Integer	The media type for the current call. The possible values are: 0 — Chat 2 — Email 3 — VCS
CCLEvent.MediaType	Integer	Same as CallData.MediaType.
CCLEvent.Status	Integer	The call reject value. Possible values are: 0 — Not rejected 3 — No Agents 7 — Invalid queue name

CosmoCall object

This object is available to all call types.

Property name	Type	Description
cosmocall.CallID	String	CosmoCall call identifier
cosmocall.LastError	String	This parameter is available as read only in the CallRejected Script where it represents the CMS reject reason: <ul style="list-style-type: none">• 0 — Call was queued / call is valid• 1 — Unknown error• 2 — Insufficient memory• 3 — No suitable agent was found• 4 — Invalid queue• 5 — Invalid Connection Server address• 6 — No CallId• 7 — Invalid external queue ID• 8 — DID agent is not free• 9 — CMS got an invalid EXTERNAL_AGENT_ID (ThirdPartyID)• 10 — CMS failed to locate logged-in agent marked for external routing

Property name	Type	Description
cosmocall.MarkAsAbandoned	String	If true, sets abandoned calls in rejected script to be reported as abandoned. If false, sets abandoned calls in rejected script to be reported as rejected (the default).

CosmoCall.CallbackData object

This object is available only for outgoing VoIP calls. All properties are Read only.

Property name	Type	Description
cosmocall.callBackData.DirectRouteAgentId	Integer	Returns the agent ID where the originating callback was routed to. Can be 0 if the originating callback call was routed via ACD rules.
cosmocall.callBackData.OutDialingAddress	String	Returns the designated callback address (usually phone number).
cosmocall.callBackData.ImmediateDialing	Boolean	Returns the immediate dialing flag. If set to true, this callout started automatically when the originating callback arrived at the agent.
cosmocall.callBackData.OriginatingCallID	String	Returns the call ID of the originating callback call.
cosmocall.callBackData.OriginatingAgentID	Integer	Returns the agent ID of the agent scheduling the originating callback call. Can be 0 if the callback was scheduled from a script.
cosmocall.callBackData.AttachmentUrl	String	Returns the attachment's URL of the originating callback call. Previously known as CallerURL.
cosmocall.callBackData.Message	String	Returns the message of the originating callback call. Previously known as CallerMessage.
cosmocall.callBackData.OriginatingQueueName	String	Returns the name of the queue of the originating callback call.
cosmocall.callBackData.RescheduleCount	Integer	Returns the reschedule count of the originating callback call.
cosmocall.callBackData.MaxRescheduleCount	Integer	Returns the maximum reschedule count of the originating callback call.

CosmoCall.TtsInfo object

This object is available only for VoIP and chat calls during TTS (Transfer to System) flow, if the selected EntryPoint includes a script. All properties are Read only.

Property name	Type	Description
CosmoCall.TtsInfo.blockName	String	Returns the block name used during the TTS (only if set via selected EntryPoint).
cosmocalldata.ttsInfo.ParamName	String	Returns the parameter name used during the TTS (only if set via selected EntryPoint).
cosmocalldata.ttsInfo.ScriptName	String	Returns the script name used during the TTS (only if set via selected EntryPoint).
cosmocalldata.ttsInfo.ApplicationID	Integer	Returns the application ID used during the TTS (only if set via selected EntryPoint).
cosmocalldata.ttsInfo.CallType	Integer	Returns the TTS call type - one of the following values: <ul style="list-style-type: none">• 0 - main call (all media)• 1 - outgoing TTS call (VoIP only)• 2 - incoming TTS call (VoIP only)
cosmocalldata.ttsInfo.ParentCallId	String	Returns the TTS parent call ID.

Line object

This object is only available for VCS calls and is destroyed before entering the call termination script.

