# Contact Center:

# Service Provider

# Admin Help - Tenant

## Version 7.4

# Contents

1: Intro

What's new

Legal disclaimer

Support

2: Overview

Log in

Password expired message

Additional login messages

Reseller view

Select the tenant to manage

Return to the Tenants page

Tenant view

Navigation pane

List view

Card view

Item properties

Mandatory fields

Error messages

Single item selection

Multiple item selection

How to search

Search generic lists

Search card-type lists

Search in the Choose items dialog

How to filter

How to use wildcards

Multiple administrators

Simultaneous updates

Simultaneous update and delete

Change your password

Log out

Additional logout messages

Session expiry

3: Configure tenant properties

Manage core tenant properties

Manage tenant URLs

Manage tenant parameters

Manage tenant SIP parameters

Manage other tenant properties

Manage tenant MS Teams parameters

CCSP Admin Help - Tenant - 7.4

4: Configure tenant resources

Manage accounts

View the Accounts list

Add an account

Edit an account

Delete an account

Manage applications

View the Applications list

Add an application

Creating rules

Set the default application

Edit an application

Delete an application

40

Manage block codes

41

View the Block Codes list

41

Add a block code

41

Edit a block code

41

Delete a block code

42

Manage campaigns

42

Manage Dashboard views

43

View the Dashboard Views list

43

Edit a Dashboard view

43

Manage the DNIS pool

44

View the DNIS Pool list

44

Add a DNIS pool

44

Assigning a DNIS pool

45

Delete a DNIS pool

45

Manage entry points

46

View the Entry Points list

46

Add an entry point

46

Edit an entry point

47

Delete an entry point

47

Manage folders

48

View the Folders list

48

Add a folder

48

Edit a folder

49

Delete a folder

49

Manage groups

51

View the Groups list

51

Add a group

51

Clone a group

52

Edit a group

52

Delete a group

52

Manage interaction profiles

53

View the Interaction Profiles list

53

Add an interaction profile

53

CCSP Admin Help - Tenant - 7.4

3

Edit an interaction profile

54

Delete an interaction profile

55

Manage legacy licenses

56

View the legacy Licenses list

56

View contact center licenses

56

Allocate legacy licenses to agents

57

Manage mailboxes

58

View the Mailboxes list

58

Add a mailbox

58

Test mailbox connectivity

60

Edit a mailbox

61

Delete a mailbox

61

Manage permission profiles

62

Default permission profiles

62

View the Permission Profiles list

63

Add a permission profile

64

Edit a permission profile

64

Delete a permission profile

64

Manage personnel

65

View the Agents list

65

Add an agent

66

Add a phone agent

67

Clone an agent

67

Edit an agent

67

Delete an agent

67

Manage personnel properties

68

Manage personnel phone settings

70

Manage personnel parameters

72

Manage phone aliases

73

Use wildcards in phone aliases

73

View the Phone Aliases list

73

Add a phone alias

74

Edit a phone alias

74

Delete a phone alias

74

Manage queues

75

View the Queues list

75

Add a queue

75

Clone a queue

76

Edit a queue

77

Delete a queue

77

Manage release codes

78

View the Release Codes list

78

Add a release code

78

Edit a release code

78

Delete a release code

79

Manage skills

80

View the Skills list

80

CCSP Admin Help - Tenant - 7.4

4

Add a skill

80

Clone a skill

81

Bulk assign agents to skills

81

Edit a skill

81

Delete a skill

81

Manage SSO permissions

82

View the SSO Permissions list

82

Add an SSO permission

82

Edit an SSO permission

83

Delete an SSO permission

83

Manage teams

84

View the Teams list

84

Add a team

84

Bulk assign agents to teams

84

Edit a team

85

Delete a team

85

Manage time zones

86

View the Time Zones list

86

Add a time zone

86

Edit a time zone

86

Delete a time zone

87

Manage wrap-up codes

88

View the Wrap-Up Codes list

88

Add a wrap-up code

88

Edit a wrap-up code

88

Delete a wrap-up code

89

CCSP Admin Help - Tenant - 7.4

5

# 1: Intro

This Help is for CCSP tenant administrators of a contact center who configure and manage their tenant

properties and tenant resources.

## What's new

Version 7.4

Personnel properties — added property to allow the agent to request a chat consultation during a call.

Requires HF74-66036. The list of available permission profiles to assign to the agent is limited to the permission

profiles that contain only the groups that are included in your own profile. Requires HF74-64411.

Manage SSO permissions — added entity to enable administrators to control the users that can access one or

more custom SSO applications. Requires HF74-61144.

Log in and Log out — added process for CCSP single sign-on. Requires HF74-61362.

Reseller view and Navigation — added reseller administrator.

Manage tenant parameters — added parameters for the use and definition of an external Identity Provider,

and the lifetimes of the CCSP SSO login session and its access and refresh. The external IdP parameters are

only available for editing by landlords. Resellers and tenant administrators are only able to edit the lifetime

parameters. Requires HF74-61362.

Manage mailboxes — added mailbox connectivity test for MSR DB. Requires HF74-61432.

## 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

Reseller view

Tenant view

Navigation pane

List view

Card view

Item properties

How to search

How to filter

How to use wildcards

Multiple administrators

Change your password

Log out

CCSP Admin Help - Tenant - 7.4

7

## Log in

The log in process is slightly different depending on 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 CCSP 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.

2. Click Go or press Enter. The Welcome to CCSP Admin page appears.

3. Click Log in. If you are already logged in, the main CCSP Admin page appears.

4. If you are not already logged in, the Auth Server Login Selector page appears.

5. Type your User name in User Principal Name (UPN) format, for example, someone@domainname.com,

and click Continue.

6. The next step depends on the identity provider configured by your administrator:

O For Auth Server, the AuthServer login page appears. Type your Password and click Log in.

Alternatively, to change the username, click Back to return to the previous page. The main CCSP Admin

page appears. If necessary, see Password expired message below.

O For ADFS, you are redirected to the ADFS login page to authenticate. Please consult your local procedures for how to proceed from this point.

O For Enghouse Identity Server (EIS), you are redirected to the EIS login page. Type your Password and click Login. The main CCSP Admin page appears.

7. If necessary, see Additional login messages below.

Next, see Reseller view or Tenant view.

## 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 Auth Server login 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.

CCSP Admin Help - Tenant - 7.4

8

## Additional login messages

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

* Failed to get authorization for Admin use. Your user profile is not permitted to use the CCSP Admin application. Only landlord, reseller, and tenant administrators are permitted to use the CCSP Admin 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 Auth Server Login Selector page to log in to the CCSP client application.
  + You entered a username on the Auth Server Login Selector 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 CCSP Admin until the issue is resolved.

CCSP Admin Help - Tenant - 7.4

9

## 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 Tenant view.

## Return to the Tenants page

To return to the Tenants page, click the left arrow in the top left corner.

CCSP Admin Help - Tenant - 7.4

10

## Tenant view

After logging in, tenant administrators see their Home page.

Some items can be viewed but not edited or deleted by tenant administrators. For details, see Configure tenant

properties.

CCSP Admin Help - Tenant - 7.4

11

## Navigation pane

Use the navigation pane as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| Icon | Description | Visible to  reseller | Visible to  tenant |
|  | Tenants — click to select the tenant to manage. | Yes | No |
|  | Home page — click to manage tenant properties. | Yes | Yes |
|  | Organization — click to manage personnel (agents), groups, teams, permission profiles, skills, or SSO permission profiles. | Yes | Yes |
|  | Routing — click to manage the DNIS pool, queues, folders, entry  points, applications, campaigns, release codes, or mailboxes. | Yes | Yes |
|  | Miscellaneous — click to manage accounts, block codes, Dashboard  views, interaction profiles, licenses (legacy), phone aliases, time zones,  or wrap-up codes. | Yes | Yes |

CCSP Admin Help - Tenant - 7.4

12

List view

Use the list view as follows.

To

Action

Add an item

Click

.

Delete an item

Select the item's check box and then click

.

Delete multiple items

Select multiple check boxes and then click

.

Bulk assign agents to

multiple skills or teams

Select multiple check boxes, click

, and then click Assign Agents to

[item].

Clone agents, groups,

queues, or skills

Select multiple check boxes, click

, and then click Clone [item].

Search for items

Type the criteria in the search box.

Sort

Click the column heading. The

indicates the sort column and sort direction.

View the next page

Scroll to the end of the page.

View/edit an item

Click the item in the list to view/edit its properties in a dialog or, for tenants,

personnel, and permission profiles, on tabs.

View counters

Total records appear in brackets after the resource name, for example, Teams

(5).

The number of:

l

Filtered records appear in brackets after the resource name, in front of the

total records, for example, Teams (2/5).

l

Selected records appear in brackets at the top of the check box column, for

example, (1).

CCSP Admin Help - Tenant - 7.4

13

Card view

Use the cards view as follows.

To

Action

Add an item

Click

.

Delete an item

Select the item's check box and then click

.

Delete multiple

items

Select multiple check boxes and then click

.

Hide the list

Click

.

Search for

items

Type the criteria in the search box. See How to search.

Sort

Click Sort by, select the field to sort by, and then click A-Z. The

indicates the sort

direction.

View the next

page

Scroll to the end of the page.

View/edit an

item

Click the item's card to view/edit its properties on tabs.

View counters

Total records appear in brackets after the resource name, for example, AGENTS (29).

The number of:

l

Filtered records appear in brackets after the resource name, in front of the total

records, for example, AGENTS (5/29).

l

Selected records appear in brackets at the top of the check box column, for example,

(3).

CCSP Admin Help - Tenant - 7.4

14

Item properties

Mandatory fields

Error messages

Single item selection

Multiple item selection

After opening the relevant resource item list, click the particular item in the list to view/edit its properties in a

dialog or, for tenants, personnel, and permission profiles, on tabs.

The following example shows the Group: G1 dialog.

Mandatory fields

Mandatory fields are marked with a red asterisk (\*).

CCSP Admin Help - Tenant - 7.4

15

Error messages

Error messages are displayed when you click OK, either at the top of the screen (error from server) or below

the field (client validation).

Single item selection

To select a single item:

l

Select the item by clicking it or pressing Enter.

l

Search by typing the item's name.

To remove a single item, unselect the item by clicking it or pressing Enter.

CCSP Admin Help - Tenant - 7.4

16

Multiple item selection

To select multiple items:

l

Select each item by clicking it or pressing Enter. Each selected item appears in a bubble.

l

Search by typing the item's name.

To remove multiple items, remove each item by clicking the X in its bubble.

When there are more than 50 items to select from and for agents,

appears next to the field.

Click

to open the Choose items dialog.

To select:

l

A single item, click the row and then click OK.

l

Multiple items, select the appropriate check boxes and then click OK.

To remove items, clear the appropriate check boxes and then click OK.

The counters show the total number of items, the number displayed, and the number selected. For details, see

the counters in Lists.

CCSP Admin Help - Tenant - 7.4

17

How to search

Search generic lists

Search card-type lists

Search in the Choose items dialog

You can search for items in generic lists, card-type lists (Agents and Permission Profiles), and the Choose items

dialog.

Each page shows up to 50 items and the total number of items appears in brackets, for example, AGENTS (210).

When you search for items or sort the list by clicking the column heading, the total number of items are

searched or re-ordered.

To view the next group of 50 items, scroll to the end of the page.

To search, type the search criteria in the search box. After 1 second of inactivity, the search starts automatically.

You cannot type additional criteria while the search result is loading.

Search generic lists

The following example searches for teams that contain "V". The number of filtered teams that match the

criteria appears in brackets, in front of the total, for example, Teams (2/6).

Search card-type lists

You can search for agents by their login ID, first name, last name, description, skills, or groups. To search by

their full name, enter [first name]+[last name], for example:

The following example searches for agents that contain "ag". The number of filtered agents that match the

criteria appears in brackets, in front of the total, for example, AGENTS (12/17).

CCSP Admin Help - Tenant - 7.4

18

Search in the Choose items dialog

The following example searches for agent names that contain "ag".

CCSP Admin Help - Tenant - 7.4

19

How to filter

Your permission profile determines which groups and teams you can filter by.

1.

Click

Filter.

2.

In the Filter by dialog, select the criteria to filter by.

Note

l

Groups, Teams, Skills, Profiles, Interaction Profiles, and IsCampaignable are AND conditions. The

selected items inside each condition are OR conditions.

l

To restart the selection, click Clear all.

l

To close the Filter by dialog, click outside it.

The following example filters for agents that are in Groups 1 or 2, AND in Teams 1 or 2, AND have the S3

skill.

3.

Click Apply. The

Active filter icon appears above the filtered list.

CCSP Admin Help - Tenant - 7.4

20

How to use wildcards

A common way to indicate a group of numbers that all have a similar pattern is by using wildcards.

If you have a series of numbers that all begin with 1631940xxxx, only the last four digits are different. To

indicate this, you can use a question mark (?) to replace the digits which can vary. For example, 1631940????

can be used to represent 16319401234, 16319409890, or 16319403214.

The ? character can appear anywhere in a string. For example, 16319405?00 could represent 16319405000,

16319405100, 16319405200, 16319405300, 16319405400, 16319405500, 16319405600, 16319405700, 16319405800

or 16319405900.

An asterisk (\*) can be used to replace any combination of characters, in any quantity. For example, 1631940\*

can be used to represent 16319401234, 16319409890, 16319401, 163194000000000.

CCSP Admin Help - Tenant - 7.4

21

Multiple administrators

Depending on how your contact center is configured, more than one administrator user may be logged on

simultaneously.

Simultaneous updates

If two administrator users open an item for update, the first user to click OK in the dialog succeeds with the

update. When the other user clicks OK, the message The record was updated by another user appears in the

dialog.

Simultaneous update and delete

If an administrator user opens an item for update, and another administrator user deletes the item, when the

first user clicks OK, the message The item does not exist appears in the dialog.

If an administrator user deletes an item, and another administrator user tries to update the item by clicking it

in the list, the warning message The item does not exist appears.

CCSP Admin Help - Tenant - 7.4

22

Change your password

1.

Log in to CCSP Admin.

2.

Click the down arrow after your user name and then click Change User Password.

3.

In the Change password dialog, type your new password and confirm it. Do not use the double quote (")

character or spaces.

4.

Click OK.

CCSP Admin Help - Tenant - 7.4

23

Log out

Note

If you are using an external identity provider such as ADFS or EIS, when you click Log out of your CCSP

session, you are also logged out of your external identity provider.

1.

At the top right of the main CCSP Admin page, click Logout.

2.

Do one of the following:

o

If you want to return to CCSP Admin, click Return to Login page.

o

If you have finished working with CCSP, click Log out of your CCSP session.

Additional logout messages

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

l

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 CCSP session again.

The application will return to the login page. Depending on the problem resolution, this may take slightly

longer than normal.

Session expiry

Your CCSP application session will become invalid when:

l

The login session time exceeds the maximum CCSP login session lifetime specified by your administrator.

l

You log out of your CCSP session using another application.

l

You clear all cookies in the browser.

When your CCSP 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 CCSP login session lifetime, you only have to enter your username on the Auth Server Login

Selector page. You will then automatically be authenticated against your external identity provider.

CCSP Admin Help - Tenant - 7.4

24

# 3: Configure tenant properties

You can configure the following tenant properties:

l

Core

l

URLS

l

Parameters

l

SIP parameters

l

Others

l

MS Teams

The landlord administrator adds the tenant and then completes all the required fields in the Tenant Properties

dialog.

Tenants can view but not change the following properties:

l

ID

l

Name

l

Account name

l

Short name

l

UPN suffix

l

Trunk group

l

All of the URLs

l

All of the SIP parameters

l

Integration of external media servers and selection of media server type

Tenants cannot change the properties shown as (landlord only) in this Help.

To configure tenant properties:

1.

On the navigation pane, click

Home page. The Tenant Properties page appears.

2.

Edit or view the properties on the Tenant Properties page. The properties are arranged on the following

tabs:

o

Core

o

URLS

o

Parameters

o

SIP parameters

o

Others

o

MS Teams

3.

Click Save changes.

CCSP Admin Help - Tenant - 7.4

25

Manage core tenant properties

On the Tenant Properties page, on the Core tab, view/enter the following information:

l

Name (required) — the name of the tenant. This name can be different from the Account name.

l

Account name — the name to use for administration only. This value is case-sensitive.

l

Short name (required) — the tenant’s public folder name. The name should not contain any spaces,

underscores, or special characters.

l

Address — the tenant's billing address.

l

Contact person — the main contact person's name.

l

Phone — the main contact person’s phone number or tenant's main number.

l

Email address — the tenant's email address.

l

UPN suffix (required) — the email and logon suffix for all users in the tenant's contact center.

l

Disable historical information for this tenant — whether to disable logging of the tenant's call data to

Historical Reports.

l

Outgoing and direct calls wrap-up timeout (sec) — for how long to wait for agents to exit wrap-up before

automatically timing out the wrap-up of an outdialed or a directly dialed incoming call. The default is 0 (no

timeout).

l

Agent to agent call timeout (sec) — for how long to wait for users to answer before automatically

canceling an agent to agent call request. This value must be between 10-3600. The default is 30.

l

Call answered script timeout (sec) — for how long to run the Call Answered Script. The script stops

running when the timeout value is reached or when the agent hangs up the call. When set to 0, the agent

that has been routed the call waits for the answer script to complete for as long as is needed. The agent can

hang up the call while the call answer script runs, whether or not there is a timeout. The default is 0 (no

timeout).

l

Enable call answered script for PD campaign — whether to play the Call Answered Script for outbound

dialing campaigns. Playing this script causes a two second delay between placing the Predictive Dialing (PD)

call and it being answer by an agent.

l

Media types — the media types available to the tenant.

Note

CCSP does not support calendar invites as a method of call delivery.

l

Aging timeout interval (mins) — for messaging-related incoming calls:

o

For how long a deferred email/voicemail call waits for processing before the system reroutes the call

o

For how long the system waits before removing a call related either to a tenant that is unassigned from

MS or an MS that is down. This allows grace time to assign another MS for a tenant or to wait for MS

recovery after an unexpected failure.

o

The default is 480 minutes.

CCSP Admin Help - Tenant - 7.4

26

Manage tenant URLs

Landlord only can change these values.

On the Tenant Properties page, on the URLs tab, view/enter the following information:

l

Intelligent outdialing URL — the URL used by the Dialing Plan:

o

For the advanced dial plan, set to http://<servername>/AdvancedDialPlan/CheckPlan

o

For the legacy dialing plan, set to http://<servername>/DialingPlan/CheckPlan.asp

l

Historical Reports URL — the URL the tenant uses to access Historical Reports. This URL must be a Fully

Qualified Domain Name (FQDN). For more information, see the CCSP Historical Reports User Guide.

l

RTR URL — the URL the tenant uses to access Dashboard. This URL must be a FQDN and must end with a

forward slash (/).

Note

l

The URL for Dashboard (2022) uses Dashboard instead of CosmoDashboard.

l

To use the new Dashboard (2022) gadget with multiple browser support, append the option

multibrowser to the Real Time Reports URL value, for example:

http://<servername>/Dashboard/?multibrowser

l

CCSP Designer URL — the URL used by the tenant to access the Designer application. This URL must be

made available to all tenants. For more information, see the CCSP Designer Help.

Note

In addition to the server that contains the Designer application, the tenant must have access to a server

to which they can publish the CallXML scripts created in Designer. This server can be a hosted service of

the system administrator, or on the tenant’s own server.

l

Replay URL — the URL of the web server where interaction recordings for the tenant are posted. This URL

must be the same URL established during installation. For more information, see the CCSP Installation Guide.

l

Voicemail URL — the URL the tenant uses to access voicemail. Set to http://<servername>/designer/Scripts

l

Elsbeth PowerContact server URL — for Elsbeth 6.x, the URL to the WebAPI service of the EPCServer

instance assigned to the tenant. From Elsbeth 9.x, the URL to the SOAP (WCF) service of the EPCServer

instance assigned to the tenant. Required only when tenant is running in Elsbeth dialer mode. For more

information, see the CCSP Dialer Guide for Elsbeth.

CCSP Admin Help - Tenant - 7.4

27

Manage tenant parameters

Note

CCSP UI was previously called TouchPoint.

On the Tenant Properties page, on the Parameters tab, view/enter the following information:

l

Block agent-to-agent calls — whether to prevent the tenant's agents calling one another.

l

Caller ID — the identity the tenant chooses to present to a called party. This value can be text or a

telephone number.

Note

l

Many called party phones can display only numeric information.

l

Many countries prohibit insertion of erroneous information in this

field.

l

Phone ANI — the identifier used for call routing or billing. This is an optional field that can be required by

your telephone carrier. ANI is different from Caller ID, but might be used by a telephone carrier to display

caller identity to the called party. The value of this field is usually provided by the telephone company and is

numeric text.

l

Maximum skill weight — the maximum value for skill weighting. The default is 100. For more information,

see Weighting skills in the CCSP Tenant Administrator Guide.

l

CallData publishing mode (required) — whether call data is published to the historical database at call

arrival, call termination, or as soon as it is created (All). The default is Call arrival only.

l

Predictive dialing — whether to allow outbound dialing campaigns. For more information, see the CCSP

Dialer Guide for Elsbeth/Sytel.

l

Preview dialing — whether to allow preview dialing, required by the agent to elect to accept an outbound

call. For more information, see the CCSP Dialer Guide for Elsbeth/Sytel.

l

Massive IVR dialing — whether to allow IVR campaigns where the numbers are dialed sequentially without

trying to determine whether an agent is available to take the call. For more information, see the CCSP Dialer

Guide for Elsbeth/Sytel.

l

Block xfer/conf to agent — whether to prevent the tenant's agents from transferring or conferencing calls

to one another.

l

Impersonate CCSP Connector — whether to pass the user name and password to CCSP Connector. If

selected, the user name and password are passed to CCSP Connector, which starts automatically if the agent

has been logged on. If not selected, the user name and password are not passed to CCSP Connector and the

user has to log on again after logging on to CCSP UI.

l

Block call/xfer to system — whether to prevent the tenant from transferring calls to entry points, including

returning calls to the IVR. For more information, see Manage entry points.

l

Agent list scope (required) — the scope of the list from which agents select other agents for conferencing

or call transfer. In the transfer dialog, CCSP UI displays only team members for team specific scope and only

group members for group specific scope. The default is Tenant-wide.

CCSP Admin Help - Tenant - 7.4

28

l

Hide agent names in RTR — whether to replace agent names in real-time reports with random numbers.

Used to comply with some regulations.

l

Inform agent when being monitored — whether to display a message in CCSP UI informing agents they

are being monitored when supervisors start monitoring.

l

Reuse CSA media leg — whether to allow the Circuit Switched Hybrid Agent (CSA) to put a call on hold

without disengaging the call. If selected, the CSA can put a call on hold without disengaging the call. The CSA

can be an agent using an external circuit switched telephone or external IP telephone. Requires that every

VCS can communicate with any gateway in the system. If not selected, the VCS tears down the CosmoCall leg

of the call.

l

Screen recording rate (fps) (required) — how fast to capture agent screens. Greater capturing speed

(bigger fps value) provides higher quality recordings, but requires more storage and may affect bandwidth

requirements in extreme cases. The default is 1.

l

Screen recording quality (%) (required) — deprecated.

l

Mailbox cleanup (days) — how long to keep voicemail that is marked as read or deleted in the database.

Agents can delete or keep voicemail that they have listened to. Deleted voicemail is marked as deleted in the

recording database. Voicemail that was read but not deleted is marked as read. Unread voicemail is marked

as new. Voicemail originates from the agents' direct telephone extensions. The default is 30.

l

Integration with external media servers — whether to use the speech capabilities in Designer. If

enabling, you must also select an External media server type (see below).

l

Email thread search mode (required) — whether and how to search for email history. The options are:

o

Automatic — the feature is always available to the agents.

o

On Demand — an agent can turn on this feature when they need it. The advantage over Automatic is to

save storage in the email database.

o

Disabled (default) — this feature is unavailable to agents.

l

External media server type (required) — the protocol used with the external media server. The options

are:

o

None (default)

o

HP OCMP — use for HP OCMP external media server.

o

Generic MRCPv1 — sessions to the external media server are established using the RTSP protocol.

o

Generic MRCPv2 — sessions to the external media server are established using the SIP protocol.

l

Dashboard SMTP server — the SMTP server name or IP address for the DashboardEmailService to send

email alerts from. Usually configured by the tenant administrator. The default is localhost.

l

Maximum call rejects — the number of call reject occurrences for a call. The default is 0 (disabled).

l

Maximum pickup failures — the number of pickup failure occurrences for a call. The default is 0 (disabled).

l

Max time allowed in queue before email call auto termination (days) — the maximum time in the

queue (between 0 and 30 days) before automatically terminating an email conversation. The default is 0

(disabled).

l

Action to take on MSR item when email call is auto terminated (required) — whether to move the email

item to the Rejected folder, move it to the Deleted folder, or delete it permanently. The default is Move to

Rejected folder.

l

Max time allowed in queue before callback call auto termination (days) — the maximum time in the

queue (between 0 and 30 days) before automatically terminating a callback call. The default is is 0 (disabled).

CCSP Admin Help - Tenant - 7.4

29

l

Action to take on MSR item when callback call is auto terminated (required) — whether to move the

callback item to the Rejected folder, move it to the Deleted folder, or delete it permanently. The default is

Move to Rejected folder.

l

Max time allowed in queue before voicemail call auto termination (days) — the maximum time in the

queue (between 0 and 30 days) before automatically terminating a voicemaill call. The default is is 0

(disabled).

l

Action to take on MSR item when voicemail call is auto terminated (required) — whether to move the

voicemail item to the Rejected folder, move it to the Deleted folder, or delete it permanently. The default is

Move to Rejected folder.

l

Recording format (required) — the recording format. The options are:

o

GSM 6.10 (default) — the GSM mono recording is stored in the recording database.

o

PCM 16 bit stereo — the recording is converted into a stereo format and saved into the directory

configured as the recording storage folder.

o

Both — the GSM mono recording is stored in the recording database and the stereo recording is saved

into the directory configured as the recording storage folder.

Note

Stereo recording is not stored in the recording database and cannot be accessed using Replay.

l

Recording storage folder — the target directory for all stereo format recordings. The default is blank.

l

Enable KCO setting — whether to enable agents to remain connected to CCSP through an alternative

device for the duration of their shift. All actions are done on the screen, so agents do not need to answer or

hang up calls on the alternative device.

l

SAML fingerprint — the fingerprint that identifies the tenant.

l

Recording encryption public key — the thumbprint of the tenant's public encryption key. The value can

contain spaces or dashes.

l

Recording encryption key server URL — the HTTPS URL of the key server web service, for example:

https://de-keyserv/keyserver/getkeys.aspx.

l

Encryption enabled — whether to enable encryption of recordings by the DTR servers.

l

Encrypted recording restrictions (required) — the restrictions that apply to supervisors when playing back

encrypted recordings. The default is Playback and save.

Note

The Encrypted recording restrictions per agent parameter value set in Personnel properties

overrides this value, unless the Personnel properties value is set to None. For more information, see

Playing back encrypted recordings in the CCSP Tenant Administrator Guide.

l

SBC IP address/hostname — the IP Address or hostname of SBC used as bridge between Agent and VCS.

Can be the address of the load balancer used to distribute connection over different SBC devices. To use

hostname, the setting for the DNS machine used to resolve the entry must exist on the Agent local machine.

l

SBC plain transport port — the port on the SBC to which connection is made from the Agent leg when

SRTP is not used. The Use SBC as proxy parameter must be enabled. The default is 5060.

l

SBC secure transport port — the port on the SBC to which connection is made from the Agent leg when

SRTP is used. The default is 5061.

CCSP Admin Help - Tenant - 7.4

30

l

Use SBC as proxy — whether to use SBC as the intermediate device between the Agent and VCS leg when

SRTP is not used.

l

Use SRTP — whether to use SRTP on the Agent leg. If SRTP is used, you don't need to select Use SBC as

Proxy, but if both are selected, Use SRTP overrides Use SBC as Proxy.

Note

If Use SRTP is not select for the tenant but is selected for an agent, SRTP is used for that agent. See

Manage personnel phone settings.

l

SRTP CA certificate name — the file name of Central Authority certificate used to validate TLS connection

from Agent to SBC, when calls are established. SRTP CA Certificate Name must be the same as the name of

relevant certificate file added under Tenants Information. TLS 1.0 is supported for SIP negotiation between

Agent and SBC.

l

Script log verbosity (landlord only) (required) — the script log verbosity. The default is Inherited.

l

Script log traces (landlord only) (required) — whether to use script log traces. The default is Inherited.

l

Tenant encryption key — an ASCII string that is unique across the system. The maximum size is 512

characters.

Note

We recommend that you change this key occasionally to ensure mailbox security.

l

Tenant messaging folder purge interval (hours.fraction) — not in use.

l

Audio file URL rebound other telco reject (Dialer for Elsbeth only) — the URL of the wave file played to

the customer when a rebound call is rejected by the system for a reason other than no queue or queue

timeout.

l

ClientUploadService process priority (required) — the process priority for the client upload service that

searches in the background for files to compress and upload. The service searches every 3 minutes and

performs 1 compression at a time. The default is BelowNormal.

Note

The client upload service should have fewer resources than the screen recording process.

l

ClientUploadService process affinity — the processors that are allowed to run the client upload service.

The default is blank (use all processors).

Note

We recommend that you use all processors so compressing and uploading finishes as fast as possible.

However, if you need to reserve processors for CCSP UI, screen recording and other processes, you can

set the client upload service processor affinity to use only half of the processors by configuring a list of

even CPU numbers, separated by a comma, for example, 0,2,4,6

l

Screen recording process priority (required) — the process priority for the screen recording service that

captures the agent's screen during CCSP UI calls. The default is Normal.

Note

The screen recording process should have more resources than the client upload service but we do not

recommend a process priority bigger than normal because this can affect OS tasks.

CCSP Admin Help - Tenant - 7.4

31

l

Screen recording process affinity — the processors that are allowed to run the screen recording service.

The default is blank (use all processors).

l

Vidyo developer key — an ASCII string provide by Enghouse Vidyo to link to the Vidyo service.

l

IVR root path — the location of the published Designer scripts, for example, \\ccsp1\ivr\t1\

l

Recording database location — the location of the recording database. The options are:

o

CCSP Recording Data Store (default) — call records will be saved to the CCSPRecording database.

o

QMS Database — call records will be sent to QMS only.

o

Both — call records will be saved to the CCSP Recording database and sent to QMS.

l

External IdP Type (landlord only) — the external Identity Provider to use. The options are:

o

None (default) — use the CCSP Auth Server to authenticate users.

o

EIS — use the Enghouse Identity Server to authenticate users.

o

ADFS — use Active Directory Federation Services to authenticate users.

l

External IdP Host URL (landlord only) — the URL where the user’s browser is redirected for authentication.

Only relevant when the External IdP Type is not None.

l

External IdP Client ID (landlord only) — the identifier by which the CCSP Auth Server is known to the

External Identity Provider. Only relevant when the External IdP Type is EIS.

l

External IdP Client Secret (landlord only) — the allocated password that the CCSP Auth Server must use

when requesting authentication from the External Identity Provider. Only relevant when the External IdP

Type is EIS.

l

Access Token Lifetime (min) — the validity period in minutes of the access token received by the client

application from the CCSP Auth Server. The default is 15.

l

Refresh Token lifetime (min) — the validity period in minutes of the refresh token received by the client

application from the CCSP Auth Server. The default is 600.

l

Login Session Lifetime (min) — the duration of the CCSP SSO login session in minutes. The default is 720.

CCSP Admin Help - Tenant - 7.4

32

Manage tenant SIP parameters

Landlord only can change these values.

On the Tenant Properties page, on the SIP parameters tab, view/enter the following information:

l

P-asserted-identity SIP — the SIP equivalent of the ANI in SIP URI format. Telephone carriers use this

information to identify authorized users, such as a particular tenant. This is not Caller ID and is not shown to

the called party. CCSP sends this information to the remote party in a SIP INVITE message. The telephone

carrier provides this value.

l

P-asserted-identity TEL — the phone number in TEL URI format. This information is similar to P-Asserted-

Identity SIP but in telephone format and is sent in a SIP INVITE message.

l

Identity of transfer from IVR — whether and how to transfer caller identity to the third party. Used only

for calls transferred from IVR script (by using the Bridge and Transfer Designer blocks). The options are:

o

Of the original call (default) — ANI, PAI, and privacy transferred to the third party are copied from the

original inbound leg of the call.

o

Anonymous — the third party does not receive the identity of the caller or the tenant.

o

Of the tenant — the third party of a transferred call receives the tenant's caller ID.

l

Identity of outgoing call — whether and how to transfer the identity of the original caller to the called

party. Used for calls transferred by agents or regular outgoing calls made by agents. The options are:

o

Of the original call (default) — if a call is transferred by agent, the identity of the original caller is

transferred to the called party. For a regular outgoing call, the identity of the individual agent, if it is

assigned, is transferred to the called party.

o

Anonymous — the identities of the original caller and the tenant are not disclosed.

o

Of the tenant — the called party receives the individual agent caller ID, if it is assigned.

l

When caller requests privacy — whether to show caller identities to the tenant. Caller identity is usually

stripped from a call by caller request even before the call arrives at CCSP. For example, if the system is

installed in an enterprise or in a service provider that is not a carrier, the setting has no effect on the system.

If installed in the core of a carrier network, CCSP may receive caller identities even if callers ask to hide it.

The system administrator may decide to show caller identities to the tenant. This parameter is also used by

H.323. The options are:

o

Do not hide (default) — pass the caller's identity and do not hide it. Ignore the caller's request.

o

Hide from agent — hide only from agents. The identity is available for scripts and reports.

o

Hide everywhere — hide everywhere. Calls where the caller asks to hide his identity appear as

Anonymous for agents, scripts, and reports.

l

Relay display name — whether to relay the display name to other call participants. The display name is a

textual description of the caller, such as the person’s name. If the caller has requested privacy and caller ID

appears as Anonymous, no display name is associated with the call, regardless of this setting.

CCSP Admin Help - Tenant - 7.4

33

Manage other tenant properties

Caution

We highly recommend that you enable Use folder segregation.

When folder segregation is disabled:

l

The following folders are not visible to agents and regular supervisors:

o

Rejected Emails, Deleted Emails, Replied, Not Replied, Drafts, Outbox, Sent Items

o

Deleted Callbacks

o

Voicemail, Archived Voicemails, Rejected Voicemails, Deleted Voicemails

l

Agents and regular supervisors can only see their own callbacks marked with "call-to-me" and their own

personal (not shared) deferred emails/voicemails.

On the Tenant Properties page, on the Others tab, view/enter the following information:

l

Media devices — the video communications devices available to the tenant. The devices support two or

more end points to a conversation.

l

Default application — the application to execute if no other application evaluates to TRUE. See Manage

applications.

l

Default time zone — the tenant's default time zone. See Manage time zones.

l

Voicemail DNIS — the DNIS for the tenant to use for personal voicemail.

l

Trunk group — the trunk group assigned to the tenant.

l

Use folder segregation — whether to use the folder segregation feature that enables selective access to

system folders for all agents and supervisors based on permissions. The default is disabled. See above

Caution.

CCSP Admin Help - Tenant - 7.4

34

Manage tenant MS Teams parameters

If you have the Teams integration, on the Tenant Properties page, on the MS Teams tab, view/enter the

following information:

l

TeamsTenantID — the Directory (tenant) ID found in Azure Portal’s App registration overview screen.

l

TeamsClientID — the Application (client) ID found in Azure Portal’s App registration overview screen.

l

CacheExpiration — the cache that hold the search results from the Teams directory is periodically checked

to clear its expired entries, at the frequency defined in this parameter.

l

ResponseLimit — the maximum number of presences to return to the agent. Must be greater than zero.

We recommend 50.

l

Polling — whether to enable polling to get presences from the Teams Graph API. The default is disabled.

l

Polling interval — the time in milliseconds between each request to get presences from the Teams Graph

API.

CCSP Admin Help - Tenant - 7.4

35

# 4: Configure tenant resources

You can configure the following tenant resources:

l

Accounts

l

Applications

l

Block codes

l

Campaigns

l

Dashboard views

l

DNIS pool

l

Entry points

l

Folders

l

Groups

l

Interaction profiles

l

Legacy licenses

l

Mailboxes

l

Permission profiles

l

Personnel

l

Phone aliases

l

Queues

l

Release codes

l

Skills

l

SSO permissions

l

Teams

l

Time zones

l

Wrap-up codes

Manage accounts

View the Accounts list

Add an account

Edit an account

Delete an account

Accounts track individual call information. Calls are assigned to accounts via scripts. Account information can

be used for billing and other purposes.

CCSP Admin Help - Tenant - 7.4

36

View the Accounts list

1.

On the navigation pane, click

Miscellaneous.

2.

If necessary, click the Accounts tab. The Accounts list appears. For navigation information, see List view

and How to search. For column descriptions, see Add an account.

Add an account

1.

On the Accounts page, click

Add. The New Account dialog appears.

2.

Enter the following information:

o

Account name (required) — the name of the account.

o

Address — the mailing address.

o

Contact person — the name of the primary contact person for this account.

o

Phone — the primary contact's or the account's phone number.

o

Email — the primary contact's or the account's email address.

3.

Click OK.

Edit an account

1.

On the Accounts page, click the item to edit. The Account: [Name] dialog appears.

2.

Edit the information. For details, see Add an account.

3.

Click OK.

Delete an account

1.

On the Accounts page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

37

Manage applications

View the Applications list

Add an application

Creating rules

Setting the default application

Edit an application

Delete an application

Applications refers here to a set of scripts created using Designer that define a call flow. You can define multiple

call flow applications. One of these applications is selected for each call.

Before running a script, CCSP applies a set of rules to determine which application to use. The rules are

checked in the order in which they appear in CCSP Admin, that is, in creation order. The first application whose

rule evaluates to True is selected. If no rule evaluates to True, then CCSP runs the default application.

The call preserves the last selected application ID that was used. However, this application ID is not preserved

when an email or voicemail call is deferred or transferred to system (TTS). If the entry point for TTS has a

configured application ID, then this application ID is preserved instead.

These rules use call parameters to determine which application to run. For voice calls, DNIS is used. For chat

calls, the calling user URL is used, and for messaging calls the email folder name is used.

View the Applications list

1.

On the navigation pane, click

Routing.

2.

Click the Applications tab. The Applications list appears. For navigation information, see List view and

How to search. For column descriptions, see Add an application.

Add an application

1.

On the Applications page, click

Add. The New Application dialog appears.

2.

Enter the following information:

o

Application name (required) — the name of the application.

o

Description — a description of the application and its use.

CCSP Admin Help - Tenant - 7.4

38

o

URL (required) — the URL where the Designer XML scripts are published.

o

Rule (required) — the rule to determine whether to select the application. See Creating rules below.

3.

Click OK.

Creating rules

Rules are used to determine which application (script) is run when a call arrives in the contact center. A rule

consists of a compilable VB script.

To select the application based on the call’s media type, use one of the following logical comparison

statements:

l

MEDIATYPE=CALLBACK

l

MEDIATYPE=VOICEMAIL

l

MEDIATYPE=EMAIL

l

MEDIATYPE=VOIP

l

MEDIATYPE=CHAT

Below are some examples of rules.

DNIS = 6319404200 AND Now > CDate(""8/4/2017 6:54:10 PM""), CDate is a date constant.

CDate(""6:54:10 AM"") < Time And Time < CDate(""6:54:10 PM"")

Date > CDate(""8/4/2017"")

DNIS = 631\* AND Date > CDate(""8/4/2017"")

When creating rules:

l

DNIS must be used if you want to specify DNIS.

l

DNIS is not supported when media type is CallBack or VoiceMail.

l

MAILFOLDER represents the last ACD folder name.

l

MAILFOLDEREMAIL represents the last ACD folder email address.

l

MAILFOLDER and MAILFOLDEREMAIL in the same rule are not supported.

l

Now means current date and time.

l

Time represents the current time.

l

Date represents the current date.

l

CDate must have double quotation marks ("") on both sides of the rule.

For email messages rules, instead of DNIS specify MAILFOLDER.

MAILFOLDER = foldername And Date > CDate(""8/4/2017"")

For chat call rules, the Application Name can be specified on the web caller while making a chat call. If an

Application Name is supplied in the web caller, and it matches one of the defined Application Names, that

application (script) is run even though Application Rule doesn't evaluate to TRUE.

Note

CCSP Admin Help - Tenant - 7.4

39

l

Wildcard matching is supported.

l

If you want exceptions to run alternate scripts and everything else to run the default application, then

the default application must be added last in CCSP Admin since rules are processed in the order of

creation. The default application is run if no other application rule evaluates to True.

Set the default application

1.

On the navigation pane, click

Home page. The Tenant Properties dialog appears.

2.

Click the Others tab.

3.

In Default application, select the application.

4.

Click Save changes.

Edit an application

1.

On the Applications page, click the item to edit. The Application: [Name] dialog appears.

2.

Edit the information. For details, see Add an application.

3.

Click OK.

Delete an application

1.

On the Applications page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

40

Manage block codes

View the Block Codes list

Add a block code

Edit a block code

Delete a block code

Block codes are used to block group or agent access to specific area codes that you select.

View the Block Codes list

1.

On the navigation pane, click

Miscellaneous.

2.

Click the Block Codes tab. The Block Codes list appears. For navigation information, see List view and How

to search. For column descriptions, see Add a block code.

Add a block code

1.

On the Block Codes page, click

Add. The New Block Code dialog appears.

2.

Enter the following information:

o

Value (required) — the code to block. This is generally an area code such as 900 or 888. To specify

multiple block codes, enter their values separated by commas, for example, 888,999,800,45.

o

Groups — select the groups to apply the block code to from the list.

o

Agents — click

Add items and select the agents to apply the block code to from the list. See Filter

for agents.

3.

Click OK.

Edit a block code

1.

On the Block Codes page, click the item to edit. The Block Code: [Value] dialog appears.

2.

Edit the information. For details, see Add a block code.

3.

Click OK.

CCSP Admin Help - Tenant - 7.4

41

Delete a block code

1.

On the Block Codes page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

Manage campaigns

Campaign creation is covered in detail in the CCSP Dialer Guide for Elsbeth/Sytel.

CCSP Admin Help - Tenant - 7.4

42

Manage Dashboard views

View the Dashboard Views list

Edit a Dashboard view

The initial name of a view is assigned by the user who created it in Dashboard. That user is the first owner and

is the first to have sole permission to edit or delete the view.

Note

In Dashboard, you can rename a view but you cannot change its owner.

In CCSP Admin, you can:

l

Rename a view — the new name will appear in the Dashboard window.

l

Change the owner of a view — only the new owner can edit or delete the view.

l

Allow other users to share a view.

View the Dashboard Views list

1.

On the navigation pane, click

Miscellaneous.

2.

Click the Dashboard Views tab. The Dashboard Views list appears. For navigation information, see List

view and How to search. For column descriptions, see Edit a Dashboard view.

Edit a Dashboard view

1.

On the Dashboard Views page, click the item to edit. The Dashboard View: [Name] dialog appears.

2.

Enter the following information:

o

Name (required) — the name of the view.

o

Description — a description of the view.

o

Owner — to re-assign the permission to edit the view, select a user with a supervisor profile.

o

Shared — whether to allow the view to be shared.

3.

Click OK.

CCSP Admin Help - Tenant - 7.4

43

Manage the DNIS pool

Note

CCSP UI was previously called TouchPoint.

View the DNIS Pool list

Add a DNIS pool

Assigning a DNIS pool

Delete a DNIS pool

The DNIS pool is added by the landlord administrator, who then allocates each DNIS to a tenant.

Note

l

You can assign the DNIS to only one user.

l

Be sure to allocate a DID to an IP Phone user who has a permanent extension and does not use CCSP

UI.

Caution

Dialer for Elsbeth only. Because EPC is the owner of the DNIS-to-campaign association, to preserve

configuration integrity, you must not create, modify, or delete any DNIS-to-campaign association via CCSP

Admin.

View the DNIS Pool list

1.

On the navigation pane, click

Routing.

2.

If necessary, click the DNIS Pool tab. The DNIS Pool list appears. For navigation information, see List view

and How to search. For column descriptions, see Add a DNIS pool and Assigning a DNIS pool. The Agent

name column shows the agent the DNIS pool is assigned to. The Campaigns column show the number of

campaigns the DNIS pool is assigned to.

Add a DNIS pool

Landlord only.

CCSP Admin Help - Tenant - 7.4

44

1.

On the DNIS Pools page, click

Add. The New DNIS Pool dialog appears.

2.

Enter the following information:

o

Value (required) — the telephone number, local extension, URL, or any other appropriate DNIS

identifier. The value can include wildcards. For details, see How to use wildcards. Separate DNISs with a

space, comma, or semicolon.

o

Description — a description of the DNIS pool.

3.

Click OK.

Assigning a DNIS pool

1.

On the DNIS Pool page, click the item to edit. The DNIS Pool: [Value] dialog appears.

2.

Enter the following information:

o

Agents — click

Add items and select the agent to assign the DNIS to. See Filter for agents.

o

Campaigns — select the campaign to assign the DNIS to.

3.

Click OK.

Delete a DNIS pool

Landlord only.

1.

On the DNIS Pools page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

45

Manage entry points

Note

CCSP UI was previously called TouchPoint.

View the Entry Points list

Add an entry point

Edit an entry point

Delete an entry point

Caution

You must also create the script entry point in your scripts. Assign it the same name you assign in CCSP

Admin. For more information, see CCSP Designer Help.

View the Entry Points list

1.

On the navigation pane, click

Routing.

2.

Click the Entry Points tab. The Entry Points list appears. For navigation information, see List view and

How to search. For column descriptions, see Add an entry point.

Add an entry point

1.

On the Entry Points page, click

Add. The New Entry Point dialog appears.

2.

Enter the following information:

o

Name (required) — the entry point name. This name appears in the user's System tab.

o

Description — a description of the entry point. This text appears as a tool tip in CCSP UI.

o

Queue — to direct a call to a particular queue, select the queue. Only one queue can be assigned.

o

Folder — to place an email call in a particular folder, select the folder. Only one folder can be assigned.

o

Application — to assign the application to which the script in this entry point belongs, select the

application. Only one application can be assigned.

o

Script name — the name of the script inside the application to execute when the agent selects this

entry point. For example, to transfer to a point in the IVR script, type IVR.xml

CCSP Admin Help - Tenant - 7.4

46

o

Block name — the name of the block inside the script where execution begins (optional). If omitted,

execution starts from the Start block of the script.

o

Attendable — whether you want the agent to choose whether to transfer or conference the call. If you

want the agent only to transfer the caller to the entry point, clear the check box.

Note

Attendable entry points are not visible for email or chat calls.

o

Parameter accepted — whether you want the parameter string of the Entry Point block to be passed to

the script. Only one string can be passed.

Note

l

If Parameter accepted is selected, the agent can type a value in the Parameter field of the CCSP

UI System tab.

l

To pass multiple parameters through the single string parameter, you can write a routine that

passes the string in the script.

o

Remove skills (voice only) — whether you want to remove skills from the original call when transferring

to TTS.

Note

If the registry RetainSkillsForTtsCall exists on the relevant VCS, this setting is ignored unless the

registry value is -1.

o

Groups — if you want to limit the availability of the entry point to particular groups, select one or more

groups. If no groups are selected, all agents can select this entry point.

3.

Click OK.

Edit an entry point

1.

On the Entry Points page, click the item to edit. The Entry Points: [Name] dialog appears.

2.

Edit the information. For details, see Add an entry point.

3.

Click OK.

Delete an entry point

1.

On the Entry Points page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

47

Manage folders

View the Folders list

Add a folder

Edit a folder

Delete a folder

View the Folders list

1.

On the navigation pane, click

Routing.

2.

Click the Folders tab. The Folders list appears. For navigation information, see List view and How to

search. For column descriptions, see Add a folder.

Add a folder

Note

The folder's Email address field must contain a valid email address. You must add any special folders that

do not require an email address directly in the database.

1.

On the Folders page, click

Add. The New Folder dialog appears.

2.

Enter the following information:

o

Public folder name (required) — the folder name. This name appears in Outlook under Short

Name/Public Folder Name. This name must not contain spaces, underscores, or special characters. It

cannot be named CallBack, Config, Dead-Letter, No-Reply, Outbox, Processed, or Reply because these

are the names of hidden system folders.

o

Email address (required) — the email address of the folder. This address must not contain spaces,

underscores, or special characters. The address can have a maximum of 64 characters before the at

sign (@) and a maximum of 254 characters after it.

o

Caller URL — the URL to send with the email. This URL appears in the agent interface.

o

Caller message — the message to send with the email. This message appears in the agent interface.

o

TQOS (sec) — the amount of time in seconds in which an email call should be answered. TQoS can be

used to specify levels of service for incoming calls.

CCSP Admin Help - Tenant - 7.4

48

o

Groups — select the groups that have access to this folder.

o

Queue — select the queue to assign to messages that arrive at this folder without a queue.

o

Skills — select the skill requirements to assign to messages that arrive at this folder without a skill.

Note

Skill settings can come from many sources, including the Folder and the Queue. Folder skills are

added to Queue skills and do not override them.

o

Mailbox — select the mailbox to assign to this folder.

o

Maximum email body size — for text mode only, the maximum email body size in characters to

expose to scripts. The default is 8192 characters.

Note

To disable body fetching for both HTML and text, set to 0.

o

Email body script format — select the format for the email body exposed to the call arrival and other

scripts. The default is HTML. For details, see EmailCall.EmailData object in the CCSP Designer Help.

Note

l

We recommend that you use text mode and control the email body size.

l

In HTML mode, the entire body is available/exposed in scripts, which may have an impact on

network bandwidth between MS and MCS, MS performance, and MCS memory growth.

l

In HTML mode, some types of embedded image links may be broken.

3.

Click OK.

Edit a folder

1.

On the Folders page, click the item to edit. The Folder: [Name] dialog appears.

2.

Edit the information. For details, see Add a folder.

3.

Click OK.

Delete a folder

Note

l

CCSP does not delete email folders that contain items. If you try to delete an email folder that contains

items or a timeout occurs, a message that the operation failed appears.

l

If the system administrator has not assigned your contact center to a Messaging Server, you cannot

delete an email folder.

CCSP Admin Help - Tenant - 7.4

49

1.

On the Folders page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion. If a warning appears that lists the folders that were not

deleted, click OK.

CCSP Admin Help - Tenant - 7.4

50

Manage groups

View the Groups list

Add a group

Clone a group

Edit a group

Delete a group

View the Groups list

1.

On the navigation pane, click

Organization.

2.

Click the Groups tab. The Groups list appears. For navigation information, see List view and How to search.

The Queues, Folders, and Agents columns show the number of items assigned to the group.

Add a group

1.

On the Groups page, click

Add. The New Group dialog appears.

2.

Enter the following information:

o

Name (required) — the name of the group.

o

Description — a description of the group.

o

Block international calls — whether to block outgoing international calls.

o

Block long distance calls — whether to block outgoing long distance calls.

o

Block local calls — whether to block outgoing local calls.

o

Agents — click

Add items and select the agents for this group. See Filter for agents.

o

Primary queues — select the primary queues for this group.

o

Secondary queues — select the secondary queue for this group.

o

Block codes — select the block codes for this group.

Note

Block codes only affect long distance calls. The Block long distance calls check box supersedes

any block codes you enter in Block codes.

CCSP Admin Help - Tenant - 7.4

51

o

Folders — select the folders for this group.

o

Release codes — select the release codes for this group. You can add the same release codes to

multiple groups.

o

No release codes — whether you do not want this group to use release codes.

Note

If you select No release codes, agents in this group can still become Available or Unavailable.

3.

Click OK.

Clone a group

1.

On the Groups page, select the item/s to clone.

2.

Click

and then click Clone Groups. The Clone Groups dialog appears.

3.

Optionally added a prefix and/or suffix for the cloned groups.

4.

Click OK.

Edit a group

1.

On the Groups page, click the item to edit. The Group: [Name] dialog appears.

2.

Edit the information. For details, see Add a group.

3.

Click OK.

Delete a group

1.

On the Groups page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

52

Manage interaction profiles

View the Interaction Profiles list

Add an interaction profile

Edit an interaction profile

Delete an interaction profile

Interaction profiles define the rules for how the CMS allocates calls to individual agents. They define the media

types a specific agent can handle, the number of concurrent calls of each type, and any blocking or interruption

rules.

Each interaction profile contains a collection of interaction profile rules, one for each media type, although not

every media type needs to be included. Each existing and new agent (personnel) is assigned to a single

interaction profile. Any new agents are assigned to the profile designated as the default profile for new agents.

The maximum total concurrent interactions per agent are limited to 9. If voice calls are configured to override

this maximum, the limit is 10.

Multi-interaction enables an agent to simultaneously handle multiple calls, up to a configured number between

one and nine. If voice calls are configured to override this maximum, the limit is 10.

In CCSP Admin, you configure the number of interactions each agent can handle using multi-interaction

profiles. You then enable multi-interaction for an agent by assigning the relevant interaction profile to the

agent in the Interaction profile property on the Properties tab of the [Agent name] page. See Manage

personnel properties. Alternatively, you can assign the agent as a member of a multi-interaction profile.

View the Interaction Profiles list

1.

On the navigation pane, click

Miscellaneous.

2.

Click the Interaction Profiles tab. The Interaction Profiles list appears. For navigation information, see

List view and How to search. For column descriptions, see Add an interaction profile. The Agents column

shows the number of agents assigned to the interaction profile.

Add an interaction profile

1.

On the Interaction Profiles page, click

Add. The New Interaction Profile dialog appears.

2.

Enter the following information:

CCSP Admin Help - Tenant - 7.4

53

o

Name (required) — the name of the interaction profile.

o

Description — a description of the interaction profile.

o

Is default profile for new agents —whether you want all new agents to be assigned to this profile.

Note

There can be only one default interaction profile. When you select the Is default profile for new

agents check box, CCSP automatically clears the corresponding check box on the previous default

profile.

o

Maximum overall interactions — a number between 0 and 9. A value of 0 means that agents assigned

to this interaction profile cannot handle any ACD media.

o

Agents — click

Add items and select the agents to assign this interaction profile to. See Filter for

agents.

3.

Under Internet, Phone, Email, Callback, and Voicemail enter the following information:

o

[Media type] rule maximum interactions — a number between 0 and 9. A value of 0 means that

agents assigned to this profile cannot handle this media type.

o

[Media type] rule call weighting — a number between 0 and 100, representing the relative weight of

this media type. A value of 100 means the highest weighting. The CMS uses this value to determine

which agent has the least load and should therefore take the next call. For more information, see How

agents are selected in the CCSP Tenant Administrator Guide.

Note

If the weight of a media type is set to zero, the agent is reported as available during the time of a

call from that media. Consequently, in reports an agent can appear as 99% available and at the

same time have handled many calls.

o

[Media type] rule media allowed — agents assigned this interaction profile can handle calls of this

media type. If you want agents assigned this interaction profile not to be allowed to handle this media

type, clear the check box. Selected by default.

o

[Media type] rule overrides maximum interactions — agents can handle chat and email even if that

agent is already handling the maximum overall interactions specified above. Not selected by default.

o

[Media type] rule blocks other media types — agents that are currently interacting with internet

media types, can be routed only the same type of media call. When all internet media type calls are

completed, the agent receives calls of another media type. Selected by default.

4.

Click OK.

Edit an interaction profile

1.

On the Interaction Profiles page, click the item to edit. The Interaction Profile: [Name] dialog appears.

2.

Edit the information. For details, see Add an interaction profile.

3.

Click OK.

CCSP Admin Help - Tenant - 7.4

54

Delete an interaction profile

Before you can delete an interaction profile, you must unassign the agents from it. If it is the default profile,

you must also first set another interaction profile as the default.

When you delete an interaction profile, CCSP also deletes its interaction profile rules.

1.

On the Interaction Profiles page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

55

Manage legacy licenses

View the legacy Licenses list

View contact center licenses

Allocate legacy licenses to users

Legacy licenses are used to determine how many agents can log on per tenant. The tenant administrator can

view the number of available agent licenses added to their contact center by the system administrator.

If the enforcement type is Block, the tenant's logged on agents cannot exceed the number of available licenses.

Note

Your agents may be prevented from logging on, even when they are within your available licenses limit, if

some tenants do not have the Block enforcement type and exceed the number of licenses allocated to

them.

For more information about license types, refer to Legacy licensing in the CCSP Tenant Administrator Guide.

View the legacy Licenses list

1.

On the navigation pane, click

Miscellaneous.

2.

Click the Licenses tab. The Licenses list appears. For navigation information, see List view and How to

search.

View contact center licenses

You can view the agent licenses added to your contact center.

1.

On the Licenses page, click the Agent - Concurrent license. The License: [Name] dialog appears.

2.

View the following information:

o

Name — the license name.

o

Available — the number of agent licenses allocated to your contact center.

o

Enforcement type — the options are:

n

None (default) — allow agent logons above this limit.

n

Block — do not allow agent logons above this limit.

CCSP Admin Help - Tenant - 7.4

56

3.

Click OK.

Allocate legacy licenses to agents

For each agent, select the appropriate license, using the following flow chart as your guide. See Manage

personnel parameters.

CCSP Admin Help - Tenant - 7.4

57

Manage mailboxes

View the Mailboxes list

Add a mailbox

Test mailbox connectivity

Edit a mailbox

Delete a mailbox

The CCSP messaging subsystem can interact with an external mailbox to receive and send emails. In this

context, external means that mailbox is stored on a mail server that is external to CCSP.

The external mailbox must be created or purchased separately and must support the IMAP, SMTP network

protocols. When the mailbox exists on a mail server, you can define it in the CCSP Admin so it can be accessed

by CCSP services. The external mailbox must have an email address. There are mail server vendors that allow

adding email address aliases to a given mailbox. In this context, an alias is just another email address given to a

mailbox. You can have one or more mailboxes defined.

View the Mailboxes list

1.

On the navigation pane, click

Routing.

2.

Click the Mailboxes tab. The Mailboxes list appears. For navigation information, see List view and How to

search. For column descriptions, see Add a mailbox. The Folders column shows the number of folders

assigned to the mailbox.

Add a mailbox

1.

On the Mailboxes page, click

Add. The New Mailbox dialog appears.

2.

Enter the following information:

o

Name (required) — the name for CCSP to use to identify the mailbox.

o

Description — a description of the mailbox.

o

Folders — select the folders that are associated with this mailbox.

Note

CCSP Admin Help - Tenant - 7.4

58

l

If more than one folder is associated with this mailbox entity, you must define an email address

alias on the mail server for this mailbox.

l

The MS performs mailbox-folder matching based on the recipient email address and the target

folder email address. If an item contains more than one recipient that matches, the item is

cloned to each matched folder.

o

Enabled — whether to allow the MS to access this mailbox.

Note

Before enabling the mailbox, confirm that the landlord system administrator has completed the

prerequisite mailbox configuration.

o

IMAP address (required) — the address of the mail server IMAP listener.

o

IMAP port (required) — the port on which the mail server is listening to the IMAP protocol. The default

is 993.

o

SMTP address — the address of the mail server SMTP listener.

o

SMTP port — the port on which the mail server is listening to the SMTP protocol. The default is 587.

o

Username — the user name to authenticate to the mail server. This name must be unique across all the

service provider's tenants.

o

Password — the password for user authentication. The value specified in this field is encrypted by the

value of the Tenant Encryption Key parameter specified in Tenant Parameters by the landlord System

Administrator.

o

Auto connect retries — the number of attempts to connect to the mail server before failing. The

default is 3.

o

Max idle time (sec) — the maximum time to hold idle connections before closing them (not currently

in use). The default is 360.

o

Max email size in bytes — the maximum size of an email that will be downloaded from the mailbox.

Emails exceeding this limit are handled according to the policy defined in the Post MSR invalid method

parameter. The default is 6000000 (bytes).

o

Max email LW size in bytes — the maximum size of a lightweight email. Emails up to this size are

downloaded in parallel with emails of size above this parameter. The default is 10000(bytes).

o

Max number of items to collect — the maximum number of emails to download from the mailbox in

each iteration. The default is 50.

o

Post MSR persisted method (required) — select what to do with the email item stored in the mailbox

inbox after it was downloaded and inserted into the Tenant Messaging Database. Possible values are:

n

Move to processed mailbox folder (default) — move the email item to a special folder on the

mailbox (the Processed folder).

n

Delete item — delete the email item from the mailbox inbox.

o

Post MSR invalid method (required) — select what to do with the email item stored in the mailbox

inbox after it was downloaded but was not inserted into the Tenant Messaging Database because it was

detected as invalid. Possible values are:

n

Move to invalid mailbox folder (default) — move the email item to a special folder on the mailbox

(the Invalid folder).

n

Delete item — delete the email item from the mailbox inbox.

CCSP Admin Help - Tenant - 7.4

59

o

Persisted emails folder name — the name of the folder on the mailbox that CCSP will create and use

to store email items that were downloaded and stored in the Tenant Messaging Database. The default

is PersistedCCSPItems.

o

Invalid emails folder name — the name of the folder on the mailbox that CCSP will create and use to

store email items that were downloaded but were detected as invalid and were not stored in Tenant

Messaging Database. The default is InvalidCCSPItems.

o

Deleted emails folder name — the name of the folder on the mailbox that CCSP will create and use

during deletion of email stored in the inbox. The default is DeletedCCSPItems.

o

IMAP secure connection type (required) — select the type of the secured connection to the mail

server when using the IMAP protocol. The options are:

n

Auto (default) — auto decide whether to use SSL or TLS.

n

SslOnConnect — the connection should immediately use SSL or TLS encryption.

n

StartTls — elevates the connection to use TLS encryption immediately after reading the greeting and

capabilities of the server.

o

SMTP secure connection type (required) — select the type of the secured connection to the mail

server when using the SMTP protocol. The options are:

n

Auto (default) — auto decide whether to use SSL or TLS.

n

SslOnConnect— the connection should immediately use SSL or TLS encryption.

n

StartTls — elevates the connection to use TLS encryption immediately after reading the greeting and

capabilities of the server.

3.

Click OK.

Test mailbox connectivity

MSR DB only.

1.

On the Mailboxes page, click the item to edit. The Mailbox: [Name] dialog appears.

2.

Click Test Connectivity. The Mailbox Connectivity dialog appears.

3.

In Max allowed time for test (seconds), enter the maximum duration of the test. This value must not be

zero or a negative number. The default is 60.

4.

Click Start.

5.

If you want to stop the test, click Stop. The test stops when it finishes the current step.

When the mailbox test completes, the results appear in the Mailbox Connectivity dialog.

The Overall Result field shows the result of the entire test, while the Result column shows the result for a

specific test step. The values for a result can be one of the following:

o

Success — the connectivity test successfully completed. CCSP can work with this mailbox.

o

Canceled — the connectivity test was cancelled or stopped by the user.

o

Failed — the connectivity test failed. See the error messages in the test steps.

o

Timed out — the connectivity test did not complete within the allowed time.

CCSP Admin Help - Tenant - 7.4

60

6.

To close the Mailbox Connectivity dialog, click Cancel or X.

Edit a mailbox

1.

On the Mailboxes page, click the item to edit. The Mailbox: [Name] dialog appears.

2.

Edit the information. For details, see Add a mailbox.

3.

Click OK.

Delete a mailbox

1.

On the Mailboxes page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

61

Manage permission profiles

Note

CCSP UI was previously called TouchPoint.

Default permission profiles

View the Permission Profiles list

Add a permission profile

Edit a permission profile

Delete a permission profile

Permissions are assigned to users through permission profiles that are designed so that a number of users can

be assigned a set of permissions simultaneously and uniformly. For detailed information about permission

profiles, see the CCSP Tenant Administrator Guide.

Note

l

Only users with the default Administrator permission profile can manage permission profiles.

l

By default, the Supervisor profile has supervisory privileges over the whole contact center. To limit a

supervisor's privileges to particular groups, create a group supervisor. See Add a permission profile.

Permission profiles control access to the following objects, so it is best to configure these objects before

creating the permission profiles:

l

Groups

l

Campaigns

l

Queues

l

Teams

Default permission profiles

The predefined permission profiles are:

l

Administrator and Agent

o

Can set up and configure anything in the tenant contact center, including new permission profiles.

o

Can use CCSP UI.

o

Cannot view CCSP reports or play back recordings.

l

Administrator and Supervisor

o

Can set up and configure anything in the tenant contact center, including new permission profiles.

o

Can view CCSP Historical Reports and Dashboard reports for the whole tenant.

l

Administrator

o

Can set up and configure anything in the tenant contact center, including new permission profiles.

CCSP Admin Help - Tenant - 7.4

62

o

Cannot view CCSP reports or play back recordings.

o

Cannot use CCSP UI.

l

Agent

o

Can use CCSP UI.

o

Cannot view CCSP reports or play back recordings.

o

Cannot set up and configure anything in the tenant contact center.

l

Supervisor (tenant-wide supervisor)

o

Can view CCSP Historical Reports and Dashboard reports for the whole tenant.

o

Can play back recordings for the whole tenant.

o

Cannot set up and configure anything in the tenant contact center.

l

Team Supervisor

o

Can supervise the teams to which they are assigned. Can also view reports and play back recordings

only for the teams to which they are assigned.

o

Can use CCSP UI.

o

Cannot set up and configure anything in the tenant contact center.

Note

Users not assigned a permission profile have the permissions of agents.

View the Permission Profiles list

1.

On the navigation pane, click

Organization.

2.

Click the Permission Profiles tab. The Permission Profiles list appears. For navigation information, see

Card view and How to search.

CCSP Admin Help - Tenant - 7.4

63

Add a permission profile

1.

On the Permission Profiles page, click

Add. The New permission profile page appears.

2.

Click the Properties tab and enter the following information:

o

Name (required) — the name of the profile.

o

Description — a description of the profile.

o

Allow create/delete — whether to allow an administrator with restricted permissions to create new

objects or delete existing objects. If not selected, the administrator can only modify the properties of

existing objects.

o

Members — click

Add items and select the agents to assign the permission profile to. See Filter for

agents.

3.

Click the Permissions tab and assign the permission to the objects, as required.

4.

Click the Groups tab and assign the Report, Administer, Playback, and/or Supervise permissions to the

groups, as required.

5.

Click the Campaigns tab and assign the Report and Administer permissions to the campaigns, as required.

6.

Click the Queues tab and assign the Report and Administer permissions to the queues, as required.

7.

Click the Teams tab and assign the Report, Administer, Playback, and/or Supervise permissions to the

teams, as required.

8.

Click Save changes.

Edit a permission profile

1.

On the Permission Profiles page, click the card of the item to edit. The [Permission profile name] page

appears.

2.

Edit the information. For details, see Add a permission profile.

3.

Click OK.

Delete a permission profile

1.

On the Permission Profiles page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

64

Manage personnel

Note

CCSP UI was previously called TouchPoint.

View the Agents list

Add an agent

Add a phone agent

Clone an agent

Edit an agent

Delete an agent

View the Agents list

1.

On the navigation pane, click

Organization.

2.

If necessary, click the Personnel tab. The Agents list appears. For navigation information, see Card view,

How to search, and How to filter.

The

icon indicates that the agent is campaignable.

CCSP Admin Help - Tenant - 7.4

65

Add an agent

Note

l

If you are adding a phone agent, see Add a phone agent before proceeding.

l

If you are adding a supervisor or administrator, add in the same way as an agent but select the

appropriate Profile on the Properties tab. See Default permission profiles.

1.

On the Agents page, click

Add. The New Agent page appears.

2.

Enter the following information, arranged on the following tabs:

o

Properties

o

Phone settings

o

Parameters

3.

Click Save changes.

CCSP Admin Help - Tenant - 7.4

66

Add a phone agent

In some instances, agents may be constrained to the use of only a circuit switched phone or IP phone. This may

be due to how your contact center is configured, or if the agent is traveling without a computer and still needs

to service calls, or perhaps in the event of a disaster.

Since the agent is operating without a computer, many of the functions that are available through CCSP UI are

not available for the Phone Agent.

Phone agents are created in exactly the same manner as other agents. You must:

l

Assign the agent a numeric Login ID and numeric Password.

l

Enter the agent's phone number into the Phone ANI field of the personnel record.

l

Configure the area code and country.

Note

l

Phone agents can use the same numeric Login ID and Password to log on normally from a

computer.

l

Phone agents can only be created on the internal VCS stack.

Clone an agent

1.

On the Agents page, select the item/s to clone.

2.

Click

and then click Clone Agents. The Clone Agents dialog appears.

3.

Optionally added a prefix and/or suffix for the cloned agents.

4.

Click Set Password. The Change password dialog appears.

5.

Type a new password that meets your AD policy, type the password again to confirm it, and then click OK.

6.

Click OK.

Edit an agent

1.

On the Agents page, click the card of the item to edit. The [Agent name] page appears.

2.

Edit the information. For details, see Add an agent.

3.

Click OK.

Delete an agent

1.

On the Agents page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

67

Manage personnel properties

Note

CCSP UI was previously called TouchPoint.

On the New Agent/[Agent name] page, on the Properties tab, enter the following information:

l

Login ID (required) — a unique name for the agent. This name must not contain spaces, underscores, or

special characters.

l

Change password — click and in the Change password dialog, type a password that meets your AD policy.

l

First name (required) — the agent's first name.

l

Last name (required) — the agent's last name.

l

Description — a description of the agent.

l

Profile — select from the a list of available permission profiles. Default profiles are enclosed in parentheses

and cannot be changed. Leave blank to assign the default agent profile. See Manage permission profiles.

Note

l

The list of available permission profiles to assign to the agent is limited to the permission profiles

that contain only the groups that are included in your own profile.

l

If you change an agent's permission profile, to avoid potential inconsistent behavior or errors inside

CCSP UI for operations regulated by different permission profiles, that agent must log off from CCSP

UI and log on again.

l

Skills — select this agent's skills from the list of available skills that you added. See Manage skills.

l

Set skills weight — click and then in the Set Skills Weight dialog, select the weight for the skills that have

weighting enabled: 0 is high and 100 is low.

Note

Skill weighting must first be enabled by the landlord in tenant properties and then enabled for the

individual skill. See Manage skills.

l

Groups — select from the list of available groups that you created. See Manage groups.

l

Block codes — select from the list of available codes that you created for blocking international, long

distance, and local calls. See Manage block codes.

l

Interaction profiles — select from the list of available interaction profiles, which determine the maximum

number and types of calls that this agent can handle. See Manage interaction profiles.

Note

If you change an agent's interaction profile, the agent's load level displayed in Dashboard changes only

when the agent finishes the current call or a new call is added to the agent.

l

Teams — select from the list of available teams that you created. See Manage teams.

l

Time zones — select from the list of available time zones that you created. See Manage time zones.

CCSP Admin Help - Tenant - 7.4

68

l

Is campaignable (Dialer for Elsbeth only) — whether to allow the agent to participate in campaigns. For

more information, refer to the CCSP Dialer Guide for Elsbeth.

l

Can request consultation — whether the agent can request a chat consultation during a call.

CCSP Admin Help - Tenant - 7.4

69

Manage personnel phone settings

Note

CCSP UI was previously called TouchPoint.

On the New Agent/[Agent name] page, on the Phone settings tab, view/enter the following information:

l

Phone ANI — the identifier used for call routing or billing. This is an optional field that can be required by

your telephone carrier. ANI is different from Caller ID, but might be used by a telephone carrier to display

caller identity to the called party. The value of this field is usually provided by the telephone company and is

numeric text.

l

Caller ID — the identity the agent presents to a called party. This value can be text or a telephone number.

l

DNIS (read only) — the DNIS assigned to the user. The DNIS serves as the personal voicemail User ID.

l

Country code — the agent's country code. This code can be a maximum of three numeric digits and can

have a leading plus sign (+).

l

Area code — the agent's area code. This code can be a maximum of 12 numeric digits.

l

Voice device type — the agent's voice device. The options are:

o

NotSet (default) — set in CCSP UI by the agent

o

SIP — CCSP UI provides a SIP voice terminal

o

H323 — CCSP UI provides an H.323 voice terminal

o

VoiceDevice — a phone

o

CommunicatorClient — not in use

l

Voice device address — the agent's phone IP or extension in E.164 or SIP URI format. For Teams, use the

agent's phone number associated with it.

l

Block local calls — whether to block outgoing local calls.

l

Block long distance calls — whether to block outgoing long distance calls.

l

Block international calls — whether to block outgoing international calls.

l

Auto release on call pickup — whether to have the agent automatically enter the released state when a

call is picked up. The agent needs to manually make themselves available when after-call work is complete.

l

Auto release on ring timeout — the number of pickup failures after which the agent automatically enters

the released state. The default is 0 (disabled).

l

Automatic answer — whether to have CCSP UI automatically answer calls when they are delivered.

l

IP phone password — the numeric password for the IP phone directly controlled by CCSP.

l

Voicemail — whether to enable personal voicemail. Personal voicemail is stored physically in the recording

database and is unrelated to ACD-routed voicemail.

l

Mailbox PIN — the initial numeric password that the agent enters to access their personal voicemail box.

The agent can change the PIN after logging on to the voicemail system. To log on to voicemail, agents use

their direct extension as the user ID and the Mailbox PIN as a password.

l

KCO mode — whether to enable the agent to remain connected to CCSP through an alternative device for

the duration of their shift. All actions are done on the screen, so agents do not need to answer or hang up

calls on the alternative device.

CCSP Admin Help - Tenant - 7.4

70

l

Shared mailbox — the agent ID of the shared mailbox. For detailed information, see Enabling personal voice

mailbox sharing in the CCSP Tenant Administrator Guide.

l

Use SRTP — whether to use SRTP on the Agent leg.

Note

If Use SRTP is select for the tenant, it is used for the agent irrespective of the above setting. See

Manage tenant parameters.

CCSP Admin Help - Tenant - 7.4

71

Manage personnel parameters

Note

CCSP UI was previously called TouchPoint.

On the New Agent/[Agent name] page, on the Parameters tab, enter the following information:

l

Login license type — the license to assign to the agent. See Allocate legacy licenses to agents.

l

CosmoCorder license — whether to assign this license.

l

CosmoConnector license — whether to assign this license.

l

ScreenRecording license — whether to assign this license.

l

External email address — the email address for delivery of personal voicemail. Agents with personal

voicemail enabled can have their personal voicemail delivered as email sound file attachments. The emails

containing the attachments are sent to this external email address, which is not related to ACD-routed

emails.

l

Agent can change password — whether to allow the agent to change their CCSP UI password. If not

selected, password settings do not appear in CCSP UI.

l

Available upon login — whether to have the agent automatically become available when logging on to

CCSP UI.

l

CRM URL — the URL of the tenant's CRM.

l

CRM user ID — the agent's user ID for the tenant's CRM.

l

CRM password — the agent's password for access to the tenant's CRM.

l

Block agent to agent calls — whether to prevent the agent from calling another agent.

l

Block xfer/conf to agent — whether to prevent the agent from transferring or conferencing calls to

another agent.

l

Block call/xfer to system — whether to prevent the agent from transferring calls to entry points.

l

Preferences edit by agent — whether the agent can edit their preferences in CCSP UI. The options are:

o

Deny — preferences are not displayed in CCSP UI.

o

Allow (default) — the agent can permanently configure CCSP UI preferences.

o

Allow per session — obsolete.

o

Read-only — preferences are displayed but selection is disabled in CCSP UI.

l

Encrypted recording restrictions — the restrictions that apply to supervisors when playing back encrypted

recordings. The default is None.

Note

Encrypted recording restrictions for the tenant are set in the Tenant Properties dialog, on the

Parameters tab (see Manage tenant parameters). To override this setting for the agent, select a value

other than None. For more information, see Playing back encrypted recordings in the CCSP Tenant

Administrator Guide.

CCSP Admin Help - Tenant - 7.4

72

Manage phone aliases

Use wildcards in phone aliases

View the Phone Aliases list

Add a phone alias

Edit a phone alias

Delete a phone alias

Phone aliases can be used to create easily memorized nicknames for particular phone numbers, or to map to

particular numbers or individuals within your organization. They can be thought of as extensions, but they can

be any set of alphanumeric characters, including words or short phrases. When assigning words or phrases to

an alias, any users who would be likely to use the alias should be made aware of its existence.

Use wildcards in phone aliases

A common use for this feature is to create shortcut dialing numbers, similar to speed dialing numbers, for a

company with offices in different locations. If, for example, all of the extensions in your local office begin with

the digit 4 and the extensions in another office begin with 7, you can assign the alias 7000 to the number

+1516557000. Then a person in your office only has to dial 7000 to connect to the person with that number in

the other office.

You can also use a question mark as a wildcard to replace any digits in both the alias and the number to be

dialed. To extend the above example further, you could assign the alias 7??? to the phone number +1516557???

to create shortcuts to all of the extensions in that other office. In a similar manner, you could assign the alias

4??? to your +1631940??? main office phone number.

View the Phone Aliases list

1.

On the navigation pane, click

Miscellaneous.

2.

Click the Phone Aliases tab. The Phone Aliases list appears. For navigation information, see List view and

How to search. For column descriptions, see Add a phone alias.

CCSP Admin Help - Tenant - 7.4

73

Add a phone alias

1.

On the Phone Aliases page, click

Add. The New phone alias dialog appears.

2.

Enter the following information:

o

Phone alias (required) — the name, number, or other significant identifier.

o

International number (required) — the + (international standard format) phone number that is

dialed when the alias is used. This number must follow the international standard format, but does not

necessarily have to be in another country.

3.

Click OK.

Edit a phone alias

1.

On the Phone Aliases page, click the item to edit. The Phone alias: [Name] dialog appears.

2.

Edit the information. For details, see Add a phone alias.

3.

Click OK.

Delete a phone alias

1.

On the Phone Aliases page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

74

Manage queues

View the Queues list

Add a queue

Clone a queue

Edit a queue

Delete a queue

Caution

The information entered in the Queue: [Name] dialog dictates call flow, sets priorities, and triggers events

effecting calls arriving at your contact center. It determines the quality of service your business strives to

provide its customers. Therefore, this information must be agreed to or provided by your contact center's

affected business units.

View the Queues list

1.

On the navigation pane, click

Routing.

2.

Click the Queues tab. The Queues list appears. For navigation information, see List view and How to

search. For column descriptions, see Add a queue.

Add a queue

1.

On the Queues page, click

Add. The New Queue dialog appears.

2.

Enter the following information:

o

Queue name (required) — the name of the queue. Queue names can be a maximum of 32 characters.

o

Description — a description of the queue.

o

Primary in groups — select the primary group or groups for this queue.

o

Secondary in groups — select the secondary group or groups for this queue.

o

Skills — select the skill requirements to assign to calls that arrive at this queue without a skill assigned

in the IVR script.

o

Wrap-up codes — if required, select the codes.

o

No wrap-up — whether the queue does not require wrap-up codes.

CCSP Admin Help - Tenant - 7.4

75

o

Timeout in queue — the maximum amount of time a caller should wait in a queue before the call times

out. When a caller's wait in queue exceeds this time, the In Queue Timeout script specified with

Designer runs. When a queue is created, its default in queue timeout is 5 minutes (300 seconds).

Note

Create a separate queue for email (MCS) messages.

For a contact center that receives many email messages, the in queue timeout should be as large as

possible. When the in queue timeout expires, the MCS takes all calls from the Call Matcher, runs the in

queue timeout script and then resubmits them back to the Call Matcher. This would consume

excessive bandwidth if there are a large number of email messages.

For example, if the contact center receives 10,000 message calls a day and it takes 2 days for all the

available agents to answer all 10,000 message calls, a reasonable in queue timeout for the message

queue would be 2 days (172800 seconds).

o

Timeout pickup — the pickup timeout for the queue. When a call rings to an agent and the ringing

time exceeds this time and the agent has not answered the call, the Pick-up Timeout script specified with

Designer runs and routes the call to the next available agent.

Caution

By default, timeout occurs at 75% of the value that you specify. For more information, see

PickUpTimeoutMultiplyer registration key in the CCSP Registry Key Reference.

Note

If you set the pickup timeout to 0, all calls routed to an agent on that queue (incoming or transfer

to system calls) ring indefinitely on the first chosen agent from that queue.

o

Timeout wrap-up — the amount of time allowed for wrap-up activity before the default wrap-up code

is selected.

o

Service level (%) — obsolete.

o

TQOS (sec) — the TQoS for the queue.

o

Agent selection rule — select the appropriate level.

3.

Click OK.

Clone a queue

1.

On the Queues page, select the item/s to clone.

2.

Click

and then click Clone Queues. The Clone Queues dialog appears.

3.

Optionally added a prefix and/or suffix for the cloned queues.

4.

Click OK.

CCSP Admin Help - Tenant - 7.4

76

Edit a queue

1.

On the Queues page, click the item to edit. The Queue: [Name] dialog appears.

2.

Edit the information. For details, see Add a queue.

3.

Click OK.

Delete a queue

1.

On the Queues page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

77

Manage release codes

View the Release Codes list

Add a release code

Edit a release code

Delete a release code

For information about release codes that are assigned to groups, see Add a group.

View the Release Codes list

1.

On the navigation pane, click

Routing.

2.

Click the Release Codes tab. The Release Codes list appears. For navigation information, see List view and

How to search. For column descriptions, see Add a release code. The Groups column shows the number of

groups assigned to the release code.

Add a release code

1.

On the Release Codes page, click

Add. The New Release Code dialog appears.

2.

Enter the following information:

o

Name (required) — the name of the release code.

o

Description — a description of the release code.

3.

Click OK.

Edit a release code

1.

On the Release Codes page, click the item to edit. The Release Code: [Name] dialog appears.

2.

Edit the information. For details, see Add a release code.

3.

Click OK.

CCSP Admin Help - Tenant - 7.4

78

Delete a release code

1.

On the Release Codes page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

79

Manage skills

View the Skills list

Add a skill

Clone a skill

Bulk assign agents to skills

Edit a skill

Delete a skill

To assign weights to skills by agent, see Manage personnel properties.

View the Skills list

1.

On the navigation pane, click

Organization.

2.

Click the Skills tab. The Skills list appears. For navigation information, see List view and How to search. For

column descriptions, see Add a skill. The Agents, Queues, and Folders columns show the number of items

assigned to the skill.

Add a skill

1.

On the Skills page, click

Add. The New Skill dialog appears.

2.

Enter the following information:

o

Name (required) — the name of the skill.

o

Description — a description of the skill.

o

Allow weight — whether to enable skill weighting for this skill.

o

Agents — click

Add items and select the agents that have this skill. See Filter for agents.

o

Queues — select the groups that require this skill.

o

Folders — select the folders that require this skill.

3.

Click OK.

CCSP Admin Help - Tenant - 7.4

80

Clone a skill

1.

On the Skills page, select the item/s to clone.

2.

Click

and then click Clone Skills. The Clone Skills dialog appears.

3.

Optionally added a prefix and/or suffix for the cloned skills.

4.

Click OK.

Bulk assign agents to skills

1.

On the Skills page, select the skills.

2.

Click

and then click Assign Agents to Skills. The Assign Agents to Skills dialog appears.

3.

Under Agents, click

Add items, select the agents, and click OK. See Filter for agents.

4.

Click OK.

Edit a skill

1.

On the Skills page, click the item to edit. The Skill: [Name] dialog appears.

2.

Edit the information. For details, see Add a skill.

3.

Click OK.

Delete a skill

1.

On the Skills page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

81

Manage SSO permissions

View the SSO Permissions list

Add an SSO permission

Edit an SSO permission

Delete an SSO permission

Landlord and tenant administrators can create an SSO permission entry to control the users that can access

one or more custom SSO applications. The SSO permission can be used to:

l

Allow specific permission profiles to be able to access custom SSO applications.

l

Override the tenant profile permissions for a custom SSO application that have been set at the landlord

level.

View the SSO Permissions list

1.

On the navigation pane, click

Organization.

2.

Click the SSO Permissions tab. The SSO Permissions list appears. For navigation information, see List view

and How to search.

Add an SSO permission

1.

On the SSO Permissions page, click

Add. The New SSO Application Permission dialog appears.

2.

Enter the following information:

o

Name (required) — the name of the SSO permission.

o

Description — a description of the SSO permission.

o

Custom SSO applications — select the custom SSO applications to which this SSO permission applies.

o

All profiles allowed — whether all tenant defined users (administrators, supervisors, agents,

permission profile users) are allowed to use the assigned custom SSO applications.

Note

This overrides any tenant permissions for the application that have been set at the landlord level.

CCSP Admin Help - Tenant - 7.4

82

o

Allowed profiles when all profiles allowed not set — select the pre-defined tenant and custom

permission profiles that are allowed to use this application. This list is only used for restricting access

when All profiles allowed is not selected.

Note

This overrides any tenant permissions for the application that have been set at the landlord level.

3.

Click OK.

Edit an SSO permission

1.

On the SSO Permissions page, click the item to edit. The SSO Application Permission: [Name] dialog

appears.

2.

Edit the information. For details, see Add an SSO permission.

3.

Click OK.

Delete an SSO permission

1.

On the SSO Permissions page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

83

Manage teams

View the Teams list

Add a team

Bulk assign agents to teams

Edit a team

Delete a team

View the Teams list

1.

On the navigation pane, click

Organization.

2.

Click the Teams tab. The Teams list appears. For navigation information, see List view and How to search.

The Agents column shows the number of agents assigned to the team.

Add a team

1.

On the Teams page, click

Add. The New Team dialog appears.

2.

Enter the following information:

o

Name (required) — the name of the team.

o

Description — a description of the team.

o

Agents — click

Add items and select the agents in the team. See Filter for agents.

3.

Click OK.

Bulk assign agents to teams

1.

On the Teams page, select the teams.

2.

Click

and then click Assign Agents to Teams. The Assign Agents to Teams dialog appears.

3.

Under Agents, click

Add items, select the agents, and click OK. See Filter for agents.

4.

Click OK.

CCSP Admin Help - Tenant - 7.4

84

Edit a team

1.

On the Teams page, click the item to edit. The Team: [Name] dialog appears.

2.

Edit the information. For details, see Add a team.

3.

Click OK.

Delete a team

1.

On the Teams page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

85

Manage time zones

View the Time Zones list

Add a time zone

Edit a time zone

Delete a time zone

For information about setting the default time zone, see Manage other tenant properties.

View the Time Zones list

1.

On the navigation pane, click

Miscellaneous.

2.

Click the Time Zones tab. The Time Zones list appears. For navigation information, see List view and How

to search. For column descriptions, see Add a time zone.

Add a time zone

1.

On the Time Zones page, click

Add. The New Time Zone dialog appears.

2.

Enter the following information:

o

Name (required) — the name of the time zone.

o

Description — a description of the time zone.

o

Offset (required) — the offset from Greenwich Mean Time (GMT) in minutes.

3.

Click OK.

Edit a time zone

1.

On the Time Zones page, click the item to edit. The Time Zones: [Name] dialog appears.

2.

Edit the information. For details, see Add a time zone.

3.

Click OK.

CCSP Admin Help - Tenant - 7.4

86

Delete a time zone

1.

On the Time Zones page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

87

Manage wrap-up codes

View the Wrap-Up Codes list

Add a wrap-up code

Edit a wrap-up code

Delete a wrap-up code

View the Wrap-Up Codes list

1.

On the navigation pane, click

Miscellaneous.

2.

Click the Wrap-Up Codes tab. The Wrap-Up Codes list appears. For navigation information, see List view

and How to search. For column descriptions, see Add a wrap-up code.

Add a wrap-up code

1.

On the Wrap-Up Codes page, click

Add. The New Wrap-Up Code dialog appears.

2.

Enter the following information:

o

Name (required) —the name of the wrap-up code.

o

Description — a description of the wrap-up code.

o

PD disposition code — select a predictive dialer code to link to the wrap-up code. For more

information, see the CCSP Dialer Guide for Elsbeth/Sytel.

3.

Click OK.

Edit a wrap-up code

1.

On the Wrap-Up Codes page, click the item to edit. The Wrap-Up Code: [Name] dialog appears.

2.

Edit the information. For details, see Add a wrap-up code.

3.

Click OK.

CCSP Admin Help - Tenant - 7.4

88

Delete a wrap-up code

1.

On the Wrap-Up Codes page, select the item/s to delete.

2.

Click

Delete and then confirm the deletion.

CCSP Admin Help - Tenant - 7.4

89

# Contact Center: Service Provider System Administrator Guide Version 7.3 and 7.4

CCSP-SAG-73/74-R6-09/2024   
Contents   
1: Intro .............................................................................................................................................7   
What’s new ................................................................................................................................................. 7   
Definitions .................................................................................................................................................. 7   
Legal disclaimer ......................................................................................................................................... 9   
Support ....................................................................................................................................................... 9   
2: CCSP overview ......................................................................................................................... 10   
Components ............................................................................................................................................. 10   
Automatic Call Distribution (ACD) Server .................................................................................... 10   
Administrator Interaction Server (Admin IS) .............................................................................. 10   
Agent Interaction Server (AIS) ...................................................................................................... 11   
CCSP Admin ..................................................................................................................................... 11   
Clients ............................................................................................................................................... 11   
CCSP Caller ....................................................................................................................................... 12   
Device Proxy Server (DPS) ............................................................................................................. 12   
Internet Chat Server (ICS).............................................................................................................. 12   
Message Connection Server (MCS) .............................................................................................. 12   
Messaging Server (MS) .................................................................................................................. 12   
Predictive Dialing Server (PDS) ..................................................................................................... 12   
Replay ............................................................................................................................................... 13   
Voicemail Server (VMS) .................................................................................................................. 13   
VoIP Connection Server (VCS) ....................................................................................................... 13   
Interaction recording .............................................................................................................................. 13   
Interaction recording components .............................................................................................. 14   
Contact center performance report tools............................................................................................ 15   
Tenant statistical views .................................................................................................................. 15   
How CCSP works ...................................................................................................................................... 16   
Logical connections ................................................................................................................................. 17   
Always On availability groups ................................................................................................................ 18   
Historical Reports............................................................................................................................ 18   
Remote site survivability ........................................................................................................................ 19   
IP telephony ............................................................................................................................................. 19   
Personal voicemail .......................................................................................................................... 19   
Interactive Voice and Video Response (IVVR) ..................................................................................... 20   
Speech-enabled contact centers ........................................................................................................... 21   
Teams integration ................................................................................................................................... 22   
Messaging modes ................................................................................................................................... 23   
Multi-tenancy concepts .......................................................................................................................... 25   
System administrator roles in a multi-tenant contact center .................................................. 25

CCSP System Administrator Guide - 7.3 and 7.4   
   
3   
Tenant roles in a multi-tenant contact center ............................................................................ 25   
Administrative hierarchy for multi-tenancy ................................................................................ 25   
System Administrator supported data................................................................................ 26   
Tenant Administrator supported data ................................................................................ 26   
Privacy protection .................................................................................................................................... 27   
Regulating information exposed to called parties .................................................................... 27   
Tenant uses SIP privacy information in scripts .......................................................................... 27   
System administrator tasks ................................................................................................................... 27   
Setting up the CCSP platform ....................................................................................................... 27   
Add a new tenant ............................................................................................................................ 28   
System operation and administration ......................................................................................... 28   
3: Additional tenant configuration ............................................................................................ 29   
Change the secret key ............................................................................................................................ 29   
Use Exchange as an external mailbox.................................................................................................. 30   
Help tenants use Designer ..................................................................................................................... 31   
Verify that the XML Interpreter is available ................................................................................ 31   
Use WebDAV to save scripts via HTTP ......................................................................................... 32   
Configure SRTP for a tenant .................................................................................................................. 34   
Configure tenant time zones ................................................................................................................. 36   
Script-based answer supervision .......................................................................................................... 38   
PBX-style answering ....................................................................................................................... 38   
Enable Web chat in a multitenant environment ................................................................................. 39   
Caller language ID .......................................................................................................................... 39   
Enable interaction recording for a tenant ........................................................................................... 39   
Resolve Dashboard cross-domain authentication issues ................................................................. 40   
Connect Provisioning Portal (PRP) to CCSP Admin ............................................................................ 40   
Remove user profile folders .................................................................................................................. 41   
Reduce agent logon time ....................................................................................................................... 41   
Change DNS records from the command line .................................................................................... 42   
Install G.729 codecs ................................................................................................................................. 43   
4: Configure CCSP Admin ........................................................................................................... 44   
Customize the CCSP Admin look and feel ............................................................................................ 44   
Customize the CCSP Admin navigation ................................................................................................ 45   
Customize the CCSP Admin grid definition ......................................................................................... 45   
Customize the CCSP Admin screen definition ..................................................................................... 46   
Preserve customer overrides................................................................................................................. 48   
5: Configure CCSP UI ................................................................................................................... 49   
Configure the CCSP UI Updater ............................................................................................................ 49   
Disable editing of the default template and system gadgets .......................................................... 50

CCSP System Administrator Guide - 7.3 and 7.4   
   
4   
Configure MS folder change notifications ........................................................................................... 51   
Configure the chat box alignment ........................................................................................................ 52   
Enable multiple interaction mode for callbacks ................................................................................. 52   
Set the inactivity timer ............................................................................................................................ 53   
Change the email draft auto-save rate ................................................................................................ 53   
Enable start in toolbar mode ................................................................................................................. 53   
Configure the wallboard ........................................................................................................................ 54   
Connect to external Web or desktop applications ............................................................................. 54   
6: Configure Dashboard ............................................................................................................. 55   
Configure the Dashboard service ......................................................................................................... 55   
Configure the Dashboard application .................................................................................................. 57   
Configure the Dashboard application language ................................................................................ 58   
7: Configure Replay ..................................................................................................................... 59   
Create the recordings download virtual directory ............................................................................. 59   
Configure the recordings download folder ......................................................................................... 60   
Control access to recordings virtual directory .................................................................................... 60   
Enable audio visualization ...................................................................................................................... 61   
Streamed file names ............................................................................................................................... 62   
Archiver Playback .................................................................................................................................... 62   
Configuration modes ..................................................................................................................... 62   
Configure Replay for Online mode .............................................................................................. 63   
Configure Replay for Offline mode .............................................................................................. 63   
Auth Server configuration ..................................................................................................... 63   
Replay Server configuration ................................................................................................. 64   
Configure Replay for On-premise mode ..................................................................................... 65   
Auth Server configuration ..................................................................................................... 65   
Replay Server configuration ................................................................................................. 65   
8: Configure IP phones ............................................................................................................... 66   
Create an FTP folder for each tenant ................................................................................................... 66   
Create and assigning tenant extensions ............................................................................................. 66   
Individual IP phone configuration files ................................................................................................ 66   
9: Configure languages .............................................................................................................. 67   
Configure the back end language ........................................................................................................ 67   
Chat Proxy Service .......................................................................................................................... 67   
ChatExtension.dll ............................................................................................................................ 67   
Dashboard (legacy) ........................................................................................................................ 68   
Dashboard Service .......................................................................................................................... 68   
SQL Server ........................................................................................................................................ 68

CCSP System Administrator Guide - 7.3 and 7.4   
   
5   
Enable different Web chat languages for tenants ............................................................................. 69   
Configure the callback dialog language .............................................................................................. 69   
10: Configure the VCS ................................................................................................................. 70   
Features and capabilities ....................................................................................................................... 70   
Gatekeeper registration................................................................................................................. 70   
Inbound/outbound dialing............................................................................................................ 70   
IVR capabilities and limitations ..................................................................................................... 71   
Monitoring capabilities .................................................................................................................. 71   
Codecs .............................................................................................................................................. 71   
Multiple VCSs ................................................................................................................................... 71   
Load balancing ................................................................................................................................ 71   
Multiple gateway service policy .................................................................................................... 72   
System administration and configuration .................................................................................. 72   
Components .................................................................................................................................... 72   
Signaling features ................................................................................................................................... 72   
Gatekeeper discovery ..................................................................................................................... 72   
Registration ..................................................................................................................................... 72   
Keep Alive interval .......................................................................................................................... 73   
Terminate connection .................................................................................................................... 73   
H.225 call setup ............................................................................................................................... 73   
H.245 control channel .................................................................................................................... 73   
IVR handling .................................................................................................................................... 73   
SIP signaling .................................................................................................................................... 73   
Set the RTP frame packet size ............................................................................................................... 74   
Debug the VCS ......................................................................................................................................... 75   
Limiting the log file size ................................................................................................................. 75   
Configure VCS load balancing ............................................................................................................... 76   
Configure the VCS for Teams ................................................................................................................. 76   
Stack configuration ................................................................................................................................. 77   
System parameters......................................................................................................................... 78   
Syntax ....................................................................................................................................... 78   
Parameters .............................................................................................................................. 78   
H.245 parameters ........................................................................................................................... 80   
Syntax ....................................................................................................................................... 80   
Parameters .............................................................................................................................. 81   
Q.931 parameters ........................................................................................................................... 82   
Syntax ....................................................................................................................................... 83   
Parameters .............................................................................................................................. 83   
RAS parameters............................................................................................................................... 84   
Syntax ....................................................................................................................................... 85   
Parameters .............................................................................................................................. 85

CCSP System Administrator Guide - 7.3 and 7.4   
   
6   
11: Configure dialing plans ........................................................................................................ 88   
Configure the VCS ................................................................................................................................... 88   
Enable agent relative number translations ......................................................................................... 89   
Enable routing code remapping ........................................................................................................... 89   
Test the dialing plan ................................................................................................................................ 90   
Customize dialing plans ......................................................................................................................... 91   
CheckPrivileges input parameters ............................................................................................... 91   
CreateCSList input parameters ..................................................................................................... 92   
12: Configure SNMP traps .......................................................................................................... 93   
Implement traps ...................................................................................................................................... 93   
Configure traps ........................................................................................................................................ 94   
13: Configure the ECS ................................................................................................................. 95   
Configure ECS services ........................................................................................................................... 95   
Restart ECS services ................................................................................................................................ 96   
Configure tenant mappings ................................................................................................................... 97   
Command line reference ............................................................................................................... 98   
Administration mode command line ................................................................................ 105   
ECS upgrade process ............................................................................................................................ 108   
14: Configure the TWS .............................................................................................................. 109

CCSP System Administrator Guide - 7.3 and 7.4   
   
7   
1: Intro   
This document is for CCSP landlord administrators who configure and manage the CCSP platform,   
and reseller administrators who manage only the tenant properties and tenant resources of their   
assigned tenants.   
What’s new   
Version 7.4   
•   
6: Configure Dashboard:   
o   
Configure the Dashboard service — added parameter FromEmailAddress.   
o   
Configure the Dashboard application — added parameter defaultLanguage.   
o   
Configure the Dashboard application language — added instructions.   
•   
13: Configure the ECS — updated for Elsbeth 9.x.   
•   
Removed CCSP Admin information from this guide. Instead, please refer to the CCSP Admin   
Help - Landlord online help or PDF.   
Definitions   
Term   
Definition   
agent   
A contact center representative, typically a customer service representative,   
collection agent, or a remote help desk support person.   
ASR   
Automated Speech Recognition — automatic conversion of spoken words to   
text.   
Call   
parameter   
A call is associated with these parameters: TqoS, skills required, queue   
destination, caller name, URL, and any number of optional parameters.   
Contact   
center   
Generally, reservation centers, help desks, collection agencies, customer service   
centers, and the like.   
Group   
A collection of agents working in the same area of expertise (skill set) in the   
contact center. The agents in a group receive calls from the same queues.   
KCO   
Keep Connection Open — enables agent to remain connected to CCSP through   
an alternative device for the duration of their shift.   
MSR   
Message Storage Repository

CCSP System Administrator Guide - 7.3 and 7.4   
   
8   
Term   
Definition   
MSR DB   
General term for referencing database type that stores data for emails,   
callbacks, and voicemail.   
•   
Active MSR DB — database type that stores data for new and ongoing   
emails, callbacks, and voicemails. Also called the Tenant Messaging   
Database.   
•   
Archive MSR DB — database type that stores data for handled, rejected, or   
deleted emails, callbacks, and voicemails. Also called the Tenant Messaging   
Archive Database.   
Priority   
Independent of TqoS, you can assign priorities to calls arriving at your contact   
center 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 calls (telephone, email,   
Internet, video) 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 small program written for a command interpreter. Scripts are used to respond   
to pre-defined events that may be triggered in CCSP.   
Skill   
A skill is an attribute of an individual agent, who is defined by the administrator   
in the agent administration procedure. Skills (skill requirements) are also   
attributes of individual calls, which may be assigned to the call in the   
construction of the call request by the web or IVR application.   
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.   
TTS   
Text to Speech — the synthesis of a spoken sound version of a text.   
Wrap up code   
Code that represents the result of a call.

CCSP System Administrator Guide - 7.3 and 7.4   
   
9   
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 content in this document, please open a case with   
Enghouse Support.

CCSP System Administrator Guide - 7.3 and 7.4   
   
10   
2: CCSP overview   
CCSP is a carrier-class contact center system, specifically designed for service providers and large   
scale enterprise users, that supports multimedia customer interactions using all media channels —   
telephone and internet, live and message-based, fixed and wireless — in one high-capacity, high-  
availability, multi-tenant system.   
Instead of a legacy, circuit-switched Automatic Call Distributor (ACD), CCSP uses industry standard   
computing elements and current software development techniques to create a completely IP-based   
contact center. The all-IP environment is a unifying force for all media and all channels, replacing   
traditional Computer-Telephone Integration (CTI) with Computer-Computer Integration (CCI).   
CCSP addresses the needs of Application Service Providers (ASPs) offering hosted contact center   
services, as well individual enterprises having their own contact centers. This capability allows the   
system owner or operator, or an enterprise IT services department, to provide hosted contact   
center services to the tenants. This service is called a virtual contact center because the tenant has   
access to all contact center resources required to provision, administer and operate a live contact   
center in the same manner as if they owned the contact center infrastructure. The overall system   
provisioning and administration is distributed between the system administrator and their tenants.   
Each has a defined role in the operation of the CCSP system.   
Components   
The CCSP system comprises the following major components.   
Automatic Call Distribution (ACD) Server   
The Automatic Call Distribution (ACD) Server routes incoming calls to agents. The ACD Server has   
subcomponents of its own, these being:   
•   
Configuration Manager (CFM) — retains data regarding clients (tenants), users, agent   
groups, queues, skills, teams, events, scripts and connection servers. All system and tenant   
data is controlled by the Configuration Manager, and resides in a single database, the   
configuration database. This includes the former IPC addressing server.   
•   
Call Management Server (CMS) — call distribution engine. Matches calls to appropriate   
agents, based upon information received from the Configuration Manager.   
•   
Logging Server — logs data for historical reporting. Each client (tenant) has its own database   
in the logging server.   
•   
Statistics Server — logs data for real-time reporting.   
Administrator Interaction Server (Admin IS)   
Administrator Interaction Server (Admin IS) is the CCSP component that provides TCP/IP-based   
communication with Administrator applications over IP sockets.

CCSP System Administrator Guide - 7.3 and 7.4   
   
11   
Agent Interaction Server (AIS)   
Agent Interaction Server (AIS) is the CCSP component that provides TCP/IP-based communication   
with CCSP UI over IP sockets. This component manages all the interactions between the CCSP UI   
applications and the rest of the system.   
CCSP Admin   
CCSP Admin queries and manipulates the configuration database. The component consist of a   
backend API and a client-side application hosted in the IIS.   
Clients   
The CCSP system services a number of types of clients:   
•   
CCSP UI — used by users working with a browser   
•   
IP Phone — used as standalone device, or in conjunction with one of the agent applications for   
synchronous call control and computer- telephone integration (CTI) capabilities. Call control   
can be accomplished through both telephone keys and, optionally, CCSP UI.   
•   
Phone Agent — used by agents working through any telephone, without CCSP UI. Call control   
is through telephone keys for Answer and Hang-up, and through Interactive Voice Response   
(IVR), for example, for logon/off and availability.   
   
CCSP UI is an application in the user's browser that enables the user to accept incoming calls and   
originate outgoing calls. No separate telephone of any kind is required.   
A CCSP UI client can work in hybrid mode with any external voice devices, such as POTS telephones   
or IP phones connected to a PBX or the PSTN.   
An agent application client can also use an IP phone as a peripheral device for voice delivery and   
call control. In this configuration, CCSP provides the full management of the IP phone without the   
need for a third party PBX.   
CCSP UI can support video calling in hybrid mode with an IP phone that supports video or a   
softphone program in the user's computer.   
Caution   
The IP phone or softphone must support SIP.   
   
The CCSP applications service three types of users:   
•   
Business user — standard office workers   
•   
ReadyRep — knowledge workers and others who sometimes receive ACD-routed calls   
•   
Phone Agent — contact center customer service representatives who spend the majority of   
their day handling ACD-routed customer contacts   
   
The Business User license provides basic unified communication functions. It can only be reached   
by direct dial or direct transfer. The business user cannot be a destination of the Automatic Call   
Distribution (ACD) function of the CCSP system or participate in outbound dialing campaigns.

CCSP System Administrator Guide - 7.3 and 7.4   
   
12   
Agent and ReadyRep can be used as ACD destinations. Agent licenses are for traditional, formal   
contact center agents. ReadyRep licenses are designed for use by people who are not solely or even   
primarily dedicated to a formal contact center, but who need to be part of the contact center   
infrastructure in most other ways. They are sometimes known as informal agents or experts, and   
their ACD use is part-time. Supervisors can monitor and remotely manage agents, but not   
ReadyReps.   
The telephones, wireless devices, and the computers illustrated in the CCSP logical connections   
support the users and the callers. See also 5: Configure CCSP UI.   
CCSP Caller   
CCSP Caller enables any web-browsing user to call a CCSP-enabled contact center without any   
software download or installation.   
Device Proxy Server (DPS)   
The Device Proxy Server (DPS) is the CCSP component that provides communication with IP phones   
and manages all the interactions between IP phones and the rest of the system. DPS enables   
customers to use their IP Phones as agents in the system without needing to install any desktop   
components. Alternatively, the KCO-DPS keeps SIP alternative devices connected for the duration of   
the agent logon.   
Internet Chat Server (ICS)   
The Internet Chat Server (ICS) is the CCSP component that manages web chat sessions. The   
Internet Chat Proxy Server is a subcomponent of this server.   
Message Connection Server (MCS)   
The Message Connection Server (MCS) enables a contact center to receive email, voicemail and   
callback messages in call context.   
Messaging Server (MS)   
The Messaging Server (MS) serves as a the CCSP abstraction layer server with different types of   
Message Repositories (MSR), isolating it from each MSR brand and its specific behaviors. The MS   
collects the non-interactive media items (email, voicemail, and callback messages) that are sent to   
the MCS servers for call context handling.   
Predictive Dialing Server (PDS)   
The Predictive Dialing Server (PDS) is used to manage telephone calling campaigns, and works with   
the Dialer Campaign Manager.   
PDS is only installed if you are using the Dialer Campaign Manager to run calling campaigns. For   
more information, refer to the CCSP Dialer Guide for Elsbeth/Sytel.

CCSP System Administrator Guide - 7.3 and 7.4   
   
13   
Replay   
Replay queries the recording database. The component consist of a backend API and a client-side   
application hosted in the IIS.   
Voicemail Server (VMS)   
The Voicemail Server (VMS) is the CCSP component that enables IP phone and CCSP UI users to   
retrieve voicemail.   
VoIP Connection Server (VCS)   
The VoIP Connection Server (VCS) allows incoming and outgoing voice calls. The VCS is a CCSP   
component that provides a standard H.323 or SIP interface to the network. The VCS also provides   
IVR, supervisory-monitoring capability and facilitates connection of telephone calls to users via an   
intermediate VoIP media gateway or network of gateways. The VCS also implements music on hold   
and voicemail handling.   
The VCS has member components of its own, these being:   
•   
The RTP Relay Server — receives RTP packets with destination addresses, and replaces the   
destination addresses with different IP address, UDP port, or both.   
•   
The Interactive Media Server (IMS) — supports delivery and receipt of DTMF transported   
through the network via H.245 signaling. The gateway detects DTMF signals and sends them   
to the VCS.   
Interaction recording   
Interaction recording is a standalone product that works in conjunction with your CCSP system to   
capture, store, and enable access to all caller-agent interactions, for all media types. It works with   
the VCS, ICS, MCS and CCSP UI to provide a complete record of all agent-caller interactions.   
All of the information about the recorded calls and the actual recordings themselves are stored in   
an SQL database. You can establish a separate database for each tenant. When a supervisor or   
tenant administrator selects a recording to view, the recorded data is retrieved from the database.   
The connection servers and subsystems operate according to the tenant's Designer scripts to   
decide which calls to record. Each connection server or subsystem handles this a little differently:   
•   
VCS — the VCS writes the call information and the audio streams of the call to a file on a   
network shared folder. The Data Transport Server (DTS) monitors the network shared drive   
while waiting to process new recordings.   
•   
ICS — the ICS writes the call information and the chat text from the call to a file on a network   
share. The DTS monitors this network share waiting to process new recordings.   
•   
MCS — the MCS collects the call information, the email content of the inbound email and the   
email content of any and all replies and/or forwards of this email. At the end of the call, the   
MCS uploads this information to the recording database via a web page.

CCSP System Administrator Guide - 7.3 and 7.4   
   
14   
•   
CCSP UI — CCSP UI is used in screen recording to capture the screen recordings from the   
user's computer and upload them via the Screen Recording Server and DTS to the recording   
database.   
Interaction recording components   
Interaction recording has components of its own:   
•   
Data Transport Server (DTS) — the DTS monitors the network shared folder(s) where the   
VCS, ICS, and SRS servers write the temporary recording information. When the DTS has   
determined that a particular file has finished being updated by the connection server, it   
processes it. This processing consists of uploading the call information (for example, time of   
call, duration of call, call type, ANI, DNIS, and email address) and the actual recording to the   
recording database. For voice calls, an additional step of combining and compressing the   
audio streams into GSM format is also performed.   
The DTS is set up during CCSP installation. For configuration instructions, refer to Configure the   
Data Transport Server in the CCSP Admin Help - Landlord online help or PDF.   
•   
Screen Recording Server (SRS) — the SRS is a virtual directory in your network that   
temporarily store the screens recorded in user sessions. It resides in one of the computers   
used as a server. The SRS receives the screen files from the Background Intelligent Transfer   
Service (BITS).   
•   
Screen Recording Service — the Screen Recording Service is a CODEC (coder/decoder)   
automatically installed on the user's computer at the time of CCSP UI installation. When screen   
recording is active, this component records screen activity and stores it in a temporary file on   
the user’s computer until it is uploaded.   
•   
Client Uploads Service — the Client Uploads Service is automatically installed on the user's   
computer at the time of CCSP UI installation. After the recording is completed or the call is   
hung up, it transfers files generated by screen recording to the Screen Recording Server, using   
BITS.   
•   
Replay — the Replay web interfaces are used by supervisors to review saved recordings. In the   
web interfaces, supervisors can sort recording by time/date, user, call type, ANI, and DNIS.   
The Replay websites are normally established during CCSP installation. They can also be   
created by the system administrator.

CCSP System Administrator Guide - 7.3 and 7.4   
   
15   
Contact center performance report tools   
CCSP provides the following types of contact center performance reports:   
•   
Real-time reports — Dashboard displays current performance statistics. For more information,   
refer to the CCSP Dashboard Help.   
•   
Historical reports — Historical Reports displays historical performance statistics in standard   
reports provided with CCSP. For more information, refer to the CCSP Historical Reports Guide.   
•   
Interaction recording — Replay enables CCSP users to capture, store, and access all caller-  
agent interactions regardless of the call type. For more information, refer to the CCSP Replay   
Help.   
   
The report interfaces and database structure allow each tenant access to their data without   
compromising the data privacy and security of other tenants. Since access to these reports is   
browser-based, the tenant's supervisory personnel have easy access to the contact center's   
operations from virtually anywhere.   
Tenant statistical views   
Tenants can display both historical and real-time data in detail.   
Data is displayed in a windowed format, which allows a supervisor within that tenant to have   
multiple views of their contact center on one screen. Each supervisor selects these views.   
These reports provide tenants detailed historical and instantaneous information about the   
performance of their contact center operations.

CCSP System Administrator Guide - 7.3 and 7.4   
   
16   
How CCSP works   
The CCSP design is based on client/server architecture connected through a local area or wide area   
network (LAN or WAN). CCSP uses the same packet switching TCP/IP-based technology that makes   
the Internet, intranets, and extranets work.   
This architecture provides a cost effective solution which is not possible with traditional ACDs,   
including direct calls from computer to computer that support voice, video, and data   
communications.   
In addition, CCSP supports email routing using CCSP's Message Communication System as well as   
private (modem-linked) dial-ins and telephone connections with IVR using CCSP's Connection   
Servers.   
Furthermore, CCSP is an open, software-based solution. This unique architecture makes it easy to   
integrate CCSP into your existing information and communications systems environment.   
   
On the Internet side, callers can use any typical computer with a current release of the Microsoft   
browser, and optionally conference software, installed. The more the computer is equipped for   
multimedia, the more communication options the caller has with the contact center users.   
With a headset, or speakers and a microphone, an Internet caller can speak with the user just as in   
a telephone call. Without a microphone, callers can chat by typing their messages while listening to   
the user. For callers without a sound card, keyboard chatting in both directions is possible. In all   
connections, the caller can see the user (if equipped with a camera), physical samples, and/or   
prepared video material. When the caller also uses a video camera-equipped computer , the user   
can see the caller and any material he or she may wish to present. The user and the caller can also   
co-browse websites and jointly fill out web forms.

CCSP System Administrator Guide - 7.3 and 7.4   
   
17   
Logical connections   
The CCSP back-end servers and services enable communications between the caller and the contact   
center clients. The CCSP back-end servers and services are:   
•   
Message Connection Service (MCS)   
•   
Internet Chat Server (ICS)   
•   
VoIP Connection Server (VCS)   
•   
Agent Interaction Server (AIS)   
   
All back-end servers and services reside in the CCSP Microsoft Windows domain. This simplifies   
administration tasks and provides better security. The following figure provides a visual   
representation of how the system is logically arranged.

CCSP System Administrator Guide - 7.3 and 7.4   
   
18   
Always On availability groups   
Note   
We recommend that you use Always On availability groups rather than database mirroring. We   
also recommend that you make Historical Reports highly available.   
   
The Always On availability groups feature is a high-availability and disaster-recovery solution that   
provides an enterprise-level alternative to database mirroring. Introduced in SQL Server 2012,   
Always On availability groups maximizes the availability of a set of user databases for an enterprise.   
An availability group supports a failover environment for a discrete set of user databases, known as   
availability databases, that fail over together. An availability group supports a set of read-write   
primary databases and one to eight sets of corresponding secondary databases.   
Optionally, secondary databases can be made available for read-only access and/or some backup   
operations.   
For information about CCSP databases, refer to Manage database groups in the CCSP Admin Help -   
Landlord online help or PDF.   
Historical Reports   
To make Historical Reports highly available, your DBA needs to add the Microsoft Reporting   
services databases (ReportServer and ReportServerTempDB) to an availability group in SQL Server   
and configure the corresponding secondary database on each server instance that hosts a   
secondary replica.   
The availability group can be different from the one that you use for the CCSP databases.   
For information about Reporting Services with Always On availability groups, refer to   
https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/reporting-  
services-with-always-on-availability-groups-sql-server.   
For information about adding a database to an availability group, refer to   
https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/availability-  
group-add-a-database.   
For information about configuring the corresponding secondary database on each server instance   
that hosts a secondary replica, refer to https://docs.microsoft.com/en-us/sql/database-  
engine/availability-groups/windows/start-data-movement-on-an-always-on-secondary-database-  
sql-server.

CCSP System Administrator Guide - 7.3 and 7.4   
   
19   
Remote site survivability   
Remote site survivability (RSS) enables the installation of backend CCSP components in different   
geographical locations (sites). If disconnected from the rest of the platform, RSS enables each site   
to continue to work independently, without a disruption to callers.   
During failover, each server communicates only within the boundaries of the site using the   
resources configured in the remote site's properties. For example, when AIS needs to route a call,   
the Remote Sites properties determine the local CMS to use. For information, refer to Manage   
remote sites in the CCSP Admin Help - Landlord online help or PDF.   
Failover to localized operation can be initiated:   
•   
Manually — the system administrator changes system state information.   
•   
Automatically — CCSP detects connection loss and performs failover.   
   