Property name	Type	Description
line.ANI	String	For incoming calls: the phone number of the caller, if available. For outgoing to external number calls: the agent ANI as declared in CCaaS Admin on the agent level. If not declared on the agent level, then the ANI declared on the tenant level. If not declared on the tenant level, then the tenant name.
line.CallerID	String	The identification of the caller, if available.
line.CallingPartyID	String	The number of the caller, if available.
line.CalledPartyID	String	The number of the VCS if available.
line.CallIdentifier	String	H323 call identifier.

Property name	Type	Description
line.DeviceProfileShortName	String	Short name of the recognized device. This variable is applicable only to H.323 video.
line.DNIS	String	For incoming calls: the phone number that the caller dialed, if available. For outgoing to external number calls: DNIS is the original dialed value before dialing plan processing.
line.UUI	String	Q.931 user-to-user data field. This variable is applicable only to H.323 and not SIP calls.
line.UUI.ex	String	Q.931 user-to-user data field with specification of language and code page. This variable is applicable only to H.323 and not SIP calls.
line.RemoteIpAddressInfo	String	Contains the IP address of caller if the caller calls from IP phone or gateway if the caller calls from PSTN.

NewLine object

This object is associated with the outbound call from a call transfer. It is similar to the [Line Object](#). It is available only after a successful [Transfer Call](#) or [Supervised Transfer](#) block has been executed.

Property name	Type	Description
line.lineID	Integer	The VCS line number that the current call is using.

Session object

This object is available to all call types.

Property name	Type	Description
Session.Digits	String	The digits received from the last getDigits block.
Session.CallerID	String	Same as Line.CallingPartyID.
Session.CalledID	String	Same as Line.CalledPartyID.
Session.ID	String	Same as CosmoCall.CallID.

7: CallXML

This section contains the following topics:

- [General elements](#)
- [Call control elements](#)
- [Test and debug elements](#)
- [CallXML high level elements](#)
- [Additional CCaaS Designer elements](#)

This section describes how CCaaS Designer supports the CallXML standard and explains how to use the additional CCaaS Designer elements to customize the scripts.

CallXML is a language based on the Extensible Markup Language (XML). Using CallXML, an organization can describe an application in terms of how the call will be handled at the Web site and how the organization will interact with the caller based on voice responses. CallXML is similar to other voice markup languages such as VoiceXML and Microsoft's WTE.

CallXML is used to describe the user interface of a phone, VoIP, or multimedia application to a CallXML browser. A CallXML browser then uses that description to control and react to the call itself.

The markup language includes:

- Media action elements such as `<playAudio>` and `<recordAudio>` to describe what to present to the user during a call.
- Call action elements such as `<answer>`, `<call>`, and `<hangup>` to describe how to control and route the call itself.
- Logic action elements such as `<assign>`, `<clear>`, and `<goto>` to describe how to modify variables and interact with traditional server-side web logic such as PERL, other CGI languages, PHP, or ASP.
- Event elements such as `<onTermDigit>`, `<onHangup>` to describe how to react to things the user can do during the call, such as pressing digits or hanging up.
- Block elements that logically group actions and events together, so that one set of event handling elements can be used for several sequential actions.

Note

When specifying URLs in CCaaS Designer or scripts, you must use the forward slash character at all times. This ensures proper resolution of the URL by the interpreter.

For example:

- Correct — `http://www.somesite.com/default.htm`
- Incorrect — `http:\\www.somesite.com\\default.htm`

Use this Help in conjunction with the CallXML specification. If there is a difference, the information here supersedes the CallXML specification. Refer to the CallXML specification at http://community.voxeo.com/cxml/docs/cxml_2.0/home.htm.

Some elements of the CallXML standard are listed as *not supported*. Unsupported elements do not cause the interpreter to fail. If the interpreter processes a CallXML file that includes unsupported elements, it gracefully ignores them.