Recovery to normal mode at the remote site is initiated manually by the system administrator.   
IP telephony   
In addition to receiving telephone calls with the built-in softphone on the user’s PC, the system also   
allows telephone calls to be delivered to alternate devices such as standard circuit-switched phones   
or IP phones behind an IP PBX, or handset behind a regular PBX.   
This flexibility is advantageous for users who do not have enough bandwidth to handle VoIP, or to   
leverage the implementation of an already existing phone setup. An additional benefit of this   
feature is that it can vary on a user-to-user basis, and the user can change it on a call-by-call basis.   
The supported devices for telephone delivery include:   
•   
Softphones – browser-based or desktop-based, either SIP or H.323   
•   
Circuit phones behind a PBX or IP PBX   
•   
Landlines, for example home phone   
•   
CCSP supports the use of SRTP for the media traffic to be encrypted on its softphone   
Personal voicemail   
In addition, users can receive Direct Inward Dial (DID) extensions with voicemail support. A contact   
center can program the voicemail service with their own prompts and create their own voicemail   
features.   
Contact centers can route calls to DID extensions via script and direct callers to leave voicemail.   
Voicemail can be retrieved through scripts. The directly dialed extensions can be used together   
with the ACD and outward dialing campaign features of CCSP.

CCSP System Administrator Guide - 7.3 and 7.4   
   
20   
Interactive Voice and Video Response (IVVR)   
Interactive Voice and Video Response (IVVR) allows callers to access contact center functions from   
video capable devices, such as video phones, mobile video phones, or computers, in a manner   
similar to that of IVR.   
The diagram below shows IVVR using the H.323 protocol to process incoming video calls. This   
means that gateways are required to convert the incoming video signal to H.323. If the devices   
from which the calls come in support H.323 natively, the Device Video Protocol gateway can be   
bypassed, but it is still a necessary component of the system, as an incoming H.323 signal cannot   
be guaranteed otherwise.   
   
When an incoming call is determined to be a video call, the caller is presented with video prompts,   
in a manner similar to IVR. These prompts may include audio and text. The caller responds using   
DTMF tones, and the displayed video and audio changes based on the caller's input. In addition,   
video on hold is available.   
When the call is routed to an agent, the agent workstation establishes a VCS-tromboned connection   
to the caller using a video client.   
Two Designer blocks are available to process video calls for IVVR:   
•   
Play Video   
•   
Stop Video   
   
IVVR video prompts display in one of two formats:   
•   
Common Intermediate Format (CIF) — a video format used in video conferencing systems,   
which supports both NTSC and PAL signals, with a data rate of 30 frames per second (fps), with   
each frame 352 x 288 pixels in width and height.   
•   
Quarter Common Intermediate Format (QCIF) — a video format in which frames are 176 pixels   
wide and 144 pixels high (one quarter the size of the CIF format).

CCSP System Administrator Guide - 7.3 and 7.4   
   
21   
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   
Speech-enabled contact centers   
Speech applications enhance Interactive Voice Response (IVR) with two capabilities:   
•   
Text-to-Speech (TTS) — a technology that converts written text to audio and enables callers to   
hear the text.   
•   
Automated Speech Recognition (ASR) — a technology that allows users of IVR systems to   
speak entries instead of making a selection through DTMF.   
   
The VCS uses the Media Resource Control Protocol (MRCP) mechanism to communicate directly as   
a client device with speech servers. The VCS also supports Hewlett Packard's (HP) OpenCall Media   
Platform (OCMP) comprehensive voice and video services system as an intermediate device to   
communicate with MRCP-supporting speech servers.   
Speech servers in the market include Nuance, IBM, Telisma, and Loquendo. Enghouse Interactive   
tests speech server integration with Nuance. Other MRCP-based speech servers should also be   
compatible — check with Enghouse Interactive for the latest certifications.   
The system administrator can grant speech capabilities to some or all tenants through CCSP Admin.   
Tenants configured with these capabilities use speech-enabling blocks in Designer to create   
applications employing TTS and ASR. These blocks are:   
•   
Set Language   
•   
Define Grammar   
•   
Play text   
•   
Speech Enabled Menu   
•   
Speech Play and Collect   
   
These blocks are translated into VXML elements and carried over the MRCP interface from CCSP to   
the speech servers. When HP OCMP is used, the VXML elements are carried through the MRCP   
interface to OCMP and then to the speech servers.   
For more information on Designer and these blocks, refer to the CCSP Designer Help.   
The system administrator can deploy speech servers supporting one or many languages and   
dialects. The language is configured through the Set Language block when tenants create   
applications.   
Note   
Inform tenants which languages are available to them.

CCSP System Administrator Guide - 7.3 and 7.4   
   
22   
To use the speech capabilities in their Designer applications, tenant administrators create a   
grammar that specifies the words and patterns of words to be listened for by a speech recognizer.   
Grammar creation rules are specific to the third-party speech engine products used with CCSP, and   
are not covered in CCSP documentation.   
Teams integration   
CCSP enables agents to use their desktop Teams client as an alternative device for voice call   
handling, to synchronize their presence with CCSP UI, to search for contacts in the Teams directory,   
and to dial Teams contacts directly from CCSP UI.   
   
The Teams Integration consists of a server component – Teams Web Service (TWS) – and a client   
component – Teams Gadget. TWS communicates with Teams Servers to get the presences for   
Teams contacts. The Teams Gadget receives the presence updates from TWS and provides the   
ability to search and to dial Teams contacts. Dialing of Teams contacts is achieved by enabling a   
route between the VCS and Teams servers through your vendor-provided SIP or PSTN gateway.   
See Configure the VCS for Teams.

CCSP System Administrator Guide - 7.3 and 7.4   
   
23   
Messaging modes   
In CCSP 7.3, there are two messaging modes:   
•   
Tenant messaging database (MSR DB) and external mailboxes — recommended for new   
installations or provisioning new tenants on existing installations. From CCSP 7.3.4 onward,   
data in each MSR DB (Active-MSR DB) can be archived to a single Archive database (Archive-  
MSR DB). The administration of Archive-MSR DB is available only via the new CCSP Admin.   
•   
Exchange 2010 as MSR — inherited from CCSP 7.2.x   
   
A Message Storage Repository (MSR) is required for the messaging media, which are:   
•   
Email — system-routed email and system-relayed outgoing email   
•   
Voicemail — system-routed voicemail   
•   
Callback — system-routed scheduled callbacks   
   
Each tenant has three separate data store types, one for each media type. Each data store specifies   
the MSR DB (or Exchange 2010 as MSR) to use for that data type.   
Each MSR DB can be used:   
•   
As a single or multiple media type store.   
•   
As a single or multiple tenants store.   
   
Each Active-MSR DB can optionally be associated with one Archive-MSR DB. Data is moved from the   
Active-MSR DB to the Archive-MSR DB according to the configuration definitions in CCSP Admin.   
The following table shows the data store use for the two modes.   
Data store   
MSR DB and external mailboxes   
Exchange 2010 as MSR   
Email   
Supported   
Supported   
Callback   
Supported   
Supported   
Voicemail   
Supported   
Not supported\*   
\* CCSP no longer sends voicemails as SMTP messages   
   
You can apply either mode, where supported, to each data store per tenant, as shown below.   
Tenant   
Data store   
Mode   
A   
Email   
MSR DB and external mailboxes   
   
Callback   
MSR DB and external mailboxes   
   
VoiceMail   
MSR DB   
B   
Email   
Exchange 2010 as MSR

CCSP System Administrator Guide - 7.3 and 7.4   
   
24   
Tenant   
Data store   
Mode   
   
Callback   
Exchange 2010 as MSR   
   
VoiceMail   
MSR DB   
C   
Email   
Exchange 2010 as MSR   
   
Callback   
MSR DB and external mailboxes   
   
VoiceMail   
MSR DB   
An external mailbox is a mailbox on a mail server. The mail server must support secured IMAP and   
SMTP protocols and can be any of the following:   
•   
Exchange 2010 of an existing installation   
•   
Mail server on a private network, for example, on customer premise Exchange 2010   
•   
Cloud-based mail service, for example, Google G Suite   
   
A tenant can use multiple mail servers and can provision and operate multiple external mailboxes   
on a chosen mail server.   
Your tenants are responsible for provisioning and operating external mailboxes, unless you host   
the mail server for your tenants. Using CCSP Admin, your tenants:   
1.   
Define each external mailbox as a separate mailbox.   
2.   
Define email folders and associate mailboxes to them. Each email folder may associate with   
one mailbox. Each mailbox may associate with multiple email folders.   
   
If the chosen mail server supports using aliases, an external mailbox can use them to handle   
several email addresses. CCSP matches items in the external mailbox based on the email address   
(that is, the alias) with the email folder that has the matching alias. Using CCSP Admin, your   
tenants:   
1.   
Define one email folder per alias.   
3.   
Assign the respective mailbox to these email folders.   
   
You can migrate content from an existing Exchange 2010 MSR to an MSR DB, per tenant. For   
details, refer to the CCSP 7.2.x to 7.3.x Upgrade Guide or CCSP 7.2.x to 7.3 Tenant Migration Guide.   
To provision MSR DB and external mailboxes, refer to the Preparing MSR DB and external mailboxes   
chapter of the CCSP 730 to 73x Prerequisites Guide.

CCSP System Administrator Guide - 7.3 and 7.4   
   
25   
Multi-tenancy concepts   
CCSP addresses the needs of Application Service Providers (ASPs) offering hosted contact center   
services, as well individual enterprises having their own contact centers. This capability allows the   
system owner or operator, or an enterprise IT services department, referred to as the system   
administrator, to provide hosted contact center services, in the form of a virtual contact center, to   
the tenants. This service is referred to as a virtual contact center because the tenant has access to   
all contact center resources required to provision, administer and operate a live contact center in   
the same manner as if they owned the contact center infrastructure. In this model, the overall   
system provisioning and administration is distributed between system administrators and their   
tenants. Each has a defined role in the operation of the CCSP system.   
System administrator roles in a multi-tenant contact center   
The system administrator is typically the owner of the CCSP infrastructure. In this role, the system   
administrator is responsible for set-up, provisioning and maintenance of the components. These   
include all Connection Servers, ACD components and ancillary equipment and services such as   
backup and restore systems. The system administrator creates and deletes tenants and provisions   
hardware and software components of the system such as VoIP connections in the Connection   
Servers and mail administration in the MCS mail server.   
The system administrator provides personnel for system administrator level administration. These   
system administrators have the ability and permissions to provision and administer the CCSP   
resources. The system administrator is also responsible for tenant determination. This means   
provisioning the system to route the call to the proper tenant upon call arrival. For example, the   
tenant determination for a telephone call can be based on the incoming trunk group or DNIS   
information.   
Tenant roles in a multi-tenant contact center   
The tenant is typically responsible for self-provisioning of all contact center resources. This is   
accomplished through the use of CCSP Admin, which ensures that the tenant can only view and   
modify their contact center.   
This includes administration of contact center administrators, users, teams, groups, skills, queues,   
scripts, wrap-up and release codes, and accounts for the contact center. Accounts are typically used   
by a tenant who is an outsourcer, to track calls for different clients within one tenant. The   
outsourcer can assign the account name to each call for their own use, such as billing.   
Optionally, the system administrator can perform some or all of the administration of the contact   
center on behalf of a tenant.   
Administrative hierarchy for multi-tenancy   
In CCSP Admin, the system administrator is at the root and controls the infrastructure and tenant   
creation branches. The system administrator also has access to all lower branches and therefore   
can configure tenant contact center resources. The tenant administrator uses the same CCSP

CCSP System Administrator Guide - 7.3 and 7.4   
   
26   
Admin, but the program displays only information relevant to the tenant, and therefore, provides   
isolation from both the system administrator and other tenants.   
System Administrator supported data   
CCSP Admin supports the following system administrator data:   
•   
Contact center name   
•   
Connection servers (server profiles)   
•   
System administrator personnel   
•   
Tenant creation and deletion   
•   
Tenant determination   
Tenant Administrator supported data   
CCSP Admin supports the following tenant administrator data:   
•   
Contact information   
•   
Personnel (agents, supervisors, and tenant administrators)   
•   
Teams   
•   
Groups   
•   
Skills   
•   
Queues   
•   
Scripts/events   
•   
Wrap-up codes   
•   
Release codes   
•   
Accounts   
•   
Block codes   
•   
Wave files   
•   
Manage files   
•   
Mailboxes

CCSP System Administrator Guide - 7.3 and 7.4   
   
27   
Privacy protection   
Based on the SIP protocol, CCSP provides a refined degree of privacy protection, which the system   
administrator configures when they provision the tenants.   
Regulating information exposed to called parties   
Through CCSP Admin, the system administrator configures how each tenant handles caller identity   
and privacy. The system administrator determines if:   
•   
The caller's identity is exposed to agents.   
•   
The caller's identity is hidden from agents but exposed to scripts and the database.   
•   
The caller's identity is hidden from everything, including agents, scripts and the database.   
   
The system administrator also configures how each tenants' own identity is exposed or protected   
in the public network, or exposed to or hidden from the called party.   
Tenant uses SIP privacy information in scripts   
Tenants can use the SIP INVITE header information in contact center scripts.   
The Designer scripting tool enables tenants to use the SIP privacy header information to route   
incoming calls through their contact center.   
System administrator tasks   
After the CCSP platform has been installed, the system administrator needs to perform certain   
procedures to set up and administer the system. These procedures can be divided into these main   
tasks:   
1.   
Set up the system for use. Create a default system administrator.   
2.   
Add tenants to the system (provisioning). This process only needs to be performed once for   
each tenant.   
3.   
Perform the following system operations and administration:   
o   
Start and stop   
o   
Backup and restore   
o   
Add and delete connection servers   
   
An enterprise with its own contact center should follow the directions in this guide for provisioning   
a tenant. The contact center is the tenant of the enterprise.   
Setting up the CCSP platform   
As part of the initial installation procedure, the system administrator must ensure that the system   
is properly configured for use. This configuration process needs to be performed only once and   
consists mainly of creating one or more system administrators, who are authorized to act on behalf

CCSP System Administrator Guide - 7.3 and 7.4   
   
28   
of the system administrator and who can perform the other tasks described in this Help. For more   
information on system installation, refer to the CCSP 712 to 72x Installation Guide.   
The DefaultTenant should be created as part of the CCSP installation.   
Add a new tenant   
This process, also referred to as provisioning, requires the following general steps:   
1.   
Add a new tenant to the database.   
2.   
Add at least one user with administrative rights to the tenant personnel.   
3.   
Perform other setup tasks as required.   
System operation and administration   
In addition to adding new tenants, system administrator tasks include:   
•   
Updating tenant contact information, such as contact name and address.   
•   
Performing other tasks at tenants' requests which they can't perform themselves due to   
security or domain issues.   
•   
Starting and stopping communication servers.

CCSP System Administrator Guide - 7.3 and 7.4   
   
29   
3: Additional tenant configuration   
Change the secret key   
Note   
•   
The Secret key should be common across the system   
•   
We recommend that you change the secret key occasionally to reduce the chance of external   
resources impersonating users by creating a fake token.   
•   
If you do not want to implement Single Sign On for CCSP UI, do not add the secret key in CCSP   
UI's web.config file.   
   
From CCSP 7.3 onwards, some of the client applications (CCSP Admin, CCSP UI, and Replay) use   
tokens to authenticate users. Tokens are created by the Authentication server, and signed using a   
secret key.   
The Secret key is like a password used by a CCSP application to authenticate a token as one   
originated by the Authentication servers. As a key, you can use any random string hashed as 256-bit   
WEP Key.   
For example:   
VgN8GWfWAsThyYVDGcqGKieMqS6jH0Mw   
0apla9wHIJsLo9yma9OxZsry5zNX3Q4s   
   
You can generate random 256 bit keys using the JWT Keys Generator UI: https://jwt-keys.21no.de   
To replace the secret key:   
1.   
Generate a random 256 bit key as described above.   
2.   
Open the Admin's web.config file and set the new OAuthSecretKey value.   
3.   
Open the Auth Server's web.config file and set the new OAuthSecretKey value.   
4.   
Open Replay's web.config file and set the new OAuthSecretKey value.   
5.   
If using SSO, open CCSP UI's web.config file and set the new OAuthSecretKey value.

CCSP System Administrator Guide - 7.3 and 7.4   
   
30   
Use Exchange as an external mailbox   
When using an email address that includes the domain suffix, Exchange accept the send attempts   
because that email address is part of the Exchange Accepted Domains, defined in Organization   
Configuration > Hub Transport.   
However, when using an email address that uses the tenant suffix, Exchange rejects the send   
attempt unless that tenant upn suffix is defined in the Exchange Accepted Domains, defined in the   
Organization Configuration > Hub Transport.

CCSP System Administrator Guide - 7.3 and 7.4   
   
31   
Help tenants use Designer   
Verify that the XML Interpreter is available   
To verify that the XML Interpreter has been properly loaded:   
1.   
On the navigation pane, click   
 Routing.   
2.   
On the menu, click Manage Files. The Manage files list appears.   
   
3.   
Click XMLInterpreter to view its properties. The File: XMLInterpreter dialog box appears.   
   
4.   
Do one of the following:   
o   
If the Size value is 0, then the XMLInterpreter.vbs file has not been loaded correctly and   
needs to be reloaded from …/CosmoCom/Server Components/setup/BLOBS.   
o   
If the Size value is greater than 0, click Cancel.

CCSP System Administrator Guide - 7.3 and 7.4   
   
32   
Use WebDAV to save scripts via HTTP   
WebDav can be used to upload Designer scripts over HTTP. On the web server that hosts Designer,   
scripts share the desired directory through web sharing.   
Since tenants generally do not have access to the contact center servers, you may be requested to   
perform this task on a tenant's behalf.   
1.   
On the server computer, right-click the Designer scripts folder and then click Properties. The   
folder Properties dialog box appears.   
   
2.   
Click the Web Sharing tab.   
3.   
Select the Share this folder check box.   
4.   
Click OK. The Edit Alias dialog box appears.

CCSP System Administrator Guide - 7.3 and 7.4   
   
33   
   
5.   
Select the Write check box.   
6.   
Select the Directory browsing check box.   
7.   
Click OK.   
8.   
Click OK.

CCSP System Administrator Guide - 7.3 and 7.4   
   
34   
Configure SRTP for a tenant   
SRTP functionality is available on the CCSP platform, using CCSP UI SIP Agents and an intermediate   
SBC device, which acts as a bridge between the VCS internal leg and the Agent application.   
Secure connection is needed only when the Agent location is outside of the protected corporate   
network. To accommodate different scenarios, you can enable or disable SRTP functionality both on   
the Tenant or Personnel level. Even if SRTP is not used, in some scenarios it is favorable to use the   
SBC device as a proxy between the Agent and the VCS. If both Use SRTP and Use SBC as Proxy are   
activated, Use SRTP takes precedence.   
Each tenant has two relevant endpoints on SBC (secure and non-secure). Particular endpoints   
should belong to only one tenant. When connecting to the secure endpoint, to establish the SRTP   
media connection, you must also protect SIP negotiation with TLS. To validate TLS connection with   
the SBC endpoint, each tenant requires a certificate signed by a Central Authority that is installed   
on the relevant SBC endpoint.   
The Central Authority certificate used for signing certificate installed on SBC also exists on Agent   
machines inside the Client Services directory. If during SIP initialization of an SRTP-enabled SIP   
Agent, a certificate with the appropriate certificate name is not found in the Client Services   
directory, or it can't be opened from some reason, a popup appears on the Agent machine and SIP   
fails to initialize.   
Note   
•   
You must configure the certificate location in ClientServiceConfig.JS. APS saves the certificates   
files in this location, creating a subfolder for each tenant ID.   
   
   
•   
If you change the certificate server location and/or want to stop the AgentWS download   
mechanism, navigate to C:\Program Files (x86)\CosmoCom\Server Components\AgentProxy and   
open Web.config in a text editor. Search for <add key="Use Certificates Mechanism"   
and set the value to 0. For example:   
   
<!-- SRTP Configuration -->   
 <add key="Use Certificates Mechanism" value="0"/>   
•   
If you stop the AgentWS download mechanism, you must physically copy the certificate file   
used in the corresponding SBC endpoint to each Agent machine inside the ClientServices   
directory for that Tenant, even if you previously uploaded the certificate via CCSP Admin.

CCSP System Administrator Guide - 7.3 and 7.4   
   
35   
To configure SRTP for a tenant:   
1.   
In Admin:   
a.   
Add a certificate.   
b.   
Configure the following Tenant parameters:   
▪   
SBC IP Address/Hostname   
▪   
SBC plain transport port   
▪   
SBC secure transport port   
▪   
Use SBC as Proxy   
▪   
Use SRTP   
▪   
SRTP CA Certificate Name — this name must match the relevant certificate file   
added under Tenants Information.   
2.   
On the CCSP server:   
a.   
Navigate to C:\Program Files (x86)\CosmoCom\Server Components\TouchPoint\ClientServices   
and open ClientServiceConfig.js in a text editor.   
b.   
Search for var certificateDirUrl = and replace "%CertificatesDirURL%"; with   
your certificate location.   
For example:   
var certificateDirUrl =   
"http://CCSP12.pj12.LOC/AgentWS/Certificates/";   
   
3.   
Tenant Administrator: in CCSP Admin, on the [Agent name] page, on the Phone settings tab,   
optionally configure Use SRTP.   
4.   
Agents: to activate the change, each agent must log on again.

CCSP System Administrator Guide - 7.3 and 7.4   
   
36   
Configure tenant time zones   
The default CCSP time zone is the time zone the system administrator established for the CCSP   
platform. This can be set to coincide with where the main platform is located, or another time zone   
that is convenient for the system administrator, for example, GMT. Administrators and supervisors   
often want to see reports based on other time zones, usually because contact centers and/or users   
are located in other geographic regions.   
Each tenant can have its own list of time zones for supervisors to choose from when generating or   
subscribing to a report. The list of time zones includes captions for each time zone, usually the   
name of each contact center location, and its time zone offset. The time zone offset is defined as   
the difference between the platform time zone and the desired time zone. For example, if the   
platform time zone is set for Dublin, Ireland but the tenant is in New York, the selected time zone   
configuration would be GMT-5:00. The time zone defined for the platform is displayed in reports as   
the Default time zone and its default time zone offset is 0.   
The list of time zones for each tenant is typically created by the system administrator, since it   
requires direct access to the tenant database, but it may also be created by a tenant administrator   
who has been granted such privileges.   
You can define a list of time zones that applies to all tenants assigned to a database.   
To create a list of time zone offsets, the system administrator adds records to the   
siTimeZoneOffsets table in the historical database. The columns of the table are:   
•   
TimeZoneOffset\_ID — int, primary key, ID of the time zone offset   
•   
TimeZoneOffsetinMinutes — time zone offset in minutes   
•   
TimeZoneOffsetName — nvarchar 255, name of the time zone offset. Usually, the name   
indicates the geographic location of the contact center.   
•   
ShowOnlyForTenant\_ID — if not NULL, indicates this offset applies only to the specified   
tenant. If NULL, the time zone applies to all tenants assigned to this database.

CCSP System Administrator Guide - 7.3 and 7.4   
   
37   
   
The default table has one record with TimeZoneOffsetName set to Default,   
TimeZoneOffsetinMinutes set to 0 offset, and ShowOnlyForTenant\_ID set to NULL.   
Compute the time zone offsets from the formula:   
Time Zone Offset (in minutes) = Requested Time Zone - Platform Time Zone   
For example, to create a time zone that adds 6 hours to the platform's time zone, specify 360 in   
TimeZoneOffsetinMinutes. To subtract time, include a negative value, for example, to display all   
reports an hour back, specify -60.   
The TimeZoneOffsetName serves as a label for the time zone determined by the offset.

CCSP System Administrator Guide - 7.3 and 7.4   
   
38   
Script-based answer supervision   
For certain ASP applications, it is necessary to begin IVR processing without sending answer   
supervision to the network. In this case, it may also be necessary for the IVR script to perform other   
call signaling activity, such as initiating answer supervision. This functionality mimics that of a PBX.   
For more information, see PBX-style answering below.   
To use script-based answer supervision, you must turn off auto-answer of a call by the VCS so that   
the IVR script can select Accept, Answer, or Reject a call.   
To turn off auto-answer:   
1.   
On the VCS computer, add the following registry subkey entry:   
HHKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\CosmoCom\Universe\CurrentVersio  
n\vcs\debug\script\_answer\_supervision   
2.   
Set the subkey to 1.   
3.   
Restart the computer.   
   
When auto-answer is turned off, you can use the following objects to handle the call in the IVR   
script:   
•   
line.AnswerCall — causes the VCS to send a connect message to the caller.   
•   
line.AcceptCall — causes the VCS to send an alerting message to the caller.   
•   
line.RejectCall — causes the VCS to send a releaseComplete message to the caller.   
PBX-style answering   
PBX-style answering is where the Q.931 Connect to the caller is not sent before the IVR, but is sent   
under control of the IVR. This allows the caller not to be billed unless the IVR decides to take the   
call.   
For example, suppose the IVR rejects calls between 19:00 and 7:00. This mode of operation allows   
the calls to be treated as busy, not as answered, by the telephone company.

CCSP System Administrator Guide - 7.3 and 7.4   
   
39   
Enable Web chat in a multitenant environment   
To configure multi-tenant support for WebCaller:   
1.   
Open the folder that contains the WebCaller files. The default location is C:\Inet-  
pub\wwwroot\CosmoCall.   
This folder contains subfolders like <NNNN>, where NNNN is the language ID. For example,   
the language ID for English (US) is 1033, so the folder will be   
C:\Inetpub\wwwroot\CosmoCall\1033.   
Each language folder has a subfolder called Default. This folder contains all the WebCaller files.   
2.   
Create a new folder under the language folder and copy all the files from the Default folder to   
the new folder. The new folder name should be identical to the tenant name, for example,   
C:\Inetpub\wwwroot\CosmoCall\1033\TenantName.   
3.   
Modify any HTML files in the new folder.   
4.   
If you need multi-language support for a new tenant, repeat steps 1-3 for each language   
folder.   
Caller language ID   
Each request for a new call has caller language ID and tenant name. ChatExtension uses caller   
language ID and tenant name to define the correct WebCaller file set. Lookup is performed by   
language ID first and by tenant name second. If the language folder does not exist, folder 1033,   
which is English, is used. If the tenant folder does not exist, the Default folder is used.   
Note   
If you change the tenant name, you must also change the folder name.   
Enable interaction recording for a tenant   
For detailed information about the recording database, refer to the CCSP Operation Guide.   
To enable a tenant to use interaction recording:   
1.   
Calculate the tenant's estimated recording database size and add to the total database size for   
all tenants. For details, see below.   
2.   
Create the tenant's recording database. For details, refer to the CCSP Installation Guide - 712 to   
72x.   
3.   
In Admin:   
a.   
Click   
 Database and add the database. In the Database type list, select Media   
Recording Database.   
b.   
Under the particular tenant's information, click   
 Tenant information and then click   
Data stores. Add the Recording Data Store.

CCSP System Administrator Guide - 7.3 and 7.4   
   
40   
Resolve Dashboard cross-domain authentication issues   
By default, for the Dashboard, IIS has both Negotiate (Kerberos) and NTLM set up as authentication   
methods.   
If you run the Dashboard client on untrusted machines (outside the CCSP domain), the Kerberos   
authentication cannot be completed due to ticket creation failure. The client machines then try to   
authenticate the Dashboard users against the local DC, using the principal name of the   
authenticated CCSP user. This name isn't recognized and is thus rejected.   
Negotiation then continues to NTLM, which works.   
Although there is no functional impact, using Kerberos authentication results in many   
authentication errors in the local DC.   
To resolve the Dashboard cross-domain authentication issues, you need to remove the Negotiate   
authentication method from the Dashboard IIS configuration.   
To remove Negotiate (Kerberos) authentication:   
1.   
Open IIS on the Web server where the Dashboard Web service is installed.   
2.   
In IIS Manager, in the Connections pane, expand Sites, expand Default Web Site, and then   
click rtrdll.   
3.   
In the Features View, under IIS, click Authentication.   
4.   
Under Authentication, right-click Windows Authentication and then click Providers.   
5.   
In the Providers dialog box:   
a.   
Under Enabled Providers, click Negotiate.   
b.   
Click Remove.   
c.   
Click OK.   
Connect Provisioning Portal (PRP) to CCSP Admin   
You can configure CCSP Admin to include an icon in the navigation pane that displays the   
Provisioning Portal (PRP) tool.   
To connect PRP to CCSP Admin:   
1.   
Navigate to \Program Files (x86)\CosmoCom\Server Components\Admin.   
2.   
Open configuration.js in a text editor.   
3.   
Search for prpAdminUrl: '',   
4.   
Type the PRP main page URL between the quote marks, for example:   
prpAdminUrl: 'http://CCSP1.pj16.loc/PRP/'   
5.   
Save the file.

CCSP System Administrator Guide - 7.3 and 7.4   
   
41   
Remove user profile folders   
The first time agents change their password, a new folder for their user profile is created in C:\Users   
on the CFM machine.   
When there are many agents, these folders consume a lot of disk space.   
To remove user profile folders:   
1.   
On the CFM machine, run gpedit.msc.   
2.   
Expand Computer Configuration, expand Windows Settings, expand Security Settings,   
expand Local Policies, and then click User Rights Assignment.   
3.   
Double-click Allow log on locally.   
4.   
Remove Users.   
Reduce agent logon time   
If you recently added a significant number of agents to CCSP or you experience prolonged logon   
times when agents try to log on, you need to update the SQL Server's statistics for the tables   
holding the logon information.   
To update the SQL Server's statistics, run the following commands from the SQL Server   
Management Studio query window:   
update statistics tPersonnel;   
update statistics tPersonnelParameters;

CCSP System Administrator Guide - 7.3 and 7.4   
   
42   
Change DNS records from the command line   
You can add or delete records on the DNS server using only the command line.   
The files for this operation are included on the distribution media in the following location:   
..\software\CosmoCom, and the filename is dnsConfig.zip.   
To remove or add records to the DNS, use the two batch files dnsAdd.bat and dnsDelete.bat.   
Each batch file accepts two parameters: Server name and Server IP. You need to edit all the other   
parameters.   
To enable this functionality:   
1.   
Unzip all the files to c:\dnsconfig   
2.   
Copy the DNSprov.DLL and dnsschema.mof to the C:\Windows\System32\wbem folder.   
3.   
At the command prompt, go to \Windows\System32\wbem folder and type mofcomp   
dnsschema.mof   
4.   
Register dnsprov.dll   
   
To add a record:   
1.   
Open the command prompt.   
2.   
Go to c:\dnsConfig   
3.   
Run dnsAdd.bat dnsRecordName dnsRecordIP   
   
To delete a record:   
1.   
Open the command prompt.   
2.   
Go to c:\dnsConfig   
3.   
Run dnsDelete.bat dnsRecordName dnsRecordIP   
   
The files should be located in the DNS server.   
You can also run the script directly, for example:   
C:\cscript dnsrecord.vbs /add A domain.com myServer myServerIP /S dnsHost /U   
administrator /W password   
C:\cscript dnsrecord.vbs /delete A domain.com myServer myServerIP /S dnsHost   
/U administrator /W password

CCSP System Administrator Guide - 7.3 and 7.4   
   
43   
Install G.729 codecs   
These instructions assume you know how to install G.729 codecs and provide information on how   
installation can affect the operation of your contact centers.   
Note   
To use the G729 codec, on the VCS server, the svccosmocall user needs to have full access to the   
following registry key:   
HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows   
NT\CurrentVersion\Drivers32   
   
Caution   
If the G.729 codec is installed while agents are connected, their connections will be broken.   
   
To avoid this problem, follow these steps:   
1.   
Coordinate the installations with the tenant administrators.   
2.   
Log off all agents.   
3.   
Stop these services:   
o   
DTS   
o   
IMS   
o   
RTP if separate from VCS   
4.   
Install the G.729 codec on these computers:   
o   
VCSs   
o   
DTR   
o   
IMS   
o   
RTP, if separate from VCS   
5.   
Restart   
o   
DTS   
o   
IMS   
o   
RTP if separate from VCS   
6.   
Reactivate the agents.

CCSP System Administrator Guide - 7.3 and 7.4   
   
44   
4: Configure CCSP Admin   
This chapter explains how to customize CCSP Admin's look and feel, the options that appear on the   
navigation pane and in the associated menus, the columns that appear in the entity list views, and   
the fields that appear in the entity-type dialog boxes (screens).   
   
Note   
•   
We recommend that you make a backup copy of a file before you edit it.   
•   
Changes to the look and feel, navigation, and definitions apply to all tenants. You cannot   
customize these settings for individual tenants.   
Customize the CCSP Admin look and feel   
You can customize CCSP Admin's look and feel by using your own customized CSS file. You must   
copy your CSS file to Server Components\Admin and specify its name in the configuration file. The   
CSS file can have any name.   
To customize the CCSP Admin look and feel:   
1.   
Navigate to Server Components\Admin.   
2.   
Copy the customized CSS file into this folder. You can use a subfolder.   
3.   
Open configuration.js in a text editor.   
4.   
Search for customizationCss. By default, this property is empty.   
5.   
Set the value to /Admin/ and the CSS filename. If the file is in a subfolder, you must specify   
the rest of the path after /Admin/.   
For example:   
customizationCss: '/Admin/customization.css',   
customizationCss: '/Admin/static/css/customization.css',   
6.   
Save the file.   
7.   
To apply the changes, in CCSP Admin refresh the page by pressing F5.   
CSS examples   
To change the ID column width in the Groups list, add the following CSS rule to the customization   
file:   
.groups table .id{ width:20px;}   
   
To hide the ID column in the Groups list, add the following CSS rule to the customization file:   
.groups table .id{ display:none;}

CCSP System Administrator Guide - 7.3 and 7.4   
   
45   
Customize the CCSP Admin navigation   
You can customize the options that appear on the navigation pane and in the associated menus by   
editing the navigation bar xml files.   
Caution   
In the navigation files, do not change any information in the rows. You can only comment out or   
delete rows.   
   
To customize the CCSP Admin navigation:   
1.   
Navigate to Server Components\Admin\bin\Navigations.   
2.   
To change the landlord navigation options (Tenants, Database, Tenant information), open   
LandlordNavigationBar.xml in a text editor.   
3.   
To change the tenant navigation options (Home page, Organization, Routing, and   
Miscellaneous), open TenantNavigationBar.xml in a text editor.   
4.   
Comment out or delete the rows that you do not require.   
5.   
Save the file.   
6.   
To apply the changes, in CCSP Admin log out and then log in again.   
Customize the CCSP Admin grid definition   
You can customize the columns that appear in the entity list views by editing the grid definition xml   
files.   
Caution   
In the grid definition files, do not change any information in the rows. You can only comment out   
or delete rows.   
   
To customize the CCSP Admin grid definition:   
1.   
Navigate to Server Components\Admin\bin\Grids.   
2.   
To change the landlord grid definitions, open LandlordGridsDefinitions.xml in a text editor.   
3.   
To change the tenant grid definitions, open GridsDefinitions.xml in a text editor.   
4.   
Comment out or delete the rows that you do not require.   
5.   
Save the file.   
6.   
To apply the changes, in CCSP Admin log out and then log in again.

CCSP System Administrator Guide - 7.3 and 7.4   
   
46   
Customize the CCSP Admin screen definition   
You can customize the fields that appear in the entity-type dialog boxes (screens) by editing the   
screen definition xml files.   
For example, the fields in the Queue dialog box are determined by the following settings in   
ScreensDefinitions.xml.

CCSP System Administrator Guide - 7.3 and 7.4   
   
47   
Each field of an entity screen has the following properties:   
•   
controlType   
•   
name   
•   
label   
   
Each controlType has its own additional properties, as shown in the following table.   
controlType   
Additional properties   
textBoxForString   
maxLength   
notAllowedChars   
textArea   
maxLength   
notAllowedChars   
multiselect   
controllerName   
listType   
checkBox   
N/A   
textBoxForInteger   
default   
maxValue   
minValue   
singleselect   
controllerName   
listType   
singleSelectRelatedData   
listType   
multiselectRelatedData   
listType   
   
You can also add the following true/false properties:   
•   
isDisabled   
•   
isHidden   
•   
isRequired   
   
Caution   
•   
If you change the value of a true/false property, you must be careful not to disable or   
hide required fields, such as the Name field that is required when adding a new entity.   
•   
You can remove fields from the screen by deleting the row. Do not comment out the   
row.   
•   
Do not change the values of the controlType, name, label, controllerName, and listType   
properties. You can change the other values.

CCSP System Administrator Guide - 7.3 and 7.4   
   
48   
To customize the CCSP Admin screen definition:   
1.   
Navigate to Server Components\Admin\bin\ScreenDefinitions.   
2.   
To change the landlord screen definitions, open LandlordScreensDefinitions.xml in a text editor.   
3.   
To change the tenant screen definitions, open ScreensDefinitions.xml in a text editor.   
4.   
Delete the rows for the fields that you do not require.   
5.   
Where allowed, edit or add properties as described above.   
6.   
Save the file.   
7.   
To apply the changes, in CCSP Admin log out and then log in again.   
Preserve customer overrides   
To preserve customizations after upgrade, create an xml file that defines what needs to be   
removed or modified and add this file’s path in the web.config.xml file. If provided, this file is read   
once on application start and stays in memory until the application is restarted.   
The example file customer\_overrides\_example.xml in the App\_Data folder has a documentation   
section that explains the purpose of each element, including its permitted attributes and allowed   
children elements. This example file includes a full example of a customization for each thing that   
can be customized (either removed or modified).   
The default customerOverridesFilePath value in the Web.config.xml is set to a non-existent file   
called customer\_overrides.xml in the App\_Data folder. You can either use this path and filename for   
your xml file or change the value.   
To change the value:   
1.   
Navigate to Server Components\Admin.   
2.   
Open Web.config.xml in a text editor.   
3.   
In the <appSettings> section, find   
<add key="customerOverridesFilePath"   
value="~/App\_Data/customer\_overrides.xml" />   
4.   
Change the path and file name.   
5.   
Save the file.

CCSP System Administrator Guide - 7.3 and 7.4   
   
49   
5: Configure CCSP UI   
For information about administering CCSP UI customizations and gadgets, including the Mini   
Wallboard, refer to the CCSP AdminPoint Help and the CCSP UI CSS Customization Guide.   
Configure the CCSP UI Updater   
Note   
•   
The initial installation of CCSP UI on an agent's computer requires Administrative rights to   
install the CCSPClientInstallationService and enable it to run under the System account.   
•   
CCSP UI supports TLS 1.0.   
   
CCSP UI Updater enables you to automatically install CCSP UI updates on an agent's computer   
using a Windows service (CCSPClientInstallationService), that runs under the System account.   
On the CCSP server machine, all CCSP UI updates are uploaded by default to   
\%ServerComponents%\TouchPoint\ClientServices.   
The CCSPClientInstallationService downloads the updates as MSI files from the CCSP server   
machine, according to the timeout interval, which is every 60 minutes by default.   
Depending on the install mode configured in ClientServicesConfig.js, the   
CCSPClientInstallationService then does one of the following:   
•   
Downloads the updates but does not install them.   
•   
Automatically installs the updates on the agent’s computer.   
•   
Asks if the agent would like to install the downloaded updates, and if yes, installs them.   
   
To configure the CCSP UI Updater:   
1.   
Navigate to \Program Files (x86)\CosmoCom\Server Components\TouchPoint\ClientServices.   
2.   
Open ClientServicesConfig.js in a text editor.   
3.   
Search for var clientServicesDirUrl and change the URL to match your installation.   
4.   
If the public address of the web server (the address used in clientServicesDirUrl) differs   
from the DNS address that the Web server knows itself by, search for var   
internalClientServicesDirUrl and change the URL to match the internal address for   
your installation.   
5.   
Search for var clientInstallationServiceTimeoutInterval and optionally, increase   
the frequency (in minutes).

CCSP System Administrator Guide - 7.3 and 7.4   
   
50   
6.   
Search for var downloadInstallMode and optionally, change the download install mode.   
The options are:   
o   
0 — always download and install (default)   
o   
1 — always download but never install   
o   
2 — download and prompt for install   
   
7.   
Search for var wcfBase and change the URL to match your installation. This value is the base   
URL for client services that are implemented as a WCF service.   
8.   
Search for the Connector URLS and change them to match your installation.   
9.   
Save the file.   
   
For information about configuring the agent's computer for CCSP UI updates, refer to the CCSP   
Tenant Administrator Guide and the CCSP UI Help.   
Disable editing of the default template and system   
gadgets   
After configuring the default template and system gadgets in AdminPoint, you can make them   
read-only for all users by setting EditDefaultSettings to false. This setting does not affect custom   
gadgets and additional templates that the customer created.   
Note   
This flag disables the option to edit the default setting of system gadgets from the Gadgets   
subtab, in addition to the default template.   
   
To disable editing of the default CCSP UI template:   
1.   
Navigate to \Program Files (x86)\CosmoCom\Server Components\AdminPoint\.   
2.   
Open Web.config in a text editor.   
3.   
Search for key="EditDefaultSettings" and change the value to false.   
4.   
Save the file.

CCSP System Administrator Guide - 7.3 and 7.4   
   
51   
Configure MS folder change notifications   
Stop   
Do not change these settings unless instructed by Enghouse Interactive Customer Support to do   
so.   
   
AgentWS maintains a local cache of the folder structure that is used for server FolderStructure API   
requests from agent clients (for example, CCSP UI) and permissions checks for the Pull and   
GetItems operations.   
You can configure how often the AgentWS is notified of changes to a tenant's folder structure by   
the Messaging Server and by how much the tenants folder structure can have changed before the   
entire folder structure must be retrieved again by the AgentWS.   
To configure MS folder change notifications:   
1.   
Navigate to C:\Program Files (x86)\CosmoCom\Server Components\AgentProxy.   
2.   
Open Web.config in a text editor.   
3.   
In <appSettings>, find:   
<add key="Messaging Server Get Events Interval" value ="20000" />   
<add key="Folder Structure Retrieval Threshold %" value ="20" />   
4.   
In Messaging Server Get Events Interval, set the interval in milliseconds at which the   
AgentWS polls for folders structure change notifications from the Messaging Server.   
5.   
In Folder Structure Retrieval Threshold (%), set the maximum number of folders as   
a percentage of total folders in a tenant’s folder structure that can change before the entire   
folder structure must be retrieved again by the AgentWS.   
6.   
Save the file.

CCSP System Administrator Guide - 7.3 and 7.4   
   
52   
Configure the chat box alignment   
To configure the alignment of the chat box:   
1.   
On the CCSP UI server, navigate to C:\Program Files (x86)\CosmoCom\Server   
Components\CosmoAgent\css.   
2.   
Open chat.css in a text editor.   
3.   
Scroll to the .ChatBoxContent section and add direction:rtl; as follows:   
.ChatBoxContent   
{   
 ...   
 overflow-x: hidden;   
 direction:rtl;   
}   
4.   
Scroll to the .ChatBoxTime section and change float:right; to float:left; as follows:   
.ChatBoxTime   
{   
 font: 8px italic;   
 float:left;   
 ...   
}   
5.   
Scroll to the .ChatBoxMessage section and change float:left; to float:right; as follows:   
.ChatBoxMessage   
{   
 float:right;   
 word-wrap: break-word;   
}   
6.   
Save the file.   
Enable multiple interaction mode for callbacks   
Note   
The callback interaction mode must be the same across all AWS instances used by a particular   
tenant.   
   
By default, single-interaction mode is enabled for callbacks.   
To enable multiple interaction mode for callbacks:   
1.   
Navigate to C:\Program Files (x86)\CosmoCom\Server Components\AgentProxy.   
2.   
Open Web.config in a text editor.   
3.   
Find add key = "Multiinteraction Callback" value = "0"   
4.   
Change to value = "1".

CCSP System Administrator Guide - 7.3 and 7.4   
   
53   
Set the inactivity timer   
If a CCSP UI client is shutdown ungracefully, the Agent Proxy Server detects there are no more   
requests for events and logs off the client. The default value of the Inactivity Timeout key is 60000   
ms.   
To set the inactivity timer:   
1.   
Navigate to C:\Program Files (x86)\CosmoCom\Server Components\AgentProxy.   
2.   
Open Web.config in a text editor.   
3. Find add key = "Inactivity Timeout" value = "60000"   
4.   
Replace 60000 with your value.   
Change the email draft auto-save rate   
CCSP UI automatically saves draft email messages every 30 seconds.   
To change the auto-save rate:   
1.   
Navigate to C:\Program Files (x86)\CosmoCom\Server Components\TouchPoint\Ajax.   
2.   
Open Config.js in a text editor.   
3.   
Search for var SaveDraftInterval = 30000;   
4.   
Replace 30000 with your value.   
Enable start in toolbar mode   
In toolbar mode, the gadgetory toolbar uses as little space as possible on the user’s screen.   
By default, start in toolbar mode is not enabled. You enable start in toolbar mode and set the   
toolbar window size by editing ClientServicesConfig.js.   
To enable start in toolbar mode:   
1.   
Navigate to \Program Files (x86)\CosmoCom\Server Components\TouchPoint\ClientServices.   
2.   
Open ClientServicesConfig.js in a text editor.   
3.   
Search for var toolbarMode and change the value to true.   
4.   
Search for var toolbarWinSize and change the value to 0.

CCSP System Administrator Guide - 7.3 and 7.4   
   
54   
Configure the wallboard   
Note   
The new Mini Wallboard is not functional with the legacy wallboard API.   
   
The new Mini Wallboard uses the new Web Agent Ajax Statistics API. The APS address is used for   
this, so in Config.js, the wallBoardAddress value is set as an empty string.   
The legacy wallboard uses the legacy Web Agent Ajax Statistics API. To use the legacy wallboard API   
instead of the new Mini Wallboard, in Config.js, set the wallBoardAddress value to the url of the   
wallboard.   
To configure the legacy wallboard API:   
1.   
Navigate to \Program Files (x86)\CosmoCom\Server Components\TouchPoint\Ajax.   
2.   
Open Config.js in a text editor.   
3.   
Search for var wallBoardAddress.   
4.   
Change the value to your wallboard address, for example:   
var wallBoardAddress =   
"http://DNCCSP12.DN12.LOC/CCWALLBOARD/wallboard.aspx";   
5.   
Save the file.   
Connect to external Web or desktop applications   
You can integrate CCSP UI with a third-party application such as a Customer Relations Management   
application. For details, refer to the CCSP UI Connector User Guide.

CCSP System Administrator Guide - 7.3 and 7.4   
55   
6: Configure Dashboard   
Configure the Dashboard service   
1.   
Navigate to \Program Files (x86)\CosmoCom\Server Components\DashboardService\DashboardWS\ and edit the parameters in   
appsettings.json.   
2.   
After changing the parameters, you must restart the application pool DashboardWSAppPool.   
Name   
Description   
Type   
Default   
CCSPServer   
URL of CCSP Server. It is the   
protocol and the IP address of the   
machine where RTR DLL Web   
Application is installed.   
String   
   
CleanProfileStatisticsInterval   
Interval to check whether it is   
necessary to remove a profile from   
the list due to inactivity   
TimeSpan   
00:05:00   
DataProviders   
List of data providers   
Array   
["DashboardExtension.RtrWebDataProvi  
der"]   
FromEmailAddress   
Valid email address to use as the   
from address when thresholds are   
met and configured to send email   
notifications.   
If left empty, the email address of   
the user that saved the dashboard   
is used.   
String   
   
HttpTrace   
Enable or disable http logs   
Boolean   
false

CCSP System Administrator Guide - 7.3 and 7.4   
56   
Name   
Description   
Type   
Default   
MaxUnauthorizedAccessCount   
BeforeSignout   
Maximum number of unauthorized   
access attempts to the RTR Web   
DLL before signout   
Int   
5   
RtrWebDataProviderHistoryUpdate   
Interval   
Refresh rate of historical statistics   
TimeSpan   
00:05:00   
RtrWebDataProviderUpdateInterval   
Refresh rate of statistics   
TimeSpan   
00:00:05   
RtrWebDllUrl   
VirtualDir of RTR   
String   
rtrdll/rtrweb.dll   
SMTPServer   
SMTP server for to send email   
alerts. Type the SMTP server name   
or IP address.   
String   
localhost   
SqlConnectionString   
Connection string for the CCSP   
database   
String   
   
SyslogLevel   
The level of logging done by the   
Dashboard Service in SysLog. The   
applicable values are the standard   
CCSP log level values.   
String   
Warning   
WorkspaceProvider   
   
String   
DashboardExtension.RtrWebWorkspaceP  
rovider

CCSP System Administrator Guide - 7.3 and 7.4   
57   
Configure the Dashboard application   
1.   
Navigate to \Program Files (x86)\CosmoCom\Server Components\DashboardService\Dashboard\assets\config\ and edit the parameters in   
app-settings.json.   
2.   
After changing the parameters, you must restart the application pool DashboardAppPool.   
Name   
Description   
Type   
Default   
apiGraphQLUrl   
HTTPS address of the service   
String   
https://localhost:5001/graphql   
defaultLanguage   
Default language of the Dashboard application. The   
possible values are:   
•   
en – English   
•   
es – Spanish   
•   
de – German   
•   
he – Hebrew   
•   
ja – Japanese   
•   
ru – Russian   
   
If left empty, English (en) is used.   
String   
en   
gadgetsLimit   
Maximum number of gadgets per report   
Int   
10   
logLevel   
Logging level. The possible values are:   
•   
TRACE   
•   
DEBUG   
•   
INFO   
•   
LOG   
•   
WARN   
•   
ERROR   
•   
FATAL   
•   
OFF   
String   
OFF

CCSP System Administrator Guide - 7.3 and 7.4   
58   
Name   
Description   
Type   
Default   
maxReconnectionAttempts   
Maximum reconnection attempts to service. When the   
number of attempts is reached, the user is logged out   
from the application.   
Int   
10   
popupVisibilityTimerInMins   
After the Dashboard is minimized, how many minutes to   
wait until displaying the reactivate message.   
Int   
240   
reportsLimit   
Maximum number of reports per view   
Int   
6   
savePassword   
Save encrypted password in the client for the current   
session.   
Boolean   
true   
viewsLimit   
Maximum number of views per workspace   
Int   
10   
Configure the Dashboard application language   
1.   
Stop the application pool DashboardAppPool.   
2.   
Navigate to \Program Files (x86)\CosmoCom\Server Components\DashboardService\Dashboard\assets\config\app-settings.json.   
3.   
Search for parameter defaultLanguage and change the value as required. For details, see the defaultLanguage description.   
4.   
Save the file.   
5.   
Restart the application pool DashboardAppPool.   
6.   
Clear the cache in your browser and reload the Dashboard.

CCSP System Administrator Guide - 7.3 and 7.4   
59   
7: Configure Replay   
Note   
•   
To sync between screen recording and audio recording, screen recording calculates the client   
offset from the CCSP UI server in seconds. Hence the CCSP UI server and Replay server must   
have the same time zone.   
•   
The ffmpeg tool is extracted by screen recording before recording starts. Because agents   
don't have admin privileges, the ffmpeg process is extracted to the temp folder under the   
drive root. For example, if the user's temp folder is L:\AppData\User\Temp, then the drive is L:   
and the folder from which the ffmpeg executable runs is L:\Temp.   
Create the recordings download virtual directory   
The recordings download folder is the folder in which Replay stores recordings when it retrieves   
them from the database prior to playback. The default location for this folder is C:\Program Files   
(x86)\CosmoCom\Server Components\Replay\Recordings.   
You can configure an alternative location for the recordings download folder by defining it as a   
virtual directory in IIS, provided that the physical folder is accessible to the network service in the   
same way as the default recordings download folder   
To create the recordings download virtual directory:   
1.   
In IIS Manager, right-click Replay and then click Add Virtual Directory. Create a virtual   
directory, for example, ClientRecordings, linked to a network-accessible physical folder, for   
example, C:\Windows\Temp\LocalRecordings.   
2.   
Set the folder permissions to include Read and List access for ALL APPLICATION PACKAGES   
and IUSR:   
a.   
Right-click the new folder and then click Edit Permissions.   
b.   
On the Security tab, click Edit. Click Add, and then click Locations. Select your local   
machine.   
c.   
Under Enter the object names to select, type All and click OK. Ensure that ALL   
APPLICATION PACKAGES has Read and List permissions allowed.   
d.   
Click Add. Under Enter the object names to select, type IUSR and click OK. Ensure that   
IUSR has Read and List permissions allowed.   
3.   
Edit the RecordingsDownloadFolder value in Web.config. See below.

CCSP System Administrator Guide - 7.3 and 7.4   
60   
Configure the recordings download folder   
To configure the recordings download folder path and maximum number of files:   
1.   
Navigate to Server Components\Replay and open Web.config in a text editor.   
2.   
If you want to use an alternative location for the recordings download folder, search for <add   
key="RecordingsDownloadFolder" and change the value from an empty string to the   
name of the virtual directory created above, for example:   
<add key="RecordingsDownloadFolder" value "ClientRecordings" />   
3.   
By default, the maximum number of processed files that are allowed in the recordings   
download folder before the oldest file is deleted to make way for a new one is 50. To change   
this number, search for <add key="MaxCachedFiles" and change the value.   
Control access to recordings virtual directory   
Only the Replay application (ReplayAppPool) and the server administrator should have access to   
the Replay recordings virtual directory.   
To control access to the Replay recordings virtual directory:   
1.   
Open Windows Explorer.   
2.   
Select the recordings directory under \Program Files (x86)\CosmoCom\Server   
Components\Replay.   
3.   
Right-click the file and then click Properties.   
4.   
Click the Security tab.   
5.   
Click Edit and then click Add.   
6.   
Click Locations and then select the local machine. (Do not select the Windows domain, if the   
server belongs to one.)   
7.   
In Enter the object names to select, type IIS AppPool\ReplayAppPool.   
   
8.   
Click Check Names.

CCSP System Administrator Guide - 7.3 and 7.4   
61   
   
9.   
Click OK.   
Enable audio visualization   
Note   
Adding the waveform display may significantly increase the processing time before audio playback   
starts, if the system resources are insufficient.   
   
To enable audio visualization:   
1.   
Navigate to Server Components\Replay and open Web.config in a text editor.   
2.   
If you want audio recordings to be played with a sound waveform display, search for   
<add key="AudioVisualisationEnabled" and change the value to "true".

CCSP System Administrator Guide - 7.3 and 7.4   
62   
Streamed file names   
The format of the names of the cached files in the recordings download folder is the name of the   
file to stream to the client plus:   
•   
CallID (multi-interaction calls)   
•   
Segment number   
•   
Outro duration (screen only)   
   
The filename formats are:   
•   
Single segment audio files:   
<original\_filename>.<segment\_num>.<segment\_offset\_from\_image\_in\_s>.specific.mp3   
•   
Combined segments audio files:   
<call\_ID>.audio.full.mp3   
•   
Video files:   
<original\_filename>.<callID>.<segment\_num>.<outro\_length\_in\_s>.mp4   
•   
Chat files:   
original\_filename.txt   
•   
Email files:   
original\_filename.eml   
Archiver Playback   
Archiver Playback is a feature that enables you to search and view recordings from archived tenant   
recording databases in addition to or instead of the tenant’s live production recording database.   
The PS Recording Archiver application creates and manages the archive recording databases. For   
information about this application, refer to the Recording Archiver User Guide.   
The PS-supplied ConfigPortal database provides access to the archiver configuration and archived   
recording databases. You must set up the ConfigPortal according to the instructions in the   
ConfigPortal Deployment Guide (version 3.0.0.1).   
To enable the Archiver Playback feature, you need to edit the Replay application configuration.   
Configuration modes   
The following modes are supported:   
•   
Off — Replay has access to the rest of the CCSP system and only the live recording databases   
for the tenants.   
•   
Online — Replay has access to the rest of the CCSP system and the live recording databases   
for the tenants. Replay accesses the archive databases through a connection to the   
ConfigPortal database.

CCSP System Administrator Guide - 7.3 and 7.4   
63   
•   
Offline — Replay has no access to the rest of the CCSP system apart from the CCSP Auth   
Server for user login. The CCSP Auth Server does not have access to the CCSP system and   
authenticates users against the local machine. The live recording databases are not available.   
Replay accesses the archive databases using connection strings in its web.config file.   
•   
On-premise (Remote Offline) — Replay has no access to the rest of the CCSP system apart   
from the CCSP Auth Server for user login. The CCSP Auth Server does not have access to the   
CCSP system and authenticates users against the local machine. The live recording databases   
are not available. Replay accesses the archive databases through a connection to the Archiving   
Master Offline database, which is a local clone of the ConfigPortal database. A Windows   
service on the local server is responsible for merging new uploaded archive databases to the   
server’s aggregated database storage.   
Configure Replay for Online mode   
For Archiver Playback feature to be available, changes to the Replay application configuration are   
required.   
1.   
Navigate to Server Components\Replay and open web.config in a text editor.   
2.   
Search for the connection string name ConfigPortalDB.   
3.   
Replace the %ConfigPortalDBConnString% placeholder with the connection string for the   
ConfigPortal database.   
4.   
Search for the ArchiverMode key.   
5.   
Change the value to "Online".   
6.   
Save the file.   
Configure Replay for Offline mode   
In this mode, only local users can be authenticated to use Replay. Details of the authorized users   
must be added to the CCSP Auth Server.   
Auth Server configuration   
1.   
Navigate to Server Components\AuthServer and open Web.config in a text editor.   
2.   
Search for the <offlineModeConfig> section.   
3.   
Add a new element as below:   
<authorizedUsers>   
</authorizedUsers>   
4.   
Within this element add an entry for each user that is authorized to use Replay in this mode.   
The following information must be provided:   
o   
name — the user login name without any domain suffix   
o   
tenantId — the numeric configuration database identifier for the tenant   
o   
tenantDisplayName — the name of the tenant as it appears in the Display name field of   
the tenant properties

CCSP System Administrator Guide - 7.3 and 7.4   
64   
For example   
<add name="t1User" tenantId="1" tenantDisplayName="t1"/>   
5.   
Save the file.   
Replay Server configuration   
1.   
Navigate to Server Components\Replay and open Web.config in a text editor.   
2.   
Search for the ArchiverMode key.   
3.   
Change the value to "Offline".   
4.   
Search for the <offlineModeConfig> section.   
5.   
Add a new element as below:   
<recDbs>   
</recDbs>   
6.   
Within this element add an entry for each database that can be accessed in this mode. The   
following information must be provided:   
o   
name — the name of the database as it should appear in Replay   
o   
tenantId — the numeric configuration database identifier for the tenant   
o   
connectionString — the full connection string for the database   
   
For example   
<recDbs>   
 <add tenantId="1" connectionString="data source =   
MyServer\MyInstance; Initial Catalog = t1Archive\_t1\_20161101; Integrated   
Security = SSPI;"   
 name="t1Archive\_t1\_20161101"/>   
 <add tenantId="1" connectionString="data source =   
10.57.34.72\SQL2016; Initial Catalog = t1Archive\_t1\_20190201; Integrated   
Security = SSPI;"   
 name="t1Archive\_t1\_20190201"/>   
 </recDbs>   
7.   
Save the file.

CCSP System Administrator Guide - 7.3 and 7.4   
65   
Configure Replay for On-premise mode   
In this mode, only local users can be authenticated to use Replay. Details of the authorized users   
must be added to the CCSP Auth Server.   
Auth Server configuration   
Follow the instructions for the Auth Server configuration in Offline mode.   
Replay Server configuration   
1.   
Navigate to Server Components\Replay and open Web.config in a text editor.   
2.   
Search for the connection string name ConfigPortalDB.   
3.   
Replace the %ConfigPortalDBConnString% placeholder with the connection string for the   
Archiving Master Offline database.   
4.   
Search for the ArchiverMode key.   
5.   
Change the value to "OnPremise”.   
6.   
Save the file.

CCSP System Administrator Guide - 7.3 and 7.4   
66   
8: Configure IP phones   
Configure IP phones describes the recommended method to configure IP phones for use with CCSP   
in typical CCSP environments.   
Tenants may want to use IP phones directly controlled by CCSP and used as standalone phones,   
and/or as peripheral devices with CCSP UI.   
You should create an FTP folder for each tenant that will contain a tenant-wide configuration file.   
Most deployments need to secure the location of the configuration file, since it may be accessible   
from the public Internet. Therefore, we recommend using FTP.   
Note   
The IP phone does not support provisioning of authentication credentials for HTTPS.   
Create an FTP folder for each tenant   
For each tenant that will use IP phones:   
1.   
Create an FTP folder.   
2.   
Create a Windows account with access rights to the FTP folder.   
3.   
Send the tenant's Login ID and password to the tenant administrator.   
4.   
Use the folder to pass the IP phone configuration files that the tenant needs.   
Create and assigning tenant extensions   
To create and assign tenant extensions:   
1.   
Ask each tenant administrator for a list of the extensions they will be using. Request the list in   
CSV format.   
2.   
Add the extension list to the DNIS pool.   
3.   
Assign the DNIS pool numbers to the appropriate tenant.   
Individual IP phone configuration files   
Each individual IP phone needs unique configuration files. You can provide these files for the   
tenants or the tenant administrators can create them themselves. For more information, refer to   
the CCSP Tenant Administrator Guide.

CCSP System Administrator Guide - 7.3 and 7.4   
67   
9: Configure languages   
In CCSP Admin, CCSP UI, and Dashboard, all the supported languages are part of the basic   
installation and use the language selected in Internet Options on the agent's computer.   
The other components of CCSP are only installed with English support. For further language   
customization information, contact Enghouse Interactive Customer Support.   
The CCSP-supplied files that support foreign languages are named according to specific   
conventions and usually include a language ID within the name itself. This is a numeric code that   
the Windows operating system recognizes and that is unique to each language.   
For example, the language ID for US English is 1033, and the language ID for Standard Italian is   
1040. Therefore the names of the Agent Phrase files for each of these languages are:   
Agent\_Phrases\_1033.txt, and Agent\_Phrases\_1040.txt, respectively.   
In the following topics, the language ID is replaced by <LocaleID>.   
Configure the back end language   
The backend language components are configured as follows.   
Chat Proxy Service   
The ACD server component ChatProxy.exe uses a ChatProxy\_Phrases\_<LocaleD>.txt file located in   
\Program Files (x86)\CosmoCom\ServerComponent\Resources\   
The default file is US English, which is ChatProxy\_Phrases\_1033.txt.   
You can add a new Chat Proxy Service resource file to the Chat Proxy Resources folder. The resource   
file name must be ChatProxy\_Phrases\_<LocaleID>.txt.   
ChatExtension.dll   
The Web Server component ChatExtension.DLL uses a directory structure of \Inet-  
pub\wwwroot\CosmoCall\<LocaleID>\<TenantName>Phrases.txt where:   
•   
<LocaleID> is the calling user language. The default is English which is 1033.   
•   
<TenantName> is the name of the tenant. The default is Default.   
You can create an IIS %Inetpub%\wwwroot\CosmoCall\<LocaleID>\Default folder and add an   
appropriate Phrases.txt file, and translated Chat Extension DLL html and templates pages. The   
tenant must provide the translated Chat Extension DLL html and templates pages.   
To present different appearances and languages per tenant, you can create a   
%Inetpub%\wwwroot\CosmoCall\<LocaleID>\<TenantName> folder.

CCSP System Administrator Guide - 7.3 and 7.4   
68   
Dashboard (legacy)   
\Program Files (x86)\Cosmocom\Server Components\CosmoDashboard\Source\Localization contains   
Strings.<Locale>.xaml files for each language, for example, Strings.en-us.xaml. When these files are   
published, they appear in \Program Files (x86)\Cosmocom\Server   
Components\CosmoDashboard\Published\Localization.   
The Internet Options on the computer running Dashboard must also be configured for the correct   
language.   
Dashboard Service   
\Program Files (x86)\Cosmocom\Server Components\DashboardService\Dashboard\assets\i18n and   
\Program Files (x86)\Cosmocom\Server Components\DashboardService\DashboardWS\Assets contain   
<Locale>.json files for each language, for example, en.json.   
The language can be chosen at login time.   
SQL Server   
\Program Files (x86)\CosmoCom\Server Components\Setup contains adminDictionary<LocaleID>.SQL   
files for each language which, when run, add the appropriate language files to the database.   
To add language files to the database:   
1.   
Log on to the computer that contains the SQL Server configuration database.   
2.   
Start Microsoft SQL Query Analyzer.   
3.   
Open the languages.sql file in the folder for the language you wish to add to the Administrator   
interface.   
4.   
Press F5 to run the query.   
5.   
Open the appropriate .SQL file for the language you wish to add to the Administrator   
interface.   
6.   
Press F5 to run the query.   
7.   
Close the Query Analyzer.   
   
The Internet Options on the computer running CCSP Admin must also be configured for the correct   
language.

CCSP System Administrator Guide - 7.3 and 7.4   
69   
Enable different Web chat languages for tenants   
You can change the messages on a tenant-by-tenant basis. Before connecting to the Cha-  
tExtension.DLL, the website determines the tenant and sets the TenantName   
Calling\_User\_Language parameters accordingly.   
To enable different web chat languages for different tenants:   
1.   
Define a new language number (locale ID) for each tenant. This must not correspond to any of   
the known numbers used by Windows.   
2.   
Edit the ChatProxy\_Phrases\_<LocaleID>.txt for each tenant, and place this file in the \Program   
Files (x86)\CosmoCom\ServerComponent\Resources directory.   
3.   
Edit the Phrases.txt for each tenant, and create a new directory structure   
\Inetpub\wwwroot\CosmoCall\<LocaleID>\<TenantName> where <LocaleID> is the new   
language number for the tenant and <TenantName> is the actual tenant name.   
   
For example, to enable different languages for two tenants, RostovCo and YMS\_SystemsCo:   
1.   
Assign RostovCo the locale ID of 8000 and assign YMS\_SystemsCo the locale ID of 8001.   
2.   
For ChatProxy.exe, edit the ChatProxy\_Phrases\_<LocaleID>.txt for each tenant:   
o   
\Program Files\CosmoCom\ServerComponent\Resources\ChatProxy\_Phrases\_8000.txt   
o   
\Program Files\CosmoCom\ServerComponent\Resources\ChatProxy\_Phrases\_8001.txt   
3.   
For the ChatExtention.DLL, make two copies of Phrases.txt and edit one for each tenant.   
4.   
Create the following directories and place the associated Phrases.txt files in each directory:   
o   
\Inetpub\wwwroot\CosmoCall\8000\RostovCo\   
o   
\Inetpub\wwwroot\CosmoCall\8001\YMS\_SystemsCo\   
   
To change chat colors, edit the settings of the following registry subkey:   
HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Cosmocom\ChatExtension\Param-  
eters\TenantName\ChatSystemMsgColor   
See also Enabling Web chat in a multitenant environment.   
Configure the callback dialog language   
To configure the language of the Schedule a Callback dialog:   
5.   
Navigate to the C:\Program Files (x86)\CosmoCom\Server   
Components\CallBackScripts\App\_GlobalResources folder.   
6.   
Make a copy of Resource.RESX and rename as Resource.<LocaleID>.RESX, for example,   
Resource.de.RESX.   
7.   
Open Resource.<LocaleID>.RESX in Visual Studio.   
8.   
Replace the default values with the translations.   
9.   
Save the file.

CCSP System Administrator Guide - 7.3 and 7.4   
70   
10: Configure the VCS   
When the VCS is properly installed, it must be configured using both CCSP Admin and by changing   
and adding some H.323 stack parameters. This configuration depends on whether you are using a   
gatekeeper environment.   
Note   
Although you can use the VCS without a gatekeeper, we recommend that you install a   
gatekeeper or multiple gatekeepers as part of your contact center.   
The gatekeeper offers centralized management of gateways and registered endpoints, is   
software-only, and is usually provided by the gateway manufacturer.   
   
To configure the VCS in CCSP Admin:   
1.   
Enter the E.164 numbers to be used by the VCS for routing (gatekeeper only).   
2.   
Create E.164 groups that contain at least one E.164 number (gatekeeper only).   
3.   
Create VCS groups that contain E.164 groups (gatekeeper only).   
4.   
Configure the VCS parameters.   
Features and capabilities   
The VCS is a software-only solution that includes the following features and capabilities.   
Gatekeeper registration   
The VCS includes the ability to register with an external gatekeeper via standard H.323 methods:   
•   
The VCS advertises the E.164 address for each of the supported dialed numbers of the contact   
center. E.164 addresses for individual agents are not supported.   
•   
The VCS can receive CLID/DNIS digits and the originally dialed number from the gateway and   
gatekeeper, subject to the gateway and gatekeeper configuration.   
•   
The RADVision stack configuration file and administrator data determines whether the system   
will register with a gatekeeper, manual vs automatic discovery, terminal type, aliases, and time   
to live values.   
Inbound/outbound dialing   
The VCS can direct voice calls to CCSP UI and provide outbound dialing capability for the agents.   
The outdialing is based on E.164 numbering through the gatekeeper and may dial any valid phone   
numbers through the gateway.   
VCS terminated and originated calls support voice media.

CCSP System Administrator Guide - 7.3 and 7.4   
71   
IVR capabilities and limitations   
The VCS has the following IVR capabilities and limitations:   
•   
The IVR provides the capability to record caller messages and play pre-recorded audio   
prompts to a caller that is being processed in the VCS.   
•   
The VCS supports delivery and receipt of DTMF, transported through the network via H.245   
signaling.   
•   
The VCS .wav files must be 16-bit, 8Khz, Mono. At startup, the Interactive Media Service (IMS)   
reads the database and creates a local copy of all the media files. If the local version of a   
particular media file exists, and it is determined to be the most current version, the file will not   
need to be recopied from the database.   
•   
Two timeouts are associated with recordings that are specified by the VoIP Call Control   
Module issuing a request. They are the total record timeout, which specifies the maximum   
amount of media that can be recorded, and the no voice data time. The no voice data timeout   
occurs when either RTP packets stop being transmitted to the IMS or the RTP packets contain   
silence. A silence detection algorithm is used to detect silence.   
•   
The VCS supports scripting languages for use in IVR applications. When the script requires that   
audio be spoken or that audio be recorded, the VCS uses the IMS to perform the task.   
Monitoring capabilities   
The VCS supports supervisor monitoring capabilities, including silent and whisper monitoring of   
calls.   
Codecs   
The VCS fully supports G.711 aLaw and muLaw, and G.729. G.723.1 is not supported with PJSIP. See   
Install G.729 codecs.   
Multiple VCSs   
The VCS architecture allows for each VCS server to use any number of RTP Relay servers.   
Load balancing   
Load balancing is achieved by using the Microsoft Network Load Balancer (NLB) as a front end to a   
number of VCSs, each registering the same E.164 aliases. The NLB is a distributed system, running   
on each of the load balanced systems.

CCSP System Administrator Guide - 7.3 and 7.4   
72   
Multiple gateway service policy   
When a VCS is connected to multiple gateways:   
•   
The VCS tries the gateways in round robin order.   
•   
When a gateway fails, the VCS tries the next one. The VCS keeps trying until the list of   
gateways is exhausted.   
System administration and configuration   
System administrators use CCSP Admin and reporting tools to administer the VCS. CCSP Admin   
treats the VCS as a type of connection server and lists the VCS under Server Profiles.   
The system supports the configuration of stack parameters. Two configuration files are used to   
configure the operation of the H.323 protocol stack, created during VCS installation. They can be   
manually edited on the local VCS, if required. See Set the RTP frame packet size.   
System administrators can view VCS performance statistics through Historical Reports.   
Components   
The VCS has three main components:   
•   
VoIP Call Control — routes H.323 messages using H.323 stack to the agent and IMS. The VCS   
implements the H.323 version 2 standard. H.225 RAS signaling, H.245 Control signaling.   
•   
Interactive Media Server (IMS) — plays prompts and records voice messages.   
•   
RTP Relay Module — receives RTP and RTCP packets and routes the data into and out of the   
contact center under control of the VoIP Call Control component.   
Signaling features   
Gatekeeper discovery   
A gatekeeper discovery message (GRQ) sent to the gatekeeper address results in a response (GCF)   
indicating a transient gatekeeper address that has been enabled for the anticipated registration   
request. The VCS sends GRQs to a configurable port and IP address on the external side of the   
gatekeeper. The address may or may not be known (multicast).   
Registration   
Registration requests are sent to the address and port indicated in the GCF. The address/port   
provided by the discovery mechanism is only valid for the duration of the subsequent registration.   
During registration the VCS advertises the E.164 address for each of the supported dialed numbers   
of the contact center, but not for the individual agents connected to the system.   
The registration needs to provide a keep alive interval that the VCS expects for re-registration   
during the initial registration process.

CCSP System Administrator Guide - 7.3 and 7.4   
73   
Keep Alive interval   
The VCS has a configured interval for Keep Alive, and re-registers at the smaller specified interval.   
The Keep Alive parameter is sent in a registration request from the VCS to its gatekeeper, to keep   
the registration active for a given amount of time. This is useful if the registration is about to expire   
based on the endpoint's Time to Live.   
Terminate connection   
The ports activated by discovery close after a timeout if the discovery is not followed by   
registration.   
H.225 call setup   
The H.225 call setup is used for call setup, alerting, tear down, and other connection oriented   
signaling.   
H.245 control channel   
The H.245 control channel is responsible for control messages governing operation of the H.323   
terminal, including capability exchanges, commands, and indications. H.245 tunneling is supported   
by default.   
IVR handling   
DTMF digits received from the PSTN are transported through the VoIP network and delivered to the   
H.323 endpoint via the H.245 message UserInputIndication, if this is supported by the gateway   
(DTMF relay).   
SIP signaling   
The VCS supports SIP signaling over UDP and TCP transport protocols.

CCSP System Administrator Guide - 7.3 and 7.4   
74   
Set the RTP frame packet size   
VCS installation creates two separate H.323 stack configuration files that are located in the   
C:\Program Files (x86)\CosmoCom\Server Components directory.   
These files are:   
•   
config.int — affects interactions between the VCS and IMS and any agents logged on to the   
contact center.   
•   
config.ext — affects external communications coming into the VCS and registration with the   
gatekeeper.   
   
You must edit the stack files so that the RTP frame packet size matches the frame size of the   
gateways you use.   
Note   
All entries are case-sensitive.   
   
To set the RTP frame packet size:   
1.   
In the config.int file, find all of the lines in the file that reference g711 and look like the   
following:   
g711Ulaw64k = nn   
g711Alaw64k = nn   
2.   
Replace the nn with the appropriate value for your gateway.   
3.   
Save the config.int file, overwriting the older version.   
4.   
In the config.ext file, find all of the lines in the file that reference g711. There should be four of   
them and they should look like the following:   
g711Ulaw64k = nn   
g711Alaw64k = nn   
5.   
Replace the nn with the appropriate value for your gateway.   
6.   
Save the config.ext file, overwriting the older version.   
   
For more information on the H.323 stack and its entries and parameters, see Stack configuration.

CCSP System Administrator Guide - 7.3 and 7.4   
75   
Debug the VCS   
The events associated with the components of the VCS (IMS, VoIP Call Control and RTP Relay) are   
logged in C:\Program Files (x86)\Common Files\CosmoCom\CosmoCall Universe\syslogd\vcs.txt.   
To turn on verbose logging of these events:   
1.   
Start the Registry editor and find the appropriate key for each of the following components:   
o   
[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\CCUInteractiveMediaServer  
\Parameters\]   
o   
[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\CCURTPRelay\Parameters\]   
o   
[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\CCUVoIPCallControl\Param  
eters\]   
2.   
Double-click the V entry and set the Value data to 255. The value of 255 sets logging to its most   
verbose level. Use a less verbose level to create smaller log files.   
Limiting the log file size   
There is no current way to limit the size of the vcs.txt file. It will continue to add lines of information   
as long as the registry is set as described above.   
However, you can create a batch file, which should be run daily to delete the vcs.txt file or any of   
the logging text files in the ..\syslogd directory.   
This file must:   
1.   
Stop the CosmoCall Universe Syslog service.   
2.   
Delete the .txt files.   
3.   
Restart the CosmoCall Universe Syslog service.

CCSP System Administrator Guide - 7.3 and 7.4   
76   
Configure VCS load balancing   
To load balance the VCS, you need at least two clustered nodes.   
Each VCS must have at least three IP addresses bound to it:   
•   
Internal stack IP   
•   
External stack IP (dedicated host IP)   
•   
NLB IP (NLB shared primary host IP)   
   
When Network Load Balancing (NLB) support is enabled, the external stack is used for outbound   
calls only. Inbound calls are sent to the shared NLB address, therefore the gateway must be   
configured accordingly.   
To configure VCS load balancing:   
1.   
In CCSP Admin, under System information, create a VCS group, which can have any name.   
2.   
Configure the VCS group as follows:   
o   
NLB IP Address — set to the primary host IP address configured within the properties of   
NLB. This is the shared NLB address.   
o   
NLB Port — set to 1720.   
o   
Under System information > Services, configure the internal stack, and external stack IP   
addresses for each VCS as you would normally.   
3.   
Select the Under NLB check box and add the VCS group you just created.   
4.   
Restart each VCS.   
Configure the VCS for Teams   
For CCSP UI to be able to dial a Teams contact using its SIP URL (for example,   
someone@company.com), the VCS must be able to resolve the domain portion of the URL to a   
defined host or gateway. The routing between the VCS and the Teams server is specific to your   
system infrastructure and will normally be through your vendor-provided SIP or PSTN gateway.   
See Teams integration.

CCSP System Administrator Guide - 7.3 and 7.4   
77   
Stack configuration   
To minimize the obstacles to interoperability with other manufacturer's VoIP devices, the   
RADVision H.323 Protocol Stack has been incorporated into the product. The protocol stack from   
RADVision is used for the following:   
•   
H.245 for call control   
•   
Q.931 for call signaling   
•   
RAS (Registration, Admissions, and Status) signaling functions   
Note   
CCSP configuration settings are stored in the config.ext and config.int files.   
These parameters are the set of values that determine the default settings under which an   
endpoint functions throughout an H.323 conference.   
The four types of stack configuration parameters are:   
•   
System   
•   
RAS   
•   
Q.931   
•   
H.245   
Each type is arranged in a separate ASN.1 tree. Each tree follows the description of the ASN.1   
structure as specified in the appropriate H.323 Recommendation. For example, the H.245   
parameters tree reflects the ASN.1 structure in the H.245 Recommendation. As soon as the   
endpoint is activated, the stack reads the stack configuration parameters and acts accordingly.   
From this stage on, the parameters are a part of the stack. Although most of the parameters   
remain constant throughout the H.323 conference, some parameters can be changed dynamically   
during runtime using the stack configuration API.   
Within these categories the manual parameters affect the way your application will use the API. The   
manual parameters include the following and are discussed below in the relevant sections:   
•   
System.manualStart.   
•   
RAS.manualRas.   
•   
RAS.AllowCallsWhenNonReg.   
•   
RAS.manualDiscovery.   
•   
RAS.manualRegistration.   
•   
Q931.manualAccept.   
•   
Q931.autoAnswer.   
•   
H.245.MasterSlaveConf.manualOperation.   
•   
H.245.MasterSlaveConf.manualResponse.   
•   
H.245.CapabilitiesConf.manualOperation.   
   
Note   
We recommend that new users of the RADVision H.323 Protocol Stack disable the manual keys   
and let the stack make the necessary choices.

CCSP System Administrator Guide - 7.3 and 7.4   
78   
System parameters   
System parameters are general configuration parameters used during initialization, for example, to   
determine memory allocation and define the number of calls and channels per call.   
You cannot change system parameters after cmInitialize().   
Syntax   
PDLSystemConfiguration ::= SEQUENCE   
{   
manualStartNULLOPTIONAL,   
pdlNameOCTETSTRINGOPTIONAL,   
cidAssociateNULLOPTIONAL,   
delimiterOCTETSTRING(SIZE(1))OPTIONAL,   
allocationsSEQUENCE   
{   
vtPoolSizeINTEGER(0..2147483647)OPTIONAL,   
vtNodeCountINTEGER(0..2147483647)OPTIONAL,   
channelsINTEGER(0..2147483647)OPTIONAL,   
chanDescsINTEGER(0..2147483647)OPTIONAL,   
messagesINTEGER(0..2147483647)OPTIONAL,   
nameChansINTEGER(0..2147483647)OPTIONAL,   
tpktChansINTEGER(0..2147483647)OPTIONAL,   
udpChansINTEGER(0..2147483647)OPTIONAL,   
protocolsINTEGER(0..2147483647)OPTIONAL,   
maxProcsINTEGER(0..2147483647)OPTIONAL,   
maxBuffSizeINTEGER(0..65536)OPTIONAL,   
}OPTIONAL,   
maxCallsINTEGER(0..2147483647)OPTIONAL,   
maxChannelsINTEGER(0..2147483647)OPTIONAL,   
maxRasInTransactionINTEGER(0..2147483647)OPTIONAL,   
maxRasOutTransactionINTEGER(0..2147483647)OPTIONAL,   
}   
Parameters   
The system parameters and structure are as follows.   
manualStart   
When this parameter is present, it indicates whether the stack or the application initiates network   
activity at initialization time. If you set manualStart, the stack does not initiate network activity and   
the application must call the cmInitialize() function, change any non-system parameters, and then   
call cmStart() to complete the stack initialization.   
pdlName   
When this parameter is present, it contains the name of the external PDL object file. When   
pdlName is not set, the stack uses an internal PDL object. If there is no internal PDL object in the   
stack, give the name of the external PDL file.

CCSP System Administrator Guide - 7.3 and 7.4   
79   
cidAssociate   
When this parameter is present, it indicates that the stack should associate calls according to their   
conferenceID value.   
delimiter   
The character used as a delimiter between fields, such as in the destAddress and srcAddress fields   
in the cmCallMake() function.   
allocations   
This structure sets the maximum values for controlling stack initialization allocations:   
•   
vtPoolSize — the maximum size of the PVT pool. The PVT pool is the memory used to store the   
PVT string. There is no need to set this parameter since it is obsolete and kept only for   
backward compatibility.   
•   
vtNodeCount — the maximum number of PVT nodes allowed in all the trees in the system.   
There should be a node for each ASN field in an ASN message. For each string in a message   
(for example, alias), a node supplies 20 bytes of storage space, meaning that the required   
nodes for string = bytes of string/20. Each node requires 28 bytes. Allocation should take into   
account the memory available to the system.   
•   
channels — the maximum number of channels allowed in the system. A channel is a link   
between a PDL state machine and the network. Each channel requires 40 bytes. Allocation   
should take into account the memory available to the system.   
•   
chanDescs — the maximum number of channel types in the system. There is no need to set   
this parameter since it is obsolete and kept only for backward compatibility.   
•   
messages — the maximum number of messages allowed for simultaneously queuing in all   
channels. Each message requires 20 bytes. Allocation should take into account the memory   
available to the system. This parameter is obsolete and kept only for backward compatibility.   
•   
nameChans — the maximum number of name channels allowed in the system. This parameter   
is obsolete and kept only for backward compatibility.   
•   
tpktChans — the maximum number of TPKT channels allowed in the system. Each TPKT   
channel requires 4.1 kilobytes. Allocation should take into account the memory available to the   
system.   
•   
udpChans — the maximum number of UDP channels allowed in the system. There is no need   
to set this parameter.   
•   
protocols — the maximum number of PDLAPI protocols allowed in the system. Each requires   
60 bytes. Allocation should take into account the memory available to the system.   
•   
maxProcs — the maximum number of PDL state machines allowed in the system. Each   
requires 1 kilobyte. Allocation should take into account the memory available to the system.   
•   
maxBuffSize — the maximum buffer required for TPKT channels. The default is 2K.   
Optional allocation parameters   
The following allocation parameters are optional:

CCSP System Administrator Guide - 7.3 and 7.4   
80   
•   
maxCalls — the maximum number of concurrent calls allowed.   
•   
maxChannels — the maximum number of concurrent channels allowed.   
•   
maxRasInTransaction — default definition for controlling the memory allocation for RAS   
transactions based on the maximum number of calls. Defined as maxCalls\*2+10.   
•   
maxRasOutTransaction — default definition for controlling the memory allocation for RAS   
transactions based on the maximum number of calls. Defined as maxCalls\*2+10.   
H.245 parameters   
H.245 parameters specify timeouts for different procedures in the H.245 protocol. The H.245   
parameters are contained in the following structures:   
•   
MasterSlaveConf — parameters for the master/slave determination procedure.   
•   
CapabilitiesConf — parameters related to information on endpoint and sending capabilities.   
•   
ChannelEntry — a list of channel entries. Each entry defines the parameters of a channel.   
•   
ModeEntry — a list of mode entries. Each entry defines the parameters of a mode.   
   
You can change the parameters at runtime unless specified otherwise.   
Syntax   
PDLH245Configuration::=SEQUENCE   
{   
masterSlaveMasterSlaveConfOPTIONAL,   
capabilitiesCapabilitiesConfOPTIONAL,   
channelsSEQUENCEOFChannelEntryOPTIONAL,   
modesSEQUENCEOFModeEntryOPTIONAL,   
channelsTimeoutINTEGER(0..65535)OPTIONAL,   
roundTripTimeoutINTEGER(0..65535)OPTIONAL,   
requestCloseTimeoutINTEGER(0..65535)OPTIONAL,   
requestModeTimeoutINTEGER(0..65535)OPTIONAL,   
mediaLoopTimeoutINTEGER(0..65535)OPTIONAL,   
}   
MasterSlaveConf::=SEQUENCE   
{   
terminalTypeINTEGER(0..255),   
manualOperationNULLOPTIONAL,   
manualResponseNULLOPTIONAL,   
timeoutINTEGER(0..65535)OPTIONAL,   
}   
CapabilitiesConf::=SEQUENCE   
{   
terminalCapabilitySetTerminalCapabilitySet,   
manualOperationNULLOPTIONAL,   
timeoutINTEGER(0..65535)OPTIONAL,   
}   
ChannelEntry::=SEQUENCE   
{   
nameIA5String(SIZE(1..128)),   
dataTypeDataType,   
isDynamicPayloadTypeNULLOPTIONAL,

CCSP System Administrator Guide - 7.3 and 7.4   
81   
}   
ModeEntry::=SEQUENCE   
{   
nameIA5String(SIZE(1..128)),   
modeModeElement,   
}   
Parameters   
The H.245 parameters and structures are as follows.   
channelsTimeout   
The timeout in seconds of the channel establishment.   
roundTripTimeout   
The timeout in seconds of the round trip procedure.   
requestCloseTimeout   
The timeout in seconds of the request close procedure.   
requestModeTimeout   
The timeout in seconds of the request mode procedure.   
mediaLoopTimeout   
The timeout in seconds of the media loop procedure.   
MasterSlaveConf   
This structure includes the parameters needed for the master/slave determination procedure:   
•   
terminalType — the terminal type, for example, 60 for a gateway. The effect of a runtime   
change is seen at the next call.   
•   
manualOperation — when this parameter is present, it cancels the automatic master/slave   
determination procedure. The application developer needs to use the CM API functions to   
execute the master/slave determination procedure manually. The effect of a runtime change is   
seen at the next call.   
•   
manualResponse — when this parameter is present, it cancels the automatic   
acknowledgement for the master/slave determination procedure. It allows the application to   
specify statusDeterminationNumber and acknowledge the procedure as required. The effect   
of a runtime change is seen at the next call.   
•   
timeOut — the maximum time in seconds that the stack waits for a response to the master/   
slave inquiry before it decides it cannot perform the master/slave determination.

CCSP System Administrator Guide - 7.3 and 7.4   
82   
CapabilitiesConf   
This structure holds the parameters related to information on endpoint and sending capabilities:   
•   
terminalCapabilitySet — holds the terminal capabilities. When running a basic H.323   
conference, the stack reads these default capabilities and acts according to them. The default   
values of the parameters in this set are suitable for most terminal types. This capability set   
should be reviewed to ensure that the parameter values are suitable for the endpoint. The   
terminalCapabilitySet is defined in the H.245 Recommendation. The effect of a runtime change   
is seen at the next call.   
The application can also use accessory and advanced functions to dynamically change default   
capabilities at runtime, in which case you may not need to change capability parameters.   
•   
manualOperation — when this parameter is present, it instructs the stack not to send the   
capabilities of an endpoint and cancels auto mode. The application developer needs to use the   
CM API functions to execute the capability exchange. The effect of a runtime change is seen at   
the next call.   
•   
timeOut — the maximum time in seconds that the stack waits before it decides that the called   
party will not acknowledge receipt of the capabilities it sent.   
ChannelEntry   
This structure provides a list of channel entries. Each entry defines the parameters of a channel:   
•   
name — the name that identifies the channel. You can use any name. The cmChannelOpen()   
function uses this name.   
•   
dataType — an internal standard structure that defines the types of information streams and   
their parameters. DataType is defined in the H.245 Recommendation.   
•   
isDynamicPayloadType — when this parameter is present, it tells the stack that the channel   
supports dynamic payload. This parameter is obsolete and kept only for backward   
compatability. The recommended parameter is cmChannelSetDynamicRTPPayloadType.   
ModeEntry   
This structure provides a list of mode entries. Each entry defines the parameters of a mode:   
•   
name — the name that identifies the mode. You can use any name.   
•   
mode — an internal standard structure that defines the types of information streams and their   
parameters.   
   
The ModeEntry is defined in the H.245 Recommendation.   
Q.931 parameters   
The Q.931 parameters configure the Q.931 protocol.   
You can change the parameters at runtime unless specified otherwise.

CCSP System Administrator Guide - 7.3 and 7.4   
83   
Syntax   
{   
responseTimeOutINTEGER(1..200),   
connectTimeOutINTEGER(1..20000),   
callSignalingPortINTEGER(0..65535),   
maxCallsINTEGER(0..65535),   
manualAcceptNULLOPTIONAL,   
notEstablishControlNULLOPTIONAL,   
earlyH245NULLOPTIONAL,   
autoAnswerNULLOPTIONAL,   
manualCallProceedingNULLOPTIONAL,   
overlappedSendingNULLOPTIONAL,   
H245TunnellingNULLOPTIONAL,   
T302INTEGER(1..200)OPTIONAL,   
T304INTEGER(1..200)OPTIONAL   
}   
Parameters   
The Q.931 parameters are as follows.   
responseTimeOut   
The maximum waiting time for the first response to a new call. If no response is received during   
this time, the stack launches the Disconnect procedure. The effect of a runtime change is   
immediate.   
connectTimeOut   
The maximum time that the stack waits for the establishment of a new call, after receiving the first   
response to the call. If the call is not established during this time, the stack launches the Disconnect   
procedure. The effect of a runtime change is immediate.   
callSignalingPort   
The number of the port that receives the calls directed to the endpoint. If this parameter is set to 0,   
the operating system looks for an available port. The effect of a runtime change is only seen after   
STOP/START is applied once, at START.   
maxCalls   
The maximum number of concurrent calls allowed. If this number is exceeded, the stack returns a   
state of busy. The effect of a runtime change is seen at the next call.   
manualAccept   
When this parameter is present, the stack does not automatically accept a call, meaning that the   
stack sends a Call Proceeding message and not an Alerting message. To accept the call, meaning to   
send an Alerting message, use the cmCallAccept() function. The effect of a runtime change is seen   
at the next call.

CCSP System Administrator Guide - 7.3 and 7.4   
84   
notEstablishControl   
When this parameter is present, it causes Q.931 not to move to the H.245 control stage. The effect   
of a runtime change is seen at the next call.   
earlyH245   
When this parameter is present, it causes Q.931 to send and accept H.245 address in the Setup   
message. The effect of a runtime change is seen at the next call.   
autoAnswer   
When this parameter is present, it causes Q.931 to automatically answer calls. You can call the   
cmCallAnswerExt() function while in the cmEvCallStateOffering state to supply display and user   
information. In auto answer mode, the only message sent in response to the Setup message is the   
Connect message.   
manualCallProceeding   
When this parameter is present, the stack does not automatically send a Call Proceeding message.   
To send the message, use the cmCallSendCallProceeding() function.   
overlappedSending   
Indicates that the application supports overlap sending.   
H245Tunneling   
Specifies that an endpoint can support H.245 tunneling. When this parameter is present, tunneling   
is always to be used unless the other party does not support tunneling, or the application   
specifically requests not to use tunneling.   
T302   
The maximum waiting time for address data on the incoming side during an overlapped sending   
procedure.   
T304   
The maximum waiting time for address data on the outgoing side during an overlapped sending   
procedure.   
RAS parameters   
The RAS parameters configure the RAS protocol.   
You can change the parameters at runtime unless specified otherwise.

CCSP System Administrator Guide - 7.3 and 7.4   
85   
Syntax   
PDLRASConfiguration::=SEQUENCE   
{   
responseTimeOutINTEGER(1..200),   
gatekeeperNULLOPTIONAL,   
manualRASNULLOPTIONAL,   
maxFailINTEGER(1..200),   
allowCallsWhenNonRegNULLOPTIONAL,   
manualRegistrationNULLOPTIONAL,   
manualDiscoverySEQUENCE   
{   
defaultGatekeeperTransportAddress   
}OPTIONAL,   
registrationInfoSEQUENCE   
{   
terminalTypeEndpointType,   
terminalAliasSEQUENCEOFAliasAddressOPTIONAL,   
gatekeeperIdentifierGatekeeperIdentifierOPTIONAL,   
endpointVendorVendorIdentifierOPTIONAL,   
timeToLiveTimeToLiveOPTIONAL   
}OPTIONAL,   
rasMulticastAddressTransportAddress,   
rasPortINTEGER(0..65535),   
compare15bitRasCrvNULLOPTIONAL,   
maxRetriesINTEGER(1..200),   
maxMulticastTTLINTEGER(0..200)OPTIONAL,   
overlappedSendingNULLOPTIONAL,   
PreGrantedArqUse::=CHOICE   
{   
directNULL,   
routedNULL,   
}OPTIONAL,   
}   
Parameters   
The RAS parameters are as follows.   
responsetimeout   
The amount of time in seconds that the stack waits until it notifies the application that the called   
party has failed to respond to a transaction. After the given number of seconds, the stack calls   
cmEvRASTimeout(). The effect of a runtime change is immediate.   
gatekeeper   
When this parameter is present, a gatekeeper application is the running application. Otherwise, an   
endpoint application is the running application. The presence or absence of this parameter   
influences stack behavior. For example, if not present, the stack adds the endpoint Id to all   
messages directed to the gatekeeper. You cannot change this parameter at runtime.

CCSP System Administrator Guide - 7.3 and 7.4   
86   
manualRAS   
When this parameter is present, the stack disregards the automatic RAS mechanism and expects   
the application to implement RAS transactions explicitly via RAS functions. When the parameter is   
absent, the stack uses the internal RAS mechanism (basic level). You cannot change this parameter   
at runtime.   
allowCallsWhenNonReg   
When this parameter is present, the stack allows the establishment of calls even when an endpoint   
cannot register with a gatekeeper. This option applies to calls with a known destination transport   
address.   
maxFail   
The maximum number of attempts to register with the gatekeeper before the stack allows call   
establishment without registration, provided that allowCallsWhenNonReg has been set.   
manualRegistration   
When this parameter is present, the stack (manually) registers an endpoint with the gatekeeper   
only when the cmRegister() function is called. When absent, the stack automatically registers the   
endpoint with the gatekeeper. The effect of a runtime change is only seen after STOP/START is   
applied once, at START.   
manualDiscovery   
This parameter is relevant for an endpoint that does not run the automatic gatekeeper discovery   
process. It provides a set of constant values that define the gatekeeper for that endpoint. The   
presence of manualDiscovery disables the automatic discovery mechanism. The elements included   
in the manualDiscovery sequence correspond to and provide all the parameters an endpoint uses   
during automatic gatekeeper discovery (such as transportAddress).   
•   
defaultGatekeeper — the gatekeeper with which you want the endpoint to register. You   
specify the defaultGatekeeper by its IP address and port.   
registrationInfo   
A sequence that includes all the parameters with which the endpoint registers with a gatekeeper.   
The effect of a runtime change is immediate.   
•   
terminalType — the type of endpoint that is registering. This value is defined according to the   
EndpointType type from the H.225.0 Recommendation.   
•   
terminalAlias — a list of alias addresses by which other terminals may identify this terminal.   
The addresses are defined according to the AliasAddress type from the H.225.0 Recommen-  
dation.   
•   
gatekeeperIdentifier — a gatekeeper ID for identifying the gatekeeper with which the terminal   
wishes to register. This value is defined according to GatekeeperIdentifier type from H.225.0   
Recommendation.

CCSP System Administrator Guide - 7.3 and 7.4   
87   
•   
endpointVendor — an endpoint vendor ID to be used in registration. This value is defined   
according to the VendorIdentifier type from the H.225.0 Recommendation.   
•   
timeToLive — when this parameter is set, determines the number of seconds before the   
registration becomes invalid.   
rasMulticastAddress   
The well-known multicast address of the RAS Multicast IP group. The effect of a runtime change is   
only seen after STOP/START is applied once, at START.   
rasPort   
The number of the port that receives all RAS transactions directed to the endpoint. You set this   
parameter to 0 to let the operating system look for an available port. The effect of a runtime   
change is only seen after STOP/START is applied once, at START.   
compare15bitRasCrv   
Causes an endpoint to ignore the CRV MSBit (the flag bit) in RAS messages. Default is 16 bit.   
maxRetries   
The number of retransmission retries. Default: 3. The effect of a runtime change is immediate.   
maxMulticastTTL   
The number of multicast TTL. Default: 20. The effect of a runtime change is only seen after   
STOP/START is applied once, at START.   
overlappedSending   
When present, the endpoint supports Overlapped Sending.   
preGrantedArqUse   
•   
direct — if equal to direct, the pre-granted ARQ feature is used for both direct and routed calls.   
•   
routed — if equal to routed, the pre-granted ARQ feature is used only for routed calls. If   
absent, the pre-granted ARQ is not used. The effect of a runtime change is seen at the next   
call.

CCSP System Administrator Guide - 7.3 and 7.4   
88   
11: Configure dialing plans   
CCSP allows for the installation of agents and VoIP Connection Servers (VCSs) with a high degree of   
geographic flexibility. Since agents and VCSs can be located anywhere around the world, it is   
necessary to choose VCSs for out-dial requests based upon the location of the dialed number. The   
dialing plan feature adds this capability to CCSP. It also allows for provisioning of calling privileges   
on a per tenant/group/agent basis.   
The VCS selection criteria and the tenant/agent/group privileges are customizable via ASP scripting.   
With the dialing plan feature turned on, each out-dial request from an agent is sent to the dialing   
plan web page. The web page determines if the agent is allowed to make the call and creates a list   
of VCSs in order of preference to be used for the out-dial. If the tenant, agent or group is blocked   
from making the call or no VCSs are found, the agent receives a r;Call rejected message.   
If more than one VCS is returned, the agent software will try to make the call using the first VCS in   
the list. If it fails for any reason, it will try each subsequent VCS in the list. If all VCSs in the list fail,   
the agent will receive a No outbound lines message.   
Dialing plans are also referred to as r;intelligent outdialing.   
Note   
•   
The maximum outgoing call number or SIP URI is 370 characters.   
•   
Each CCSP UI must be minimally configured with the correct dialing location for any dialing   
plan to work. For details, see the CCSP UI Help.   
Configure the VCS   
You must specify the location information for all VCSs that are configured in your contact center.   
To configure each VCS:   
1.   
In the Control Panel, click Phone and Modem.   
2.   
On the Dialing Rules tab, click New. The New Location dialog box appears.   
3.   
In Location name, type the name.   
4.   
In the Country/region list, click the country or region from which the dialing location origi-  
nates.   
5.   
In Area code (or city code), type the new area code.   
6.   
Click OK. The Phone and Modem dialog box appears.   
7.   
Click the name of the new location and click OK.   
   
Note   
To exclude a VCS from the dialing plan, in CCSP Admin, delete both the country code and the   
area code from the VCS properties.

CCSP System Administrator Guide - 7.3 and 7.4   
89   
Enable agent relative number translations   
To translate all numbers to their shortest form relative to the agent before further processing by   
the dialing plan, enable agent relative number translations.   
To enable agent relative number translations:   
1.   
On the IIS server, navigate to /DialingPlan/ and open CheckPlan.asp in a text editor.   
2.   
Set var enableAgentRelativeNumberTranslations to true.   
3.   
Restart the IIS.   
Enable routing code remapping   
Some gateway providers require special codes to be sent through the gateway preceding a normal   
dialable number to apply a certain behavior. For example, a gateway provider may require that you   
prefix numbers with \*67 so that your phone number is withheld and the called party cannot see it.   
However an asterisk (\*) is not a standard dialable digit for E.164 numbers as it is not present on   
some phones. This code always has a translated form that is made up from the normal dialable   
digits 0-9; such that the code \*67 may be equivalent to 1167.   
You enable remapping of the routing code and optionally configure the remapping in   
CheckPlan.asp.   
The routing modes are as follows.   
Code   
Meaning   
0   
Disabled — the routing code has already been remapped or there is no remapping   
configured.   
1   
Standard — removes the routing code from the dialled number before submitting the   
number for standard checkplan processing. Finally the remapped code is prepended to   
the front of the translated number.   
2   
Preprocess — remaps the routing code in the dialled number field before any other   
checkplan processing. This mode remaps the code before any national or international   
dialling codes are added or removed.   
3   
Smart — removes the routing code as in Standard mode, but instead of replacing it at   
the front of the translated number, it injects the number after any added country code.

CCSP System Administrator Guide - 7.3 and 7.4   
90   
To enable routing code remapping:   
1.   
On the IIS server, navigate to /DialingPlan/ and open CheckPlan.asp in a text editor.   
2.   
Set var enableRoutingCodeRemapping to true.   
3.   
If required, configure the values in var routingCodeRemappingTable. The default values are:   
[   
{ match: /\\*67(\d+)/, code: "1167", remappingMode: 3 } //translates   
numbers of the form \*67xyz to 1167xyz   
, { match: /\\*68(\d+)/, code: "1268", remappingMode: 3 } //translates   
numbers of the form \*68xyz to 1268xyz   
, { match: /00(\d+)/, code: "+", remappingMode: 2 } //translates   
numbers of the form 00xyz to +xyz   
];   
4.   
Restart the IIS.   
Test the dialing plan   
You can test the dialing plan independently of the agent by using the test.asp page included in the   
installation.   
To test the dialing plan:   
1.   
Use a browser to load the test.asp page in the dialing plan virtual directory on the web server,   
for example, http://universe.cosmocall.cosmocom.com/DialingPlan/Test.asp   
2.   
Enter the information:   
o   
Tenant Name — select from the list.   
o   
Agent's Country Code — type a valid country code.   
o   
Agent's City Code — type a valid city code.   
o   
Phone Number — type a phone number. The number should not contain any format   
characters, such as parentheses, dashes, commas, or spaces. It must also be in a valid   
format for the country and city that you are dialing from, for example, 0016319404200   
3.   
The result appears as an XML document. Confirm the following:   
o   
The <DialResponse Allow="TRUE"> line specifies whether the out-dial request was   
accepted or rejected. If Allow="FALSE" then no other lines appear.   
o   
The following lines specify the CSs that were chosen to do the out-dial:   
<ConnectionServer Type="VCS" Server="VCS1" ID="10" TranslatedNumber="99404200"   
OrigNumber="9404200" />   
o   
The lines are listed in order of preference. The ID field is the connection server ID of the   
VCS. The TranslatedNumber field is the translated phone number. It has been translated   
from the agent's location perspective to the VCS’s location perspective. It has also been   
prepended with the VCSs access codes, if any exist. The OrigNumber field is the original   
number that was dialed by the agent.

CCSP System Administrator Guide - 7.3 and 7.4   
91   
In the following example, the agent is in the United Kingdom and is dialing a phone number in   
Melville, New York. One VCS is also located in Melville, New York and the other is located in the   
United Kingdom, so the phone number for the first VCS has been translated from an international   
call to a local call.   
<?xml version="1.0" ?> =   
<DialResponse Allow="TRUE">   
<ConnectionServer Type="VCS" Server="VCS1" ID="10" TranslatedNumber="99404200"   
OrigNumber="9404200" />   
<ConnectionServer Type="VCS" Server="VCS2" ID="11"   
TranslatedNumber="900163199404200" OrigNumber="9404200" />   
</DialResponse>   
Customize dialing plans   
Information about agent/group dialing privileges is stored in the configuration database. This   
information can be accessed through the CCSP Config Com object.   
Only the system administrator can customize the agent privileges determination and the VCS   
selection process.   
There are two ASP (VBScript) functions in the C:\Program Files (x86)\CosmoCom\Server   
Components\Dialing Plan\DialingPlan.inc file, CheckPrivileges() and CreateCSList(), which can be   
modified or rewritten.   
The only restrictions are that the file name, and the functions' input parameters and output types   
must remain the same.   
By default the CheckPrivileges() function uses the DP\_Privileges database table in the configuration   
database to determine if the agent or group is allowed to make the specified call.   
By default the CreateCSList() function will do a simple Least Cost Routing determination based on   
the dialed number and each VCS's location. It will first try to find a VCS with the same country code   
and city code as the dialed number. Then it will try to find any VCS with the same country code.   
Finally, it will take any remaining VCSs.   
CheckPrivileges input parameters   
The CheckPrivileges input parameters are:   
•   
CallType — type of call request. Its value will be one of INTL\_CALL, LD\_CALL, LOCAL\_CALL, or   
OTHER\_CALL.   
•   
AgentId — the agent’s ID.   
•   
GroupId — the agent’s group ID.   
•   
TenantId — the agent’s tenant ID.   
•   
CityCode — the city code portion of the dialed number.   
•   
LocalNum — the local number portion of the dialed number.

CCSP System Administrator Guide - 7.3 and 7.4   
92   
•   
Output: ReturnValue — this function should return TRUE if the agent is allowed to make the   
requested out-dial, otherwise return FALSE.   
CreateCSList input parameters   
The CreateCSList input parameters are:   
•   
CallType — type of call request. Its value will be one of INTL\_CALL, LD\_CALL, LOCAL\_CALL, or   
OTHER\_CALL.   
•   
TenantId — the agent’s tenant ID.   
•   
CntryCode — the country code of the dialed number.   
•   
CityCode — the city code of the dialed number.   
•   
LocalNum — the local number portion of the dialed number.   
•   
TransferID — if this parameter is not an empty string (" "), then it indicates that the outdial   
request is a transfer. A transfer must only select the VCS from which the inbound call is   
originating. This ID will be that of the VCS’ ID from which the inbound call is coming.   
•   
Output: ReturnValue — this function should return a string that contains colon-delimited   
records. Each record will contain the VCS Server ID, the string "VCS" and the phone number to   
dial. The server ID being the VCS connection server ID, and the phone number being the   
translated phone number to be dialed by the VCS.   
The fields are separated by a semi-colon, ";", and the records are separated by a colon, ":", for   
example:   
2;VCS;99404200:1;VCS;916319404200:3;VCS;90016319404200   
To cause the call to be rejected if no VCSs meet the correct criteria, return an empty string, "".

CCSP System Administrator Guide - 7.3 and 7.4   
93   
12: Configure SNMP traps   
Simple Network Management Protocol (SNMP) is the protocol governing network management   
and the monitoring of network devices and their functions. It is used to examine and change   
configuration parameters and counters of LAN- and WAN-connected devices.   
SNMP applications running on monitored equipment, use Management Information Bases (MIBs),   
which communicate with management stations. A MIB is a formal description of a set of network   
objects that can be managed using SNMP. The format of the MIB is defined as part of SNMP.   
The CCSP MIB is used in conjunction with third-party SNMP-compliant management applications. It   
specifies whether the CCSP connection servers, including the AIS, have started or been stopped for   
any reason.   
The SNMP protocol supports only the following functions, the first three of which can be managed,   
and the last of which is sent by the SNMP agent as an unsolicited message:   
•   
Get — a specified variable’s current value from an agent   
•   
GetNext — get the next variable’s value   
•   
Set — a variable   
•   
Trap — CCSP UI sends a message when a threshold is exceeded   
Implement traps   
You use third-party SNMP management applications to monitor traps from your servers.   
To implement traps:   
•   
The SNMP Service must be installed on the servers you want to monitor, for example, VCS,   
MCS, and ICS, and on the computers designated to receive trap messages.   
•   
The concurrent SNMP Trap service must be enabled or disabled, depending on your   
requirements.   
•   
The CosmoCom.mib file must be added to your third-party SNMP management application   
and then be compiled for use. The SNMP Trap service may need to be restarted, depending on   
the requirements of the third-party application you use.   
   
The trap agents are assigned what is called a community string (weak password), which is not   
encrypted when it is sent over the network. By default, both the read-access and the write-access   
community strings are public. The write-access community string may instead be private. If you   
have any concerns about security, you should change the default passwords.

CCSP System Administrator Guide - 7.3 and 7.4   
94   
The CosmoCom.mib file supports three traps:   
•   
ServiceStart   
•   
ServiceStop   
•   
Message   
   
These traps send a message to the computer that is running the SNMP management application   
when any CCSP server or service starts or stops.   
You must add the CosmoCom.mib file to your SNMP management application and then compile it   
for use.   
Configure traps   
To use SNMP, it must be installed and started on the servers you want to monitor. It must also be   
installed on the computer that will run the monitoring service.   
Note   
If you change existing SNMP settings, your changes take effect immediately. The SNMP   
Service does not need to be restarted for your settings to take effect.   
   
To verify that the SNMP Service and the SNMP Trap Service are installed:   
1.   
In Server Manager, on the Tools menu, click Services.   
2.   
Scroll through the list of services to see if SNMP Service and SNMP Trap appear. If not, install   
the SNMP Service.   
3.   
Confirm that:   
o   
SNMP Service is set to start automatically.   
o   
SNMP Trap is set to start manually.   
   
To add destinations for trap messages:   
1.   
In Server Manager, on the Tools menu, click Computer Management.   
2.   
In the navigation pane, expand Services and Applications, and click Services.   
3.   
In the details pane, click SNMP Service.   
4.   
On the Action menu, click Properties.   
5.   
On the Traps tab, under Community name, type the case-sensitive community name to which   
this computer will send trap messages. The default is public.   
6.   
Click Add to list.   
7.   
In Trap destinations, click Add.   
8.   
In Host name, IP or IPX address, type the IP address of the computer that runs the third-party   
SNMP management application or the script that is run when a message is received, and click   
Add.   
9.   
Repeat steps 5 through 8 until you have added all the communities and trap destinations you   
want.

CCSP System Administrator Guide - 7.3 and 7.4   
95   
13: Configure the ECS   
A CCSP system requires a single instance of the ECS server. This single instance provides   
synchronization of campaign-related data between the CCSP configuration database and EPC   
databases for all tenants running in Elsbeth dialer mode.   
One installation of the ECS server deploys two Windows services, by default named CCSP Elsbeth   
Config Sync CT and CCSP Elsbeth Config Sync SS. Each of these services performs a different part of   
the configuration synchronization task but they work as a pair and cannot complete the full   
synchronization process if both are not running properly.   
The CCSP Elsbeth Config Sync CT service performs change detection and processing using the SQL   
Server Change Tracking feature, which is enabled on target databases and tables. The service polls   
for changes in configurable intervals (default is 30 seconds). The detected changes are organized   
and stored in workset tables.   
The CCSP Elsbeth Confing Sync SS service performs the actual data synchronization between the   
databases. It retrieves the lists of changed data from the workset tables populated by the CCSP   
Elsbeth Config Sync CT service and synchronizes those changes between the databases using the   
administration/configuration APIs of CCSP (AdminServer) and Elsbeth (EPCServer). This change   
retrieval and synchronization is done in configurable intervals (default is 30 seconds).   
A system can have multiple ECS server instances, 1 per machine, but only one of those can be   
running at a given time. Other instances can be used for failover.   
Configure ECS services   
Note   
After any change in the configuration file, the file must be saved and both ECS services must be   
manually restarted for changes to take effect.   
   
You configure the single ECS server instance by editing the ECS.exe.config XML file located in the   
installation folder of the ECS services. Both of the ECS windows services in an installation use the   
same configuration file.   
The configuration file contains the following configuration keys of interest:   
•   
LogLevel — the level of logging done by both ECS services. The applicable values are the   
standard CCSP log level values. The default is 7 (Info).   
•   
AuthServerURL — the URL pointing to the CCSP AuthServer.   
•   
AdminServerURL — the URL pointing to the CCSP AdminServer.   
•   
SynchronizerServiceClientID — the client ID registration value assigned to the ECS server   
instance during the installation process, which is used together with   
SynchronizerServiceSecret as part of the service authentication mechanism. This value must   
not be modified manually.

CCSP System Administrator Guide - 7.3 and 7.4   
96   
•   
SynchronizerServiceSecret — the secret key assigned to the ECS server instance during the   
installation process, which is used together with SynchronizerServiceClientID as part of the   
service authentication mechanism. This value must not be modified manually.   
•   
CcspConfigurationDatabseConnectionString — the connection string to the CCSP   
configuration database.   
Restart ECS services   
When requested to start, the ECS services CT and SS only validate basic dependency requirements.   
If those are satisfied, the service is declared as successfully started and begins full initialization.   
There are two initialization failure modes exhibited by ECS services.   
The first one is where the services start and continue running but the synchronization for mapped   
and enabled tenants, that failed these initialization steps, is unavailable. In this case an error event   
log and system log entries are created that provide an initialization summary of which mapped   
enabled tenants passed initialization and which failed. These initialization steps include:   
For ECS SS service:   
•   
Check CCSP AdminServer availability   
•   
Check EPC Server availability   
•   
Check EPC Server version compatibility   
•   
Retrieve EPC DB parameters from EPC Server   
   
For ECS CT service:   
•   
Retrieve table resync requests from CCSP configuration DB   
•   
Check EPC Server availability   
•   
Check EPC Server version compatibility   
•   
Retrieve EPC DB parameters from EPC Server   
   
This behavior is to allow the ECS services to provide synchronization for those mapped enabled   
tenants for which initialization succeeded until the issues that caused initialization failures for the   
other mapped enabled tenants are resolved. After this is done for one or more mapped enabled   
tenants that failed the initialization, ECS services must be restarted.   
The second failure mode is one where the services generate a fault, write custom fault information   
into the system event log and regular logs, and exit (stop), qualifying for automatic service recovery   
per recovery configuration for that service instance. The initialization steps whose failure triggers   
this behavior are primarily:   
•   
ECS configuration file loading   
•   
Initial tenant mapping retrieval from CCSP configuration DB   
•   
Initial service authentication with CCSP Auth service and initial access token retrieval   
(applicable only to ECS Synchronizer service).

CCSP System Administrator Guide - 7.3 and 7.4   
97   
After the services successfully complete the initialization and start operating:   
•   
An info event log entry is created which provides the initialization summary.   
•   
The services start by performing a full tenant synchronization, in case of initial tenant   
synchronization or expired database change tracking retention period, followed by   
subsequent delta synchronizations.   
•   
The services continue performing delta tenant synchronizations if change tracking information   
is available in the tenant's database and its retention period has not expired.   
Configure tenant mappings   
Note   
•   
As part of the installation or update process, the ECS installer automatically suspends all   
tenant mappings and automatically makes a pending re-synchronization request for all   
existing tenant mappings.   
•   
After modifying any tenant mapping, you must manually restart both ECS services to apply   
the changes. You can restart the services in any order.   
   