General elements

Use the general elements as follows.

assignvar

```
<assignvar="ttt"_default: ""  
    value="123">_default: "" />
```

This element follows the CallXML standard.

clear var

```
<clear var=""ttt" default: none />
```

This element follows the CallXML standard.

cleardigits

```
<cleardigits />
```

This element follows the CallXML standard.

getDigits var

```
<getDigits var="pager_msg"_default: none  
    maxDigits="9"_ default: "no limit "  
    termDigits="#*"_default: ""  
    includeTermDigit="TRUE|FALSE"_default: "false"  
    clearDigits="TRUE|FALSE"_default: "false"  
    maxTime="30s" _default: maxSilence* maxDigits  
    maxSilence="5s"_default: "3s"_ />
```

This element follows the CallXML standard.

goto value

```
<goto value="http://w.v.n/next.voxeo#block"_default: ""  
    submit="" default: ""  
    method="get" default: "get" />
```

This element differs from the CallXML standard as follows.

Submit only supports all "*" or no "" variables. You cannot specify a list of variables to send to the goto destination.

Method only supports "get".

You can go to ASP (Active Server Pages) as well as XML pages provided the ASP pages generate appropriate CallXML output so that the interpreter knows what to do next.

run value

```
<run value="http://w.v.n/next.voxeo|#block"_default: ""
    submit="*|x,y,z" default: "*"
    method="get|post" default: "get"
    var="varForReturnedSessionID"_default: "" />
```

This element is not supported.

sendEvent value

```
<sendEvent value="msg_call_answered"_default: ""
    session="sss"_ default: "" />
```

This element is not supported.

Call control elements

Use the call control elements as follows.

answer

```
<answer />
```

This element is not supported. In the CCaaS system the call is already answered before the first script event is fired. Therefore there a script control to answer a call is not required.

call value

```
<call value="pstn:18314395130" default: ""
    callerID="pstn:1234567890" default: "session.callerID" maxTime="30s" default:
    "30s" />
```

This element is not supported. Use the transferCall element to dial out.

conference targetSessions

```
<conference targetSessions="sessionID1, sessionID2" default: ""
```

```
termDigits="#" default: "" />
```

This element is not supported.

hangup

```
<hangup />
```

This element follows the CallXML standard.

playAudioformat

```
<playAudioformat="audio/wav" default: "audio/wav"  
value="http://www.ttt.com/sample.wav" default: ""  
termDigits="1234567890*#" default: ""  
clearDigits="TRUE|FALSE" default: "false" />
```

This element differs from the CallXML standard as follows.

The only supported format is **audio/wav**. The audio files must be in the format of Microsoft wave file 8kHz-16bit-mono.

The **value** attribute can be one of the following formats:

- A fully qualified file name. This means that the wave file is located on the local hard drive of each VCS (VoIP Connection Server). The path specified must be accessible to every VCS in the CCaaS system.
- A URL. This means that the wave file is located on a web server. All wave files that are retrieved from a URL are cached locally on each VCS. This limits the amount of data that needs to be downloaded from the web servers.
- A CCaaS file descriptor. This means that the wave file is stored in the CCaaS database. The file descriptor name must match the name that is specified in CCaaS Admin for the particular file.

playDateformat

```
<playDateformat="ymmddhhnn" default: as shown below  
value="9901111200" default: ""  
termDigits="1234567890*#" default: ""  
clearDigits="TRUE|FALSE" default: "false" />
```

The supported formats are:

- yyyyymmddhhnn
- yyyyymmdd
- yymmddhhnn
- yymmdd
- mmddhhnnss
- mmddhhnn

- hhmmss
- hhmm

Where:

- yyyy is a 4 digit year
- yy is a 2 digit year
- mm is a 2 digit month
- dd is a 2 digit day of the month
- hh is a 2 digit hour
- nn is a 2 digit minute
- ss is a 2 digit seconds

playMoneyformat

```
<playMoneyformat"us" default: "us"
  value="1.25" default: ""
  termDigits="1234567890*#" default: ""
  clearDigits="TRUE|FALSE" default: "false" />
```

The only supported format is US (United States).

playNumber

```
<playNumber
  format="digits|number" default:0 "digits"
  value="12345" default: ""
  termDigits="1234567890*#" default: ""
  clearDigits="TRUE|FALSE" default: "false" />
```

This element follows the CallXML standard.

recordAudioformat

```
<recordAudioformat="audio/wav" default:"audio/wav"
  value="msg.wav" default: ""
  termDigits="1234567890*#" default: ""
  clearDigits="TRUE|FALSE" default: "FALSE"
  maxTime="30s" default: "30s"
  maxSilence="5s" default: "3s"
  beep="1" default: "0" />
```

This element differs from the CallXML standard as follows.

The only supported format is **audio/wav**. The audio files are recorded in the format of Microsoft wave file 8kHz-16bit-mono.

The **value** attribute can be one of the following values:

- A fully qualified filename. This means that the file is recorded to the local hard drive of the VCS servicing the call in the directory specified. In this case the file is *not* deleted at the end of the call.
- A non-qualified filename. This means that the file is recorded to the local hard drive of the VCS servicing the call in the Windows %TEMP% directory. In this case the file *is automatically* deleted at the end of the current script.

The **beep** attribute can be one of the following values:

- 0 — for no beep
- 1 — for a short beep
- 2 — for a normal length beep
- 3 — for a long beep

text format

```
<text format="?" default: "connected"
    termDigits="#" default: ""
    clearDigits="TRUE|FALSE" default: "false" >
```

This element is not supported.

waitForConferenceEnd

```
<waitForConferenceEnd />
```

This element is not supported.

Test and debug elements

The **simline value** element writes to the log file and event log as follows.

```
<simline value="Any text for the system log" />
```

This element writes to a log file on the local hard drive of the CCaaS Connection Server servicing the call. The log files are located in the %TEMP%/CosmoDesigner directory.

If the beginning of the string in the value attribute starts with **ERROR:** or **WARNING;**, then the message is also written to the Windows application Event Log of the Connection Server.

CallXML high level elements

The following high level elements follow the CallXML standard.

block label

```
<block label="anyname" default: ""
  repeat="?" default: "1"
  clearDigits=>"TRUE|FALSE" > default: "false"
```

menu label

```
<menu label="main_menu" default: ""
  repeat="3" default: "1"
  format="audio/wav" default: "audio/wav"
  value="http://w.v.n/msg.wav" default: ""
  clearDigits="TRUE|FALSE" default: "false"
  termDigits="1234567890*#" default: ""
  maxTime="15s" default: "30s" >
```

inputAudio label

```
<inputAudio label="leave_message" default: ""
  repeat="3** default: "1"
  var="myaudio** default: ""
  value="http://w.v.n/msg.wav" default: ""
  format="audio/wav" default: "audio/wav"
  termDigits="1234567890*#" default: ""
  clearDigits="TRUE|FALSE" default: "false"
  maxTime="15s** default: "30s"
  maxSilence="5s" default: "5s"
  beep="TRUE|FALSE" default: "true" >
```

inputDigitslabel

```
<inputDigitslabel="input_pin" default: ""
  repeat="3" default: "1"
  var="pager_msg" default: ""
  format="audio/wav" default: "audio/wav"
  value="http://w.v.n.msg.wav" default: ""
  termDigits="1234567890#*" default: ""
  clearDigits="TRUE|FALSE" default: "false"
  includeTermDigit="TRUE|FALSE" default: "true"
```

```
maxDigits="4" default: no limit
maxTime="15s" default: maxSilence * maxDigits
maxSilence="5s" default: "3s" >
```

Additional Designer elements

Use the additional CCaaS Designer elements as follows.

addSkillvalue

```
<addSkillvalue="defaultSkill" default: "" />
```

Use this element to add CCaaS skills to the current call.

arrivalTimeBiasvalue

```
<arrivalTimeBiasvalue="30" default: "" />
```

Use this element to modify the arrival time bias for the current call. The lower the value, the higher you move the call in the queue.