Tenant mapping is a key configuration process through which a link is established between a CCSP   
tenant (defined by its ID) and an EPC mandant/tenant (defined by its EPCServer instance).   
The ECS server provides synchronization functionality only for tenants for which mappings have   
been provided and activated.   
When a tenant is mapped for the first time or re-mapped (a sequence of unmap and map   
operations for a previously mapped tenant, issued via ECS.exe command line), a full re-  
synchronization is explicitly requested for that tenant. Standard re-synchronization rules are   
applied for this re-synchronization request.   
The tenant mapping settings for ECS purposes are stored in the ecs.TenantMapping table of the   
CCSP configuration database while resynchonization requests are stored in   
ecs.ResyncTableRequests table of CCSP configuration database.   
The tenant mappings are controlled through the ECS service executable command line. To access   
it:   
1.   
Open a console window on the ECS server machine. Because ECS connects to the CCSP   
configuration database as part of the mapping process using the   
CcspConfigurationDatabseConnectionString value specified in the ECS.exe.config file, the   
console window must be opened under a user with sufficient credentials to allow such   
database operations if the connection string specifies integrated security options.   
2.   
Navigate to the ECS installation folder.   
3.   
Execute ECS.exe with the appropriate tenant mapping parameters. For full command-line   
documentation and examples, refer to the rest of this section.

CCSP System Administrator Guide - 7.3 and 7.4   
98   
Command line reference   
Note   
From Elsbeth 9.x, the epc\_password (-ep) parameter is deprecated. The EPC Server service password must be encrypted via EPC’s   
PowerContactTextEncrypter tool and then supplied to ECS via the new epc\_encrypted\_password (-eep) parameter.   
   
Command line options, both full-form and their aliases, must be prefixed with either a double hyphen (--), single hyphen (-), or forward   
slash (/).   
Values should be separated on the command line from the command line option using an equals sign (=) or colon (:).   
Configuration mode command line   
The configuration command set is initiated using configure, as described below.   
Command line option   
Alias   
Value   
Mandatory   
Description   
configure   
c   
n/a   
Y   
Put ECS in configuration mode. This mode must be   
specified to map, remap, or unmap a tenant mapping.   
change\_tracking\_interval   
cti   
Numeric   
N   
Optionally set the change tracking interval (in seconds) of   
the tenant mapping. Default value is 30.   
create\_mapping   
map   
n/a   
Y1   
Tell ECS to create a new tenant mapping or edit an existing   
tenant mapping.   
delete\_mapping   
unmap   
String   
Y1   
Tell ECS to delete an existing tenant mapping.   
ecs\_CT\_epc\_user\_password   
ectup   
String   
Y2, 7, 8   
Specify the plaintext password for the dedicated ECS CT   
service EPC user   
"SystemReservedApplicationUser\_ECS\_CT".   
Elsbeth 6.x: EPC Server will not be contacted.   
From Elsbeth 9.x: EPC Server will be contacted.8.

CCSP System Administrator Guide - 7.3 and 7.4   
99   
Command line option   
Alias   
Value   
Mandatory   
Description   
ecs\_CT\_epc\_user\_encrypted\_password   
ectuep   
String   
Y2, 7, 8, 9   
From Elsbeth 9.x: Specify the encrypted password for the   
dedicated ECS CT service EPC user   
"SystemReservedApplicationUser\_ECS\_CT".   
EPC Server will not be contacted.8’ 9   
Note: Use either ecs\_CT\_epc\_user\_password or   
ecs\_CT\_epc\_user\_encrypted\_password.   
ecs\_SS\_epc\_user\_password   
essup   
String   
Y2, 7, 8   
Specify the plaintext password for the dedicated ECS SS   
service EPC user   
"SystemReservedApplicationUser\_ECS\_SS".   
Elsbeth 6.x: EPC Server will not be contacted.   
From Elsbeth 9.x: EPC Server will be contacted.8.   
ecs\_SS\_epc\_user\_encrypted\_password   
essuep   
String   
Y2, 7, 8, 9   
From Elsbeth 9.x: Specify the encrypted password for the   
dedicated ECS SS service EPC user   
"SystemReservedApplicationUser\_ECS\_SS".   
EPC Server will not be contacted.8’ 9   
Note: Use either ecs\_SS\_epc\_user\_password or   
ecs\_SS\_epc\_user\_encrypted\_password   
epc\_password   
ep   
String   
Y2   
Elsbeth 6.x: Specify the EPC Server services password that   
ECS should use for this tenant mapping.5   
From Elsbeth 9.x: This command is deprecated.   
epc\_encrypted\_password   
eep   
String   
Y 9   
From Elsbeth 9.x: Specify the EPC Server service encrypted   
password that ECS should use for this tenant mapping.   
EPC Server will not be contacted.8’ 9   
epc\_server   
es   
String   
Y2,3   
Specify the EPC Server address that ECS should use for this   
tenant mapping.5

CCSP System Administrator Guide - 7.3 and 7.4   
100   
Command line option   
Alias   
Value   
Mandatory   
Description   
epc\_username   
eu   
String   
Y2   
Specify the EPC Server services username that ECS should   
use for this tenant mapping.5   
mandant\_id   
mi   
String   
Y4   
Set the Elsbeth Third Party ID for this tenant mapping,   
stored in the dbo.tTenants table of CCSP configuration   
database.   
mapping\_enabled   
   
true/false   
N   
Optionally set the enabled state of the tenant mapping.   
Default value is true.   
preserve\_mapping\_status   
pms   
true/false   
N   
When this flag is specified, the tenant mapping suspension   
status does not get automatically cleared (if set). If the   
switch is omitted, any configuration mode command clears   
the tenant mapping suspension status (if set).   
synchronization\_interval   
si   
Numeric   
N   
Optionally set the synchronization interval of the tenant   
mapping. Default value is 30.   
tenant\_id   
t   
Numeric   
Y3   
Specify the CCSP Tenant ID of the tenant mapping.   
Note: If updating an already mapped tenant, to apply the   
change, all the tenant’s agents that are using EPC Gadget   
at that moment must log in again.   
   
   
Notes   
1. Either create\_mapping (map/remap) or delete\_mapping (unmap) must be specified. Not both.   
2. The EPC details are only mandatory when creating a new mapping. When remapping they are only required if you are changing   
one of the values, and when unmapping they are not required at all.   
3. Either the tenant\_id (t) or epc\_server (es) must be specified to find the mapping you are editing or deleting. When creating a new   
mapping you need to specify both, when remapping or unmapping you can specify either. If both are specified, the tenant\_id takes   
precedent

CCSP System Administrator Guide - 7.3 and 7.4   
101   
4. The ECM mandant name value, stored in the dbo.tTenants table of the CCSP configuration database, is not required for proper   
functioning of the configuration synchronization functionality provided by ECS. However, it is mandatory for overall Elsbeth   
integration with CCSP because it is used in general dialer workflows to provide coupling between the CCSP system and Elsbeth   
dialer.   
5. The URL must point to the WCF API host address of the target EPC Server. This value is specified during the EPC Server installation   
process in the General Settings > SOAP Service (BASEADDRESSSOAPSERVICE) parameter.   
o   
Elsbeth 6.x: It can be found after EPC Server installation in the the PowerContactServer.exe.config configuration file of the EPC   
Server instance in “configuration/system.serviceModel/services/service/host/baseAddresses” element group. For more information,   
refer to the EPC Installation Guide and CCSP Dialer Guide for Elsbeth.   
o   
From Elsbeth 9.x: It can be found after EPC Server installation in the “WcfApi” element group of the   
PowerContactServer.appsettings.json configuration file of the EPC Server instance. There must be no trailing forward slashed in   
the URL. For more information, refer to the EPC Installation Guide and CCSP Dialer Guide for Elsbeth.   
6. The EPC Server service user credentials are specified during EPC Server installation in the General Settings > Service ID   
(SERVICESUSER) and General Settings > Service PWD (SERVICESPASSWORD) parameters.   
o   
Elsbeth 6.x: They can be found after EPC Server installation in the PowerContactServer.exe.config configuration file of the EPC   
Server instance, in “configuration/applicationSettings/ Sarix.PowerContact.Properties.Settings/” element group in “setting”   
elements “ServicesUser” and “ServicesPassword”.   
o   
From Elsbeth 9.x: They can be found after EPC Server installation in the EPC database table PWC.SETTINGS\_SERVICES as   
SERVICES\_USER and SERVICES\_PASSWORD field values respectively. For more information, refer to the Configuring section of the   
EPC Installation Guide.   
7. EPC has two system-reserved users: ECSCTApplicationUserEPCUserPassword and ECSSSApplicationUserEPCUserPassword. Both   
users are intended to be used for authentication purposes by the ECS ChangeTracker and ECS Synchronizer services. The   
passwords for these users are changeable from the EPC Admin tool but must be provided by ECS when authenticating. These   
passwords are stored in the ecs.TenantMapping table. The password for these new EPC users is initially the same default value as   
for EPC’s “admin” user, “powercontact”, but can be changed through EPC Admin.   
Existing tenant mappings in CCSP version 7.3 and 7.3.1 must be updated to enable these two passwords to be set for each tenant.   
8. From Elsbeth 9.x: Either the plaintext password variant or the encrypted password variant of the command can be used at the   
same time. If the plaintext version is used, EPC Server is contacted by ECS to perform the encryption but a valid EPC Server service   
password must be known to ECS: either by supplying it via the “-eep” parameter or through an existing tenant mapping entry for   
the specified tenant. If EPC Server cannot be contacted when specifying the password in plain text, the password assignment fails   
to consider either fixing the EPC Server issue immediately and trying again or using the variant of this command that accepts an   
encrypted password, so EPC Server does not get contacted.

CCSP System Administrator Guide - 7.3 and 7.4   
102   
9. EPC password encryption is performed by the PowerContactTextEncrypterUI (user interface) or PowerContactTextEncrypterCLI   
(command-line interface) tools provided as part of the EPC installation pack, in the TextEncrypter subfolder. For details, refer to the   
EPC Text Encrypter Guide. The same tool is used to generate all encrypted passwords for EPC (including the EPC Gadget, as   
configured via the Campaign gadget’s Service password parameter in CCSP UI Admin) and the same plaintext input always   
generates the same encrypted output.   
   
Mapping examples   
All command line options marked mandatory must be specified when creating a new mapping.   
For Elsbeth 6.x:   
ecs --configure --create\_mapping --tenant\_id:12 --epc\_server:"http://alt1.pj16.loc:7931/services/" --  
epc\_username:myUser --epc\_password:secret --ecs\_SS\_epc\_user\_password:powercontact --  
ecs\_CT\_epc\_user\_password:powercontact --mandant\_id:ElsbethTenant1   
As above, but using aliases:   
ecs -c -map -t:12 -es:"http://alt1.pj16.loc:7931/services/" -eu:myUser -ep:secret -essup:powercontact -  
ectup:powercontact -mi:ElsbethTenant1   
As above, but using aliases and including some optional fields:   
ecs -c -map -t:12 -es:"http://alt1.pj16.loc:7931/services/" -eu:myUser -ep:secret -essup:powercontact -  
ectup:powercontact -mi:ElsbethTenant1 -si:40   
   
For Elsbeth 9.x:   
ecs --configure --create\_mapping --tenant\_id:12 --epc\_server:"http://alt1.pj16.loc:7931/services/" --  
epc\_username:myUser --epc\_encrypted\_password:9CA089A518D9235C0E010711C1EFB7C2   
--ecs\_SS\_epc\_user\_password:powercontact --ecs\_CT\_epc\_user\_password:powercontact --mandant\_id:ElsbethTenant1   
As above, but using aliases:   
ecs -c -map -t:12 -es:"http://alt1.pj16.loc:7931/services/" -eu:myUser -  
eep:9CA089A518D9235C0E010711C1EFB7C2 -essup:powercontact -ectup:powercontact -mi:ElsbethTenant1   
As above, but using aliases and including some optional fields:   
ecs -c -map -t:12 -es:"http://alt1.pj16.loc:7931/services/" -eu:myUser -  
eep:9CA089A518D9235C0E010711C1EFB7C2 -essup:powercontact -ectup:powercontact -mi:ElsbethTenant1 -si:40

CCSP System Administrator Guide - 7.3 and 7.4   
103   
As above, but using encrypted passwords for all password-related fields:   
ecs -c -map -t:12 -es:"http://alt1.pj16.loc:7931/services/" -eu:myUser -  
eep:9CA089A518D9235C0E010711C1EFB7C2 -essuep:3822C690457F3A57D6A861B48D82E7B7 -  
ectuep:3822C690457F3A57D6A861B48D82E7B7 -mi:ElsbethTenant1 -si:40   
Remapping (updating an existing mapping) examples   
Use the map command to remap or change mappings. Only the tenant\_id or the epc\_server must be specified because they are unique   
entries in the TenantMapping table, so the mapping can be found. If both are specified, the tenant\_id is considered the primary search   
value. If the tenant mapping is found, the corresponding epc\_server is set to the new value.   
Edit the epc server address for the mapping for tenant id 12:   
ecs -c -map -t:12 -es:"http://alt1.pj16.loc:7931/services/"   
   
Edit the epc password for the mapping for epc server "http://alt1.pj16.loc:7931/services/":   
For Elsbeth 6.x:   
ecs -c -map -es:"http://alt1.pj16.loc:7931/services/" -ep:"don't tell anyone"   
For Elsbeth 9.x:   
ecs -c -map -es:"http://alt1.pj16.loc:7931/services/" -  
eep:F5F971479BE91AA927231EF4EA1CE0983C8E9294A1BF021E717CDD536C0747CF   
   
Change the change tracking and synchronization intervals to 20 seconds both for tenant id 12:   
ecs -c –map -t:12 -cti:20 -si:20   
   
Disable the mapping for tenant id 12:   
ecs -c -map -t:12 -mapping\_enabled:false   
   
Update a tenant mapping and automatically clear the suspension flag (if set):   
ecs --configure -map --tenant\_id:4 -ectup:powercontact

CCSP System Administrator Guide - 7.3 and 7.4   
104   
   
Update a tenant mapping and retain the suspension flag:   
ecs --configure -map --tenant\_id:4 -ectup:powercontact –pms   
Unmapping examples   
Use the unmap command to remove existing mappings. Only the tenant\_id or the epc\_server must be specified so the mapping can be   
found. Do not specify both.   
   
Delete the mapping for tenant id 12:   
ecs --configure --delete\_mapping --tenant\_id:12   
   
Delete the mapping for epc server "http://alt1.pj16.loc:7931/services/":   
ecs -c -unmap -es:"http://alt1.pj16.loc:7931/services/"

CCSP System Administrator Guide - 7.3 and 7.4   
105   
Administration mode command line   
The administration command set is initiated using administer, as described below.   
For command line options that are part of a set, only one of the options can be used at a time, or none if the option set is optional.   
Command line option   
Alias   
Value   
Mandatory   
Set name   
Description   
administer   
a   
n/a   
Y   
Mode   
Put ECS in   
administration mode.   
This mode must be   
specified to:   
•   
Suspend or resume   
tenants   
•   
Request full resync   
for tenants   
all\_tenants   
n/a   
n/a   
Yes (1 of   
set)   
Common   
Apply the command to   
all tenants in   
ecs.TenantMapping   
table.   
clear   
n/a   
n/a   
Yes (1 of   
set)   
ResyncRequestControlOperation   
Clears the resync   
request for the selected   
tenant or tenants.   
enabled \_tenants   
n/a   
n/a   
Yes (1 of   
set)   
Common   
Apply the command to   
all enabled tenants in   
ecs.TenantMappign   
table (IsEnabled flag set   
to true).   
resume   
n/a   
n/a   
Yes (1 of   
set)   
ExecutionControlOperation   
Clears the suspension   
status (resumes) for the   
selected tenant   
mapping or mappings.

CCSP System Administrator Guide - 7.3 and 7.4   
106   
Command line option   
Alias   
Value   
Mandatory   
Set name   
Description   
set   
n/a   
n/a   
Yes (1 of   
set)   
ResyncRequestControlOperation   
Sets the resync request   
for the selected tenant   
or tenants.   
suspend   
n/a   
n/a   
Yes (1 of   
set)   
ExecutionControlOperation   
Suspends the selected   
tenant mapping or   
mappings.   
tenant\_execution\_control   
texc   
n/a   
Yes (1 of   
set)   
CommandType   
enables   
ExecutionControlOperation set   
Use the tenant   
execution control   
command to control the   
tenant mapping status   
(between suspended or   
resumed).   
tenant\_id   
n/a   
Tenant id from   
ecs.TenantMapping   
table   
Yes (1 of   
set)   
Common   
Apply the command to   
the specified tenant in   
ecs.TenantMapping   
table.   
tenant\_resync\_request\_control   
trrc   
n/a   
Yes (1 of   
set)   
CommandType   
Enables   
ResyncRequestControlOperation   
set   
Use the tenant resync   
request control   
command to set or clear   
full resync requests.   
Suspend resume examples   
Suspend all tenants:   
ecs.exe --administer -texc --suspend --all\_tenants   
Resume all tenants:   
ecs.exe --administer -texc --resume --all\_tenants

CCSP System Administrator Guide - 7.3 and 7.4   
107   
Suspend enabled tenants:   
ecs.exe --administer -texc --suspend --enabled\_tenants   
Resume enabled tenants:   
ecs.exe --administer -texc --resume --enabled\_tenants   
   
Suspend selected tenant:   
ecs.exe --administer -texc --suspend --tenant\_id:4   
Resume selected tenant:   
ecs.exe --administer -texc --resume --tenant\_id:4   
Request resync examples   
Set resync request for all tenants:   
ecs.exe --administer -trrc --set --all\_tenants   
Clear resync request for all tenants:   
ecs.exe --administer -trrc --clear --all\_tenants   
Set resync request for enabled tenants:   
ecs.exe --administer -trrc --set --enabled\_tenants   
Clear resync request for enabled tenants:   
ecs.exe --administer -trrc --clear --enabled\_tenants   
Set resync request for selected tenant:   
ecs.exe --administer -trrc --set --tenant\_id:4   
Clear resync request for selected tenant:   
ecs.exe --administer -trrc --clear --tenant\_id:4

CCSP System Administrator Guide - 7.3 and 7.4   
108   
ECS upgrade process   
A specific version of ECS is designed to work only with a specific version of EPC Server. Each CCSP   
release is paired with a corresponding EPC Server release.   
When upgrading a CCSP system with Elsbeth integration and ECS:   
1.   
Perform the CCSP and Elsbeth upgrade in any order.   
2.   
Do one of the following:   
o   
If there are changes specific to the CCSP/EPC version being upgraded that require tenant   
mapping reconfiguration (for details, refer to the specific release note), reconfigure   
tenant mappings via the ecs.exe command line as follows:   
"ecs --configure -map [other mandatory parameters]")   
Note that if “-pms” flag is omitted the tenant mapping status is   
automatically set to “resumed“   
o   
If there no tenant mapping changes required, just resume the tenant mappings via the   
ecs.exe command line as follows:   
"ecs.exe --administer -texc --resume [other mandatory parameters]"   
3.   
Restart ECS services.

CCSP System Administrator Guide - 7.3 and 7.4   
109   
14: Configure the TWS   
The Teams Web Service (TWS) is a self-hosted web application that runs as a windows service. You   
configure TWS by editing the appsettings.json file located in the installation folder of the TWS.   
Note   
After any change in the configuration file except SyslogLevel, you must restart the service for   
changes to take effect.   
   
The configuration file contains the following configuration keys of interest:   
•   
SyslogLevel — the level of logging done by TWS in SysLog. The applicable values are the   
standard CCSP log level values. The default is 7 (Info). This value can be changed without   
having to restart the application.   
•   
TraceLogLevel — the level of the Windows logging output that can be viewed using   
DebugView The default is 1 (Emergency).   
•   
https\_port — the HTTPS port where the service is listening.   
•   
default\_port — the HTTP port where the service is listening.   
•   
Certificate   
o   
Store — the name of the store from which to get the root certificate needed for HTTPS.   
The default is My.   
o   
StoreLocation — the location of the certificate store. The default is LocalMachine.   
o   
StoreSubject — the subject name of the certificate. The default is the FQDN of the local   
machine.   
•   
Token   
o   
OAuthSecretKey — the secret key of CCSP’s AuthServer or the secret key of the token   
issuer when using an issuer different from CCSP’s AuthServer.   
o   
Issuer — the token issuer when using an issuer different from CCSP’s AuthServer.   
o   
Audience — the intended recipient (token audience) when the token issuer is different   
from CCSP’s AuthServer.   
•   
Logging:LogLevel:Microsoft — the ASP.NET Core Logging level. Possible values are:   
Trace, Debug, Information, Warning, Error, Critical and None. The default is Warning.   
•   
CORS-Allowed — comma separated URLs allowed by CORS validation. Only requests from   
these URLs are valid when accessing TWS. For example:   
http://aps1.pj16.loc,https://aps1.pj16.loc,http://ccsp1.pj16.loc,https://ccsp1.pj16.loc.   
•   
MaxUsersPerSubscription — the maximum number of Teams users per subscription. The   
Graph API limits this value to 650. The default is 650.   
•   
SubscriptionDuration — the duration of a subscription in minutes. The Graph API limits this   
value to 60. The TWS renews the subscriptions before they expire. The default is 60.   
•   
ConfigNotificationCheckIntervalSeconds — the interval in seconds to check for changes in   
the list of CCSP agents. The default is 600.   
•   
ConnectionStrings:CCSP\_DB — the connection string to the CCSP configuration database.

CCSP System Administrator Guide - 7.3 and 7.4   
110   
•   
PublicEndpoint — the endpoint to receive presence change notifications from the MS Teams   
Servers.   
•   
TenantsFilter — the filter to limit the CCSP tenants that TWS monitors. This filter is a comma   
separated list of tenant UPN suffixes, for example: “t1.loc, t2.loc”. By default, this property is   
empty, meaning that all configured tenants are monitored.   
•   
Proxy   
o   
ProxyAddress – the address of the proxy, for example:   
http://proxy.server:808   
o   
ProxyUser – the user to log in to the proxy   
o   
ProxyPassword – the password to log in to the proxy

# Contact Center: Service Provider Prescriptive Architecture Version 7.3 and 7.4

CCSP-PAG-73/74-R7-06/2024   
Contents   
1: Intro .............................................................................................................................................6   
What’s new ................................................................................................................................................. 6   
Definitions .................................................................................................................................................. 6   
Legal disclaimer ......................................................................................................................................... 8   
Support ....................................................................................................................................................... 8   
2: Overview .....................................................................................................................................9   
CCSP overview .......................................................................................................................................... 10   
Requirements ........................................................................................................................................... 10   
Fundamentals .......................................................................................................................................... 11   
Microsoft technologies .................................................................................................................. 12   
Overview of deployment strategies ..................................................................................................... 13   
Hosted platform ...................................................................................................................................... 13   
Single site — Type 1 ........................................................................................................................ 13   
Distributed — Type 2 ...................................................................................................................... 14   
3: Redundancy ............................................................................................................................. 15   
N+1 ............................................................................................................................................................. 18   
Round robin DNS ............................................................................................................................ 20   
Hardware load balancing .............................................................................................................. 21   
Gateway/SIP trunk load balancing ............................................................................................... 22   
SIP Proxy load balancing ............................................................................................................... 23   
SoftSwitch load balancing ............................................................................................................. 24   
High availability ....................................................................................................................................... 25   
Microsoft clustering ........................................................................................................................ 25   
SQL Server Always-On Availability Groups .................................................................................. 26   
SQL mirroring .................................................................................................................................. 27   
Replication ....................................................................................................................................... 27   
CCSP proprietary ............................................................................................................................. 27   
Network equipment redundancy .......................................................................................................... 28   
Firewalls ............................................................................................................................................ 28   
Switches ............................................................................................................................................ 29   
Routers ............................................................................................................................................. 29   
Application switch ........................................................................................................................... 30   
Best practice .................................................................................................................................... 30   
Disaster recovery solution ..................................................................................................................... 36   
Two-site deployment ...................................................................................................................... 36   
Bandwidth considerations ............................................................................................................. 37   
Failover mechanisms and triggers ............................................................................................... 38

CCSP Prescriptive Architecture - 7.3 and 7.4   
3   
Remote site survivability ................................................................................................................ 38   
4: Network/firewall infrastructure ............................................................................................. 39   
VLANs and connectivity .......................................................................................................................... 39   
IPC network leg ............................................................................................................................... 40   
Internal network leg ....................................................................................................................... 44   
External network leg ...................................................................................................................... 53   
CCSP VLAN usage ............................................................................................................................ 53   
Firewall port configuration .................................................................................................................... 54   
FW1 ................................................................................................................................................... 54   
FW2 ................................................................................................................................................... 55   
FW3 ................................................................................................................................................... 60   
Distributed architecture ......................................................................................................................... 64   
Bandwidth requirements ....................................................................................................................... 65   
5: Connectivity ............................................................................................................................. 66   
Media connectivity .................................................................................................................................. 66   
VoIP ................................................................................................................................................... 66   
VoIP media connectivity ................................................................................................................. 69   
MRCP ................................................................................................................................................. 70   
Back-end server connectivity ................................................................................................................. 71   
SQL Server ........................................................................................................................................ 71   
Messaging media and MSRs ......................................................................................................... 75   
Backup and maintenance .............................................................................................................. 79   
6: CCSP Dialer .............................................................................................................................. 80   
Elsbeth mode ........................................................................................................................................... 80   
Setup ................................................................................................................................................. 81   
Dialer ports ...................................................................................................................................... 83   
Sytel mode ................................................................................................................................................ 85   
Setup ................................................................................................................................................. 85   
Dialer ports ...................................................................................................................................... 86   
7: Virtualization ............................................................................................................................ 87   
8: IP multimedia subsystem ....................................................................................................... 88   
Overview ................................................................................................................................................... 88   
CCSP integration ...................................................................................................................................... 90   
Interfaces ......................................................................................................................................... 90   
Setup ................................................................................................................................................. 91

CCSP Prescriptive Architecture - 7.3 and 7.4   
4   
Appendix A: System environment configuration ..................................................................... 92   
Active Directory and domain controller ............................................................................................... 92   
MSDCS section in DNS server ....................................................................................................... 93   
Global catalog .................................................................................................................................. 93   
Replication ....................................................................................................................................... 93   
Flexible Single-Master Operation roles ....................................................................................... 95   
DNS ............................................................................................................................................................ 99   
DNS architecture ............................................................................................................................. 99   
Internal CCSP DNS servers − setup and zone replication ....................................................... 101   
Hard drive configuration ...................................................................................................................... 102   
SQL .................................................................................................................................................. 103   
Server requirements ............................................................................................................................. 104   
Client requirements .............................................................................................................................. 105   
Appendix B: SQL Server Always-On and mirroring ................................................................ 106   
Synchronous mode ............................................................................................................................... 106   
Asynchronous mode ............................................................................................................................. 108   
Disk image .............................................................................................................................................. 110   
Blade servers, SAN configuration and redundancy ......................................................................... 110   
Voice quality ........................................................................................................................................... 110   
Appendix C: QoS ........................................................................................................................ 111   
DiffServ .................................................................................................................................................... 111   
Microsoft QoS Packet Scheduler ......................................................................................................... 113   
Packet scheduler ........................................................................................................................... 113   
Packet shaping .............................................................................................................................. 113   
MOS ......................................................................................................................................................... 114   
Jitter ......................................................................................................................................................... 114   
Echo................................................................................................................................................. 115   
Packet loss .............................................................................................................................................. 115   
Packet delay ........................................................................................................................................... 117   
Appendix D: Component security settings ............................................................................. 118   
Svccosmocall user ................................................................................................................................. 118   
Dedicated users ..................................................................................................................................... 118   
System services security change mapping ........................................................................................ 119   
System web services security change mapping ............................................................................... 120   
Appendix E: CCSP-Elsbeth integration .................................................................................... 121   
Elsbeth CommunicationManager (ECM) ................................................................................... 121   
Elsbeth PowerContact (EPC) ........................................................................................................ 122

CCSP Prescriptive Architecture - 7.3 and 7.4   
5   
Elsbeth Configuration Synchronization Server (ECS) .............................................................. 123   
Elsbeth Agent Gadget service (EPC Gadget)............................................................................. 123   
How to plan an Elsbeth system for CCSP ........................................................................................... 124   
Definitions ...................................................................................................................................... 124   
System requirements ................................................................................................................... 125   
System sizing ................................................................................................................................. 126

CCSP Prescriptive Architecture - 7.3 and 7.4   
6   
1: Intro   
This document is for system architects and the CCSP system administrator.   
What’s new   
Version 7.4   
•   
Requirements — updated for Elsbeth 9.x.   
•   
Dialer ports — updated the ports used by the EPC Gadget and EPC Server.   
•   
Appendix E: CCSP-Elsbeth integration — updated for Elsbeth 9.x. Added the EPC Gadget (from   
Elsbeth 9.x only).   
Definitions   
Term   
Definition   
Adapter   
Adapter service is a proxy between the ECM Gate and CCSP systems.   
ADP   
Advanced Dial Plan — backward-compatible replacement for the legacy   
dialing plan component.   
AIS   
Agent Interaction Server — handles all the platform’s agent connections.   
ALS   
Agent List Server — provides agent state and configuration data for A2A calls   
and transfers, the Supervisor window, and assist functions.   
APS   
Agent Proxy Server — proxy server between CCSP UI application and the AIS.   
Auth Server   
Creates tokens used to authenticate users in Replay, CCSP Admin and CCSP   
UI.   
CCSP Admin   
Queries and manipulates the configuration database. The component consist   
of a backend API and a client-side application hosted in the IIS.   
CFM   
Configuration Manager — manages the entire CCSP contact center   
configuration and connects to the SQL database. Publishes any changes to all   
other services.   
CM   
Campaign Manager — management interface to the Dialer for Sytel.   
CMS   
Call Management Server — call routing engine of the CCSP platform. CMS   
distributes inbound calls to the individual agents.

CCSP Prescriptive Architecture - 7.3 and 7.4   
7   
Term   
Definition   
CCSP Publisher   
Manages the versions of the components and performs remote updates.   
Dialer   
CCSP Dialer for Sytel/Elsbeth — dialing systems that provide an array of   
campaign management tools.   
DTS   
Data Transport Server — converts audio stream and pushes it to the   
recording database.   
DPS   
Device Proxy Server — registrar and SIP Proxy between agent IP phones and   
the CCSP platform.   
ECM   
Elsbeth CommunicationManager — implements the dialer algorithm and   
manages the call state.   
ECM Gate   
ECM Gate service — an interface to the media layer.   
Elsbeth   
A brand of dialer supported by the CCSP platform.   
EPC   
Elsbeth Power Contact — provides agent management (agent client) and   
the campaign manager.   
ICS   
Internet Connection Server — handles all incoming web chat calls.   
IVVR   
Interactive Voice and Video Response — plays videos and prompts on   
demand as requested by the VCS IVR scripts.   
KCO-DPS   
Keep Connection Open-DPS — keeps SIP alternative device connected for the   
duration of the agent logon.   
LCS   
License Control Server — enforces system-wide license control.   
LS   
Logging Server — inserts information into the historical database.   
MCS   
Message Connection Server — handles all non-persistent media calls.   
MS   
Messaging Server — handles MSR’s adaptation to the CCSP platform   
MSR   
Message Storage Repository   
MSR DB   
General term for referencing database type that stores data for emails,   
callbacks, and voicemails.   
Active MSR DB   
Database type that stores data for new and ongoing emails, callbacks, and   
voicemails. Also called the tenant messaging database.   
Archive MSR DB   
Database type that stores data for handled/rejected/deleted emails,   
callbacks, and voicemails. Also called the tenant messaging archive database.

CCSP Prescriptive Architecture - 7.3 and 7.4   
8   
Term   
Definition   
PDS   
Predictive Dialer Server — service connects the outbound dialer and the CCSP   
platform.   
Replay   
Queries the recording database. The component consist of a backend API   
and a client-side application hosted in the IIS.   
RTP Relay   
Real-Time Transport Protocol Relay Server — handles the RTP media stream   
negotiated in the VoIP signaling.   
SRS   
Screen Recording Server — component that receives and manages agent   
screen recording and sends it to the DTS server.   
SS   
Statistics Server — provides real-time statistics to the Real Time Reports (RTR)   
web application.   
VCS   
Voice Connection Server — VoIP interface of the platform. All incoming and   
outgoing voice calls go through the VCS. Also manages call sessions and   
handles IVR scripts.   
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 content in this document, please open a case with   
Enghouse Support.

CCSP Prescriptive Architecture - 7.3 and 7.4   
9   
2: Overview   
•   
CCSP overview   
•   
Requirements   
•   
Fundamentals   
•   
Overview of deployment strategies   
•   
Hosted platform   
   
This document provides an architectural overview to design the platform for a cloud contact system   
deployment. The Prescriptive Architecture Platform Design is the reference architecture model that   
enables customers to build scalable, reliable, secure, and manageable environments. It is intended   
primarily for customers’ system architects and IT personnel who need to prepare their budget and   
platform design for the deployment of the Contact Center: Service Provider (CCSP) platform.   
This reference model targets the following objectives:   
•   
Scalability — all components of the architecture must scale to support continuously growing   
user demand and business requirements.   
•   
Availability — no single point of failure. Components of the architecture must provide   
redundancy or functional specialization to contain faults.   
•   
Security — the architecture must provide an end-to-end security model that protects data and   
the infrastructure from malicious attack or theft.   
•   
Manageability — ease of configuration, ongoing health monitoring, and failure detection are   
vital to the goals of availability, scalability, and security. The ability to manage the solution   
must also match the planned growth of the environment.   
•   
Agility — the architecture should have the ability to quickly adapt to changes based on   
business needs. Agility is an essential aspect of maximizing the benefit to the service provider   
and its customers.   
•   
Customizability — the architecture is customizable to meet the different needs of service   
providers. It is a sound, established starting point from which to build to meet the needs of an   
organization.

CCSP Prescriptive Architecture - 7.3 and 7.4   
10   
CCSP overview   
Contact Center: Service Provider (CCSP) is a next-generation contact center platform that allows   
service providers to develop, offer, and deploy contact center services in a unique, highly flexible,   
and profitable way. Unlike conventional contact center offerings that require complex and   
expensive premise-based equipment, CCSP is cloud-based by design and requires agents and   
supervisors to have only a computer, a browser, a headset, and a connection to an IP network. IP   
phones (SIP) and hybrid (PSTN) agents are also supported.   
Requirements   
The following table show the CCSP server operating system and software requirements.   
Computer   
Requirement   
CCSP Server components   
and Elsbeth server   
components   
Microsoft Windows Server 2012 R2 or 2016 — you can install   
separate components on different OS versions.   
Microsoft SQL Server 2016 SP2 or higher   
Microsoft Active Directory 2016   
Elsbeth 6.x: Microsoft .NET Framework 4.7 or later   
Elsbeth 9.x: Microsoft .Net 6.0   
If using Exchange 2010 as   
MSR   
Microsoft Exchange Server 2010 SP3 (latest rollup update)   
Microsoft Windows Server 2012 R2   
Microsoft .NET 3.5

CCSP Prescriptive Architecture - 7.3 and 7.4   
11   
The following table shows the CCSP client operating system and browser requirements.   
OS/   
Browser   
CCSP Admin   
CCSP Administrator   
(legacy)   
CCSP Dashboard   
CCSP Dashboard   
(legacy)   
CCSP Designer   
CCSP Designer   
(legacy)   
CCSP Historical   
Reports   
CCSP Replay   
CCSP UI8   
CCSP UI Admin   
Microsoft   
Windows 10   
Y   
Y   
Y   
Y4   
Y   
Y   
Y   
Y   
Y   
Y   
Microsoft   
Windows 11   
Y   
N   
Y   
N   
Y   
N   
Y   
Y   
Y   
Y   
Microsoft   
Edge   
Y   
Y1   
Y   
Y1   
Y   
Y1   
Y   
Y   
Y   
Y   
Google   
Chrome   
Y   
Y2   
Y   
Y5   
Y   
Y2   
Y   
Y   
Y   
Y   
Firefox3   
Y   
N   
Y   
N   
N   
N   
Y   
Y   
Y   
Y   
Ubuntu   
client   
Y   
N   
N6   
N   
N   
N   
N   
Y   
Y7   
Y   
1. CCSP supports the latest public version, in IE mode.   
2. CCSP supports the latest public version, with IE Tab extension.   
3. CCSP supports the latest public version.   
4. Requires .NET 4.7.   
5. Must be launched and used outside of CCSP UI.   
6. If Ubuntu client includes Firefox, it will be automatically supported.   
7. In web mode only, without Client Services. SIP devices, and screen recording are not   
available in this mode.   
8. CCSP UI was previously called TouchPoint.   
Fundamentals   
Before working with this document, it is important that you become familiar with basic Microsoft   
architecture fundamentals and further technologies. This chapter contains the basic knowledge   
that is necessary for a proper understanding of the CCSP platform. Use the information provided   
here as a starting point for further reading.

CCSP Prescriptive Architecture - 7.3 and 7.4   
12   
Microsoft technologies   
The CCSP platform uses the following Microsoft technologies.   
Microsoft Windows Server   
Microsoft Windows Server 2016 is currently the main server for the platform.   
If using Exchange 2010 as MSR, you also need Microsoft Windows Server 2012 R2 (required for   
Exchange 2010).   
Microsoft Active Directory /DC concepts   
The CCSP platform relies on Microsoft Active Directory (AD) services to manage tenants and user   
identities, to authenticate agents, and to define access rules and permissions within the CCSP   
domain.   
Microsoft IIS   
Internet Information Server (IIS) is the web server that holds the web applications, and delivers and   
executes the applications running on top of it.   
Microsoft Exchange   
A Message Storage Repository (MSR) is required for the messaging media (email, voicemail, and   
callbacks).   
In CCSP 7.3, the supported MSRs are:   
•   
Tenant Messaging Database (MSR DB) - recommended   
•   
Tenant Exchange Database for Microsoft Exchange   
   
For details, see Messaging media and MSRs.   
Microsoft SQL Server   
SQL Server is one of the key third-party components of the cloud contact platform. It provides   
repositories of the configuration, messaging media, call events, and call recordings.   
SQL Server supports a wide variety of redundancy. For details, see 3: Redundancy.   
Microsoft Reporting Services   
Microsoft Reporting Services (MSRS) are part of SQL Server and are used to present the historical   
statistics of the contact center for each tenant.   
It also contains the call and licensing statistics of the complete platform from the landlord level.

CCSP Prescriptive Architecture - 7.3 and 7.4   
13   
Overview of deployment strategies   
The components can be logically divided into two groups:   
•   
Front-end (F/E) components — contain all the front-end services that provide interfaces to   
administrate, monitor and use the software. Front-end components are typically deployed   
inside a demilitarized zone (DMZ).   
They can be shared between all tenants or can be assigned as dedicated.   
•   
Back-end (B/E) components — contain all the internal logic, call routing functionality, event   
logging, and data storage units. Back-end components are typically deployed behind a DMZ   
and are shared between all tenants.   
   
As shown in the following sections, these logical groups can be scaled, distributed and replicated in   
different patterns, allowing customers and service providers to meet their different needs.   
The email system in previous CCSP versions used the Microsoft Exchange FE/BE solution for that   
part of the system separation.   
From CCSP 7.x onward, the entire FE/BE system separation no longer relies on the Microsoft FE/BE   
solution. Instead, the system FE/BE solution relies on agent/system basic separation handled by   
system proxy services.   
Hosted platform   
Single site — Type 1 is easier to maintain and to administrate. The single hosted platform uses   
cloud concepts to provide the service.   
Distributed — Type 2 is optimized for different customers. We recommend using the distributed   
hosted platform when one or more customer has a large amount of usage and/or the call delivery   
needs to be handled locally.   
Single site — Type 1   
This is the most basic deployment scenario. The CCSP software is inside its network and is used by   
one or more customers. Each customer becomes a tenant on the CCSP platform, essentially sharing   
resources with other tenants or using its own dedicated resources (optional).

CCSP Prescriptive Architecture - 7.3 and 7.4   
14   
   
   
   
Figure 1: CCSP hosted single platform   
Distributed — Type 2   
The distributed hosted platform involves replication of all front-end components across multiple,   
geographically-separated sites for local access. Back-end components remain in one or two central   
locations, typically sharing identical configuration for business-continuity purposes and   
synchronized via a dedicated WAN link.   
   
Figure 2: CCSP hosted distributed platform

CCSP Prescriptive Architecture - 7.3 and 7.4   
15   
3: Redundancy   
•   
N+1   
•   
High availability   
•   
Network equipment redundancy   
•   
Disaster recovery solution   
   
All CCSP components benefit from carrier-grade high availability and redundancy.   
The following table provides a summary of the redundancy types that are offered depending on the   
component.   
CCSP components   
   
High availability   
Notes   
With load   
balancing   
(N+1)   
With   
clustering   
With   
Always   
On   
Groups   
With   
mirroring   
Adapter   
Y   
N   
N   
N   
Hosted within ECM   
Gate   
Advanced Dial Plan   
(ADP)   
Y\*   
N   
N   
N   
\* RR DNS is not   
required.   
Agent Interaction   
Server (AIS)   
Y   
N   
N   
N   
   
Agent List Server   
(ALS)   
Y   
N   
N   
N   
   
Agent Proxy Server   
(APS)   
Y   
N   
N   
N   
   
Auth Server   
Y   
N   
N   
N   
   
Call Management   
Server (CMS)   
N   
Y   
N   
N   
Proprietary HA   
solution   
Campaign Manager   
(CM)   
N   
Cold/warm   
standby   
N   
N   
   
CCSP Admin   
Y   
N   
N   
N

CCSP Prescriptive Architecture - 7.3 and 7.4   
16   
   
High availability   
Notes   
With load   
balancing   
(N+1)   
With   
clustering   
With   
Always   
On   
Groups   
With   
mirroring   
Configuration   
Manager (CFM)   
Y   
N   
Y   
Y   
   
Data Transfer Server   
(DTS)   
Y   
N   
Y   
Y   
   
Device Proxy Server   
(DPS)   
Y   
N   
N   
N   
   
Elsbeth   
Communication   
Manager (ECM)   
N   
N   
N   
N   
   
ECM Gate   
Y   
N   
N   
N   
   
Elsbeth   
PowerContact (EPC)   
N   
N   
N   
N   
   
Interactive Voice-  
Video Response   
(IVVR)   
Y   
N   
N   
N   
   
Internet Connection   
Server (ICS)   
Y   
N   
N   
N   
   
License Control   
Server (LCS)   
Y\*   
N   
N   
N   
\*RR DNS is not   
required   
Logging Server (LS)   
Y   
N   
Y   
Y   
   
Messaging   
Connection Server   
(MCS)   
Y   
N   
N   
N   
   
Messaging Server   
(MS)   
N   
N   
N   
N   
   
Predictive Dialing   
Server (PDS)   
Per   
Tenant   
Cold/warm   
standby   
N   
N   
   
Real-Time Protocol   
Server (RTPS)   
Y   
N   
N   
N

CCSP Prescriptive Architecture - 7.3 and 7.4   
17   
   
High availability   
Notes   
With load   
balancing   
(N+1)   
With   
clustering   
With   
Always   
On   
Groups   
With   
mirroring   
Real Time Report   
Server (RTR)   
Y   
N   
N   
N   
   
Replay   
Y   
N   
N   
N   
   
Statistics Server (SS)   
N   
Y   
N   
N   
Proprietary HA   
solution   
Sytel Dialer   
N   
Cold/warm   
standby   
N   
N   
   
Voice/Video   
Connection Server   
(VCS)   
Y   
N   
N   
N   
External VLAN   
interface only1   
1. NetBIOS on the external NIC must be disabled.   
Microsoft components   
   
High availability   
Notes   
With load   
balancing   
(N+1)   
With   
clustering   
With   
Always   
On   
Groups   
With   
mirroring   
Active Directory   
(AD/DC)   
Y   
N   
N   
N   
Multimaster   
Domain Name Server   
(DNS)   
Y   
N   
N   
N   
AD integrated zones   
Exchange Server   
N   
Y   
   
   
   
Internet Information   
Server (IIS)   
Y   
N   
N   
N   
N   
SQL Server   
N   
Y   
Y   
Y   
N

CCSP Prescriptive Architecture - 7.3 and 7.4   
18   
N+1   
N+1 means that components are redundant and load balanced. The 1 stands for an extra server   
added to the load-balanced group so capacity is preserved if any server crashes.   
Having an extra server is also useful for software upgrades, as there is no effect on performance or   
the capacity of the platform when servers are temporarily taken out of production, one by one, to   
be upgraded.   
Load balancing solutions are described below.   
The following table provides a comparison of different load balancing technologies.   
Feature   
RR DNS   
Hardware   
load balancer   
Gatekeeper   
SIP Proxy or   
SIP server   
Gateway load   
balancer   
SIP + RTP compliant (VCS)   
Y1   
Y2   
Y   
Y   
Automatically removes a   
failed node from the cluster   
N   
Y   
Y3   
Y   
Works in a LAN environment   
Y   
Y   
Y   
Y   
Works in a WAN environment   
Y   
Y4   
Y   
Y   
Does not require static ARP   
table entries in switches and   
routers   
Y   
Y   
Y   
Y   
SSL offloading and L7 load   
balancing (HTTP/HTTPS)   
N   
Y   
N   
N   
Does not increase LAN   
bandwidth   
Y   
Y5   
Y   
Y   
Dynamically detects server   
load   
N   
Y   
N   
N   
Dynamically detects   
application sluggishness or   
unresponsiveness   
N   
Y   
Y6   
Y   
Allows for zero downtime   
upgrades   
Y   
Y   
Y   
Y   
Speed of packet routing   
Medium   
Fast   
Medium   
Fast   
OSI layer   
L3   
L3   
L7   
L7   
Call load awareness   
N   
Y   
Y   
N

CCSP Prescriptive Architecture - 7.3 and 7.4   
19   
Feature   
RR DNS   
Hardware   
load balancer   
Gatekeeper   
SIP Proxy or   
SIP server   
Gateway load   
balancer   
1. A record timeout should be set as low as possible, for example, to 1 second.   
2. Can only be used on the internal interface. Must be configured with client IP stickiness.   
Cannot be used between gateway and VCS.   
3. Can only be used if it is SIP/SDP-aware and can also redirect RTP media to the same VCS.   
4. Depends on the models. Sometimes port forwarding must be configured on inter-domain   
routers.   
5. Heartbeats can be sent on a separate connection.   
6. Depends if vendor solution implements some kind of detection mechanism, such as ping, SIP   
call, or establishment tracking.   
   
It is important to understand that different load balancing and keep alive mechanism can and   
should be used for the several services.   
In general this should be a combination of hardware load balancing and SIP gateway balancing.   
The hardware load balancers should be inside the IPC and the internal network.

CCSP Prescriptive Architecture - 7.3 and 7.4   
20   
Round robin DNS   
Round robin DNS (RR DNS) means configuring multiple Host A records in DNS with the same FQDN   
name but different IP addresses. Each IP address belongs to a different server in the N+1 load-  
balanced group. Each time a query is made for this A record, the DNS server returns a different IP   
address from the group, in round-robin fashion.   
DNS  
DNS  
DNS  
DNS   
Server  
Connection  
Server  
Server  
Server  
Client  
Client  
Client  
   
Figure 3: Round robin DNS   
Note   
•   
The TTL on each of these entries needs to be set to 1 second to expire the DNS entry in the   
client’s BIND table, and to force a new DNS query.   
•   
RR DNS does not discover a failed node and continues to resolve the IP address of the failed   
component, which can cause routing delays until the SIP connection retries time-out. To   
resolve this, ensure your SNMP monitor software can automatically remove stale DNS   
entries.   
•   
As DNS RR is the most simple way to create load balancing and is free, we recommend it for   
internal platform load balancing for IVR scripts, web servers, and AIS servers.   
•   
We recommend this solution for small-medium platforms of up to 20 servers.   
•   
We do not recommend DNS RR for SIP load balancing.

CCSP Prescriptive Architecture - 7.3 and 7.4   
21   
Hardware load balancing   
Hardware load-balancing solutions are very robust and we recommend them for large networks   
with high throughput. Besides having a lot of built-in features such as dynamic detection of node   
failure, they benefit from fast processing times and low latency.   
Hardware load balancers can be used in all three networks for different purposes:   
•   
Inside the IPC network — used by the VCS servers to reach the scripts.   
•   
On the external network — used for monitoring and choosing the right VCS to connect the   
voice call to.   
•   
In the internal network — used to create, establish and hold the agent session agent on the   
AIS/APS server to which the agent is connected.   
   
Figure 4: Hardware load balancing   
Note   
•   
You load balance the DPS in the same way as the APS.   
•   
When selecting a hardware load-balancer for CCSP, make sure that it is SIP-aware so that IP   
stickiness is also preserved for RTP stream. IP stickiness should also be configurable for the   
agents connection to the APS.   
•   
Always use two hardware load balancers for high availability. If one fails, the other should   
transparently take over, provided both hardware load balancers are in the same LAN.

CCSP Prescriptive Architecture - 7.3 and 7.4   
22   
Gateway/SIP trunk load balancing   
This configuration is the most simple and the preferred one for VoIP load balancing. Support for   
load balancing is integrated inside each gateway. The gateway itself is responsible for placing calls   
across multiple VCSs. The only drawback in this approach is that each gateway is not aware of the   
total call load on the VCS, so it only considers those calls that it has placed. Other solutions like SIP   
Proxy or SoftSwitch keep track of all calls.   
Gateways use a keep alive mechanism to detect a failed VCS and remove it from the load-balanced   
group.   
VCS  
VCS  
VCS  
Distribute Calls  
GW  
   
Figure 5 Gateway load balancing

CCSP Prescriptive Architecture - 7.3 and 7.4   
23   
SIP Proxy load balancing   
A SIP Proxy or Registrar can be configured to act as a round robin load-balancing device. All VCSs   
register to it and gateways are configured with a single endpoint IP such that all inbound calls are   
sent to that IP via the SIP Proxy. The SIP Proxy in turn maps this IP to one of the many registration   
addresses it has received. When a VCS out-dials, the SIP Proxy can provide the reverse action of   
load-balancing gateways. A SIP Proxy normally removes a stale IP address after a failed call   
attempt.   
   
Figure 6: SIP Proxy load balancing

CCSP Prescriptive Architecture - 7.3 and 7.4   
24   
SoftSwitch load balancing   
SoftSwitches are used to provide flexible and intelligent call routing in a VoIP network. Using a   
SoftSwitch to load balance VCS is very similar in functionality to using a gatekeeper. All calls are   
sent to the SoftSwitch, which redirects the call using a SIP 3xx REDIRECT command. Each redirect   
contains a different VCS IP address. SoftSwitches are not aware of endpoint failure, although some   
are equipped with ping or even application-level testing capabilities to monitor endpoints.   
   
Figure 7: SoftSwitch load balancing

CCSP Prescriptive Architecture - 7.3 and 7.4   
25   
High availability   
High availability means having a two or more servers configured with a single primary node, and   
one or more secondary nodes. Only the primary node is active and accepting connections, while the   
others, known as secondary nodes, are all in standby. Should the primary node fail, one of the   
standby nodes immediately takes over in a seamless manner for the connected clients, and   
assumes the role of the new primary node for the system. There are many ways of achieving high   
availability, such as clustering, Always-On Availability Groups, mirroring, or through CCSP’s   
proprietary technology.   
Microsoft clustering   
Clustering allows one or multiple services to run in an active-standby formation on two or more (up   
to eight) cluster nodes. Should the active node fail, the clustered services automatically start on the   
standby node, and the standby node becomes active. The two nodes are connected by an interface   
called heartbeat, which monitors server and service health status.   
The cluster shares a single virtual IP between both nodes, which is what other clients in the   
network see. Both nodes must therefore be on the same LAN segment. To a client, the service   
running on a cluster appears as if it is the only instance.   
CCSP uses clusters for high availability of SQL.   
   
Figure 8: Microsoft clustering   
The cluster requires a shared storage with both nodes connected to the same disk array, essentially   
pointing to the same data. When the active node fails, the standby is promoted and receives read-  
write access to the same data. The cluster software uses the Quorum to store cluster configuration   
and keep service state.   
In previous CCSP versions, Exchange servers deployment also used clusters. From version 7.x   
onward, CCSP uses Microsoft Exchange 2010 and no longer needs to provide a high-availability   
solution for working with Exchange. This is because Exchange 2010 encapsulates clustering and   
other high availability mechanisms within its architecture in a way that makes it transparent to   
systems integrating with it.

CCSP Prescriptive Architecture - 7.3 and 7.4   
26   
For clustering, CCSP requires SQL Server 2016 SP1 or higher. We recommend using the Enterprise   
Edition, otherwise you are limited to only two failover cluster nodes.   
For detailed configuration information, see Appendix A: System environment configuration.   
SQL Server Always-On Availability Groups   
The SQL Server Always-On Availability Groups feature is a high-availability and disaster-recovery   
solution that provides an enterprise-level alternative to database mirroring.   
Availability Groups allows for up to two secondary copies running in synchronous mode and up to   
two secondary copies running in asynchronous mode. Mirroring only allows for a single mirror   
copy that runs in either synchronous or asynchronous mode.   
Availability Groups uses the newer Windows Server Failover Clustering services (WSFC). Unlike   
regular clustering solutions with SQL Server that work against the server as a whole, Availability   
Groups work with groups of databases much like mirroring works with individual databases.   
You can configure different groups to provide the required failover control. Failure of an Availability   
Group only triggers its own failover to one of the secondary nodes, while all other databases or   
groups remain active on the existing node.   
CCSP requires either the Standard or Enterprise editions of SQL Server 2016 to provide the Always-  
On capability.   
   
Note   
•   
No shared storage is required. Each node has its own copy of the databases in any   
Availability Groups, so each node added to the group increases the storage space required   
by a factor of 1.   
•   
Some of the secondary nodes can be configured for read-only access, making them good for   
running reports against.   
   
Availability Groups can be merged with standard SQL Server clustering that uses WSFC. The   
following table provides a comparison.   
   
SQL Server Failover Clustering   
Instance nodes (FCI)   
Availability Group nodes   
Uses WSFC cluster   
Yes   
Yes   
Protection level   
Instance   
Database   
Storage type   
Shared   
Non-shared1   
Storage solutions   
Direct attached, SAN, mount points,   
SMB   
Depends on node type   
Readable secondary   
nodes   
No2   
Yes

CCSP Prescriptive Architecture - 7.3 and 7.4   
27   
   
SQL Server Failover Clustering   
Instance nodes (FCI)   
Availability Group nodes   
Applicable failover policy   
settings   
•   
WSFC quorum   
•   
FCI-specific   
•   
Availability group settings3   
•   
WSFC quorum   
•   
Availability group   
settings   
Failed-over resources   
Server, instance, and database   
Database only   
1. Although the nodes in an availability group do not share storage, if a node is also part of an   
FCI, that node requires a shared storage solution as required by that FCI. That storage   
solution is shared only by nodes within the FCI and not between replicas of the availability   
group.   
2. When running an Availability Group on top of SQL Server Failover Clustering, when an FCI-  
hosted database on the active FCI node belongs to an availability group, if the local replica is   
running as a readable secondary replica, the database is readable.   
3. Failover policy settings for the availability group apply to all replicas, whether it is hosted in a   
standalone instance or an FCI instance.   
   
For a full discussion of Availability Groups and SQL Server Failover Clustering in SQL Server, refer to   
https://msdn.microsoft.com/en-us/library/ms190202(v=sql.110).aspx.   
For detailed information about CCSP works with availability groups and how they are supported   
and configured from a CCSP perspective, refer to the CCSP Always On Availability Groups document.   
SQL mirroring   
Microsoft has moved database mirroring to maintenance mode. This means that while it works in   
all current versions of SQL Server, it may be removed as a feature in future versions. Therefore, it is   
no longer recommended to deploy systems using this feature.   
Replication   
CCSP accepts different kinds of replication solutions, provided the replication solution includes all   
available functionality as its source.   
Replication is used mainly to provide high availability across different networks without using a   
cluster. CCSP addresses each of the replications without knowing if it is the original copy or the   
replication.   
CCSP proprietary   
CCSP can also achieve high availability through a proprietary mechanism supported by its   
architecture, called Clusterless.   
This high-availability architecture assumes any pair of servers can be handled based on heartbeat.   
The mechanism includes a witness handled by the configuration database via the CFM load-  
balanced server group.

CCSP Prescriptive Architecture - 7.3 and 7.4   
28   
The servers in the pair each start with an initial role (active or passive) and, based on the witness   
decision, exchange their role. The configuration layer monitors the active/passive status between   
the services handled under that solution.   
If a manual or automatic failover occurs, that same architecture layer manages propagation of the   
failover event to other CCSP components as part of its addressing awareness role.   
CCSP has two types of services that use this type of high availability solution and can be configured   
for it:   
•   
Call Matching Server (CMS)   
•   
Statistics Server (SS)   
Network equipment redundancy   
Various solutions and combinations are possible, depending on budget and fault tolerance. We   
recommend using network equipment redundancy to maintain a network architecture with   
sophisticated load balancing and without any single point of failure.   
The solution should support the following types of teaming:   
•   
Automatic   
•   
Dual-channel   
•   
802.3ad dynamic   
•   
SLB   
•   
TLB   
•   
 NFT   
Firewalls   
When deploying redundant firewalls, make sure that all packet streams within a session are sticky   
based on source IP and/or protocol and/or ports. Firewalls are stateful, therefore if not well   
configured, splitting a single data stream (single session) between two or more firewall devices   
might result in dropped sessions by the firewall.   
Firewalls should have good high availability functionality. They should support active/active and/or   
active/passive configurations, make a seamless failover between the firewalls and keep the passive   
firewall updated with the current session’s state. Failover should be done under one second.   
We recommend working with clustered firewalls in active/passive mode to keep the configuration as   
simple as possible. Active/active architectures might result in dropped packets when using VoIP.   
We recommend choosing firewalls with dual LAN, WAN, and DMZ links for full redundancy.   
You must choose a FW2 that is SIP/H.323-aware, and has good VoIP and NAT capabilities –  
Application Layer Gateway (ALG).

CCSP Prescriptive Architecture - 7.3 and 7.4   
29   
Switches   
Always use switches equipped with 100 MB interfaces and above. We recommend at least 1 GB   
interfaces for IPC VLAN and External VLAN.   
All switches must have the ability to configure port mirroring (analyzer port).   
All switches should support QoS for prioritization of network traffic according to at least the   
source/destination IP address and/or DiffServ.   
If using Network Load Balancing Service (NLBS) in multicast mode (only for VCS external load   
balancer from gateway), consider support for IGMP snooping on the VCS switches.   
We recommend that all switches support Rapid Spanning Tree (RSTP) IEEE 802.1w and be configured   
accordingly.   
All platform switches should support 802.3ad — dynamic dual channel load balancing (also referred   
to as dynamic dual channel teaming), switch-assisted dual channel load balancing (also referred to   
as dual channel teaming) and the advanced teaming features of active path failover and fast path   
failover.   
We highly recommend using some version of 802.3ad, 802.1s switch fabric redundancy, Virtual   
Switch Redundancy Protocol (VSRP), either active/active or active/passive. This eliminates the   
dependence on First Hop Redundancy Protocols (FHRP) and Spanning Tree Protocol. This   
technology delivers deterministic, sub-200 millisecond layer 2 link recovery through inter-chassis   
stateful failovers and the predictable resilience of Etherchannel. Each brand has its own specific   
protocol name that is based on VRRP, for example, Nortel Split MLT and Cisco VSS.   
Use 802.1q (VLAN Tagging) to achieve full redundancy, for inbound and outbound traffic with   
teamed network cards on all servers.   
If necessary, you can use two layers of switches. The first layer is directly connected to the servers’   
teamed NICs. The second layer aggregates all traffic from the first layer switches and also other   
traffic towards the firewalls and the WAN. The customer should verify that all switches are not   
creating a single point of failure.   
Caution   
Never use network hubs.   
Routers   
Routers should support VRRP (RFC 3768) for redundancy and load balancing in active/passive and   
active/active modes (multiple virtual router support). Using VRRP, you can define several routers as   
one single virtual router (cluster-like) even in different WAN sites over MPLS or Ethernet. As such,   
the failover between routers is seamless and routing redundancy is achieved.   
We recommend using two redundant networks between the contact centers and the platform   
(WAN), if possible from two different providers. The two networks can be active/passive or   
active/standby in both directions, depending on the routing configuration. You can give one   
network priority over the other or, if using a load balancer, use it to load balance the routers and   
therefore the networks.

CCSP Prescriptive Architecture - 7.3 and 7.4   
30   
Application switch   
The application switch (hardware load balancer) can be configured for load balancing in multiple   
VLANs at a time.   
Incoming packets destined for a service are sent to the IP address of the hardware load balancer.   
Outside world connections should all reach the load balancer IP with no direct access to the   
servers.   
In some cases, such as HTTPS offloading or Layer7 load balancing, the source IP is rewritten by the   
IP of the hardware load balancer or other changes can be made to the packets according to the   
desired configuration.   
If using multiple sites for business continuity, we highly recommend using the load balancers as a   
Global traffic Manager (GTM) or Global Server Load Balancing (GSLB). The load balancer   
automatically and seamlessly directs traffic from clients to a secondary data center if failure occurs   
or load balance between two data centers. Most of the failover procedure between sites are done   
automatically by those load balancers.   
The load balancers should support the high availability mechanism in active/active and/or   
active/passive modes over WAN.   
Best practice   
The best practice solutions are as follows.   
DMZ and servers redundancy − full   
The following table lists the advantages and disadvantages.   
Advantages   
Disadvantages   
Outbound traffic from each server is redundant   
Maintenance   
Inbound traffic to the servers is redundant   
Complexity   
All network equipment is active/passive or active/active   
and fully redundant   
Operation   
WAN redundancy   
   
Easy management of traffic using hardware load   
balancer   
   
   
Note   
•   
CCSP servers hosting services use NIC teaming.   
•   
Rapid STP should be configured on the switches to lower the link failover time.   
•   
VLAN trunk must be setup between switches and on all server ports that are teamed.   
•   
The load balancer can be used for several network improvements: firewalls and routers can   
be load balanced and therefore more throughput can be handled by the platform, WAN

CCSP Prescriptive Architecture - 7.3 and 7.4   
31   
redundancy (the load balancer chooses a router and a specific ISP), site failover, global load   
balancing between sites, and servers can be load balanced.   
•   
We recommend configuring the load balancer IP as the default gateway for the front-end   
servers and thus load balancing the routers and WAN connections. If a load balancer is not in   
use, configure the router’s VRRP address as the default gateway.   
   
   
Figure 9: Network and servers architecture − fully redundant

CCSP Prescriptive Architecture - 7.3 and 7.4   
32   
DMZ and servers redundancy – medium   
In the following example the network equipment is partially redundant.   
The servers are divided between two switches that are connected by VLAN trunks. If one switch is   
down, the servers connected to the other switch seamlessly continue functioning.   
The firewall and routers are redundant.   
The following table lists the advantages and disadvantages.   
Advantages   
Disadvantages   
No single points of failure   
Partial network redundancy   
Partial network redundancy   
No NIC redundancy   
Reduced costs since no load balancer and no   
expensive network equipment is required   
No traffic load balancing from and to the   
servers   
   
No load balancer   
   
No ISP redundancy

CCSP Prescriptive Architecture - 7.3 and 7.4   
33   
   
Figure 10: Network and servers architecture – medium redundancy

CCSP Prescriptive Architecture - 7.3 and 7.4   
34   
DMZ and servers redundancy − small   
For small scale platforms that are not service provider platforms, do not require high redundancy,   
and are not designed to support very high loads, to lower the costs you can design the network   
with fewer components.   
The switch must be a managed switch that supports VLANs.   
On each server, each VLAN consists of a physically separate network card and no teaming or   
trunking is used.   
You can use only two VLANs: Internal/IPC and External. IPC and Internal VLANs can be combined   
into one VLAN.   
If possible the designer should maintain the no single point of failure concept as much as possible.   
The following table lists the advantages and disadvantages.   
Advantages   
Disadvantages   
Reduced costs   
No network redundancy   
   
Many single points of failure   
   
No servers connection redundancy   
   
No load balancer   
   
Note   
We highly recommend creating some redundancy even in a small platform for only a few agents   
and maintaining the principle of no single point of failure.

CCSP Prescriptive Architecture - 7.3 and 7.4   
35   
   
Figure 11: Network and servers architecture – no redundancy

CCSP Prescriptive Architecture - 7.3 and 7.4   
36   
Disaster recovery solution   
The disaster recovery solutions are as follows.   
Two-site deployment   
CCSP fully supports disaster recovery/business continuity by using geographically dispersed   
architecture. This architecture is designed to maintain the business continuity of the CCSP platform,   
even if a full site fails or is taken down for maintenance.   
Architecture description   
The CCSP platform is built out of two active sites (active/active). Each site can contain front-end   
and/or back-end components for full redundancy between the sites. Each site can potentially   
handle 100% of the contact center’s traffic, but it can also be designed to support a reduced   
amount of traffic, for example, each site can be designed to support 75% of the full capacity of the   
system. Consequently, each site can support N+1 redundancy and high availability within the site   
itself or consider the second site as its +1 or high availability solution.   
The front-end components are load balanced between the sites, preferably using a hardware global   
server load balancer (GSLB) that manages the traffic to the sites.   
By using a GSLB, the traffic can be load balanced between the sites or, if failure occurs, be diverted   
to only one of the sites. The GSLB monitors the sites, network, and specific servers for failure and   
automatically adapts the traffic accordingly.   
Back-end services are load balanced internally in the CCSP platform and do not require external   
load balancing.   
Using this architecture, there are two modes of operation for CMS and SS, the CCSP high availability   
services:   
•   
The first mode is when the service has a single instance within each site. The redundancy for   
the service is therefore in the remote site. One site hosts the active instance and the second   
site hosts the passive instance of the specific service.   
•   
The second mode is when the service is clustered within each site (active/passive Microsoft   
Cluster), and using the internal CCSP high availability between the two clusters. One site hosts   
the active instance on the active cluster node and the second site hosts the passive instance on   
the active cluster node. If failure occurs, the failover is either within the local cluster (the active   
service stays in the same site) or failover between the sites (the active service moves to the   
other site). As a result, the sites might switch roles. The complete operation is seamless and   
does not require any manual interference.   
SQL   
For replication of SQL over different networks, we strongly recommend that for further   
information, installation and configuration advice, you read Microsoft’s SQL Server Replication   
documentation.   
from Microsoft SQL and Microsoft Exchange over different networks.

CCSP Prescriptive Architecture - 7.3 and 7.4   
37   
MSR   
For replication of Exchange over different networks, we strongly recommend that for further   
information, installation and configuration advice, you read Microsoft’s Exchange Replication   
documentation.   
CCSP services   
CCSP services are not site-aware and therefore front-end services communicate with back-end   
services in the remote site. For example, the VCS in site B sends IPC messages to a logging server   
on site A, and the AIS in site A sends configuration requests to a CFM on site B.   
We recommend that you have N+1 LCSs per site.   
Bandwidth considerations   
The bandwidth required for this kind of architecture varies according to the CCSP platform load.   
Some of the parameters that strongly affect the required bandwidth are shown in the following list:   
•   
Max simultaneous calls   
•   
Calls per second   
•   
Agent state change frequency   
•   
Max simultaneous logged in agents   
•   
Number of configuration changes   
•   
Number of received and sent emails and their size   
   
The main bandwidth users in this kind of architecture are:   
•   
SQL replication   
•   
Exchange replication   
•   
IPC communications   
•   
Database transactions:   
o   
CFM → database   
o   
LS  → database   
o   
DTS  → database   
License Control   
The LCS measures 56K bit/sec Peak and 12K bit/sec Avg. for 40 agent login request/sec. The LCS is   
not site-specific, so the APS in site A may communicate with the LCS in site B when checking license   
status. Provision sufficient WAN bandwidth for traffic between APS (Zone 1) and LCS (Zone 2) for   
this communication type.   
Minimizing the bandwidth   
For minimizing the VoIP traffic between the sites, we highly recommend that the VoIP gateways   
only use the local VCS servers. In this way, no VoIP traffic occurs between the remote sites.

CCSP Prescriptive Architecture - 7.3 and 7.4   
38   
We recommend configuring the DTS to monitor only local VCS, MCS, and ICS shared storages to   
minimize the communication between the sites.   
Chat Extension should use only local IPS services.   
APS should use only local AIS.   
Most replication applications use a compressions module between the remote replication sites   
when configured in WAN operation mode. This kind of module can significantly reduce the required   
bandwidth between the sites and we highly recommend using it if bandwidth is an issue. The third-  
party application vendor should be able to supply all the required details before the   
implementation.   
The replication software should be able to compare data and transport only the data that has   
changed. Sending bytes across a network reduces the impact on the network and eliminates the   
need for considerable large bandwidth.   
Failover mechanisms and triggers   
The failover triggers and mechanisms can be varied according to the third-party software. Each   
software supports different mechanisms, triggers, and generic applications as Exchange and SQL   
server failovers.   
Triggers can vary from specific event viewer messages, service down, server down, network issues   
or using NMS any other SNMP alert that can trigger the failover command. We recommend setting   
a minimum number of triggers to begin with and with time adding more cases.   
In general, we recommend using the Exchange and SQL cluster failover before failing over to the   
secondary site using the replication software.   
Remote site survivability   
Remote site survivability (RSS) enables the installation of back-end CCSP components in different   
geographical locations (sites). If disconnected from the rest of the platform, RSS enables each site   
to continue to work independently, without a disruption to callers.   
During failover, each server communicates only within the boundaries of the site using the   
resources configured in the remote site's properties.   
Failover to localized operation can be initiated:   
•   
Manually — the system administrator changes system state information.   
•   
Automatically — CCSP detects connection loss and performs failover.   
   
Recovery to normal mode at the remote site is initiated manually by the system administrator. After   
resolving all network issues between the remote and the central sites, including any potential DNS   
issues, all services at the remote site are stopped, log shipping from the central site's configuration   
database to the remote site's copy of the database is restored, and all services on every server at   
the remote site are restarted.

CCSP Prescriptive Architecture - 7.3 and 7.4   
39   
4: Network/firewall infrastructure   
•   
VLANs and connectivity   
•   
Firewall port configuration   
•   
Distributed architecture   
•   
Bandwidth requirements   
VLANs and connectivity   
CCSP does not allow the use of hubs in its prescriptive architecture under any circumstances, given   
the nature of traffic that flows through the networks. The typical deployment is comprised of three   
different mandatory VLANs and two optional VLANs. However not all servers are directly connected   
to them, as shown below. For details, see the diagrams in Front-end zone and Back-end zone.   
You can have either separated physical NICs for each VLAN or 2-3 teamed NICs that are configured   
for all VLANs and connected as a trunk to the switches.   
   
Figure 12: Example of VLAN connectivity (4 VLANs)

CCSP Prescriptive Architecture - 7.3 and 7.4   
40   
The CCSP VLANS are as follows:   
•   
External (VoIP only, mandatory) — connects VCSs to the PSTN gateway or to a SIP trunk.   
Connects KCO-DPS to the SIP Gateway. This VLAN is accessible from outside. Private or public   
address space (see NAT traversal considerations).   
•   
Internal (all front-end services, mandatory) — connects VCS, AIS, IIS, DPS, and internal and/or   
external agents. Is accessible from outside. Private or public address space (see NAT traversal   
considerations), unless agents are local, in which case the address space should be private.   
•   
IPC — connects all CCSP servers together.   
•   
Management and Syslog (optional, recommended) — connects all CCSP servers to a central   
Syslog server (optional) and is used for all management and system administrators remote   
connections. Only internal access. Private address space.   
•   
SQL back end (optional, for very large service provider platforms with high load) — connects   
LS, CFM, and DTR servers to the SQL database. Internal access only. Private address space.   
IPC network leg   
The IPC network leg is the network that is used exclusively for communication between the several   
CCSP services.   
CCSP services are typically deployed in two zones:   
•   
Zone 1 — a secure DMZ where front-end services reside.   
•   
Zone 2 — a secure internal zone where back-end services reside.   
   
Both zones are protected by firewalls (FWs). The following diagrams show a simplified   
representation of this dual-zone architecture.   
Front-end zone   
The front end of the platform contains all the services accessible to the platform users.   
   
Note   
Typically FW1 and FW2 perform Network Address Translation (NAT). The CCSP platform supports   
installation of front-end servers behind such NAT devices.   
However, FW3 should not be configured for dynamic NAT. Preferably, this firewall does not   
perform NAT at all. If NAT is required, only static mapping can be used.   
If FW2 performs NAT, it must be SIP-aware to allow external agents to connect to the VCSs   
behind the firewall. For details, see FW2.   
The primary reason to have SIP-aware NAT is that NAT changes the address of the SIP Agent   
reachable by VCS. During establishment of SIP session, Agent sends its internal address in SIP   
Contact header (not address exposed by firewall). Due to nature of SIP protocol, VCS later on   
sends BYE to address denoted inside Contact header. Because of this, VCS is unable to reach   
Agent when call is hung up (BYE sent from VCS).

CCSP Prescriptive Architecture - 7.3 and 7.4   
41   
   
   
Figure 13: DMZ deployment − Zone 1

CCSP Prescriptive Architecture - 7.3 and 7.4   
42   
Back-end zone   
The back-end zone contains all the services that are not accessed directly from the client side.   
Zone 2 is typically protected behind two sets of firewalls: the one in Zone 1 and the intra-zonal   
firewall. FW3 should not perform NAT and should only open CCSP ports. Zone 2 contains mission   
critical services, hence the level of protection.   
Note   
Cluster heartbeats or data mirroring must be on a separated and dedicated interface and not   
separated by a firewall.   
Clusters should be built on SANs with RAID protected disks. NAS is also possible but only for   
small contact centers of less than 100 agents, due to possible performance implications.

CCSP Prescriptive Architecture - 7.3 and 7.4   
43   
   
Figure 14: Back-end deployment − Zone 2

CCSP Prescriptive Architecture - 7.3 and 7.4   
44   
Internal network leg   
The internal network leg is mainly for connecting customers and agents media with the system.   
To work properly, the VoIP Connection services must have a connection to this network leg.   
It is also used to connect the speech server applications to the platform, for example, MRCP.   
Agent delivery options   
CCSP supports four different agent delivery mechanisms:   
•   
Direct desktop delivery   
•   
Alternate delivery (hybrid)   
•   
IP phone   
•   
Phone-only   
CCSP UI to platform connectivity   
CCSP UI connects to the platform in several ways:   
•   
Firewall to firewall VPN   
•   
Internet   
•   
Per agent VPN   
Firewall to firewall VPN   
The customer can connect its tenants to the main platform using a firewall to firewall VPN from the   
contact center to the platform. The contact center firewall should create a permanent VPN   
connection to the platform firewall.   
Note   
•   
Both firewalls should be redundant — no single point of failure.   
•   
Firewalls should support enough VPN bandwidth for all contact center agents and enough   
connections passing through the VPN.   
   
The following table lists the advantages and disadvantages.   
Advantages   
Disadvantages   
Secure connection for all agents   
Only agents in the contact center premises can   
connect to the platform   
Easy to manage

CCSP Prescriptive Architecture - 7.3 and 7.4   
45   
Internet   
The customer can connect its tenants to the main platform through the Internet. The platform’s   
FW2 should be configured accordingly.   
The following table lists the advantages and disadvantages.   
Advantages   
Disadvantages   
Agents can connect from anywhere   
Less secure to open connections to the   
platform from the Internet   
No dedicated links to the platform are required   
No QoS can be guaranteed for the VoIP traffic,   
therefore low voice quality can be observed   
   
Note   
Firewalls or NAT devices not in the service provider’s control might block RTP and create audio   
issues.   
Per agent VPN   
It is possible that each agent will connect using a VPN connection to the platform’s firewalls. In this   
case, the firewall should support enough dedicated connections from all agents.   
The following table lists the advantages and disadvantages.   
Advantages   
Disadvantages   
Agents can connect from anywhere   
Load on the firewalls requires more expensive   
firewalls   
No dedicated links to the platform are required   
   
QoS can be configured and therefore high   
voice quality maintained   
   
Direct desktop delivery   
Both media and application control go to the agent’s desktop. Media can only be voice. The control   
interface allows for a very rich integrated feature set, including screen-pops, web mail, chat, various   
CRM integrations, real-time reports, and click-to-dial.   
Direct desktop delivery supports Microsoft Windows 7and Windows 8 and requires Microsoft .NET   
4.5.1.   
The following diagram shows CCSP UI.

CCSP Prescriptive Architecture - 7.3 and 7.4   
46   
   
   
Figure 15: Agent direct media delivery   
Alternate delivery (hybrid)   
In the hybrid mode, the agent interface control goes to the agent desktop and the media to a third-  
party device such as an IP phone, analog phone, soft phone, or video-capable software. Because of   
the wide variety of applications and devices supported, hybrid agent delivery is extremely flexible   
and versatile.   
Simple IP phone with no shared call appearance   
In the simple hybrid delivery mode scenario, CCSP delivers media to a third-party device. This   
device is not registered and managed through CCSP and has no synchronized call appearance. The   
agent may independently use the endpoint device to make an outbound call or to receive a   
personal call, without the ACD knowing. From the ACD’s perspective, the agent still looks available.

CCSP Prescriptive Architecture - 7.3 and 7.4   
47   
   
Figure 16: Agent hybrid with no shared call appearance   
Enhanced IP phone with shared call appearance via DPS   
The Device Proxy Server (DPS) is a SIP component between the IP phone and the VCS. Its role is to   
act as SIP registrar and signaling intermediary between the IP phone and the CCSP system. Its main   
function is to translate call actions invoked on the phone, for example, Hold, Transfer, and   
Conference, into CCSP call controls. The agent GUI is also synchronized, so the DPS puts the IP   
phone in a state that matches the call controls initiated via the Agent GUI. The result is a complete,   
shared call appearance between the Agent computer application and the IP phone. All call actions   
are communicated via the CCSP proprietary messaging mechanism between the DPS and the AIS.   
The KCO-DPS keeps the SIP alternative device connected for the duration of the agent logon.   
The following diagram shows how the signaling path has been diverted to go through the DPS   
element.

CCSP Prescriptive Architecture - 7.3 and 7.4   
48   
   
Figure 17: Agent hybrid synchronized via DPS   
The DPS feature supports two modes of operation that are described in the following sections.   
Permanent extensions permanently associate each IP phone with a particular user, and thus with   
that user's associated direct extensions. The phone can be plugged into any Ethernet network port   
that provides access to the CCSP servers, and the phone will register and receive calls for that user.   
A lobby phone or conference room phone can be defined and appear as a user with an appropriate   
descriptive name. For presence and reporting purposes, these users always show as Logged In.   
Those using CCSP UI can indicate their presence status, such as whether or not they are available,   
from CCSP UI. When these users are not logged into CCSP UI, they show in presence status and   
reports as logged in and unavailable.   
Session-based extensions require a user to be logged into the IP phone to place or receive calls   
from the phone. Users enter their extension number and password into the phone using the user   
logon feature of the phone. Once logged in, users can control their availability status from both the   
phone and CCSP UI. Upon logon, the phone automatically acquires the direct extension of the user,   
so direct inbound calls are routed to the phone based on the user's identity, permitting extension   
mobility. A user can log out from the phone, for example, at the end of a shift.

CCSP Prescriptive Architecture - 7.3 and 7.4   
49   
Session-based extension users can either log out from their IP phone or from CCSP UI. If users log   
out from their IP phone, their CCSP UI application is automatically logged out and closed. If users   
log out from CCSP UI, they can be presented with the choice to also log off the phone, or to close   
only the CCSP UI application and keep the phone logged in. Alternatively, this choice can be   
preconfigured by the administrator. Only when users are logged out from both CCSP UI and their   
IP phone, are they marked as logged out for reporting purposes.   
The following table summarizes the differences between the two modes of CCSP extension.   
   
Permanent extension   
Session-based extension   
Standard features   
   
   
Inbound DID   
Always   
When user is logged on to   
phone   
Outbound   
Always   
When user is logged on to   
phone   
Hot seating   
No   
Yes   
Manage Available/Unavailable   
Via CCSP UI   
Via CCSP UI or phone   
ACD features   
   
   
Receive ACD-routed inbound   
calls   
When logged into CCSP UI and   
status is Available   
When logged into phone and   
status is Available (optionally   
also logged into CCSP UI)   
User logoff   
No   
Via CCSP UI or phone   
Receive outbound campaign   
calls   
Yes   
Yes   
CCSP UI for ACD features   
Required   
Optional   
Permanent extensions   
When using the permanent extension mode, registration parameters, including Agent Direct   
Extension and PIN are configured within the configuration preferences of the phone. The IP phone   
does not have to be configured with the ACD feature turned on.   
The following tables outline the different media delivery scenarios depending on the IP phone   
state.

CCSP Prescriptive Architecture - 7.3 and 7.4   
50   
Media delivery when dialing agent DID   
   
Agent logged in via GUI   
Agent logged out from GUI   
IP phone not registered   
CCSP UI   
Y   
IP phone registered   
IP phone   
IP phone   
   
Media delivery when receiving ACD routed call   
   
Agent logged in via GUI   
Agent logged out from GUI   
IP phone not registered   
CCSP UI   
Y   
IP phone registered   
IP phone   
Y   
   
Note   
When the IP phone is successfully registered, the agent media delivery is automatically changed   
from the desktop (direct) to the IP phone (alternate).   
Session-based extensions   
When using this mode, the IP phone must be configured with the ACD feature turned on.   
The following tables outline the different media delivery scenarios depending on the IP phone   
state.   
Media delivery when dialing agent DID   
   
Agent logged in via GUI   
Agent logged out from GUI   
IP phone not registered   
CCSP UI   
Y   
IP phone registered and agent   
not ACD logged-in   
CCSP UI   
Y   
IP phone registered and agent   
is ACD logged-in   
IP phone   
IP phone

CCSP Prescriptive Architecture - 7.3 and 7.4   
51   
Media delivery when receiving ACD routed call   
   
Agent logged in via GUI   
Agent logged out from GUI   
IP phone not registered   
CCSP UI   
Y   
IP phone registered and agent   
not ACD logged-in   
CCSP UI   
Y   
IP phone registered and agent   
is ACD logged-in   
IP phone   
IP phone   
   
Note   
•   
When the agent successfully registers via the IP phone’s ACD logon, the agent media delivery   
is automatically changed from the desktop (direct) to the IP phone (alternate).   
•   
When the agent logs out of the IP phone’s ACD feature, he is also automatically logged out   
from the desktop GUI. If he needs to have his media delivered directly to the desktop, he may   
then log in again to the desktop GUI.   
DPS and NAT   
This scenario supposes the IP phone to be installed behind a NAT device. We strongly recommend   
that you have a SIP-aware NAT firewall.   
Constraints and requirements   
The NAT feature is only supported for UDP transport.   
To support NAT, IP phone endpoints are required to support rport/received fields in the topmost   
SIP VIA header.   
For the residential NAT to keep the port alive, the Registration refresh or SIP session timer should   
be less than the port expiration.   
Corporate NAT   
The firewall has to look into the SIP messages and manipulate them to remove all private IP   
addresses on the public side. The result is complete NAT transparency.   
Residential NAT   
DPS has the ability to take care of NAT if the SIP message is correctly formatted.   
If the stack has rport/received populated, then the stack knows how to send response on this   
transaction.   
The DPS also detects if the device is behind NAT since the transport IP/Port values are different   
from the ones negotiated in the SDP body.

CCSP Prescriptive Architecture - 7.3 and 7.4   
52   
When sending SIP requests to the device, the DPS uses the remote IP address and port from the   
transport layer as the contact address or port to communicate with the device.   
The DPS automatically sets the behindNAT property for the IP phone to true.   
For incoming calls to the device, the DPS uses the transport contact address/port to send the call.   
The DPS inserts the ccunat parameter in the VIA header of the 1xx response. The parameter value   
should be ignored and the presence of the parameter should be a trigger to the RTP relay to sniff   
the first packet.   
For originating calls from the device, the DPS inserts the ccunat parameter in the VIA header of   
the INVITE message. The parameter value should be ignored and the presence of the parameter   
should be a trigger to the RTP relay to sniff the first packet.   
Phone Agent   
Phone Agent allows an agent with a legacy analog phone and without a computer application to   
participate in ACD-routed calls. The agent dials in to a special CCSP IVR, and logs in via DTMF digits.   
All presence state is subsequently done through the IVR interface. Only media (voice) goes to the   
analog device.   
Phone Agent is particularly useful for roaming agents using a cell phone or for remote agents who   
have limited or no Internet access.   
   
Figure 18: Phone Agent

CCSP Prescriptive Architecture - 7.3 and 7.4   
53   
External network leg   
The external network leg is mainly used to connect the VoIP Connection Servers to the voice   
delivery. This can be a gateway, a trunk, or an SBC.   
For detailed information about the connection of the VCS to the voice delivery device and the   
supported and recommended architecture, see 3: Redundancy.   
The external network leg is also used to connect the KCO-DPS servers to the SIP Gateway.   
CCSP VLAN usage   
For a complete list of the VLAN (s) to configure for each CCSP component, refer to the CCSP VLAN   
usage appendix in the CCSP Prerequisites Guide.

CCSP Prescriptive Architecture - 7.3 and 7.4   
54   
Firewall port configuration   
In a typical CCSP deployment, there are three main firewalls. FW1 could be omitted if SIP trunking is   
not used. The following sections describe the recommended CCSP signaling and VoIP traffic   
settings for each firewall.   
For a complete list of the services, protocols, and ports to consider for firewall port configuration,   
refer to the CCSP service connectivity map appendix in the CCSP Prerequisites Guide.   
FW1   
FW1 mainly serves to initiate and receive calls over a SIP trunk. If there is no trunking present, FW1   
could be entirely omitted from the network.   
The following table shows the FW1 port configuration.   
Service   
Protocol   
Port   
Direction   
SIP   
TCP/UDP   
5060   
Both   
PJSIP   
TCP/UDP   
40000-65565   
Both   
RTP   
UDP   
60000-65535\*   
Both   
RTSC   
UDP   
60000-65535\*   
Both   
H.225 (RAS)   
UDP   
1718/1719   
Both   
H.225 (Q.931)   
TCP   
1720   
Both   
H.245 (Call Control)   
TCP   
1024-65535   
Both   
\* Dynamic ports. See following note on how to restrict them.   
   
Note   
•   
This firewall may perform Network Address Translation (NAT).   
•   
Only open the ports for the VoIP protocols you are using.   
•   
If the ports for RTP/RTSC should be restricted, this can be configured by registry settings on   
each VCS server.

CCSP Prescriptive Architecture - 7.3 and 7.4   
55   
FW2   
FW2 is placed between the Internet (and any remote agents) and the CCSP platform. It allows the   
connection of end user applications for configuration, call control, and media streaming.   
Depending on the media delivery type, the signaling is initiated in different directions:   
•   
Direct delivery — media is delivered to CCSP UI. In this case the signaling (SIP, H.323) is   
always initiated by the agent, with the exception of outbound dialing campaign calls. VoIP   
signaling between VCS and agents is initiated in the Internal VLAN.   
•   
Hybrid delivery and phone-only — media is delivered to an IP phone or other third-party   
application. In this case the signaling (SIP, H.323) is always initiated by the VCS. VoIP signaling   
is initiated only in the External VLAN.   
   
The following table shows the FW2 port configuration.   
Service   
Protocol   
Port   
Direction   
SIP   
TCP/UDP   
5060   
Either   
PJSIP   
TCP/UDP   
40000-65565   
Either   
RTP   
UDP   
60000-65535\*   
Both   
RTSC   
UDP   
60000-65535\*   
Both   
H.225 (RAS)   
UDP   
1718/1719   
Either   
H.225 (Q.931)   
TCP   
1720   
Either   
H.245 (Call Control)   
TCP   
1024-65535   
Either   
HTTP/HTTPS   
TCP   
80/443   
Contact   
center/Internet->Zone   
1   
Exchange front end   
-   
-   
-   
\* Dynamic ports. See note regarding FW1 on how to restrict them.   
   
Note   
•   
This firewall should perform NAT (see below for more NAT considerations).   
•   
We recommend always encrypting HTTP traffic.   
•   
The firewall must be VoIP-aware.

CCSP Prescriptive Architecture - 7.3 and 7.4   
56   
NAT traversal considerations   
CCSP does not support STUN/TURN protocols on any SIP interface (VCS and agent). For NAT   
support, consider the following requirements. These requirements are compatible with symmetric   
firewalls, which are the most common type in today’s networks. There are three main scenarios   
regarding agent access, as described below.   
Note   
•   
For direct media delivery and firewalled VCS, FW2 should be SIP-aware so external agents can   
initiate connections to the VCS behind the firewall. A SIP and RTP forwarding rule needs to be   
programmed into FW2 to route all SIP signaling to the appropriate VCS. Furthermore, a   
separate registry key needs to be programmed into each VCS, containing the public IP   
address of FW2, so that the agents can actually reach it.   
•   
If there is no SIP-aware firewall, we recommend configuring VCSs with a public IP on the   
Internal VLAN.   
•   
Both agents and VCS use symmetrical RTP, which means transmitting and receiving RTP   
packets on the same port — Connection Oriented Media (COMEDIA). This allows for replies to   
go back to the same source address port.   
•   
VCS provides transport checks for both UDP and TCP. This means that if the parameters   
inside the SDP packet do not match those on the receives transport source IP and port, the   
VCS always uses the transport address to send replies to.   
•   
Normally, if no Session Border Controllers (SBCs) are used, the entity behind the firewall   
should be the one initiating the signaling. Since CCSP supports different types of media   
delivery methods that involve signaling from either end, care should be taken when choosing   
the delivery method paired up with the proper firewall configuration.   
•   
If using UDP for signaling transport, make sure the session timer is smaller than the firewall   
session timeout. RE-INVITEs should be sent at a regular interval to avoid teardowns. The   
same technique could be used with REGISTER messages if using an SBC.

CCSP Prescriptive Architecture - 7.3 and 7.4   
57   
Agent behind NAT firewall and VCS on public IP   
Remote agents behind NAT are very common. In this case, the VCS internal IP has a public address.   
This greatly facilitates connections from agents.   
   
Figure 19: Agents behind NAT   
The following table shows agents behind NAT signaling.   
Media   
delivery   
SIP direction   
RTP   
direction   
Notes   
Direct   
Agent->VCS   
TCP or UDP   
Either   
initiates   
VCS replies to transport IP address and port for   
SIP signaling (agent public IP).   
VCS replies to transport IP address and port for   
RTP signaling (agent public IP).   
VCS can’t establish RTP connection first.   
Alternate (IP   
phone)   
VCS->IP   
phone   
TCP or UDP   
Either   
initiates   
Agent alternate delivery needs to be a public IP   
and port. Agent firewall needs to have static   
forwarding rule to relay SIP to IP phone.   
Works better if agent firewall is SIP-aware, else IP   
phone needs to support COMEDIA.   
Use IP phone’s STUN capabilities if available.   
   
Note   
Trivial NAT traversal can be implemented on the engineering level by limiting the range of the   
ports on the agent side and statically mapping the same range of ports on the NAT on a per   
agent basis. Microsoft RTC client (SIP client used on the agent side) allows you to set the range of   
ports used for signaling and media.

CCSP Prescriptive Architecture - 7.3 and 7.4   
58   
Agent with public IP and VCS behind NAT firewall   
   
Figure 20: VCS behind NAT   
The following table shows the VCS behind NAT signaling.   
Media   
delivery   
SIP direction   
RTP   
direction   
Notes   
Direct   
Agent->VCS   
TCP or UDP   
Either   
initiates   
Agent contacts VCS on public IP, as configured in   
the registry key.   
Agent contacts VCS on public RTP IP as   
configured in the registry key but SIP static   
forwarding port is needed for RTP on FW2.   
VCS can also establish RTP session with agent.   
Alternate (IP   
phone)   
VCS->IP   
phone   
TCP or UDP   
Either   
initiates   
VCS can establish both SIP and RTP media to the   
IP phone.   
IP phone contacts VCS on public RTP IP as   
configured in the registry key but SIP static   
forwarding port is needed for RTP on FW2.   
Best if IP phone supports COMEDIA.   
   
Note   
We highly recommend having a VoIP-aware firewall in this scenario.

CCSP Prescriptive Architecture - 7.3 and 7.4   
59   
Both agent and VCS behind NAT firewalls   
This is the most difficult scenario to work with.   
   
Figure 21: Agent and VCS behind NAT   
The following table shows the agent and VCS behind NAT signaling.   
Media   
delivery   
SIP direction   
RTP   
direction   
Notes   
Direct   
Agent->VCS   
TCP or UDP   
Either   
initiates   
Agent contacts VCS on public IP, as configured in   
the registry key.   
Agent contacts VCS on public RTP IP as   
configured in the registry key but SIP static   
forwarding port is needed for RTP on FW2.   
VCS can’t establish RTP connection first.   
Alternate (IP   
phone)   
VCS->IP   
phone   
TCP or UDP   
Either   
initiates   
Agent alternate delivery needs to be a public IP   
and port. Agent firewall needs to have static   
forwarding rule to relay SIP to IP phone.   
Works better if agent firewall is SIP-aware,   
otherwise the IP phone needs to support   
COMEDIA.   
Use IP phone’s STUN capabilities if available.   
IP phone contacts VCS on public RTP IP as   
configured in the registry key but SIP static   
forwarding port is needed for RTP on FW2.   
   
Note   
We highly recommend having a VoIP-aware firewall in this scenario.

CCSP Prescriptive Architecture - 7.3 and 7.4   
60   
SBC   
Session Border Controllers (SBCs) provide an easy solution to the dual firewall configuration and   
neither firewall needs to be SIP-aware. SBCs typically sit between firewalls in the public network   
and intercept and modify the SIP signaling to provide both endpoints with the illusion that they are   
communicating with a public IP device.   
   
Figure 22: SBC between NAT devices   
SBCs provide the following advantages:   
•   
End-to-end SIP security since only registered devices are allowed to communicate.   
•   
Fully NAT-aware.   
•   
Clients must register with the SBC. During registration, the private IPs are discovered.   
•   
Modifies the SDP packet in real-time, presenting a public IP address to both clients.   
•   
Forces RTP media to flow through the SBC. Handles RTCP channel mapping.   
•   
Provides transcoding, if required.   
   
Note   
For details about configuring VCS with SBC, see the CCSP Registry Key Reference.   
FW3   
FW3 protects the back-end services. We recommend that you either:   
•   
Open only the specific ports needed for each application, also using IP source addresses to   
restrict access.   
•   
Open ports based only on IP source address (less secure).   
   
The following tables show the core ports used between Zone 1 and Zone 2.   
For a more detailed list, see Appendix A: CCSP service connectivity map in the CCSP Prerequisites Guide.

CCSP Prescriptive Architecture - 7.3 and 7.4   
61   
FW3 port configuration   
Service   
Protocol   
Port   
Direction   
Adapter   
TCP   
14022   
Both   
AdminIS IPC   
TCP   
14300   
Both   
ADP IPC   
TCP   
80   
Zone 1->Zone 2   
AIS IPC   
TCP   
14005   
Both   
ALS IPC   
TCP   
14030   
Both   
CFM IPC   
TCP   
14001   
Zone 1->Zone 2   
CMS IPC   
TCP   
14002   
Both   
DPS IPC   
TCP   
14014   
Both   
ECM Gate   
TCP   
1500   
Both   
ICS IPC   
TCP   
14006   
Both   
IMS IPC   
TCP   
14011   
Both   
LCS IPC   
TCP   
9000   
Zone 1->Zone 2   
LS IPC   
TCP   
14003   
Zone 1->Zone 2   
MCS IPC   
TCP   
14007   
Both   
MS IPC   
TCP   
14020   
Both   
PDS IPC   
TCP   
14015   
Both   
RTP IPC   
TCP   
14012   
Both   
SS IPC   
TCP   
14004   
Zone 1->Zone 2   
VCS IPC   
TCP   
14010   
Both   
FW3 Active Directory communication ports   
Service   
Protocol   
Port   
Direction   
LDAP to Active   
Directory   
TCP/UDP   
389   
Zone 1->Zone 2   
LDAP to Global   
Catalog Server   
TCP   
3268   
Zone 1->Zone 2

CCSP Prescriptive Architecture - 7.3 and 7.4   
62   
Service   
Protocol   
Port   
Direction   
Kerberos   
Authentication   
TCP/UDP   
88   
Zone 1->Zone 2   
RPC Endpoint Mapper   
/ CCSP Publisher   
TCP   
135   
Zone 1->Zone 2   
SMB   
TCP   
139   
Zone 1->Zone 2   
SMB (Microsoft-DS) /   
CCSP Publisher   
TCP   
445   
Zone 1->Zone 2   
NetBIOS related   
UDP   
137/138   
Zone 1->Zone 2   
AD Logon and   
replication   
TCP   
1025/1026   
Zone 1->Zone 2   
RPC Services   
TCP   
>= 1024\*   
Zone 1->Zone 2   
DNS   
TCP/UDP   
53   
Zone 1->Zone 2   
\* See note below on how to limit the port range.   
FW3 other ports   
Service   
Protocol   
Port   
Direction   
Exchange back end   
-   
-   
-   
Microsoft RS (SQL   
queries)   
TCP   
1433   
Zone 1->Zone 2   
Terminal Services   
Access   
TCP   
3389   
If VPN inside Zone 1   
   
Note   
•   
This firewall should not perform NAT.   
•   
Limit RPCs across the firewall by editing the registry of all your DCs.   
•   
IPC ports in FW3 port configuration are based on default CCSP installation values. These can   
be modified as needed during installation.

CCSP Prescriptive Architecture - 7.3 and 7.4   
63   
Dialing Plan (DP)/Advanced Dial Plan (ADP)   
Caution   
The dial plan is a trusted internal web service. It does not include any authentication and so must   
not be accessible from the public internet.   
   
The dial plan should be installed in Zone 2 (backend) with FW3 firewall rules in place to allow   
HTTP(S) access from the following frontend servers:   
•   
AIS   
•   
VCS   
•   
DPS (including KCO)   
   
The following servers also query the dial plan but are already in Zone 2 (backend).   
•   
PDS   
•   
Elsbeth   
   
Each service queries the dial plan via a basic HTTP request (port 80) and the server responds with   
XML that includes a list of voice servers and gateways.   
For small systems that require a smaller footprint, it is possible to install the dial plan in Zone 1   
(frontend) on an existing IIS machine. However, this requires additional IIS configuration to protect   
any external access to the dial plan.

CCSP Prescriptive Architecture - 7.3 and 7.4   
64   
Distributed architecture   
CCSP’s distributed environment is adapted for deployments in multiple global locations.   
   
Figure 23: Distributed architecture   
The front-end servers (VCS, AIS, IIS/APS, SSRS, DTR, and DPS) are the ones typically distributed and   
load-balanced across sites, whereas the back-end servers reside in one (or two, if using the   
Business Continuity model) central locations. These central locations can also have their own front-  
end servers, if required.   
If connectivity to one of the remote sites fails, agents in that geographical region may be redirected   
to other Zone 1 sites by giving them a different logon URL that points them to another IIS/AIS   
server farm.   
If one of the back-end sites fails, this should be transparent for all agents, provided business   
continuity is planned.   
Depending on the bandwidth to the back-end site, it could be useful to host the recording database   
on each remote location. This avoids sending substantial volumes of data over the WAN link.   
We recommend establishing a VPN mesh between Zone 1 and Zone 2 for better security and also to   
avoid having to configure two additional Internet-facing firewalls.

CCSP Prescriptive Architecture - 7.3 and 7.4   
65   
Bandwidth requirements   
The client bandwidth for the G711 codec and with different frame sizes from 10ms to 60ms can be   
computed as:   
•   
G711 with 10ms frame size   
o   
(64kbps \* 10 ms )/ 8bit = 80 bytes (of voice coder data per 10 ms)   
o   
[(54 Bytes of Overhead) + (80 Bytes of Voice Coder Data)] \* (8bits/byte) \* (100 Packets Per   
Second) = 104.7kbps   
   
•   
G711 with 20ms frame size   
o   
(64kbps \* 20 ms )/ 8bit = 160 bytes (of voice coder data per 20 ms)   
o   
 [(54 Bytes of Overhead) + (160 Bytes of Voice Coder Data)] \* (8bits/byte) \* (50 Packets   
Per Second) = 83.6kbps   
   
•   
G711 with 30ms frame size   
o   
(64kbps \* 30 ms )/ 8bit = 240 bytes (of voice coder data per 30 ms)   
o   
 [(54 Bytes of Overhead) + (240 Bytes of Voice Coder Data)] \* (8bits/byte) \* (33 Packets   
Per Second) = 75.8kbps   
   
•   
G711 with 30ms frame size   
o   
(64kbps \* 40 ms )/ 8bit = 320 bytes (of voice coder data per 40 ms)   
o   
 [(54 Bytes of Overhead) + (320 Bytes of Voice Coder Data)] \* (8bits/byte) \* (25 Packets   
Per Second) = 73.1kbps   
   
•   
G729   
o   
(8kbps \* 10 ms) / 8 bit = 10 bytes (of voice coder data per 164\*80 ms)   
o   
 [(54 Bytes of Overhead) + (10 Bytes of Voice Coder Data)] \* (8bits/byte) \* (50 Packets Per   
Second) = 25.6kbps   
   
These bandwidths are for each direction.   
The CCSP UI connection to the AIS is:   
•   
Log on and go available — 167Kb of traffic   
•   
Get a call and pick it up — 7Kb of traffic   
•   
Stay idle for 1 minute — 1.5Kb of traffic

CCSP Prescriptive Architecture - 7.3 and 7.4   
66   
5: Connectivity   
•   
Media connectivity   
•   
Back-end server connectivity   
•   
Backup and maintenance   
Media connectivity   
The CCSP platform requires the following media connectivity.   
VoIP   
CCSP requires the following VoIP connectivity.   
Gateways   
The following basic features are needed from the VoIP gateway.   
One or many of the CCSP-supported codecs. For details, see Codecs.   
Support for the following standards:   
•   
SIP (RFC 3261)   
•   
H.323 (IEEE)   
   
DTMF transmission support in these methods:   
•   
DTMFs as RTP events (RFC 2833)   
•   
SIP INFO or H.245 facility   
   
Note   
In-band DTMF is not supported by CCSP.   
   
Voice features (voice quality features):   
•   
Echo canceller   
•   
Jitter buffer   
•   
Packet lost concealment   
•   
Voice volume level settings   
•   
Comfort noise generation to the PSTN   
•   
Call routing and load balancing   
•   
Ability to register to External SIP Proxy or H.323 Gatekeeper. Here the load balancing between   
VCS servers is done by the proxy or gatekeeper.   
•   
Ability to register with multiple SIP Proxies or H.323 Gatekeepers for backup.

CCSP Prescriptive Architecture - 7.3 and 7.4   
67   
•   
Ability to load balance calls between different VoIP devices. The gateway does the load   
balancing between VCS servers.   
•   
Ability to find the availability of VoIP devices on the call routing table by ping or other keep   
alive mechanism.   
•   
Packet marking for QoS   
•   
DiffServ marking   
•   
Security   
•   
Management white list   
•   
Call offer white list and black list   
•   
Gateway management   
•   
SNMP support   
•   
CDR generation   
•   
Backup and restore of the gateway configuration   
Codecs   
The following codecs are supported by the CCSP platform.   
Signaling protocols:   
•   
H.323 (both voice and video exist)   
•   
SIP (voice exists and video is planned)   
•   
MSCML (media resource control)   
SIP   
The following table shows the support of codecs in SIP communication.   
   
Codec   
VCS   
IVR   
CCSP UI   
integrated   
CCSP UI CSA   
H.263   
Y   
Y   
N   
Y   
H.264   
Y   
N   
N   
Y   
G711a   
Y   
Y   
Y   
Y   
G711u   
Y   
Y   
Y   
Y   
G.723.1   
Y   
Y   
Y   
Y   
G.729B   
Y   
Y   
Y   
Y

CCSP Prescriptive Architecture - 7.3 and 7.4   
68   
H.323   
The following table shows the support of codecs in H.323 communication.   
   
Codec   
VCS   
IVR   
CCSP UI   
integrated   
CCSP UI CSA   
H.263   
Y   
Y   
N   
Y   
H.264   
N   
N   
N   
Y   
G711a   
Y   
Y   
N   
Y   
G711u   
Y   
Y   
N   
Y   
G.723.1   
Y   
Y   
N   
Y   
G.729B   
Y   
Y   
N   
Y   
   
Video resolution (screen size):   
•   
CIF (352 x 288)   
•   
QCIF (176 x 144)   
   
Audio:   
•   
MP4 (voicemail)   
   
Video frame rates:   
•   
1 - 30 fps, dynamically adjusted to the source file or device   
•   
Variable bit-rate (VBR) support

CCSP Prescriptive Architecture - 7.3 and 7.4   
69   
VoIP media connectivity   
Although CCSP includes full functional IVR/IVVR functionality, in some cases external media servers   
are used. CCSP’s IVR scripts integrate with external media servers through the use of Transfer or   
VXMLTransfer blocks. Media servers return information to CCSP using the SIP INFO method.   
   
Figure 24: Media server connection   
   
In both transfer cases however, the call mechanics are identical:   
1.   
The incoming SIP call is established with the gateway.   
2.   
A new SIP call is placed on the outbound leg towards the media server.   
3.   
When the outbound call leg is established, media is bridged between the two call legs.   
4.   
The media server collects user digits and commands and returns results inside a SIP INFO   
packet.   
VXML INVITE format   
Media servers running VXML scripts can have these invoked directly inside the SIP INVITE method   
by providing a new VXML-specific header.   
sip:dialog@mediaserver.example.com; \   
voicexml=http://appserver.example.com/promptcollect.vxml; \   
maxage=3600;maxstale=0   
   
The media server is responsible for retrieving the .vxml file from a web server and executing it.

CCSP Prescriptive Architecture - 7.3 and 7.4   
70   
IP INFO   
The media server must return a SIP INFO message, which contains data in the following format:   
INFO sip:2143302100@172.17.2.33 SIP/2.0   
Via: SIP/2.0/UDP 172.80.2.100:5060   
From: <sip:9724401003@172.80.2.100>;tag=43   
To: <sip:2143302100@172.17.2.33>;tag=9753.0207   
Call-ID: 98407215401962@172.80.2.100   
CSeq: 25634   
INFO Supported: 100rel   
Supported: timer   
Content-Length: 78   
Content-Type: application/ccu-integration   
CustomerName=John Smith   
Account=1234567   
SSN=111-22-3333   
   
The content type must always be set to application/ccu-integration and the data is composed of a   
series of name-value pairs, separated by an equal sign (=), with one pair per line. Lines may end in   
CR or CR/LF. Once the data is received by the CCSP IVR application, it is extracted via the   
GetUserData block.   
Note   
Another block called SendUserData can also be used to relay information to the media server via   
SIP INFO. CCSP uses this technique to wrap the MSCML control protocol. If your media server   
understands MSCML, CCSP IVR can remotely drive it with MSCML commands. For more   
information, contact Enghouse Customer Support.   
MRCP   
Media Resource Control Protocol (MRCP) is a communication protocol defined by IETF, which is   
used to connect to speech servers to use several services.   
In CCSP, these services are speech recognition and speech synthesis.   
The speech server must be located in the internal network where the agents reside.   
MRCP is currently available in two versions. Version 1 does not specify the underlying protocol and   
version 2 is SIP-based. We recommend using version 2 when connecting with CCSP.   
The communication between the speech server and CCSP is handled by the VCS servers.   
Here the signaling port can be configured through CCSP Admin and the ports for the RTP traffic are   
dynamically assigned and used.   
We recommend that you use Nuance and have verified the following versions for CCSP.

CCSP Prescriptive Architecture - 7.3 and 7.4   
71   
Component   
Version   
OS   
Nuance Speech Recognizer   
11.0.3   
Win2016   
Nuance Speech Server   
7.1.0   
Win2016   
Nuance License Manager   
11.14.3   
Win2016   
Nuance Vocalizer Version   
7.2.7   
Win2016   
Nuance Languages Version   
3.3.7   
Win2016   
Back-end server connectivity   
The CCSP platform requires the following back-end server connectivity.   
SQL Server   
CCSP requires the following SQL Server connectivity.   
Multiple concurrent instances   
Running SQL Server in multiple instances allows for better process isolation and fault tolerance for   
memory resources and disk I/O. Very busy instances, if running at the same process priority, yield   
CPU time to other instances, making the entire platform more responsive. When the system has   
grown enough, each instance should run on its own dedicated server.   
The following diagram represents a general rule of thumb regarding SQL instances.   
   
Figure 25: SQL server instances   
\* If recording more than 50% of agents, run the recording database under its own instance.

CCSP Prescriptive Architecture - 7.3 and 7.4   
72   
Note   
•   
On big platforms, we strongly recommend separating the historical database into two   
instances, where one stores the historical information and the other presents it to the   
supervisors. For transferring the data, use the standard replication features of Microsoft SQL   
Server.   
•   
Having multiple SQL Server instances running on the same processor does not affect the   
number of licenses. SQL licenses are tied strictly to the number of processors, not the   
number of SQL copies.   
•   
When running SQL Server on a cluster, divide instances between nodes.   
•   
For each SQL Server instance running on the same server, add another 2 GB RAM.   
Microsoft Reporting Services   
Microsoft Reporting Services is used for generating and displaying historical data information.   
SQL Server Reporting Service   
CCSP uses SQL Server Reporting Service (SSRS) technology to generate historical reports. SSRS   
offers a complete, server-based platform for creating, managing, and delivering traditional and   
interactive reports, which can be configured based on specific tenant requirements. The reports   
represent an aggregated view of contact center events that have been logged in the historical   
database during a certain time interval. Each tenant has access to its own historical reporting   
portal, which resides on the IIS server in Zone 1.   
SSRS deployments   
SSRS can be deployed in a variety of ways. However, because it relies on SQL Server technologies,   
SSRS requires its own SQL license.   
There are three main types of deployments:   
•   
Type 1 — only for very small contact centers of <50 agents where all CCSP components run on   
a single server. In this case, only a single SQL Standard license is required.   
•   
Type 2 — for larger contact centers where the CCSP platform has been divided into Zone 1 and   
Zone 2. The advantage of this scenario is that the SQL Server database stays in the back end   
and is not exposed to a potential attack from the outside. Also, report generation is offloaded   
to the IIS server. This type requires a SQL Standard license on both the SSRS and the SQL   
server.   
•   
Type 3 — also used for larger contact centers and is identical to Type 2 in terms of deployment   
architecture. However, the SSRS service is now protected by load-balancing. Also because of   
multiple source synchronization, a load-balanced SSRS requires a SQL Enterprise license, which   
makes it less cost effective. The back-end SQL server should use a SQL Enterprise license.

CCSP Prescriptive Architecture - 7.3 and 7.4   
73   
   
Figure 26: SSRS deployments   
   
Type 3 is also called a scale-out deployment. If you need to use a Type 2 deployment for cost   
reasons, but the SSRS needs to be scaled to accommodate a larger number of tenants, it is usually   
more cost effective to add more processors and more RAM to a single server, than to move to Type   
3. The advantage of doing this is that you still keep a single SSRS instance while purchasing more   
SQL Standard licenses (one for each additional processor), which are much cheaper than Enterprise   
ones. This is called a scale-up approach.   
Transactional replication   
If there are more than 1000 agents, we strongly advise to mainly use the SQL Server Analysis   
Service to create and view reports.   
Running reports directly on a production server is only acceptable if the frequency of the reports,   
the amount of data mined, and the average number of transactions on the SQL server is relatively   
low.   
We recommend that above 1000 agents, reports be run from a different reporting database.   
Transactional replication is used to create a replica of the production database that is relatively   
recent and refreshed regularly. To offer high availability to report clients, the replicated data should   
also be made redundant.

CCSP Prescriptive Architecture - 7.3 and 7.4   
74   
   
Figure 27: Transactional replication   
   
In transactional replication there are five main data flows:   
1.   
The CCSP LS receives call events and logs them to the historical database.   
2.   
A Publisher service reads the transaction logs on the historical database and pushes them at   
configurable time intervals to the Distributor database.   
3.   
The Distributor in turn, pushes the logs to one or more Subscribers. For redundancy, we   
recommend having a secondary Subscriber.   
4.   
The Reporting Service Catalog database, which also resides on the primary Subscriber, is   
mirrored to the secondary Subscriber for redundancy.   
5.   
The SSRS finally queries the Subscriber database to create its reports.   
   
Note   
•   
The Distributor must run on the reports server to offload the production SQL server.   
•   
The Publisher should run at most every 15 minutes.   
•   
Instead of having two Subscribers, you can have a single Subscriber that mirrors the   
historical database to a secondary server (mirror), in the same way that the RS Catalog   
database is mirrored.

CCSP Prescriptive Architecture - 7.3 and 7.4   
75   
Messaging media and MSRs   
From CCSP 7.x onward, a Message Storage Repository (MSR) is required for the messaging media,   
which are:   
•   
Email — system-routed email and system-relayed outgoing email   
•   
Voicemail — system-routed voicemail   
•   
Callback — system-routed scheduled callbacks   
   
In CCSP 7.3, the supported MSRs are:   
•   
Tenant messaging database (MSR DB) — recommended   
o   
Each MSR DB can be used as single or multiple media store   
o   
Each MSR DB can used as single or multiple tenants store   
o   
From CCSP 7.3.4 onward, data in each MSR DB (the active MSR DB) can be archived to a   
single archive database (the archive MSR DB)   
•   
Tenant Exchange database   
o   
Exchange 2010 SP3 (latest rollup update).   
   
From CCSP 7.x onward, the APS component is the frontend for all the messaging media. The MSR,   
Message Connection Server, and Messaging Service are at the backend. This architecture is valid   
for every organization-type that uses messaging media for ACD.   
Starting from CCSP 7.3, you can activate each media separately per tenant, by assigning an MSR to   
a specific tenant’s media store.   
Access to the messaging media content is based on media clients that are part of CCSP UI. Access is   
via the APS using HTTP/HTTPS protocols.   
Starting from CCSP 7.3.4, the APS component can access messaging media stored in an active MSR   
DB or an archive MSR DB. CCSP UI toggles between the active and archive MSR DBs, according to   
the type of messaging item the agents selects.   
Backward compatibility   
•   
The frontend and backend server topology used for Exchange in earlier CCSP versions is no   
longer required.   
•   
Private email service capabilities are no longer part of the CCSP services and are managed   
separately.   
•   
OWA or other Mail Clients Applications (with Public Folders support) may be used when   
working with Exchange 2010 as MSR for maintenance purposes only and will be replaced in a   
later CCSP version. To avoid a collision with the system media management, we do not   
recommend that you alter the active messaging content.

CCSP Prescriptive Architecture - 7.3 and 7.4   
76   
The following diagram shows the frontend/backend deployment scenario.   
CCSP Touchpoint  
(remote site)  
FW2  
Front End: Agent Web-Server  
(N+1 load balanced)  
Back End: Message Servers  
(MS per tenant)  
T1  
T2 & T3  
Back End: Messaging Server Repositories (MSR)  
(MSR per tenant per media)  
T1 – Callback & Voicemail  
T1 - Email  
T2 – All Media  
T3 – All Media  
FW3  
CCSP Touchpoint  
(On premise)  
   
Figure 28: Messaging services deployment   
Email routing   
Routing email to and from the platform can be done in several ways. It should be planned in detail   
before installing the platform, taking into consideration tenant's MSR type selection. Each MSR type   
has its own considerations as detailed below.   
Authentication via Active Directory   
CCSP uses Active Directory to identify each tenant. Each tenant has its own UPN and all its users   
share the same UPN suffix.   
If email needs to be routed to the platform using a tenant UPN suffix that is already a public   
domain of the tenant corporate domain, for example, @microsoft.com, you must configure the   
corporate MSR to relay the email that needs to reach the platform to the MSR of the platform. The   
From field should always contain the sender address.   
Tenant messaging database method   
Working with MSR DB introduces a new requirement for using an external/internal mail server for   
email routing, and introduces a new messaging connector to relay from/to it using standard   
secured IMAP and SMTP protocols.