callerMessagevalue

```
<callerMessagevalue="This is a test call." default: "" />
```

Use this element to configure a custom message to be displayed to the agent when the caller is connected.

callerNamevalue

```
<callerNamevalue="John Doe" default: "" />
```

Use this element to configure the caller's name.

callSetNamevalue

```
<callSetNamevalue="defaultQueue" default: "" />
```

Use this element to identify the CCaaS queue to submit the current call to. If this element is identified multiple times, it overrides all previous settings. Only the last setting takes effect.

callerURLvalue

```
<=http://www.cosmocom.com default: "" />
```

Use this element to configure the caller's URL. When the caller is connected to the agent, this URL can be displayed to the agent.

conditionalexpr

Use this element to control the flow of the application. It has two forms.

In the following form the expression evaluates to either True or False. Thus the only "Case" sub-elements specified are True and False. If the expression evaluates to True, then execution jumps to the "value" attribute specified in the True Case. Similarly, if it evaluates to False, then execution jumps to the "value" attribute specified in the False Case.

The expression is in the form of a standard VBScript expression. Any variable names referenced in the expression must have been previously assigned using the "assign" element. The "value" attributes are in the same form as the value attribute of the "goto" element.

```
<conditionalexpr="a > 3" default: "" >
value="#nextBlock" default: "" />
value="Ivr.xml" default: "" />
```

In the following form the "conditional" element acts more like a switch/case statement. In this statement the expression can evaluate to multiple values. It is usually just a variable name.

```
<conditionalexpr="line.DNIS" default: "" >
value="#nextBlock" default: "" />
value="Sales.xml" default: "" />
value="Support.xml" default: "" />
*...
*value="Info.xml" default: "" />
```

optionalParametervar

```
<optionalParametervar="VarName" default: ""
value="abc123" default: "" />
```

Use this element to configure CCaaS optional parameters. Optional parameters are variables that are assigned to the caller and can be accessed at any time during the call, even after it is connected to an agent. Typically additional information about the caller is set in optional parameters, such as, account number, or phone number.

removeSkillvalue

```
<removeSkillvalue="defaultSkill" default: ""/>
```

Use this element to remove CCaaS skills from the current call.

sendEmail server

```
<sendEmail server="MailServer" default: ""
recipient=you@somewhere.com default: ""
cc=aaa@cosmocom.com default: ""
```

```

bcc=bbb@yahoo.com default: ""
sender=me@msn.com default: ""
subject="Test email" default: ""
body="Hi, just testing." default: ""
attachment="test.wav" default: "" />

```

Use this element to send an email. It uses SMTP (Simple Mail Transport Protocol).

The following table describes the **sendEmail server** attributes.

Attribute	Description
server	The SMTP mail server name.
recipient	The email address of the person to send the email to. For multiple recipients separate the address using a semi-colon (;).
cc	The email address of any carbon copy recipients. For multiple recipients separate the address using a semi-colon (;).
bcc	The email address of any blind carbon copy recipients. For multiple recipients separate the address using a semi-colon (;).
subject	The subject of the email.
body	The body of the email.
attachment	The names of files to attach to the email. . For multiple files, separate them using a semi-colon (;). The files must reside on the local hard drive of the CCaaS Connection Server.

unQueue and reQueue

```

call cosmocall.unqueue
'now that the call is unqueued you can do whatever
'such as 'set cosmocall.terminate = true and let the call be terminated immediately
'or change the skills and do this:
call cosmocall.requeue
'which resubmits the call (possibly rejected or rerouted...)

```

Use **unQueue** and **requeue** in the Music on Hold While in Queue script. These functions allow the script to remove a call from a queue and re-queue the call if desired. The unQueue function immediately removes a call from the queue. An example of how this may be used is to unqueue the call and send it to voicemail. The reQueue function resubmits the call to the system by calling the Call Arrival script. These features may be applied to any call type. There are no arguments for either the unQueue or reQueue functions.

See also the Note for [Unqueue](#).