CCSP Prescriptive Architecture - 7.3 and 7.4   
77   
FW2  
FW3  
External Mail-Server  
Back End: Message Servers  
(MS per tenant)  
T1  
T2 & T3  
On Premise Mail-Server  
   
Figure 29: Working with a mail server   
The mail server includes single or multiple mailboxes, where each mailbox has a single email   
address or multiple email addresses (known as email aliases). CCSP tenant configuration   
introduces a new entity called Mailbox, which is used to define (among other parameters) the   
connectivity details for both secure IMAP and SMTP for these mailboxes. There is a Mailbox entity   
for each mailbox on the mail server.   
The mail server is responsible for inbound/outbound email relay and must be set up in advance.   
When using an external mail server for a tenant, you must ensure that trust exists between the   
CCSP platform and the chosen mail servers. When working with an on premise mail server, to allow   
email relay to/from the internet, the External leg should be considered.   
For inbound emails, if email media is activated, CCSP periodically polls the configured mailboxes on   
the mail server for new items using the standard secured IMAP protocol. CCSP uses the recipient   
address on the item itself to match it to its designated folder, downloads the item, and stores it in   
this folder on the MSR. After the item is persisted on the MSR, it is either deleted from the mailbox   
or marked as persisted on the mailbox, by moving it to a specific folder on the mailbox to avoid   
retrieving the item on the next cycle.   
For outbound emails, CCSP relays the item using the standard secured SMTP protocol through the   
configured mailbox. The mailbox is selected according to the item sender's email address. After the   
item is sent, a copy of it is persisted on the MSR.   
For callback over SMTP, CCSP considers the item a callback only if the email item was properly set   
with callback-specific extended headers (for details, refer to the CCSP Web Support integration   
appendix in the CCSP APIs Guide.). The designated callback item is created in the MSR with relevant   
scheduling and other callback data. Note that for callback over SMTP, email media must be   
activated so the mailbox is scanned for new items.

CCSP Prescriptive Architecture - 7.3 and 7.4   
78   
FW2 firewall   
On the Internet firewall, FW2, the following ports must be opened.   
Exchange external ports   
Service   
Protocol   
Port   
Direction   
HTTPS   
TCP   
443   
Inbound   
SMTP   
TCP   
25   
Inbound   
SMTP secure   
TCP   
587   
Inbound/Outbound   
IMAP secure   
TCP   
993   
Inbound   
FW3 firewall   
On FW3, which connects the DMZ (front-end) and the back-end servers, the following ports must be   
opened.   
Exchange communication ports   
Service   
Protocol   
Port   
Direction   
HTTP   
TCP   
80   
DMZ->Back end   
Link State Algorithm   
routing   
TCP   
691   
DMZ->Back end   
SMTP   
TCP   
25   
DMZ->Back end   
IMAP\*   
TCP   
143   
DMZ->Back end   
POP\*   
TCP   
110   
DMZ->Back end   
\* Only if enabled on the F2 firewall.   
   
   
Active Directory communication ports   
Service   
Protocol   
Port   
Direction   
LDAP to Active   
Directory   
TCP/UDP   
389   
DMZ->Back end   
LDAP to Global   
Catalog Server   
TCP   
3268   
DMZ->Back end   
Kerberos   
Authentication   
TCP/UDP   
88   
DMZ->Back end   
   
You must also configure your Exchange front-end server to connect to the DC and GC by editing the   
server properties in Exchange System Manager. For this, the following ports are needed.

CCSP Prescriptive Architecture - 7.3 and 7.4   
79   
DNS and GC communication ports   
Service   
Protocol   
Port   
Direction   
DNS   
TCP/UDP   
53   
DMZ->Back end   
RPC Endpoint Mapper   
TCP   
135   
DMZ->Back end   
RPC Services   
TCP   
>= 1024\*   
DMZ->Back end   
\* See note below on how to limit the port range.   
   
   
If you are using IPSec between front-end and back-end servers, you have to open the following   
ports.   
IPSec communication ports   
Service   
Protocol   
Port   
Direction   
IKE   
UDP   
500   
Both   
Authentication   
Header (AH)   
TCP/UDP   
51   
Both   
Encapsulation   
Protocol (ESP)   
TCP/UDP   
50   
Both   
Security   
You can position the front-end server as the single point of access on or behind an Internet firewall   
(FW2) that is configured to allow only traffic to the front end from the Internet. Because the front-  
end server has no user information stored on it, it provides an additional layer of security for the   
organization. Also, the front-end servers authenticate requests before proxying them, protecting   
the back-end servers from denial-of-service (DoS) attacks.   
Improved public folder access and features   
A front-end Exchange server increases the robustness of accessing public folders, as it knows the   
state of back-end servers and can use multiple referrals to access public folder data. This includes   
system data such as calendar free/busy information. Also, a front-end Exchange server enables   
agents using OWA to reply to or forward posts in public folders. Without a front-end server, public   
folder post can only be read.   
Backup and maintenance   
For advice about backup and maintenance, contact Enghouse Customer Support.

CCSP Prescriptive Architecture - 7.3 and 7.4   
80   
6: CCSP Dialer   
•   
Elsbeth mode   
•   
Sytel mode   
   
CCSP Dialer is an outbound call processing system designed to maintain a high level of utilization   
and cost efficiency in the contact center. The dialer automatically calls a list of telephone numbers   
in several different dialing modes and then connects a waiting customer with the next available   
agent in the contact center.   
CCSP supports two different dialing machines to support the dialing feature: Elsbeth mode and   
Sytel mode.   
Elsbeth mode   
In Elsbeth mode, the CCSP Adapter service hosted in an Elsbeth CommunicationManger (ECM) Gate   
is the component that connects and manages the communication between the dialers and the   
CCSP platform.   
The CCSP Adapter establishes an IPC connection with the:   
•   
CFM to retrieve configuration   
•   
AIS for agent state handling   
•   
VCS for call control and media connectivity management of every call attempt   
   
It is N+1 and you should deploy more than one CCSP Adapter for system capacity enhancement.

CCSP Prescriptive Architecture - 7.3 and 7.4   
81   
Setup   
The following diagrams show the Elsbeth mode architecture.   
   
Figure 30: Elsbeth mode VLAN architecture

CCSP Prescriptive Architecture - 7.3 and 7.4   
82   
   
Figure 31: Elsbeth mode architecture   
Elsbeth CommunicationManger (ECM)   
ECM implements the dialer algorithm and manages the call state. ECM has several interfaces:   
•   
Gate service HW API, which is an interface to the media layer.   
•   
Agent Frontend API, which is an interface to the agent layer.   
•   
Monitor API, for reporting.   
•   
Backend API, not relevant for the integration.   
   
For the Elsbeth integration, ECM exposes the APIs through the CCSP Adapter, which is available on   
the ECM Gate service.   
Elsbeth PowerContact (EPC)   
EPC provides the agent management (agent client) and the campaign manager. It has a client –   
server architecture. The EPC Client is implemented in CCSP UI as the EPC Gadget, which interacts   
with the EPC Server.

CCSP Prescriptive Architecture - 7.3 and 7.4   
83   
EPC Server   
The EPC Server is a campaign manager that retrieves numbers to dial from a database and handles   
campaigns per tenant.   
Each EPC Server instance requires its own unit, which means that each tenant requires its own   
server. The tenant's EPC Server may be located in the provider site or remotely on the tenant   
premise site.   
ECM dialer   
The ECM dialer is a multi-tenant system and serves all EPC Servers and CCSP in a centralized   
manner.   
To communicate with CCSP and the frontend, it employs N+1 Gate services, each hosting a CCSP   
Adapter.   
Dialer ports   
The following ports are used by the CCSP Dialer for Elsbeth.   
Service   
Direction   
Protocol   
Port   
CFM IPC   
Adapter->CFM   
TCP   
14001   
AIS IPC   
or Adapter IPC   
Adapter->AIS   
TCP   
14005   
AIS->Adapter   
TCP   
14022   
VCS IPC   
or Adapter IPC   
Adapter->VCS   
TCP   
14010   
VCS->Adapter   
TCP   
14022   
HTTP/HTTPS   
Adapter->DP   
TCP   
80/443   
Elsbeth PowerContact   
(EPC) Gadget   
EPC 6.x->EPC Gadget 6.x   
TCP   
8888   
EPC 9.x->EPC Gadget 9.x   
TCP   
7927   
EPC Gadget 6.x ->EPC 6.x   
TCP   
8888   
EPC Gadget 9.x ->EPC 9.x   
TCP   
7927   
Elsbeth Communication   
Manger (ECM)   
   
EPC->ECM   
Gate->ECM   
TCP   
1500   
EPC->ECM   
Gate->ECM   
TCP (SSL)   
1600   
Engine Provider->ECM   
TCP   
1501   
Engine Provider->ECM   
TCP (SSL)   
1601

CCSP Prescriptive Architecture - 7.3 and 7.4   
84   
Service   
Direction   
Protocol   
Port   
Gate->ECM   
TCP   
1502   
Service Manager->ECM   
Gate->ECM   
SNMP Provider->ECM   
TCP   
1510   
Service Manager->ECM   
Gate->ECM   
SNMP Provider->ECM   
TCP (SSL)   
   
1610   
Elsbeth Engine Provider   
Service Manager->Engine Provider   
TCP   
1511   
Service Manager->Engine Provider   
TCP (SSL)   
1611   
Elsbeth Gate   
Service Manager->Gate   
TCP   
1512   
Service Manager->Gate   
TCP (SSL)   
1612   
Elsbeth Data Manager   
(optional)   
Service Manager->Data Manager   
TCP   
1515   
Service Manager->Data Manager   
TCP (SSL)   
1615   
Elsbeth Email Manager   
(optional)   
Service Manager->Email Manager   
TCP   
1517   
Service Manager->Email Manager   
TCP (SSL)   
1617   
Elsbeth SNMP Provider   
(optional)   
Service Manager->SNMP Provider   
TCP   
1518   
Service Manager->SNMP Provider   
TCP (SSL)   
1618   
Elsbeth PowerContact   
Server (EPC Server)   
EPC Server<->ECM   
TCP   
1500   
EPC Server<->ECM   
TCP (SSL)   
1600   
EPC Server<->ECS   
HTTP   
7927   
EPC Server<->ECS   
TCP   
7928   
EPC Server<->ECS   
HTTPS   
7980   
EPC Server 6.x<->EPC Gadget 6.x   
HTTP   
8888   
EPC Server 9.x<->EPC Gadget 9.x   
HTTP   
7927

CCSP Prescriptive Architecture - 7.3 and 7.4   
85   
Sytel mode   
In Sytel mode, the Predictive Dialer Service (PDS) is the component that connects and manages all   
communication between Dialers and the CCSP platform.   
It establishes an IPC connection with the:   
•   
CFM to retrieve configuration   
•   
AIS for agent state handling   
•   
Dialing Plan to find the right VCS for a specific phone number   
•   
VCS to place the call   
Setup   
The following diagram shows the Predictive Dialer architecture.   
   
Figure 32: Predictive Dialer architecture   
CCSP’s Campaign Manager (CM) server handles tenant campaigns and retrieves numbers to dial   
from a database. In the preceding diagram, each tenant CM is color-coded. CM can be deployed in   
three different ways:   
•   
On dedicated servers — each tenant can have one or more dedicated servers running a CM.   
Typically one CM can handle multiple campaigns, however it is perfectly possible for large   
systems or for distributed environments to have multiple CMs per tenant, each serving a   
separate group of campaigns.   
•   
Collocated — multiple CM instances can be deployed on the same server, each one   
configured to listen on a specific port. The CMs could belong to one or more tenants, but each

CCSP Prescriptive Architecture - 7.3 and 7.4   
86   
CM must belong to a single tenant. We recommend not having more than 8 CMs on a single   
server.   
•   
Remote — each tenant hosts his own CM and campaign database. This is typical of large   
enterprises who want to have full control of their campaigns.   
   
The following restrictions apply when sizing a Dialer:   
•   
There is a 1-to-1 match between a PDS server and a Dialer. When scaling a system, you must   
always add PDS/Dialers in pairs.   
•   
PDS servers can connect to any number of AIS/VCS/Dialing Plan servers. For the Dialing Plan it   
uses the Web Farm load-balanced IP.   
•   
A Dialer can connect to one or more CMs. However a CM can only connect to a single Dialer.   
•   
To increase business continuity, a cold-standby pair must be used and started if failure occurs.   
Active/passive high availability option is planned.   
•   
Multiple CMs cannot run the same campaigns. Doing this inevitably leads to an overlap in   
dialed numbers.   
•   
Multiple dialers cannot run campaigns for the same tenant.   
Dialer ports   
The following ports are used by the Predictive Dialer.   
Service   
Protocol   
Port   
Direction   
CFM IPC   
TCP   
14001   
PDS->CFM   
AIS IPC   
TCP   
14005   
PDS->AIS   
or PDS IPC   
TCP   
14015   
AIS->PDS   
VCS IPC   
TCP   
14010   
PDS->VCS   
or PDS IPC   
TCP   
14015   
VCS->PDS   
HTTP/HTTPS   
TCP   
80/443   
PDS->DP   
Dialer control\*   
TCP   
6498 to 6502   
PDS->Dialer   
State Viewer   
TCP   
6503   
Mgt. Client->Dialer   
CM protocol   
TCP   
6565   
Dialer->CM   
Namespace   
TCP   
6808 to 6810   
Mgt. Client->Dialer   
Other   
TCP   
7400, 7401   
Mgt. Client->Dialer   
\* 6498 and 6499 are used between the Dialer and PDS.

CCSP Prescriptive Architecture - 7.3 and 7.4   
87   
7: Virtualization   
CCSP supports production platform installation on VM Ware ESX Server. Development and test   
platforms can be also installed on Windows-based VMWare Server. You must have enough   
hardware resources for each virtual machine. It is also critical to monitor the usage of the following   
parameters on each VM:   
•   
Multiple cores   
•   
RAM   
•   
Network Bandwidth   
•   
Disk I/O   
   
Virtualization can reduce operating costs and provide additional high availability at the virtual   
server level.   
Note   
•   
Operating system requirements from physical server-based installation are also applicable to   
VMWare-based platforms.   
•   
Due to virtualization, CPU and memory consumption on VMWare ESX may increase up to 10%   
and on VMWare Server up to 30% higher than in regular server installation.   
•   
Non-CCSP components such as SQL and Exchange servers, installed as a part of the platform,   
should adhere to their vendor’s guidelines.   
•   
CCSP currently supports only alternative delivery for virtual desktops.   
   
When using virtualization for the platform, we recommend optimizing the configuration for the   
requirements of each server.

CCSP Prescriptive Architecture - 7.3 and 7.4   
88   
8: IP multimedia subsystem   
•   
Overview   
•   
CCSP integration   
   
CCSP VCS acts as an IP Multimedia Subsystem (IMS) Application Server (AS). It can also be used as a   
specialized Media Resource Function (MRF) for the contact center functionality, such as recording   
and conferencing, or be integrated with third-party MRFs.   
Overview   
The following IMS elements are important in the CCSP deployment:   
•   
Application Servers (AS) — contain all the user services that are deployed across the IMS   
network. From the IMS perspective, CCSP can be viewed as a contact center Application Server.   
Calls to and from the contact center are routed via IMS to the end customers.   
•   
Interrogating Call Session Control Function (I-CSCF) — is a query SIP function located at the   
edge of an administrative domain. Its IP address is published in the DNS of the domain using   
NAPTR and SRV type of DNS records. This enables the remote servers to find it and use it as a   
forwarding point for SIP packets to this domain, for example, for registering. The I-CSCF   
queries the HSS using the Diameter Cx interface to retrieve the user location and then routes   
the SIP request to its assigned S-CSCF.   
•   
Serving-CSCF (S-CSCF) — is the central node of the signaling plane. It is a SIP server but also   
performs session control. It is always located in the home network. It uses the Diameter Cx   
interface to the HSS to download and upload user profiles.   
The S-CSCF’s main functions are:   
o   
Handles SIP registrations, which allows it to bind the user location, for example, the IP   
address of the phone, and the SIP address.   
o   
Sits on the path of all signaling messages and can inspect every message.   
o   
Decides to which application server(s) the SIP message will be forwarded.   
o   
Provides routing services, typically using Electronic Numbering (ENUM) lookups.   
o   
Enforces the policy of the network operator.   
   
You can have multiple S-CSCFs in the network for load distribution and high availability.   
•   
Proxy-CSCF (P-CSCF) — is a SIP Proxy that is the first point of contact for the IP phone. It can   
be located either in the visited network, in full IMS networks, or in the home network, when   
the visited network is not yet IMS compliant. Some networks may use a Session Border   
Controller for this function. The IP phone either discovers its P-CSCF with DHCP or it is   
assigned statically as a SIP Proxy in the phone configuration.   
The P-CSCF’s main functions are:   
o   
Is assigned to an IMS terminal during registration, and does not change for the duration   
of the registration. As such it acts as a Registrar.

CCSP Prescriptive Architecture - 7.3 and 7.4   
89   
o   
Sits on the path of all signaling messages and can inspect every message.   
o   
Authenticates the user and establishes an IPSec security association with the IMS terminal   
(optional). This prevents spoofing attacks and replay attacks and protects the privacy of   
the user. Other nodes trust the P-CSCF and do not have to authenticate the user again.   
o   
Can also compress and decompress SIP messages using SigComp, which reduces the   
round-trip over slow radio links (optional).   
o   
May include a Policy Decision Function (PDF), which authorizes media plane resources, for   
example, quality of service (QoS) over the media plane. It is used for policy control and   
bandwidth management. The PDF can also be a separate function.   
o   
Generates charging records.   
   
•   
Home Subscriber Server (HSS) or User Profile Server Function (UPSF) — is a master user   
database that supports the IMS network entities that actually handle calls. It contains the   
subscription-related information (user profiles), performs authentication and authorization of   
the user, and can provide information about the user's physical location.   
•   
Media Gateway Controller Function (MGCF) — does call control protocol conversion   
between SIP and ISUP (SS7 network signaling protocol). It also controls the resources in a   
Media Gateway (MGW) across an H.248 interface.   
•   
Media Gateway (MGW) — interfaces with the media plane of the PSTN network by converting   
between RTP and PCM. It can also transcode when the codecs do not match, for example,   
CCSP might use G.729 while PSTN might use G.711.

CCSP Prescriptive Architecture - 7.3 and 7.4   
90   
CCSP integration   
CCSP integration requires the following interfaces and setup.   
Interfaces   
The following diagram shows how the CCSP solution fits into the IMS network.   
   
Figure 33: IMS deployment   
The following interfaces are important in this integration.   
Interface   
Protocol   
Description   
ISC   
SIP   
Used to exchange messages between S-CSCF and AS using   
predefined routing rules or URI (default behavior).   
Ma   
SIP   
Used to directly forward SIP requests that are destined to a Public   
Service Identity (PSI) hosted by the AS. Between I-CSCF and AS.   
Mg   
SIP   
MGCF converts ISUP signaling to SIP signaling and forwards SIP   
signaling to I-CSCF.   
Mw   
SIP   
Used to exchange messages between CSCF servers.

CCSP Prescriptive Architecture - 7.3 and 7.4   
91   
Interface   
Protocol   
Description   
Cx   
Diameter   
Used to communicate between I-CSCF/S-CSCF and HSS. Used to   
query user data or routing information.   
Gm   
SIP   
Interface between the external IP networks and the P-CSCF. This is   
the interface the agent IP phones use to connect to the IMS.   
Ut   
HTTP(S)   
This is an external interface to the IMS that is used for CTI and call   
control.   
-   
RTP   
All media is routed outside of the IMS network. This includes RTP   
voice streams.   
Setup   
The following configuration is required for CCSP to route calls in an IMS network:   
1.   
A SIP entity must be defined in HSS for each service phone number (SIP URI based on DNIS)   
handled by the CCSP platform. Typically there is one or more DNIS for each tenant. The SIP   
entities can be configured as one of the following:   
o   
Routing triggers in the HSS that get downloaded into the S-CSCF when the VCS registers   
with the S-CSCF.   
o   
By configuring static Public Service Identifiers (PSIs) for each tenant.   
   
2.   
The VCS load-balanced IP address must be resolvable through DNS.   
3.   
Agent identities (aliases) must be provisioned in the HSS. Only inbound calls (VCS to agent)   
should be allowed, otherwise, if an agent is allowed to out-dial, it may lead to synchronization   
problems between the IMS state and the contact center state.   
4.   
Agent IP phones must register to the IMS network via P-CSCF using the agent’s IMS identity.   
5.   
All outbound calls from the VCS must be routed by the S-CSCF to the appropriate terminal. This   
includes both the outbound call leg and the agent leg.   
6.   
CCSP UI must communicate with the AIS server through an external Ut link.

CCSP Prescriptive Architecture - 7.3 and 7.4   
92   
Appendix A: System environment   
configuration   
•   
Active Directory and domain controller   
•   
Flexible Single-Master Operation roles   
•   
DNS   
•   
Hard drive configuration   
•   
Server requirements   
•   
Client requirements   
Active Directory and domain controller   
CCSP’s security relies on Windows Integrated Authentication for all user access to the platform.   
Each CCSP platform represents an entire Active Directory forest. DNS servers must run on each   
AD/DC server to take advantage of Active Directory integration and seamless replication. Each CCSP   
can contain several MSRs that do not all need to be part of the domain.   
The Active Directory structure is created automatically and should not be re-arranged. Tenants   
have their own individual Organizational Unit (OU) in which contact center agents, groups, and   
supervisors are created. There are some preconfigured groups in each OU, that have to remain   
unchanged to ensure correct platform operation.   
The following diagram shows the Active Directory structure.   
   
Figure 34: Active Directory structure

CCSP Prescriptive Architecture - 7.3 and 7.4   
93   
Note   
•   
When creating a second forest for upgrading, no forest trusts should be established.   
•   
CCSP uses a dedicated Active Directory and cannot use the organization’s Active Directory. If   
required, there are methods to synchronize between the two Active Directories.   
MSDCS section in DNS server   
When the DNS root domain of a new Active Directory forest is created on a Windows Server-based   
domain controller, two DNS zones are automatically created:   
•   
A zone for the forest root domain. This zone is replicated between all domain controllers in   
that domain.   
•   
A zone for the \_msdcs.ForestName subdomain. This zone is stored in the forest-wide DNS   
application directory partition. This partition replicates to all Windows Server-based domain   
controllers in the forest that are running the Windows Server DNS Server service.   
Global catalog   
The global catalog (GC) is a distributed data repository that contains a searchable, partial   
representation of every object, in every domain, in a multi-domain Active Directory forest.   
In a single site deployment, for redundancy, we recommend making both domain controllers   
Global Catalog servers. In a redundant site deployment, there should be at least one GC on each   
site.   
Replication   
Replication of updates to Active Directory objects is transmitted between multiple domain   
controllers to keep replicas of directory partitions synchronized. In a distributed CCSP environment,   
domain controllers for the same domain are commonly placed in more than one site, which makes   
AD replication crucial.   
Therefore, replication must often occur both within sites and between sites to keep the CCSP   
domain and forest data consistent among domain controllers that store the same directory   
partitions.   
Customers should verify that the bandwidth used between sites is sufficient for the platform traffic,   
including AD/DC/DNS synchronization. The latter is negligible compared to the other applications   
using the inter-site bandwidth, such as SQL replication and IPC traffic.   
Do not use site configuration in the AD/DC/DNS. The platform should be configured as if in a single   
site.   
Typically in a single site CCSP environment, two AD/DC servers are required, both serving GCs.   
Replication is seamless. The same configuration should be verified in multi-site CCSP environments   
where a WAN link is used between sites.   
The following diagram shows single site Active Directory replication.

CCSP Prescriptive Architecture - 7.3 and 7.4   
94   
   
Figure 35: Single site Active Directory replication   
Inter-site   
The following diagram shows multi-site Active Directory replication.   
   
Figure 36: Multi-site Active Directory replication   
   
If there is a firewall between replicated sites, the following ports should be opened.   
Active Directory inter-site replication ports   
Service   
Protocol   
Port   
Direction   
RPC endpoint mapper   
TCP/UDP   
135   
Both   
NetBIOS name service   
TCP/UDP   
137   
Both   
NetBIOS datagram service   
UDP   
138   
Both   
NetBIOS session service   
TCP   
139   
Both

CCSP Prescriptive Architecture - 7.3 and 7.4   
95   
Service   
Protocol   
Port   
Direction   
RPC static port for AD   
replication   
TCP   
1024-65535\*   
Both   
RPC static port for FRS   
TCP   
1024-65535\*   
Both   
SMB over IP (Microsoft-DS)   
TCP/UDP   
445   
Both   
LDAP   
TCP   
389   
Both   
LDAP ping   
UDP   
389   
Both   
LDAP over SSL   
TCP   
636   
Both   
Global catalog LDAP   
TCP   
3268   
Both   
Global catalog LDAP over SSL   
TCP   
3269   
Both   
Kerberos   
TCP/UDP   
88   
Both   
DNS   
TCP/UDP   
53   
Both   
WINS resolution (not   
required)   
TCP/UDP   
1512   
Both   
WINS replication (not   
required)   
TCP/UDP   
42   
Both   
\* See note below on how to configure a specific port.   
   
Note   
•   
In multi-site replications, there must be at least one GC server at each site.   
•   
If both sites are linked with a high-bandwidth VPN connection (mandatory), no Bridge Heads   
are necessary and replication occurs seamlessly as if it were a single site.   
•   
Limit RPCs across the firewall by editing the registry of all your DCs.   
Flexible Single-Master Operation roles   
Active Directory domain controllers in the CCSP domain support multi-master updates for the   
replication of objects, such as user and computer accounts, in the Active Directory. In a multi-  
master model, objects and their properties can originate on any domain controller in the domain   
and become authoritative with replication.   
However, certain domain and enterprise-wide operations that are not well-suited to multi-master   
placement reside on a single domain controller in the domain or forest. The advantage of single-  
master operation is that it prevents the introduction of conflicts while an operation master is   
offline. Having a single-operation master means, however, that the Flexible Single-Master

CCSP Prescriptive Architecture - 7.3 and 7.4   
96   
Operation (FSMO) role owner must be available when dependent activities in the domain or   
enterprise take place, or to make directory changes associated with that role.   
Overview   
There are five FSMO roles that must reside on a single AD/DC in the domain:   
•   
Schema Master (SM) — the schema master DC controls all updates to and modifications of   
the schema. Once the schema update is complete, it is replicated from the schema master to   
all the other DCs in the directory. To update the schema of a forest, you must have access to   
the schema master. There is one SM per forest.   
•   
Domain naming master (DNM) — the DNM DC controls the addition or removal of domains   
in the forest. This DC is the only one that can add or remove a domain from the directory. It   
can also add or remove cross references to domains in external directories. There is one DNM   
per forest.   
•   
Infrastructure Master (IM) — when an object in one domain is referenced by another object   
in another domain, it represents the reference by the GUID, the SID (for references to security   
principals), and the DN of the object being referenced. The IM role holder is the DC   
responsible for updating an object's SID and distinguished name in a cross-domain object   
reference. There is one IM per domain.   
•   
Relative ID (RID) Master — the RID master is responsible for processing RID pool requests   
from all domain controllers in a particular domain. When a DC creates a security principal   
object such as a user or group, it attaches a unique Security ID (SID) to the object. This SID   
consists of a domain SID (the same for all SIDs created in a domain) and a relative ID (RID) that   
is unique for each security principal SID created in a domain. Each DC in a domain is allocated   
a pool of RIDs that it is allowed to assign to the security principals it creates. When a DC's   
allocated RID pool falls below a threshold, that DC issues a request for additional RIDs to the   
domain's RID master. The domain RID master responds to the request by retrieving RIDs from   
the domain's unallocated RID pool and assigns them to the pool of the requesting DC. There is   
one RID per domain.   
•   
PDC Emulator — the PDC emulator is needed to synchronize time in an enterprise. Windows   
includes the W32Time (Windows Time) time service that is required by the Kerberos   
authentication protocol. All Windows-based computers within an enterprise use a common   
time. The purpose of the time service is to ensure appropriate common time usage by   
ensuring that the Windows Time service uses a hierarchical relationship that controls authority   
and does not permit loops. There is one PDC emulator per domain.

CCSP Prescriptive Architecture - 7.3 and 7.4   
97   
The following diagram shows FSMO roles.   
   
   
Figure 37: FSMO roles   
The following restrictions apply when assigning FSMO roles:   
•   
SM and DNM must always reside together and on the GC DC.   
•   
PDC and RID are best installed on a non-GC DC since they can be fairly resource consuming.   
•   
IM must never be installed on a GC DC. Furthermore, the IM DC must have a direct replication   
partner with a GC DC.   
FSMO failures   
FSMO roles do not automatically transfer to another DC if failure occurs. This must be done   
manually. Because FSMO roles are not critical to the Active Directory operation, there is a certain   
time window available for trying to restore a failed FSMO DC.   
The following table describes the consequence of each FSMO failure for a long period of time.

CCSP Prescriptive Architecture - 7.3 and 7.4   
98   
FSMO failure consequences   
FSMO role   
Loss implications   
SM   
The schema cannot be extended. However, in the short term no one will notice a   
missing Schema Master unless you plan a schema upgrade during that time.   
DNM   
Unless you are going to run dcpromo, you will not miss this FSMO role.   
PDC   
The existing DCs should have enough unused RIDs to last some time, unless   
you're building hundreds of users or computer object per week.   
RID   
Will be missed soon. There will be no time synchronization in the domain, you will   
not be able to change or troubleshoot group policies and password changes will   
become a problem.   
IM   
Group memberships may be incomplete. If you only have one domain, there will   
be no impact.   
   
For an FSMO DC failure or site failure, the following steps must be taken into consideration:   
•   
Always try to bring online the failed server. If the failed server can come online but is not   
stable, while online, transfer the FSMO roles to the failover partner. We always recommend   
transfering FSMO roles when both DCs are online and operational. This action is called   
Transferring roles.   
•   
If the failed FSMO DC cannot be brought back online, convert the failover DC into a FSMO   
master. Once this is done, you must ensure that the original FSMO is never brought online   
again in the same domain/forest. See FSMO sizing restrictions. This action is called Sizing roles.   
   
The following tables show the FSMO sizing restrictions and user access rights.   
FSMO sizing restrictions   
FSMO role   
Restrictions   
SM   
Original DC server must never be brought back online. Windows operating system   
must be reinstalled before reusing the server.   
DNM   
RID   
PDC   
Original server can be brought back online and FSMO roles re-transferred.   
IM

CCSP Prescriptive Architecture - 7.3 and 7.4   
99   
FSMO user access rights   
FSMO role   
Administrator must be a member of   
SM   
Schema admins   
DNM   
Enterprise admins   
RID   
Domain admins   
PDC   
IM   
DNS   
CCSP must be installed in its own dedicated forest and domain. This domain can either be public or   
private.   
Because of multi-tenancy and to facilitate zero-downtime tenant migrations, including email   
routing changes, we recommend that the platform resolves as a subdomain of a public domain.   
This domain does not need to be the server’s domain itself. Migrating a tenant from one platform   
to another is mainly a matter of updating public DNS record entries. Although only one domain is   
active during normal operations, while performing upgrades both domains serve incoming   
connections as tenants are split between platforms.   
Note   
•   
It is very important to decide on the domain name prior to the installation.   
•   
You cannot change a domain name after the domain has been created and CCSP has been   
installed. It is also very important when planning a domain name to considerer how this will   
relate to the DNS domain name. The CCSP platform needs only one domain with two domain   
controllers for high availability and fault tolerance. Platform preparation includes installation   
of two Windows 2012 DCs and DNS servers. We highly recommend integrating the platform   
DNS in the Active Directory. Other DNS servers should not be integrated into the Active   
Directory.   
DNS architecture   
The way you plan to use the CCSP platform affects the DNS architecture.   
For security reasons, we recommend that the platform domain itself not be resolvable from the   
Internet. The platform server’s domain can be totally private while agents are routed to the   
platform using public DNS entries.

CCSP Prescriptive Architecture - 7.3 and 7.4   
100   
The following diagram shows an example DNS setup.   
   
Figure 38: DNS setup example   
   
Note   
•   
We recommend that you have a private internal DNS and a public external DNS.   
•   
CNAME records cannot be the same as the top SOA domain name in the zone file.   
•   
Make sure all MX records have an associated PTR record. Some mail forwarders or spam   
filter software block email from sources that cannot be reversely resolved.   
•   
Public UPNs of tenants and CCSP FQDN of external interfaces do not change during a   
migration since DNS record updates are done at the host.com level.   
•   
DNS servers in the host.com domain are not AD/DC integrated and are simply used for pure   
naming resolution purposes.   
•   
For information about configuring MX records and routing email, see Email routing.

CCSP Prescriptive Architecture - 7.3 and 7.4   
101   
Internal CCSP DNS servers − setup and zone replication   
Zone replication scope should be set to All DNS servers in the same Active Directory domain or to All   
Domain Controllers in the same Active Directory domain.   
Verify that zone transfer is used between all DNS servers. For full redundancy, use replmon, which   
comes with the Windows support tools.   
Default DNS timeout settings must be modified to accommodate CCSP’s Trap Monitor. The new   
settings should permit DNS to perform verification every 1 to 5 seconds.   
Required CCSP DNS entries   
For non-service-provider platforms, the following DNS entries should be set up on the local DNS   
servers managed by CCSP. Entries in the following table do not take into account SOA, CNAME, PTR,   
MX, and NS records and show only A records.   
Interfaces   
RR Type   
CFM   
A (Load Balanced IP)\*   
AIS (Internal VLAN)   
A (Load Balanced IP)\*   
AIS (IPC VLAN)   
A (Load Balanced IP)\*   
VCS (External VLAN)   
A (Load Balanced IP)\*   
VCS (Internal VLAN)   
A   
RTR/MSRS/WEB (Internal VLAN)   
A (Load Balanced IP)\*   
RTR/MSRS/WEB (IPC VLAN)   
   
MCS (IPC VLAN)   
A   
LS (IPC VLAN)   
A   
DPS (Internal VLAN)   
A (Load Balanced IP)\*   
CMS 1,2 (IPC VLAN)   
A   
StatServer 1,2 (IPC VLAN)   
A   
All other IPC interfaces   
A   
\* If using DNS Round-Robin, one CNAME + A RR per server is needed to map all FQDNs.   
   
For service provider platforms where agents do not access the DNS servers, no load balanced DNS   
entries are necessary except for CFM (IPC VLAN)and web servers (IPC or Internal VLAN).   
Forwarders must be configured in the CCSP-managed DNS servers to point back up to the root   
domain.

CCSP Prescriptive Architecture - 7.3 and 7.4   
102   
Dynamic IP addresses   
You must have DNS servers on the customer side that are configured to the static IP addresses of   
the CCSP servers.   
Hard drive configuration   
Hard drive redundancy is an extremely important part of the CCSP platform. Not only can it   
positively impact the performance of the contact center but it can also improve upgrades and   
recovery situations. Since not all servers can accommodate the same number of hard drives, we   
can only recommend best practices. When purchasing hardware to build a system, administrators   
should always be aware of the proposed hard drive setup.   
Use SAS or FC drives as much as possible for better performance, especially inside a SAN enclosure   
for network data such as clusters and the CCSP databases. See SQL.   
Keep the operating system and CCSP components on separate RAID-1 array (mirrored drives - two   
drives total). This setup provides a quick option for rollback if critical files have been changed or   
corrupted.   
If possible, have four or more hard drives for all servers. Create a separate local array (RAID-1) for   
any dynamic folders such as CCSP logs, IIS logs, and temporary files as in the LS or DTS.   
The following table shows the hard drive setup.   
CCSP service   
Total number of drives   
Comments   
LS   
OS + CCSP (RAID-1)   
Syslog + TMP files (RAID-1)   
LS stores all messages locally if SQL   
database is not reachable, therefore we   
recommend a separate drive for high-  
volume writes.   
VCS, RTP, IVVR   
OS + CCSP(RAID-1)   
Syslog + Recordings (RAID-1)   
Add extra RAID-1 array for local   
recordings and logs.   
DTS   
OS + CCSP(RAID-1)   
Syslog + TMP (RAID-1)   
Add extra RAID-1 array for recording files   
that are copied and processed from the   
VCS.   
Microsoft SQL   
OS + SQL (RAID-1)   
Page File (RAID-10)   
Add a separate faster RAID-10 drive for   
the Windows page file. The page file   
should be at least 2xRAM size. Database   
drives are described separately below.   
Microsoft Exchange   
OS + Exchange (RAID-1)   
Page File (RAID-10)   
Add a separate faster RAID-10 drive for   
the Windows page file. The page file   
should be at least 2xRAM size. Mail   
folders and queues are described   
separately below.

CCSP Prescriptive Architecture - 7.3 and 7.4   
103   
Note   
Never implement disk mirroring in software. Always use RAID controllers to offload the CPU.   
SQL   
Because SQL server is at the heart of the CCSP platform, it is very important to use the best   
performance options when designing volumes on SAN drives or local drives.   
CCSP has the following types of database:   
•   
System administrator database — stores generic data about the CCSP platform such as   
system-wide constants.   
•   
Configuration database — stores all the sever, tenant and agent configurations. This table   
typically has a lot of changes initially when new tenants are provisioned, but afterwards tends   
to be more read-oriented.   
•   
Tenant Recording database — stores all agent recordings. High volume of writes, low volume   
of reads.   
•   
Tenant Historical database — stores all contact center events. High volume of writes, low   
volume of reads, depending on the Reports.   
•   
Tenant Data warehouse — stores aggregated contact center events. Low volume of writes,   
low to high volume of reads, depending on the Reports.   
•   
Tenant messaging database (active MSR DB) — stores data for new/ongoing emails,   
callbacks, and voicemails.   
•   
Tenant messaging archive database (archive MSR DB) — stores data for handled/rejected   
emails, callbacks, and voicemails.   
•   
Tenant Designer database — stores all call routing scripts used by Designer. Low volume of   
writes, low to high volume of reads.   
   
RAID-10 has faster read-write performance than RAID-5, so it should be used mainly for transaction   
log files. RAID-5 is more economical for large amounts of stored data that is less performance   
sensitive.   
Use separate physical disks for each volume, do not partition drives. Do not stripe across volumes   
so that each database is completely separate from all others.   
For very large service provider platforms and above 1000 agents, we recommend separating the   
Tempdatabase of the historical instance to dedicated RAID-10 for the MDF and RAID-10 for the LDF.   
This is in case historical reports are used extensively and generated from the historical databases   
themselves. In smaller platforms this is unnecessary and the Tempdatabase can reside on the same   
array as the historical database but should be split into MDF and LDF on the matching arrays.   
For enterprise-level storage, you can use storage virtualization technology as follows:   
•   
The heartbeat should always be on a separate NIC, configured in half duplex, 10 MB. We highly   
recommend a cross-over cable (not switch mediated) for a two-node cluster.   
•   
No other applications should utilize the Quorum disk. This disk should only be used for the   
actual cluster and cluster database.   
•   
Minimum size of the Quorum disk partition is 500 MB.

CCSP Prescriptive Architecture - 7.3 and 7.4   
104   
•   
Always name Quorum disk Q:\.   
•   
Each major component on the cluster should be configured in its own resource group: SQL   
cluster group (for each instance of SQL), MSTDC group and Exchange cluster group.   
•   
Each cluster group should have its own hard drive resource on the storage.   
•   
Each cluster resource should have its own IP address and virtual machine name.   
•   
For the administrator’s convenience, create a reverse lookup for all the virtual IPs and virtual   
instance names used in the cluster in DNS.   
•   
Always disable NetBIOS and the WINS client on the heartbeat interface of the cluster.   
•   
Make sure that the heartbeat interface does not register with DNS. Typically the heartbeat   
connection should be a point-to-point connection between the two nodes.   
   
Ensure that the public network NIC is the first binding (at the top of the binding list) in the TCP/IP   
bindings.   
For small systems of up to a few hundred agents, shared storage should be a SCSI attached storage   
for two or four nodes. For large scale systems, storage should be part of a SAN architecture using   
fiber channel or iSCSI solutions with fiber channel storage solutions. SAN architecture can be used   
for large scale systems with a unified storage solution and disaster recovery architectures in which   
the SAN is distributed between two remote sites.   
Server requirements   
The following table shows the CCSP components and their recommended configuration.   
Service   
CPU cores   
Memory   
Disk   
ACD   
2   
4   
80   
AD   
2   
4   
80   
ADP   
2   
4   
80   
AIS   
2   
4   
80   
CFM+Logging   
2   
4   
80   
DTR   
2   
4   
80   
LCS   
2   
4   
80   
MCS   
4   
4   
80   
MS   
4   
4   
80   
RTR + Dashboard WS   
2   
8   
80   
VCS   
4   
4   
80

CCSP Prescriptive Architecture - 7.3 and 7.4   
105   
For messaging system components, we recommend the following minimum specifications, per   
machine:   
•   
Install the MS on a separate machine with 4 CPU cores and 4 GB RAM   
•   
Install the MCS on a separate machine with 4 CPU cores and 4 GB RAM   
•   
Install the active MSR DB on a separate machine with 4 CPU cores and 16 GB RAM   
•   
Install the archive MSR DB on a separate machine with 4 CPU cores and 16 GB RAM   
•   
If using Exchange as MSR, install MS Exchange on a separate machine with 4 CPU cores and 16   
GB RAM   
   
For SQL Server machines, we recommend the following minimum specifications, per machine:   
•   
For the configuration database, installed on separate machine: 2 CPU cores and 8 GB RAM.   
•   
For the recording, historical, and system administrator databases installed on the same   
machine: 4 CPU cores and 16 GB RAM   
   
For the integration with Elsbeth, see How to plan an Elsbeth system for CCSP, in Appendix E: CCSP-  
Elsbeth integration.   
Client requirements   
Note   
•   
Minimal requirements are not optimal requirements, so processing may be slow.   
•   
CCSP Replay supports multi-interaction recording using FFmpeg, which encodes screen   
recording to MP4 using the H264 codec.   
•   
If the screen resolution is higher than 2048×1536, for the encoder to work, screen recording   
forces the width and height to be 2048×1536. We therefore recommend a maximum of   
1920x1080 resolution.   
   
The minimum specifications for clients are:   
•   
2 CPU cores   
•   
With screen recording, 2 GB RAM. Otherwise, 1 GB RAM.   
•   
1x1 GB network card   
•   
10 GB hard drive   
•   
Screen resolution – although CCSP supports all resolutions, recording may be slower for   
resolutions higher than 1920x1080 pixels

CCSP Prescriptive Architecture - 7.3 and 7.4   
106   
Appendix B: SQL Server Always-On and   
mirroring   
•   
Synchronous mode   
•   
Asynchronous mode   
•   
Disk image   
•   
Blade servers, SAN configuration and redundancy   
•   
Voice quality   
Synchronous mode   
In synchronous mode, in both Always-On and mirroring, transaction logs are sent to the secondary   
or mirror nodes where the data is stored on the disk, before the client application is allowed to   
send more data to the principal. This ensures that 100% of the transactional data is preserved and   
replicated. The witness server is the arbiter that monitors SQL database availability and controls   
automatic failovers. When using mirroring, the system can be configured without a witness server.   
However, failover is then a manual process, which we do not recommend for CCSP services.   
A Quorum, or logical pair relationship, must be formed at all times between at least two nodes in   
the Always-On or mirrored system. The database accepts connections as long as a Quorum is   
maintained.

CCSP Prescriptive Architecture - 7.3 and 7.4   
107   
The following diagram shows synchronous mode.   
   
Figure 39: Synchronous mode   
   
The diagram shows the following scenarios:   
1.   
Operation under normal circumstances. The primary node accepts client data and replicates to   
all secondary nodes. Quorum is formed between primary and secondary.   
2.   
When the secondary node fails, the primary node forms Quorum with the witness and any   
other secondary nodes, and keeps accepting data from client applications. Replication stops to   
any failed secondary nodes. When the failed nodes are restored, they are resynchronized.

CCSP Prescriptive Architecture - 7.3 and 7.4   
108   
3.   
When the witness fails, Quorum is maintained between the primary node and any secondary   
nodes, and log replication continues. However should the primary node fail, failover will not   
happen automatically.   
4.   
When the primary node is down, one of the secondary nodes assumes the role of primary and   
establishes Quorum with the witness and any other secondary nodes. Client applications are   
automatically diverted to the new primary node by the witness.   
5.   
Entire network fails. No Quorums can be formed. All nodes become disconnected. Client   
applications cannot write to the database.   
   
Note   
•   
Use synchronous mode when both participating nodes are in the same LAN and if the   
bandwidth between them is robust enough (> 1 Gbps).   
•   
If possible, the witness server should be installed on a third server, such as Exchange, the   
AD/DC, or a dedicated witness server. When using mirroring, it can be installed on the mirror   
if another server is not available.   
•   
When using mirroring, if the witness is installed on a separate computer, the witness server   
only needs the free SQL Express edition. SQL Server does not need to run on the witness   
server when using Always-On.   
Asynchronous mode   
Asynchronous mode is used in environments that cannot afford to wait for secondary nodes to   
store the data. This increase in performance comes with a price, which does not guarantee 100%   
data integrity. Furthermore, failovers to asynchronous secondary nodes is a manual process, which   
makes it a more tedious process for a server crash.   
The following diagram shows asynchronous mode.

CCSP Prescriptive Architecture - 7.3 and 7.4   
109   
   
Figure 40: Asynchronous mode   
The diagram shows the following scenarios:   
1.   
Operation under normal circumstances. Replication happens on a best-try basis.   
2.   
If a secondary node fails, the primary node still accepts connections from the client   
application.   
3.   
If the primary node fails, secondary nodes remain in a disconnected state and do not accept   
any connections. Manual intervention is required to promote a secondary node to the role of   
primary node, but there might be loss of data if not all logs have been propagated. If the   
primary node is restored without activating the secondary node, then data replication   
resumes.   
   
Note   
•   
Use synchronous mode when bandwidth between participating nodes is limited and when   
there is network latency.   
•   
When using mirroring, do not use a witness, as there is no guarantee that all data has been   
replicated and this can result in a loss of data if there is a failover.   
•   
Requires SQL Enterprise edition.   
   
Using Always-On allows nodes to be configured in a combination of synchronous and   
asynchronous mode, providing more options when configuring your environment. Mirroring only   
allows for each database to have a single secondary node, so you must choose whether the primary   
and secondary should be replicated synchronously or asynchronously.

CCSP Prescriptive Architecture - 7.3 and 7.4   
110   
Disk image   
When there is a stable CCSP installation in place, we highly recommend creating a ghost image of   
the OS/CCSP drive and backing up the image files on a NAS or SAN outside of the local server. This   
can help when the hard drive is replaced or system files have been corrupted and the server needs   
to be quickly reinstalled.   
Note   
After any change of configuration or files on the server, a new image file of the drive should be   
made.   
Blade servers, SAN configuration and redundancy   
Higher availability can be achieved by implementing the system on blade servers with SAN storage.   
When one of the blades fails, a redundant blade replaces the failing server. If the hard drive of the   
failed server resides on the SAN and not internally, the extra blade can replace the failing server   
and use the former server’s hard drive. The server down time is reduced to almost zero.   
Blade servers are easier to manage, consume less power/cooling, and consume less space in the   
server room.   
Also, because all cabling is built into the blades enclosure, network cabling is more secure and   
fault-tolerant.   
We recommend implementing CCSP platforms on blades.   
Voice quality   
Voice quality is one of the most important aspects of customer satisfaction in contact center   
products. Poor sound, coupled with signaling delays and network packet loss, can severely impact   
user experience. It is therefore paramount to understand and properly tune the VoIP network to   
achieve maximum performance, even under load conditions.

CCSP Prescriptive Architecture - 7.3 and 7.4   
111   
Appendix C: QoS   
•   
DiffServ   
•   
Microsoft QoS Packet Scheduler   
•   
MOS   
•   
Jitter   
•   
Packet loss   
•   
Packet delay   
   
When installing CCSP, it is important that all end points that participate in the VoIP call can mark   
VoIP traffic with DiffServ. We highly recommend that all network elements that transit VoIP traffic   
can prioritize the traffic according to DiffServ. The result should be an end-to-end QoS network.   
We also highly recommend that network elements can prioritize VoIP traffic according to the   
source and/or destination IP address or TCP/UDP ports.   
DiffServ   
Differentiated Services (Diffserv) is a method for adding QoS to an IP network and is an IETF   
standard.   
Diffserv operates solely at the OSI layer 3, the network layer, using the first seven bits of the IP Type   
of Service (TOS) field as the Diffserv byte. The Diffserv byte is used for traffic classification and is   
defined in IETF RFC 2474 and RFC 2475.   
Other RFCs define the particular DiffServ markings or DiffServ CodePoint (DSCP) field values.   
Among those are per-hop behavior (PHB) classes, Assured Forwarding (AF) as defined in RFC 2597,   
and Expedited Forwarding (EF) as defined in RFC 3246. Real-time multimedia systems need DSCP   
values that are generally recognized and accepted across the industry as appropriate DSCP values   
for call signaling, audio, and video transport.

CCSP Prescriptive Architecture - 7.3 and 7.4   
112   
The following diagram shows an IP Header with the DiffServ bit field.   
   
Figure 41: IP Header with the DiffServ bit field   
The following table defines the generally agreed DiffServ VoIP values used in the industry today,   
mostly documented in RFC 4594.   
Application   
DSCP   
Binary value   
Call signaling   
CS5   
101000   
Audio   
EF   
101110   
Video   
AF41   
100010   
All other   
EF   
000000   
   
Each of the classes listed above has its own characteristics. Class-Selector 5 (CS5) is used because of   
its backwards compatibility to older network hardware and software. Expedited Forwarding (EF) is   
designed to create a transmission with little or no delay. EF packets are guaranteed a configurable   
rate or amount of bandwidth to ensure that those packets are transmitted in a time-sensitive   
manner. The default DiffServ Per Hop Behavior (PHB) is nothing more than a best-effort type of   
transmission. If there is enough available bandwidth, it does not experience many problems, but if   
the network is loaded down, it may experience a large packet-loss percentage.   
Basically VoIP payload traffic (EF PHB) takes priority over all other traffic, which is then followed by   
the VoIP control traffic (CS5 PHB), and then all other traffic (Default PHB), which is sent on a best-  
effort basis, with no guarantee of service.

CCSP Prescriptive Architecture - 7.3 and 7.4   
113   
Note   
If CCSP IVR/IVVR is transmitting both audio and video, arguably both media streams should have   
equal precedence and, therefore, packets for both streams should be marked with the value   
AF41. However, if audio is more important than video to a particular application, audio should   
still be given higher precedence (EF).   
Switches/routers should use two methods of QoS: Low Latency Queuing (LLQ) and Link   
Fragmentation and Interleaving (LFI). LLQ provides strict priority for Class-based Weighted Fair   
Queuing (CBWFQ), with priority placed on voice traffic. LFI is used to break down or fragment the   
voice streams into smaller delay-sensitive frames, which allow them to take priority over larger   
transmission types such as file transfers (FTP and TFTP). The combination of these two allows for   
less jitter and delay in the voice transmission and produces far superior voice quality for VoIP calls.   
Microsoft QoS Packet Scheduler   
To ensure excellent voice quality, you must use Windows QoS Packet Scheduler in all CCSP   
components: VCS, DPS, and Agent desktops.   
Note   
If the QoS Packet Scheduler is not installed on an agent’s computer, a system warning message   
displays when the Agent application is installed or updated.   
Packet scheduler   
The Windows QoS packet scheduler is a software component that acts as a traffic controller to   
regulate the amount of data packets that an application is allowed within a certain period of time.   
This is a QoS function that provides preferential treatment to higher-priority traffic and that   
ensures that network transmission of packets begins with data that has been deemed most   
important.   
The Packet Scheduler also helps to schedule incoming data packets, which can greatly improve the   
quality of multi-media sessions.   
If any portion of the network path that the data packets traverse is QoS-aware, by using Resource   
Reservation Protocol (RSVP) that data can request a certain amount of reserved bandwidth, thereby   
improving the flow of the media stream.   
Packet shaping   
Part of the packet scheduler's responsibility is shaping the way packets are transmitted from a   
network device, a capability often referred to as traffic shaping. A factor that contributes to network   
congestion is the burst nature of computer data transmissions, a side-effect of the inherent "send   
it all out right now" nature of IP transmission. Traffic shaping can help alleviate at least some of the   
effects of such activity by spacing out QOS-enabled packet transmissions, and by smoothing   
transmission peaks over a given period of time.

CCSP Prescriptive Architecture - 7.3 and 7.4   
114   
MOS   
One of the greatest ways to view the difference between VoIP calls using QoS and those without is   
to compare their Mean Opinion Scores (MOS). The industry standard for an acceptable call is   
somewhere around a MOS score of 3.5.   
The MOS scale is the International Telecommunication Union’s (ITU) recommendation for defining   
voice quality scores. The basis of the scale is the quality of a voice sample by a wide range of   
listeners on a scale of 1-5: 1 being bad and 5 being excellent. This scale monitors live customer   
calls, to produce a quality score based on the ITU-T P.862 standard for objective speech quality   
assessment. MOS scales are available to license for integration into network management   
equipment, VoIP devices, and network infrastructure.   
The following table shows the MOS for various voice quality tests.   
Score   
Opinion   
scale:   
Conversation   
test   
Difficulty   
scale   
Opinion   
scale:   
Listening   
test   
Listening effort scale   
Loudness   
preference scale   
5   
Excellent   
-   
Excellent   
Complete relaxation   
possible, no effort   
required   
Much louder than   
preferred   
4   
Good   
-   
Good   
Attention necessary,   
no appreciable effort   
required   
Louder than   
preferred   
3   
Fair   
-   
Fair   
Moderate effort   
required   
Preferred   
2   
Poor   
-   
Poor   
Considerable effort   
required   
Quieter than   
preferred   
1   
Bad   
Yes   
Bad   
No meaning   
understood with any   
reasonable effort   
Much quieter than   
preferred   
0   
-   
No   
-   
-   
-   
Jitter   
Every voice packet is sent from its origin in strictly equal intervals, usually of tens of milliseconds.   
Each packet travels to the destination in a slightly different amount of time. This is particularly   
significant in multi-route networks, such as the Internet, where each packet can take different   
routes. The packets arrive at the destination at different intervals or even in a different order. The   
destination device, for example, an IP phone, aggregates not-in-order packets in the buffer. Jitter is

CCSP Prescriptive Architecture - 7.3 and 7.4   
115   
statistical variance of inter-arrival delay. High jitter levels cause large numbers of packets to be   
discarded due to overflow of the jitter buffer.   
Note   
During busy hours, we recommend maintaining the network with values lower than 20   
milliseconds.   
Echo   
There are two sources of echo:   
•   
Acoustic echo — arises when sound from a speaker phone or a hands-free phone enters the   
phone’s microphone. The problem exists almost in any communication scenario where there   
is a speaker and a microphone due to physical coupling. The vibrations of the loudspeaker   
transfer to the microphone via the handset casing.   
•   
Line echo — is caused by the PSTN network by the transition of a call from analog telephony   
to digital telephony. This transition usually involves the change from two wires as transport in   
analog telephony to four wires as transport in digital telephony.   
   
Two basic characteristics of echo are as follows: the louder the echo (echo amplitude) or the longer   
the round-trip delay (echo starts later), the more disturbing and apparent it is. To make the echo   
less apparent, decrease the delay on the network.   
To reduce acoustic echo that originates from CCSP UI, use a headset with echo cancelation   
features.   
To reduce line echo that originates from the caller side, or line echo that originates from CCSP UI   
(for an alternative device), use an echo cancelation mechanism on the VoIP gateway. Usually you   
can configure the Echo Canceller coverage or tail. It should be configured to 32-64 milliseconds.   
Setting a higher value may cause fewer available channels on the VoIP gateway. Echo Canceller   
coverage (tail) specifies the length of time that the echo canceller stores its approximation of an   
echo. It is the maximum echo delay that an echo canceller can eliminate.   
Packet loss   
Packet loss in data networks is both common and expected. Many data protocols use packet loss to   
determine the condition of the network and by measuring it, to reduce the number of packets   
being sent.   
Voice codecs used in VoIP compression can handle small random losses, since every codec has a   
packet loss concealment function, which simply detects a missing packet and replays the previous   
packet. The receiving station waits for a period of time, per its jitter buffer, and then runs a   
concealment strategy. This concealment strategy replays the last packet received, so the listener   
does not hear gaps of silence. Because the lost speech is only 20 milliseconds, the listener most   
likely does not hear the difference. You can accomplish this concealment strategy only if one packet   
is lost. If multiple consecutive packets are lost, the concealment strategy is run only once until   
another packet is received.

CCSP Prescriptive Architecture - 7.3 and 7.4   
116   
Loss of less than 1% of packets has no effect on voice quality. 1%-3% are acceptable values with a   
slight effect on the quality, although sufficient for convenient communication. Values higher than   
3% negatively affect the voice quality.   
However, packet loss starts to be a real problem when the percentage of the lost packets exceeds a   
certain threshold, roughly 5% of the packets, or when packet losses are grouped together in large   
packet bursts. In those situations, even the best codecs can’t hide the packet loss from the user,   
resulting in degraded voice quality.   
The following diagram shows a decrease in MOS due to packet loss.   
   
Figure 42: Decrease in MOS due to packet loss   
Note   
•   
When using voice on data networks, it is important to build a network that can use DiffServ to   
successfully transport voice in a reliable and timely manner.   
•   
When troubleshooting voice-quality issues, it is important to know both the percentage of   
lost packets, and whether these losses are grouped into packet bursts.

CCSP Prescriptive Architecture - 7.3 and 7.4   
117   
Packet delay   
The ITU considers network delay for voice applications in Recommendation G.114. This   
recommendation defines three bands of one-way delay.   
The following table shows the ITU packet delay specification.   
Range in   
milliseconds   
Description   
0 – 150   
Acceptable for most user applications.   
150 – 400   
Acceptable provided that administrators are aware of the   
transmission time and the impact it has on the transmission quality of   
user applications.   
> 400   
Unacceptable for general network planning purposes. However, it is   
recognized that in some exceptional cases this limit is exceeded.   
   
These recommendations are for connections with echo adequately controlled. This implies that   
echo cancellers are used. Echo cancellers are required when one-way delay exceeds 25 milliseconds   
(G.711).   
These recommendations are for national telecom administrations and consequently more stringent   
than when normally applied in private voice networks. When the location and business needs of   
end users are well-known to the network designer, more delay can prove acceptable. For private   
networks, 150 milliseconds (one way) of delay is a reasonable goal and 250 milliseconds a limit. All   
networks need to be engineered such that the maximum expected voice connection delay is known   
and minimized.   
When trying to debug QoS issues on a VoIP network, it is important to take into consideration all   
possible sources of delay. There are basically two distinct types:   
•   
Fixed — fixed delay components add directly to the overall delay on the connection. These   
could be transcoders or poor I/O on line cards.   
•   
Variable — variable delays arise from queuing delays in the egress trunk buffers on the serial   
ports connected to the WAN or in switch ports. These buffers create variable delays, called   
jitter, across the network. Variable delays are handled through the de-jitter buffer at the   
receiving router/gateway.   
   
Note   
When using VLAN trunking in the network, make sure the trunks have adequate bandwidth to   
account for all the aggregated links. This is often a major source of packet delays.

CCSP Prescriptive Architecture - 7.3 and 7.4   
118   
Appendix D: Component security settings   
•   
Svccosmocall user   
•   
Dedicated users   
•   
System services security change mapping   
•   
System web services security change mapping   
   
To increase the security of the CCSP system environment in the field, the svccosmocall user’s   
domain-wide capabilities have been reduced and service-specific users and accounts have been   
created or adopted for different service classifications.   
After deploying the CCSP 7.x system, databases, services, and web services, you manually configure   
the service-specific users for each service.   
Svccosmocall user   
CCSP system deployment was previously based on a superuser named svccosmocall.   
This user has strong system-wide rights and admin rights for every task that a service or operator   
of the system requires, including installation, maintenance, and runtime tasks.   
You still use the svccosmocall account for new deployments and installations. However, after   
completing these activities, you should disable this account until further such activities are   
required.   
If you re-enable svccosmocall for additional deployment tasks, it also requires access to the   
database, so you must also enable its database rights.   
Dedicated users   
The following users and accounts where created or adopted for different service classifications:   
•   
cfmUser   
•   
CorderUser   
•   
DTRUser   
•   
ecsUser   
•   
LSUser   
•   
ReducedSVC   
•   
msUser   
•   
LOCAL\_SYSTEM (existing)   
•   
LOCAL\_SERVICE (existing)   
   
Each user type can be shared among different services, that is, the same user for the same class of   
services throughout CCSP.

CCSP Prescriptive Architecture - 7.3 and 7.4   
119   
System services security change mapping   
The following table shows the new security setting for each system service.   
System service   
Current security setting   
New security setting   
AdminIS   
Svccosmocall   
LocalService   
AIS   
Svccosmocall   
LocalService   
APS   
svccosmocall   
LocalSystem   
CFM   
Svccosmocall   
cfmUser   
CIS   
Svccosmocall   
LocalSystem   
CMS   
svccosmocall   
LocalSystem   
DTR   
Svccosmocall   
DTRUser   
DPS   
Svccosmocall   
reducedSVC   
ECS   
Svccosmocall   
ecsUser   
ICS   
svccosmocall   
LocalSystem   
IMS   
Svccosmocall   
LocalService   
IPS   
svccosmocall   
LocalSystem   
LoggingServer   
svccosmocall   
LSUser   
MCS   
Svccosmocall   
reducedSVC   
MS   
Svccosmocall   
msUser   
PDS   
svccosmocall   
reducedSVC   
RtpRelay   
Svccosmocall   
LocalService   
STAT   
svccosmocall   
LocalSystem   
VCS   
Svccosmocall   
reducedSVC   
VMS   
svccosmocall   
reducedSVC

CCSP Prescriptive Architecture - 7.3 and 7.4   
120   
System web services security change mapping   
The following table shows the new security setting for each system web service.   
System service   
Current security setting   
New security setting   
AdminTP   
   
localSystem   
AgentWs   
svccosmocall   
LocalSystem   
CosmoCorder   
svccosmocall   
CorderUser   
CosmoDesigner   
   
localSystem   
Dashboard   
svccosmocall   
LocalService   
DialingPlan   
svccosmocall   
localSystem   
RTRweb   
svccosmocall   
LocalService   
tenantProvisioner   
   
localSystem   
TouchPoint   
svccosmocall   
LocalSystem   
voicemailWS   
   
PrivateVMUser   
WebAdministrator   
Svccosmocall   
IdentityApplicationPool   
   
For detailed information, refer to the CCSP Security Guide.

CCSP Prescriptive Architecture - 7.3 and 7.4   
121   
Appendix E: CCSP-Elsbeth integration   
CCSP 7.3 introduces support for Enghouse’s Elsbeth dialer solution.   
You need the following documents to plan, prepare, install, and configure the Elsbeth dialer   
solution for CCSP:   
•   
ECM Installation Guide for CCSP   
•   
Elsbeth 6.x: EPC Installation Guide for CCSP   
From Elsbeth 9.x: EPC Installation Guide   
   
On CCSP platforms, the Elsbeth dialer solution consist of the following:   
•   
Elsbeth CommunicationManager (ECM) — the dialing engine   
•   
Elsbeth PowerContact (EPC) — the campaign manager   
•   
Elsbeth Configuration Synchronization Server — a CCSP service for Elsbeth   
•   
From Elsbeth 9.x: Elsbeth Agent Gadget (EPC Gadget) — the UI service for agent campaign   
handling   
   
The following sections provide detailed definitions and principles for the above.   
Elsbeth CommunicationManager (ECM)   
•   
ECM is the dialing engine.   
•   
The ECM-CCSP integration supports multi-tenancy.   
   
ECM consist of the following services.   
Service   
Description   
ECM-CS   
Elsbeth Communication Manager Service (ECM-CS)   
•   
Is a mandatory ECM component   
•   
Is the core service for dialing logic   
•   
Communicates with   
o   
ECM-GS for agents and calls   
o   
EPC for campaigns   
ECM-GS   
ECM Gate Service (ECM-GS)   
•   
Is a mandatory ECM component   
•   
Is the communication bridge between ECM and CCSP   
•   
Communicates with   
o   
ECM-CS   
o   
CCSP over CCSP Adapter for Elsbeth

CCSP Prescriptive Architecture - 7.3 and 7.4   
122   
Service   
Description   
CCSP Adapter   
CCSP Adapter for Elsbeth   
•   
Is a mandatory ECM-GS component   
•   
Is a module used by ECM-GS   
•   
Is the campaigns contractor between ECM-GS and CCSP components   
CCSP provides the adapter; it is not part of ECM installation media.   
ECM-EP   
ECM Engine Provider Service (ECM-EP)   
•   
Is a mandatory ECM component   
•   
Hosts call flow engines of the ECM tenants   
ECM-DM   
ECM Data Manager Service (ECM-DM)   
•   
Is an optional ECM component   
•   
Manages historical report data of ECM   
ECM-EM   
ECM Email Manager Service (ECM-EM)   
•   
Is an optional ECM component   
•   
Sends emails ECM call flows generate   
ECM-SP   
ECM SNMP Provider Service (ECM-SP)   
•   
Is an optional ECM component   
•   
Provides KPIs of ECM via SNMP interface   
ECM-SM   
ECM Service Manager Application (ECM-SM)   
•   
Is an optional ECM component   
•   
Is the GUI application for managing ECM services   
Elsbeth PowerContact (EPC)   
•   
EPC is the Elsbeth campaign manager.   
•   
EPC is dedicated per tenant.   
   
EPC consist of the following services.   
Service   
Description   
EPC   
Elsbeth PowerContact   
•   
EPC communicates with   
o   
EPC databases   
o   
ECM-CS   
EPC-LR   
EPC License Reader (EPC-LR)   
•   
Is a mandatory EPC component   
•   
Must be executed one time on the EPC machine

CCSP Prescriptive Architecture - 7.3 and 7.4   
123   
Service   
Description   
EPC-Admin   
EPC Administrator (EPC Admin)   
•   
Is a mandatory EPC component   
•   
Is a GUI application for monitoring and configuring campaigns,   
workflows, roles, rights, and etc.   
EPC-SS   
EPC Scheduling Service (EPC-SS)   
•   
Is an optional EPC component   
•   
Is a Windows service for executing scheduled tasks defined in EPC   
Administrator   
EPC-SNMP   
EPC SNMP Service (EPC-SNMP)   
•   
Is an optional EPC component   
•   
Is a Windows service for handling SNMP requests   
EPC-M   
EPC Monitor (EPC-M)   
•   
Is an optional EPC component   
•   
Is a GUI application for real time monitoring   
Elsbeth Configuration Synchronization Server (ECS)   
•   
Is a CCSP component.   
•   
Is provided by CCSP; it is not part of ECM installation media.   
•   
Provides continuous synchronization for a subset of configuration parameters between CCSP   
configuration database and EPC databases, for mapping tenants in Elsbeth dialer mode on the   
two systems.   
Elsbeth Agent Gadget service (EPC Gadget)   
•   
From Elsbeth 9.x.   
•   
Is the Elsbeth UI service for agent campaign handling.   
•   
Supports multi-tenancy.   
•   
Is installed as a child service of CCSP UI (TouchPoint) service instances.

CCSP Prescriptive Architecture - 7.3 and 7.4   
124   
How to plan an Elsbeth system for CCSP   
Definitions   
An Elsbeth system for CCSP   
Server   
Description   
ECM-CS server   
1 Server with ECM-CS and ECM-EP   
•   
1 ECM-CS server is supported per ECM system   
•   
You can plan ECM-CS server redundancy at virtual machine level   
ECM-GS servers   
2 or more servers with ECM-GS   
•   
Minimum is 2 servers for redundancy   
o   
ECM-CS distributes outgoing calls to ECM-GS in round-robin   
fashion   
o   
CCSP Adapter on ECM-GS distribute call load using the CCSP   
Dialing Plan   
EPC servers   
1 server per tenant   
•   
Each EPC server communicates with one EMC-CS only   
•   
You can plan EPC server redundancy at virtual machine level   
EPC Gadget (from   
Elsbeth 9.x)   
1 IIS service for each TouchPoint IIS service instance   
•   
Each EPC Gadget service communicates with a tenant-specific EPC   
Server instance   
•   
You can plan EPC Gate redundancy in the same way as for its parent   
CCSP UI service   
A CCSP platform working with Elsbeth   
Server   
Description   
Elsbeth system   
1 or more Elsbeth system(s)   
ECS servers   
1 Server with ECS per CCSP platform   
•   
ECS supports multitenancy   
•   
ECS does not support N+1   
•   
You can plan ECS server redundancy at virtual machine level

CCSP Prescriptive Architecture - 7.3 and 7.4   
125   
System requirements   
Server   
Specification   
ECM-Core Services (CS) machine   
Hardware1:   
•   
8 CPU cores   
•   
8 GB RAM   
•   
100+ GB storage2   
OS (Elsbeth 6.x):   
•   
Windows Server 2008 SP2 64-bit   
•   
Microsoft .NET Framework 4.7.2   
OS (Elsbeth 9.x):   
•   
Windows Server 2012 R2 64-bit   
•   
Microsoft .NET 6.0   
Network:   
•   
Attach to IPC network   
ECM Gate service machine   
Hardware1:   
•   
4 CPU cores with 2 GB RAM   
or 2 CPU cores with 4 GB RAM   
•   
100+ GB storage2   
OS (Elsbeth 6.x):   
•   
Windows Server 2008 SP2 64-bit   
•   
Microsoft .NET Framework 4.7.2   
OS (Elsbeth 9.x):   
•   
Windows Server 2012 R2 64-bit   
•   
Microsoft .NET 6.0   
Network:   
•   
Attach to IPC network   
EPC Server and EPC database installed on   
the same machine   
Hardware3:   
•   
8 CPU cores (3.5Ghz)   
•   
16 GB RAM   
•   
100 GB storage2   
OS (Elsbeth 6.9.x):   
•   
Windows Server 2008 SP2 64-bit   
•   
Microsoft .NET Framework 4.7.2   
OS (Elsbeth 9.x):   
•   
Windows Server 2012 R2 64-bit   
•   
Microsoft .NET 6.0   
Networks:   
•   
Attached to IPC and INT networks

CCSP Prescriptive Architecture - 7.3 and 7.4   
126   
Server   
Specification   
ECS machine   
Hardware1:   
•   
2 CPU cores   
•   
2 GB RAM   
•   
40 GB storage2   
OS (Elsbeth 6.x):   
•   
Windows Server 2016 64-bit   
•   
Microsoft .NET Framework 4.6.1   
OS (Elsbeth 9.x):   
•   
Windows Server 2016 64-bit   
•   
Microsoft .NET Framework 4.8.0   
Networks:   
•   
Attached to IPC network   
1. Hardware specification varies depending on the amount of agents, and number of   
call attempts per sec. Specification is for a typical 400 dialer agents multi-tenant   
system.   
2. Storage space requirements depending on the amount of calls per day, and the   
number of days of data to store for log and/or csv files, additional space may be   
required.   
3. Hardware specification varies depending on factors such as:   
o   
Number of agents for the chosen tenant   
o   
Number of campaigns and their sizes   
o   
Number of addresses (contacts in campaign tables) in the system   
o   
The average call duration   
o   
Number of simultaneous reports/exports/imports that are running while there   
are active campaigns   
System sizing   
Scenario 1: Multi-tenancy with 400 dialer agents and typical call attempts   
Requirements   
•   
Multiple CCSP tenants working with Elsbeth dialer service, and   
•   
Total number of dialer agents does not exceed 400, and   
•   
Aggregated dialer call attempts is typical (for example, 15 CAPS)   
Solution   
Plan for one Elsbeth system, refer to How to plan an Elsbeth system for CCSP

CCSP Prescriptive Architecture - 7.3 and 7.4   
127   
Scenario 2: Multi-tenancy with over 400 dialer agents and typical call attempts   
Requirements   
•   
Multiple CCSP tenants working with Elsbeth dialer service, and   
•   
Total number of dialer agents exceeds 400, and   
•   
Aggregated dialer call attempts is typical (for example, 15 CAPS)   
Solutions   
If no tenants exceed 400 dialer agents:   
•   
Put tenants into logical groups such that no group has 400 or more dialer agents, and leave   
some headroom for growth   
•   
Plan multiple ECM systems, refer to How to plan an Elsbeth system for CCSP   
•   
Give each group its own ECM system   
   
If one or more tenants exceed 400 dialer agents:   
•   
For each large tenant that exceeds 400 dialer agents:   
o   
Divide the large tenant into a group of smaller tenants. For example, if BigCo has 1500   
dialer agents, provision 4 separate CCSP tenants: BigCo-A, BigCo-B, BigCo-C, and BigCo-D.   
o   
Put agents into the smaller tenants such that no smaller tenant has 400 or more dialer   
agents, and leave some headroom for growth   
o   
Plan multiple ECM systems, refer to How to plan an Elsbeth system for CCSP   
o   
Give each smaller tenant its own ECM system   
•   
For the other tenants that do not exceed 400 dialer agents, follow the first solution.

# Contact Center: Service Provider CCSP UI Help - Agent Version 7.4

Contents  
1: Intro  
6  
What's new  
6  
Legal disclaimer  
6  
Support  
6  
2: Overview  
7  
Application overview  
7  
Types of connections  
8  
Types of CCSP UI users  
8  
Configure your browser  
9  
Clear your browser's cache  
9  
Log in  
10  
Password expired message  
11  
Additional login messages  
11  
Main page  
12  
Agent view  
12  
Gadgetory  
12  
Calls Menu  
13  
Gadgets and buttons  
14  
Keyboard shortcuts  
16  
Notification area icons and menu  
16  
CCSP UI icon colors  
16  
Screen recording upload service icons  
16  
Menu of available actions  
17  
Use an alternative device  
18  
Log in and be available  
19  
Answer a call on an alternative device  
20  
Hold a call on an alternative device  
20  
Make a call on an alternative device  
21  
Transfer or conference a call on an alternative device  
21  
End a call on an alternative device  
21  
Retrieve voicemail on an alternative device  
21  
Use an alternative device remotely  
22  
View the Mini Wallboard  
23  
Set your preferences  
25  
Change your password  
28  
Expired password  
28  
Report a problem  
29  
Send an outgoing email  
30  
Request supervisor assistance  
31  
Receive broadcast messages  
31  
Log out  
32  
Additional logout messages  
32  
CCSP UI Help - Agent - 7.4  
2

Session expiry  
32  
Remote logout  
33  
3: Calls  
34  
Become available or released  
35  
Become available  
35  
Become released  
36  
Answer a call  
37  
Refuse a call  
38  
Defer an email or a voicemail call  
39  
Process a voice call  
40  
Process a chat call  
41  
Process an email call  
43  
Process a callback call  
45  
Process a voicemail call  
46  
View call information  
47  
Control media recording  
48  
Control screen recording  
49  
Hold a call  
50  
Make an outgoing call  
51  
Transfer or conference a call  
54  
Hang up from a conference call  
56  
Request a chat consultation  
57  
Adjust the volume  
60  
Mute a call  
60  
Schedule a callback  
61  
Use canned phrases  
63  
Co-browse (Web collaboration)  
64  
Co-browse example  
65  
Wrap up a call  
71  
4: Messages  
72  
View messages  
73  
Regular agent view  
74  
View subfolders  
75  
Page navigation  
76  
Filter messages  
77  
Groups filters  
77  
Defer type filter  
77  
Filter by text contained in the items  
78  
Filter by date and time  
78  
Sort messages  
79  
Reset folder to default filter and sort settings  
80  
Pull messages  
81  
Remove messages  
82  
CCSP UI Help - Agent - 7.4  
3

Access archived messages  
83  
5: Personal voicemail  
84  
Retrieve personal voicemail  
84  
Set up your personal mailbox  
85  
Set up or edit your mailbox preferences  
85  
Change the password  
85  
Record a greeting/name  
85  
Change your PIN  
86  
Personal voicemail quick reference  
87  
Log on  
87  
Play Messages menu  
87  
Mailbox preferences  
88  
6: Contact History  
89  
Contact History language  
89  
Open Contact History  
90  
History dashboard  
91  
Search and filter panel  
91  
Contact actions  
92  
Interaction actions  
92  
Search for an interaction  
93  
Search all fields  
93  
Search using filters  
94  
Manage interactions  
97  
Add an unregistered contact  
98  
View interaction details  
100  
Play a recording of an interaction  
101  
Make an outbound voice or email call  
101  
Send an outbound message to a social channel  
102  
Contacts dashboard  
103  
Contacts panel  
103  
Details tab  
103  
History tab  
104  
Dashboard tab  
105  
Direct Message history  
108  
Fields tab  
108  
Related Accounts tab  
109  
Search for a contact  
110  
Manage contacts  
114  
Add a new contact  
115  
Edit a contact  
117  
Delete a contact  
118  
Add a contact identifier  
119  
Edit a contact identifier  
120  
CCSP UI Help - Agent - 7.4  
4

Delete a contact identifier  
121  
Import contacts  
122  
Export contacts  
123  
Send direct message  
124  
Send Direct Message as a standalone gadget  
125  
View errors  
127  
Error notifications  
127  
Errors dashboard  
128  
Common errors  
128  
CCSP UI Help - Agent - 7.4  
5

1: Intro  
The CCSP user interface (UI) is a browser-based application that enables users to receive voice, chat, email,  
callback, and voicemail calls that arrive at a contact center. Calls are automatically directed to particular users  
because they are available and have the expertise to handle the calls.  
CCSP UI was previously called TouchPoint.  
Agent Help is for CCSP agents, ReadyReps, and business users that have CCSP UI licenses. Unless otherwise  
specified in the instructions in this Help, the term agent refers to both agents and ReadyReps.  
What's new  
Version 7.4  
Contact History — added Contact History, a CCSP add-on for managing contacts and contact account details.  
Request a chat consultation — an agent can request a chat consultation during a call (if enabled). Requires  
HF74-66036.  
Set your preferences — if the selected SIP input and/or output device becomes unavailable during your CCSP  
UI login session (for example, it is unplugged from your computer), a warning appears in both CCSP UI and the  
notification area of your task bar. Requires HF74-64991.  
Refuse a call — when you refuse a call, you may automatically become released if this has been configured by  
your system administrator. Requires HF74-64991.  
Log in and Log out — added process for CCSP single sign-on. Requires HF74-61362.  
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.  
CCSP UI Help - Agent - 7.4  
6

2: Overview  
This section contains the following topics and subsection:  
l  
Quick reference cards  
l  
Configure your browser  
l  
Clear your browser's cache  
l  
Log in  
l  
Main page  
l  
Gadgets and buttons  
l  
Keyboard shortcuts  
l  
Notification area icons and menu  
l  
Use an alternative device  
l  
View the Mini Wallboard  
l  
Set your preferences  
l  
Change your password  
l  
Report a problem  
l  
Send an outgoing email  
l  
Request supervisor assistance  
l  
Receive broadcast messages  
l  
Log out  
l  
Remote logout  
Application overview  
Types of connections  
Types of CCSP UI users  
Supervisor tools  
CCSP UI is a browser-based application that enables users to receive calls that arrive at contact center phone  
numbers. Calls are automatically directed to particular users because they are available and have the expertise  
to handle the calls. Users can also receive chat calls.  
If the system administrator provides email service, emails are also directed to the user from email coming into  
the business. Users may also be able to receive voice calls on a direct extension.  
CCSP can also direct outbound calls to users who participate in outbound dialing campaigns, a capability  
provided by the Dialer component of CCSP. As with email, the system administrator must provide this  
capability.  
CCSP UI Help - Agent - 7.4  
7

Types of connections  
CCSP provides Automatic Call Distribution (ACD) and direct inward dialing (DID) voice connections to users, in  
addition to other media such as email and chat.  
ACD automatically queues outside calls and then routes them to those users best able to handle the calls. ACD  
also works with CCSP Dialer to route calls for outbound dialing campaigns to users.  
DID enables outside callers to dial a user’s specific phone number.  
Types of CCSP UI users  
Various types of users of the CCSP system can use CCSP UI. The available features differ for each user type and  
some features may not apply to your environment.  
The types of CCSP UI users are:  
l  
Business users — standard office workers  
l  
ReadyReps — knowledge workers and others who sometimes receive ACD-routed calls  
l  
Agents — contact center customer service representatives who spend the majority of their day handling  
ACD-routed customer contacts  
l  
Supervisors — agents with supervisory duties over other agents. Agents are assigned to groups according  
to their expertise and, if the contact center needs an administrative collection of agents different from  
groups, to teams. Supervisors have supervision responsibilities and CCSP capabilities only for the agents  
that are in their groups or teams. A supervisor not restricted to specific groups or teams has supervisory  
capabilities over all agents in the contact center.  
The Business User software license provides basic unified communication capabilities. A business user can only  
be reached by direct dial or direct transfer. A business user cannot be a destination of CCSP ACD or participate  
in outbound dialing campaigns. A business user without a CCSP UI license is not a CCSP UI user and cannot log  
on to CCSP UI.  
Agents and ReadyReps can be used as ACD destinations. Agent licenses are for traditional, formal contact  
center agents. ReadyRep licenses are designed for use by people who are not solely or even primarily  
dedicated to a formal contact center, but who need to be part of the contact center infrastructure in most other  
ways. They are sometimes known as informal agents or experts, and their ACD use is part-time. Supervisors can  
monitor and remotely manage agents and ReadyReps.  
The instructions in this Help are for all users of CCSP UI. If a specific instruction is for an agent or a supervisor,  
the instruction addresses agents or supervisors.  
CCSP UI Help - Agent - 7.4  
8

Configure your browser  
Before running CCSP UI from your browser, some initial preparation is required:  
l  
For Microsoft Edge,you must enable localhost access on your computer by running the following command  
from an administrator command prompt:  
CheckNetIsolation LoopbackExempt -a -n=Microsoft.MicrosoftEdge\_8wekyb3d8bbwe.  
l  
For Ubuntu client, to play voicemail, you need to install a codec for MP4.  
l  
To start CCSP UI in Toolbar mode, you must allow pop-ups.  
l  
Do not disable Auto Complete in your browser. CCSP UI uses this option to allow redials and to display your  
call history.  
l  
Do not disable cookies in your browser. CCSP UI uses this option to save your audio preferences.  
l  
For Chrome and Firefox, no specific settings are required.  
Note  
CCSP UI supports TLS 1.0.  
Clear your browser's cache  
After you update CCSP UI by running the CCSP Client Services Installer, you must clear your browser's cache to  
ensure that you run the updated version of CCSP UI.  
If you use Edge:  
1.  
On the Edge menu, click Settings.  
2.  
Click Privacy, search, & services.  
3.  
Under Clear browsing data, click Choose what to clear every time you close the browser.  
4.  
Select the Cached images and files option.  
If you use Chrome:  
1.  
On the Chrome menu, click Settings.  
2.  
Click Show advanced settings.  
3.  
Under Privacy, click Clear browsing data.  
4.  
In the Clear browsing data dialog, select the time period, select the Cached images and files check box,  
and then click Clear browsing data.  
If you use Firefox:  
1.  
On the Firefox menu, click Settings and then click History.  
2.  
Click Clear recent history.  
3.  
In the Clear Recent History dialog, select the time range to clear.  
4.  
Expand Details and then select the Cache check box.  
5.  
Click Clear Now.  
CCSP UI Help - Agent - 7.4  
9

Log in  
The log in process is slightly different depending on the identity provider configured by your administrator.  
Note  
l  
If Teams is enabled, it must be running before you log on to CCSP UI.  
l  
To start CCSP UI in Toolbar mode, you must allow pop-ups.  
l  
If you use a voice device with Keep Connection Open (KCO), and you are on a call when your browser  
crashes or your computer loses its network connection, you must complete the active call before you log  
on to CCSP UI again.  
l  
If your permission profile is changed, you must log off and log on again.  
1.  
In your browser, in the Address bar, type the URL supplied by your supervisor or administrator, for  
example, https://<touchpoint host/TouchPoint/  
2.  
Click Go or press Enter. The Welcome to CCSP UI page appears.  
3.  
Click Log in. If you are already logged in to CCSP, the main CCSP UI page appears.  
4.  
If you are not already logged in to CCSP, the Welcome to CCSP page appears.  
5.  
Type your User name in User Principal Name (UPN) format, for example, someone@domainname.com and  
click Continue.  
6.  
The next step depends on the identity provider configured by your administrator:  
o  
For Auth Server, on the Welcome to CCSP page, type your Password and click Log in. Alternatively, to  
change the username, click Back to return to the previous page. If necessary, see Password expired  
message below.  
o  
For ADFS, you are redirected to the ADFS login page to authenticate. Please consult your local  
procedures for how to proceed from this point.  
o  
For Enghouse Identity Server (EIS), you are redirected to the EIS login page. Type your Password and  
click Login.  
7.  
If necessary, see Additional login messages below.  
8.  
If your administrator has configured the Client Services options to be presented, the option to continue  
with or skip Client Services appears:  
o  
To run CCSP UI with Client services, click Continue with Client Services. The Client Services are  
installed and activated, depending on the configuration by your administrator.  
o  
To run CCSP UI without Client services, click Skip Client Services.  
The main CCSP UI page appears.  
CCSP UI Help - Agent - 7.4  
10

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 Auth Server login 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:  
l  
Auth Server is not available.  
l  
Failed to get authorization for TouchPoint use. This can occur because:  
o  
Your user profile is not permitted to use the CCSP UI application. Only agents, supervisors and profiles  
with supervisor permissions are permitted to use the CCSP UI application.  
o  
Your user has not been assigned to a group. Users must be allocated to a group to use CCSP UI.  
o  
Your user has no skills assigned. Users must have at least one skill assigned to use CCSP UI.  
o  
Your user has been assigned to a group with no allocated queue. A user must be belong to a group that  
has at least one queue assigned to it.  
l  
Failed to login to AIS. If one of the following messages appears, contact your administrator:  
o  
License Control Server Warning: The License Control Server is not working. Soon you will not be able to login.  
o  
Login Failed: Unable to login. The maximum number of concurrent agents exceeded the license definition,  
please try again later.  
o  
Login Failed: Unable to login. The maximum number of user logins have been exceeded, please try again later.  
l  
You are not authorized to view this page. This can occur because:  
o  
You are logged in to an external identity provider with a different username than the one you entered  
on the Auth Server Login Selector page to log in to the CCSP client application.  
o  
You entered a username on the Auth Server Login Selector page and were redirected to an external  
identity provider, where you log on with a different username.  
CCSP UI Help - Agent - 7.4  
11

Main page  
The CCSP UI page provides access to all of the capabilities of CCSP UI.  
Agent view  
Gadgetory  
In CCSP UI, the toolbar is called the gadgetory. Notification appear on top of the gadget icons in the gadgetory,  
for example, how many active calls you have.  
Depending on your CCSP UI configuration, you can change the applications state to one of the following:  
l  
Tab — use for applications that need maximum screen space.  
l  
Float — use for applications that you want to view side-by-side.  
l  
Pop out — use for applications that you want to view on a separate screen.  
l  
Minimize  
To change the application state, click the down arrow on the tab and then click the state.  
For information about the individual gadgets, see Gadgets and buttons.  
CCSP UI Help - Agent - 7.4  
12

Calls Menu  
The Calls Menu shows active interactions and opens when you click  
or, if you selected Answer Calls  
Automatically in your preferences, when a new call arrives.  
The Calls Menu contains the following call types:  
l  
Voice  
l  
Chat  
l  
Email  
l  
Callback  
l  
Voicemail  
To close the Calls Menu, click the left arrow at the top of the menu or click  
again.  
CCSP UI Help - Agent - 7.4  
13

Gadgets and buttons  
The specific gadgets and buttons that you see in CCSP UI depend on your permissions.  
Gadget  
Description  
Available — click to become released.  
Released — click to become available.  
Calls Menu - click to view active interactions.  
Outgoing Call — click to make an outgoing call.  
Outgoing Email — click to send an outgoing email.  
Message View — click to handle messages.  
Callback — click to schedule a callback.  
Assistance — click to request supervisor assistance.  
Voicemail — click to retrieve personal voicemail.  
Contact History — click to manage interactions and contacts.  
Mini Wallboard — click to view queue statistics.  
Settings — click to set your preferences, change your password, report a  
problem, or open the Help.  
Screen Recording — click to manually control screen recording.  
Logout — click to log out.  
Button  
Description  
Answer — click to answer a call.  
CCSP UI Help - Agent - 7.4  
14

Button  
Description  
Call — click to call the selected user or endpoint.  
Call information — click to view call details.  
Callback — click to schedule a callback.  
Canned phrases — click to use canned phrases in email and chat  
calls.  
Defer or pull call— click to defer or pull an email or voicemail call.  
Dial pad — click to open the dial pad in voice calls.  
Hang up — click to refuse a call.  
Headset volume — hover over to adjust the volume. Click to mute.  
Hold — click to put a call on hold.  
Microphone volume — hover over to adjust the microphone  
volume. Click to mute.  
More options — click to view more options.  
Record — click to manually control media recording.  
Web collaboration — click to co-browse web pages with the caller in  
chat calls.  
Wrap up — click to wrap up a call.  
Xfer/Conf — click to transfer or conference a call.  
CCSP UI Help - Agent - 7.4  
15

Keyboard shortcuts  
l  
Ctrl+Backspace — to make a new call  
l  
Win+F2 — to hold or release hold  
l  
Win+F3 — to toggle availability  
l  
Win+F6 — to answer  
l  
Win+F7 — to transfer or conference  
l  
Win+F9 — to maximize CCSP UI  
l  
Win+F12 — to hang up  
Notification area icons and menu  
Note  
You can shut down your computer while the upload service is working. The next time you log on to  
Windows, the upload service reprocesses any incomplete screen recording uploading tasks.  
When you run CCSP UI, icons appears in the notification area of your task bar. These icons change, depending  
on whether you are in call, in an unavailable, released, or available state, if you have lost the connection to the  
contact center, and if screen recordings are still uploading in the background.  
CCSP UI icon colors  
Color  
Meaning  
CCSP UI is connected and available for calls.  
CCSP UI is unavailable (released).  
CCSP UI is in call.  
CCSP UI is not connected.  
Screen recording upload service icons  
Note  
l  
After you log on to Windows, the upload icon appears.  
l  
After you log on to CCSP UI, the upload icon gets the status of the upload service, whether it is idle or  
currently uploading screen recordings.  
CCSP UI Help - Agent - 7.4  
16

Icon  
Meaning  
Idle — the upload service is not currently working.  
Uploading — the upload service is working.  
Menu of available actions  
To view a menu of available actions, right-click the icon that appears in the notification area of your task bar.  
CCSP UI Help - Agent - 7.4  
17

Use an alternative device  
This subsection contains the following topics:  
l  
Log in and be available  
l  
Answer a call  
l  
Hold a call  
l  
Make a call  
l  
Transfer or conference a call  
l  
End a call  
l  
Retrieve voicemail  
l  
Use an alternative device remotely  
If you have an alternative device managed directly by the CCSP system, you can use the alternative device  
stand-alone without CCSP UI or alongside CCSP UI. When you use the alternative device with CCSP UI, you can  
place and manage calls from both the phone and CCSP UI, and CCSP synchronizes call status on both the  
phone and CCSP UI.  
For instructions for your IP phone, see the user guide for your specific phone model. For Teams instructions,  
see the Microsoft Teams Help.  
CCSP UI Help - Agent - 7.4  
18

Log in and be available  
If you have Teams enabled, your CCSP UI agent status and Teams presence are synchronized. See Become  
available and released.  
If you have an IP phone, it is set up as one of the following:  
l  
Permanent extension — is always assigned to a single telephone number.  
l  
Session-based extension — can be assigned different telephone numbers and therefore to different users. A  
user begins a session by logging on to the phone.  
The extension type determines how you log on and become available.  
Permanent extension  
If your phone is set up as a permanent extension, the number is automatically assigned when it is first started  
so that you can receive direct calls. Optionally, you can also log on to CCSP UI to control your IP phone.  
You cannot receive Automatic Call Distribution (ACD)-routed calls without using CCSP UI. If your position  
requires you to receive ACD-routed calls, log on to CCSP UI and make yourself available there. You always  
become available or unavailable through CCSP UI. See Become available and released.  
When you exit CCSP UI, the ACD considers you logged on but unavailable for reporting purposes.  
If you are not set up to receive ACD calls, you may still control your availability in CCSP UIfor reporting  
purposes, but the availability setting will not affect your ability to receive direct inward dialed calls to your  
phone number.  
Session-based extension  
If your phone is set up as a session-based extension, the phone can only be used when you are logged on. If  
you try to place a call when the phone is not logged on, you receive an error message, such as Call Failed, and  
an audible fast busy signal.  
You have a numeric user ID, that is usually your phone extension number, as well as a numeric PIN. After you  
log on to the phone, your phone number is assigned to the phone. If your administrator permits it, you can  
receive direct inward dialed calls to your phone number.  
Optionally, you can use the phone with CCSP UI. To do so, log on to CCSP UI with your alphanumeric user name  
and alphanumeric password, that are usually different from the ID and PIN used for the IP phone.   
To make yourself available to receive ACD-routed calls, do one of the following:  
l  
Make yourself available from the IP phone.  
l  
Make yourself available from CCSP UI. See Become available and released.  
When you change the status from one device, the status is automatically updated in the other device.  
The mechanisms for logging on and becoming available vary from phone to phone.  
Aastra phone  
If you have an Aastra phone, it should have an ACD key that you use to manage the phone-based ACD features:  
CCSP UI Help - Agent - 7.4  
19

l  
Logging in  
l  
Logging out  
l  
Becoming available  
l  
Becoming unavailable  
To log on to the phone:  
1.  
Press the ACD key. A request for your user ID appears.  
2.  
Enter your user ID and select Log In. A request for your PIN appears.  
3.  
Enter your PIN and select Log In.  
When you are logged on, you are placed in the unavailable status. To make yourself available from the IP  
phone:  
1.  
Press the ACD key.  
2.  
Select Avail.  
To make yourself unavailable from the IP phone:  
1.  
Press the ACD key.  
2.  
Select Unavail.  
The Aastra ACD key provides a visual indicator of the ACD state, as follows.  
Visual indicator  
Indication  
Unlit  
Logged out  
Blinking  
Logged in and unavailable  
Lit  
Available  
Answer a call on an alternative device  
When you receive a call, the alternative device and CCSP UI both ring. Answer the call by:  
l  
IP phone — picking up the phone’s handset or pressing the IP phone's Speaker button.  
l  
Teams — clicking Answer in Teams.  
If you selected Answer Calls Automatically in the CCSP UI preferences, then the phone also goes immediately  
off-hook on the speakerphone, headset, or in Teams.  
Hold a call on an alternative device  
You can put a call on hold through CCSP UI or an alternative device.  
You must release the call from hold with the same device that you used to put the call on hold.  
For CCSP UI, see Hold a call. For an alternative device, see the instructions for your phone.  
CCSP UI Help - Agent - 7.4  
20

Make a call on an alternative device  
When you make a call from CCSP UI, your alternative device rings and you must answer it before the call can be  
connected at the destination.  
If the alternative device is configured to automatically pick up, the call is connected immediately.  
Transfer or conference a call on an alternative device  
Note  
To transfer a call to the system, perform all the steps from CCSP UI.  
You can transfer or conference a call through CCSP UI or an alternative device.  
You must leave a conference or complete a transfer from the same device. For CCSP UI, see Transfer or  
conference a call. For an alternative device, see the instructions for your phone.  
End a call on an alternative device  
You can end the call from either CCSP UI or an alternative device.  
If using CCSP UI and you can display wrap-up codes, then select the wrap-up code from CCSP UI.  
Retrieve voicemail on an alternative device  
To receive voicemail, your administrator or supervisor must give you a voice mailbox and a PIN. The PIN must  
be a number.  
When you have voicemail, the message waiting indicator appears in CCSP UI.  
1.  
Dial the voice mailbox. An audio message asks for your PIN.  
2.  
Click/press your PIN numbers on the device's dialpad.  
3.  
To send your PIN, click/press #. An audio message states the number of read messages and the number of  
new messages.  
4.  
Do one or more of the following:  
o  
To listen to your messages, click/press 1.  
o  
To delete the message, click/press 7.  
o  
To save the message, click/press 9.  
o  
For more options, click/press 0.  
o  
To replay the current message, click/press 4.  
o  
To hear the message envelope, click/press 5.  
o  
To return to the main menu, click/press \*.  
o  
To repeat the menu, click/press #.  
o  
To replay a message from the start, click/press 1 while the message is still playing.  
CCSP UI Help - Agent - 7.4  
21

Use an alternative device remotely  
You may be able to take your alternative device with you when you travel. Check with your administrator to see  
if this option is available to you. If it is available, then your device should work in any location where you have a  
public Internet connection.  
Alternatively, you may want to travel without your alternative device, and use either the built-in CCSP softphone  
feature or the Use Voice Device option to deliver your phone calls to any telephone number. If you use a  
permanent extension, this option is not available to you. If you use a session-based extension, this option is  
available to you only when you are not logged on to an alternative device.  
If you need to remotely log off from an alternative device:  
1.  
Log on to CCSP UI.  
2.  
Log off from CCSP UI. This message appears: IP phone still logged in. Do you want to log it out?  
3.  
Click Yes.  
4.  
Log on to CCSP UI again. CCSP UI will now deliver voice according to your CCSP UI preference settings.  
CCSP UI Help - Agent - 7.4  
22

View the Mini Wallboard  
If you have the Mini Wallboard enabled, you can view the queue statistics of the queues associated with your  
group.  
Note  
For detailed information about the measures, refer to the CCSP UI Admin Help.  
1.  
On the gadgetory, click  
Mini Wallboard. The Mini Wallboard appears.  
You can view the name of the primary measure in the tooltip.  
2.  
To expand the Mini Wallboard, click  
. The secondary measures appear.  
3.  
To view additional information, in the bottom right corner, click  
.  
CCSP UI Help - Agent - 7.4  
23

4.  
To minimize the Mini Wallboard, click  
.  
5.  
To view the Mini Wallboard on a separate screen, click  
.  
CCSP UI Help - Agent - 7.4  
24

Set your preferences  
Note  
In Remote Desktop Services environments, SIP does not work as expected.  
The first time you start CCSP UI, you may have to set up your preferences to ensure that you can receive and  
make calls.  
1.  
On the gadgetory, click  
Settings and then click Preferences. The Preferences dialog appears.  
If you select SIP, the Preferences dialog appears as follows.  
CCSP UI Help - Agent - 7.4  
25

2.  
In Country Code, type your country code. This code can be a maximum of three numeric digits and can  
have a leading plus sign (+).  
3.  
In Area Code, type your area code. This code can be a maximum of 12 numeric digits.  
Note  
To make calls, you must enter the Country Code and Area Code information.  
4.  
Under Communication Type, do one of the following:  
o  
If you use SIP for call delivery and origination, click SIP. In Input Device and Output Device, select the  
device from the list. Test and adjust the microphone volume and headset volume.  
Note  
If the selected SIP input and/or output device becomes unavailable during your CCSP UI login  
session (for example, it is unplugged from your computer), a warning appears in both CCSP UI and  
the notification area of your task bar. You cannot receive or make voice calls using SIP until either  
the device is available again and you have re-selected it in Preferences, or you have selected  
another input/output device.  
o  
If you use a voice device for call delivery and origination, click Voice Device. In Voice Device Address,  
type your phone IP or extension. If Keep Connection Open (KCO) is enabled for your company, select  
the Keep Voice Device Connection Open check box.  
Note  
CCSP UI Help - Agent - 7.4  
26

If you select Keep Voice Device Connection Open, CCSP automatically logs you on again. If you  
enabled Auto Available, you will automatically become available.  
5.  
Under Answer Options:  
o  
If you want to automatically become available when you log on to CCSP UI, select the Auto Available  
check box.  
o  
If you want CCSP UI to automatically answer calls when they are delivered, select the Answer Calls  
Automatically check box.  
6.  
In External Email Address, type your email address.  
7.  
In Mailbox Pin, type the four digits of your new PIN.  
8.  
If you want to use a chat greeting, select the Use greeting check box and type the greeting in the text box,  
for example, Hello, my name is Harry, how may I assist you?  
9.  
Click Save.  
CCSP UI Help - Agent - 7.4  
27

Change your password  
1.  
On the gadgetory, click  
Settings and then 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.  
Expired password  
If you are already logged on and receive a message that your password has expired and must be changed, log  
off and log on again. CCSP UI prompts you to change your password.  
CCSP UI Help - Agent - 7.4  
28

Report a problem  
1.  
On the gadgetory, click  
Settings and then click Report a Problem. The Report a Problem dialog  
appears.  
2.  
Type a description of the problem.  
3.  
Click Send.  
CCSP UI Help - Agent - 7.4  
29

Send an outgoing email  
Note  
l  
Separate email addresses with a semi-colon (;). CCSP UI does not support email addresses that contain  
double quotation marks (") or are separated by a comma (,).  
l  
CCSP UI auto-saves the draft every 30 seconds.  
l  
Email address autocomplete is only available if it is enabled in CCSP UI Admin.  
l  
If you insert two or more images using drag and drop, CCSP UI only inserts the last image.  
l  
In Firefox and Edge, CCSP UI inserts canned phrases at the bottom of the email.  
l  
Background text color may not work as expected.  
l  
You cannot include a digital signature in an outbound email.  
1.  
On the gadgetory, click  
Outgoing Email. The Outgoing Email tab appears.  
2.  
In To, and optionally in CC and BCC, start typing the email address. If email address autocomplete is  
enabled, an alphabetical list of matching email addresses appears as you type. If autocomplete is not  
enabled, click  
Address lookup to see this list after you have started typing.  
By default, 20 email addresses appear in the list. If you cannot see the complete email address you want to  
use, continue typing until it appears in the list. To use an address from the list, click on it.  
3.  
Write the email. To include canned phrases, click  
. See Use canned phrases.  
4.  
Click Send.  
CCSP UI Help - Agent - 7.4  
30

Request supervisor assistance  
Note  
If you are a business user, supervisor assistance is not available.  
1.  
On the gadgetory, click  
. The Assistance dialog appears.  
2.  
Type the request in the text box.  
3.  
Click Send.  
4.  
Optionally type another questions in the text box and click Send.  
5.  
To end the assistance session, click End session.  
Receive broadcast messages  
Messages broadcast from your supervisor appear in the Assistance dialog.  
CCSP UI Help - Agent - 7.4  
31

Log out  
Note  
If you are using an external identity provider such as ADFS or EIS, when you click Log out of your CCSP  
session, you are also logged out of your external identity provider.  
1.  
On the gadgetory, click  
Logout.  
2.  
If the Logout Confirmation dialog appears, click OK.  
3.  
Do one of the following:  
o  
If you want to return to CCSP UI, click Return to Login page.  
o  
If you are logged on with SSO and have finished working with CCSP, click Log out of your CCSP  
session.  
Additional logout messages  
When you click Log out of your CCSP session, the following message may appear:  
l  
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 CCSP session again.  
The application will return to the login page. Depending on the problem resolution, this may take slightly  
longer than normal.  
Session expiry  
Your CCSP UI session will not become invalid when the login session time exceeds the maximum login session  
lifetime specified by your administrator. If you are using Teams however, your CCSP UI status will no longer be  
synchronized with your Teams client presence and you will no longer be able to search or display Teams  
contacts.  
CCSP UI Help - Agent - 7.4  
32

Remote logout  
If your supervisor logs you out of the contact center, a message similar to the following example appears.  
To close the message, click OK.  
CCSP UI Help - Agent - 7.4  
33

3: Calls  
Note  
l  
For Elsbeth campaign calls, the caller name only appears on the call card. For Sytel campaign calls, the  
caller name and number do not appear on the call card. The campaign server provides this information  
in the call information, which is automatically opened for campaign calls.  
l  
The call duration timer on the call card counts the time from when you pick up the call until you hang up.  
It does not count the time the call spent in the ringing state, pickup-pending state, and in-call  
transferred state.  
l  
If you are on one or more calls when you are unexpectedly disconnected from the network, before trying  
to log on again, complete and hang up the active calls.  
l  
The maximum outgoing call number or SIP URI is 370 characters.  
This section contains the following topics:  
l  
Become available or released  
l  
Answer a call  
l  
Refuse a call  
l  
Defer an email or a voicemail call  
l  
Process a voice call  
l  
Process a chat call  
l  
Process an email call  
l  
Process a callback call  
l  
Process a voicemail call  
l  
View call information  
l  
Control media recording  
l  
Control screen recording  
l  
Hold a call  
l  
Make an outgoing call  
l  
Transfer or conference a call  
l  
Request a chat consultation  
l  
Adjust the volume  
l  
Mute a call  
l  
Schedule a callback  
l  
Use canned phrases  
l  
Co-browse (Web collaboration)  
l  
Co-browse example  
l  
Wrap up a call  
CCSP UI Help - Agent - 7.4  
34

Become available or released  
Note  
If you have Teams enabled:  
l  
Your CCSP UI agent status and Teams presence are synchronized. When you log on to CCSP UI, it  
synchronizes with your Teams presence. If your Teams status is Available, CCSP UI automatically sets  
your status to Available.  
l  
If you have Teams enabled and you select Auto Available, Teams automatically becomes available.  
l  
If you select a CCSP UI release code that does not have a corresponding presence mapping in Teams, or  
the other way around, then CCSP UI shows Released and Teams shows Away.  
l  
Call-initiated status changes only apply to voice calls.  
To see for how long you have been in the current state, point to Available  
or Released  
.  
Become available  
Do one of the following:  
l  
To automatically become available when you log on, select Auto Available in preferences or the status  
menu.  
l  
To manually become available, in the gadgetory, click  
. Your status changes to Available  
.  
l  
Press Win+F3.  
l  
In the notification area, right-click the CCSP UI icon and then click Go Available.  
CCSP UI Help - Agent - 7.4  
35

Become released  
Note  
l  
When you become released, CCSP UI does not deliver queued interactions.  
l  
During an outbound dialing campaign, when you try to become unavailable, your state changes to  
Pending Unavailable  
. You can continue working in your company's Customer Relationship  
Management (CRM) program or other applications connected to CCSP UI.  
If you are not in-call, the system will try to route new calls to other agents who are available. If other  
agents are not available, a call may arrive during this time. If you are in-call or you receive another call  
after trying to become unavailable, you could remain in the Pending Unavailable state for the whole  
duration of the call. Your state could change to Unavailable only after you finish the wrap up.  
Do one of the following:  
l  
In the gadgetory, click  
and then click the release code. Your status changes to Released  
.  
l  
Press Win+F3.  
l  
In the notification area, right-click the CCSP UI icon and then click Go Released.  
CCSP UI Help - Agent - 7.4  
36

Answer a call  
To automatically answer calls, select Answer calls automatically in preferences or the status menu.  
To manually answer a call, click the green button on the Ringing call card or press Win+F6.  
Call type  
Click...  
Next...  
Voice  
See Process a voice call.  
Chat  
See Process a chat call.  
CCSP UI Help - Agent - 7.4  
37

Call type  
Click...  
Next...  
Email  
See Process an email call.  
Callback  
See Process a callback call.  
Voicemail  
See Process a voicemail call.  
Refuse a call  
When you refuse a call, you may automatically become released if this has been configured by your system  
administrator. Depending on this configuration, you may:  
l  
Remain released until you either make yourself available or are automatically made available after a set  
time.  
l  
See a warning that you have been made released, along with the number of seconds before you are  
automatically made available again.  
To refuse a call, click the red button in the Ringing call card or press Win+F12.  
For examples of Ringing call card types, see Answer a call.  
CCSP UI Help - Agent - 7.4  
38

Defer an email or a voicemail call  
Note  
Supervisors can see personal and shared deferred calls under the deferred folder.  
1.  
Answer the call. See Answer a call.  
2.  
On the call card, click  
and then click  
. The Defer Call dialog appears.  
3.  
Do one of the following:  
o  
If you do not want other agents to see the deferred call under the deferred folder, click Personal.  
o  
If you want other agents in the same group to see the deferred call, under the deferred folder, click  
Shared.  
4.  
Click Save.  
For information about handling deferred calls, see Pull messages.  
CCSP UI Help - Agent - 7.4  
39

Process a voice call  
After you answer a voice call, the call card changes from Ringing to Active and the call duration timer starts.  
If CCSP is automatically recording the call, a red dot appears.  
To process a voice call, on the call card in the Calls Menu, do one or more of the following.  
To...  
Click...  
Next...  
Hold the call  
See Hold a call.  
Transfer or conference the call  
See Transfer or  
conference a call.  
Record the call, if CCSP is not  
automatically recording it  
and then  
See Record a call.  
Wrap up the call before hanging  
up  
and then  
See Wrap up a call.  
Schedule a callback  
and then  
See Schedule a callback.  
Open the dial pad  
and then  
See Make an outgoing  
call.  
Adjust the volume, hover over...  
See Adjust the volume.  
Mute the call  
See Mute a call.  
View call information  
See View call information.  
Hang up the call  
See Wrap up a call.  
CCSP UI Help - Agent - 7.4  
40

Process a chat call  
Note  
l  
The chat box turns orange in one corner when there are pending changes or after the maximum  
inactivity timeout.  
l  
When you close the chat box, the call is not hung up. To reopen the chat box, click the chat call in the  
Calls Menu.  
l  
You cannot conference a chat call.  
After you answer a chat call, the call card changes from Ringing to Active, the call duration timer starts, and the  
chat dialog appears. A pencil indicates that the caller is typing a message.  
To process a chat call, on the call card in the Calls Menu or in the chat dialog, do one or more of the following.  
CCSP UI Help - Agent - 7.4  
41

To...  
Click...  
Next...  
Hold the call  
See Hold a call.  
Transfer the call  
See Transfer or conference a call.  
Record the call, if CCSP is not  
automatically recording it  
and then  
See Record a call.  
Wrap up the call before hanging up  
and then  
See Wrap up a call.  
Schedule a callback  
and then  
See Schedule a callback.  
Co-browse web pages with the caller  
and then  
See Co-browse (Web collaboration).  
View call information  
See View call information.  
Resize a floating chat box  
Use canned phrases  
See Use canned phrases.  
Reopen a closed chat box  
The chat call in the Calls Menu.  
Hang up the call  
See Wrap up a call.  
CCSP UI Help - Agent - 7.4  
42

Process an email call  
After you answer an email call, the call card changes from Ringing to Active, the call duration timer starts, and  
the Email tab appears.  
To process an email call, on the call card in the Calls Menu do one of the following.  
To...  
Click...  
Next...  
Transfer the call  
See Transfer or conference  
a call.  
Record the call, if CCSP is not  
automatically recording it  
and then  
See Record a call.  
Wrap up the call before hanging  
up  
and then  
See Wrap up a call.  
Schedule a callback  
and then  
See Schedule a callback.  
Defer the call  
and then  
See Defer an email or a  
voicemail call.  
View call information  
See View call information.  
Hang up the call  
See Wrap up a call.  
CCSP UI Help - Agent - 7.4  
43

Note  
l  
Separate email addresses with a semi-colon (;). CCSP UI does not support email addresses that contain  
double quotation marks (") or are separated by a comma (,).  
l  
Background text color may not work as expected.  
l  
If you insert two or more images using drag and drop, CCSP UI only inserts the last image.  
l  
In Firefox and Edge, CCSP UI inserts canned phrases at the bottom of the email.  
l  
You can receive a digitally signed inbound email. However, you cannot include a digital signature in an  
outbound email (reply, forward, or new).  
To reply to or forward the email:  
1.  
On the Email tab, at the bottom, click Reply, Reply All or Forward.  
2.  
Write the email. CCSP UI auto-saves the draft every 30 seconds.  
3.  
Optionally, to include canned phrases, click  
Canned Phrases. See Use canned phrases.  
4.  
Click Send.  
5.  
On the call card, click  
.  
6.  
Wrap up the email call. See Wrap up a call.  
CCSP UI Help - Agent - 7.4  
44

Process a callback call  
To process a callback call, on the call card in the Calls Menu or in the Outgoing callback call dialog, do one or  
more of the following.  
To...  
Click...  
Next...  
Make the call  
Reschedule an unsuccessful callback  
and then  
See Schedule a callback.  
View call information  
See View call information.  
Hang up the call  
See Wrap up a call.  
After you answer a callback call, another instance of the Outgoing call dialog appears, the call card closes  
automatically, and your CCSP UI agent status changes to Released.  
1.  
On the Outgoing call dialog, click  
.  
2.  
When you hang up, the Wrap Up dialog appears. See Wrap up a call.  
CCSP UI Help - Agent - 7.4  
45

Process a voicemail call  
Note  
For Ubuntu client, to play voicemail, you need to install a codec for MP4. Ask your administrator for the  
installation instructions.  
After you answer a voicemail call, the call card changes from Ringing to Active, the call duration timer starts,  
and the message plays.  
To process a voicemail call, on the call card in the Calls Menu do one or more of the following.  
To...  
Click...  
Next...  
Hold the call  
See Hold a call.  
Stop playing the message  
Transfer or conference the call  
See Transfer or conference a  
call.  
Wrap up the call before hanging  
up  
and then  
See Wrap up a call.  
Schedule a callback  
and then  
See Schedule a callback.  
Defer the call  
and then  
See Defer an email or a  
voicemail call.  
Adjust the volume, hover over  
See Adjust the volume.  
View call information  
See View call information.  
Hang up the call  
See Wrap up a call.  
CCSP UI Help - Agent - 7.4  
46

View call information  
Depending on your CCSP UI configuration, call information may always open when a new call arrives.  
To view call information, on the call card in the Calls Menu, click  
. The information appears underneath the  
call card.  
CCSP UI Help - Agent - 7.4  
47

Control media recording  
Note  
l  
You cannot start media recording until the call is connected.  
l  
If you click record in the middle of a voice call, CCSP records only the part of the voice call after the click.  
l  
If you click hold in the middle of a voice call and only you and the caller are present on the call, when you  
take the call off hold, a separate media recording starts and a new media file is created in the database.  
l  
For email recordings, CCSP captures the recording at the call termination point, so you cannot partially  
record email.  
l  
For chat calls, you can simultaneously record the text and screens of each chat.  
CCSP may automatically record all your CCSP UI calls, depending on how your system administrator configured  
recording.  
When CCSP is recording a call's media, a red dot appears on the call card.  
To manually record a call's media and its screens, after answering the call, on the call card, click  
and then  
click  
.  
To manually record only a call's screens, see Control screen recording.  
CCSP UI Help - Agent - 7.4  
48

Control screen recording  
Note  
l  
You can start screen recording by clicking the Screen Recording gadget while the call is ringing.  
However, you cannot start screen recording when there are no calls.  
l  
Clicking the Screen Recording gadget starts screen recording but not media recording.  
l  
The maximum size for screen recording is 2048 x 1536. If you use multiple screens with the Extend  
option, the screen recording includes all the screens.  
l  
In Remote Desktop Services environments, screen recording does not work as expected.  
l  
If there is a problem with the screen recording service, the following warning appears: Screen Recording  
has been selected, but is not available. Ensure that CCSP Screen Recording service is properly installed.  
This warning indicates that the background client services required for screen recording were  
unexpectedly terminated on your computer and the screen recording process is no longer active for you.  
To restart the screen recording process, log out of CCSP UI and then log back in to CCSP UI.  
When there are multiple calls, CCSP automatically records screen only if all the calls should be recorded. Screen  
recording stops if a call starts that should not be recorded or if you stop screen recording to avoid recording  
sensitive data such as credit cards.  
Screen recording also stops or starts if another agent participating in a call with you stops or starts the  
recording.  
When CCSP is recording a call's screens, the Screen Recording gadget is red.  
To manually stop recording screens, on the gadgetory, click  
Screen Recording.  
To manually start recording screens, on the gadgetory, click  
Screen Recording.  
CCSP UI Help - Agent - 7.4  
49

Hold a call  
To hold a call, click  
on the call card or press Win+F2.  
To take a call off hold, click  
 again on the call card or press Win+F2 again.  
CCSP UI Help - Agent - 7.4  
50

Make an outgoing call  
Note  
l  
When you make a call to a system endpoint, Call to the System appears on the call card.  
l  
Received campaign calls appear in the Incoming calls list of the  
tab as either Predictive Call or  
Preview Call. You cannot dial a campaign call from the incoming calls list.  
1.  
On the gadgetory, click  
or press Ctrl+Backspace. The Outgoing Call dialog appears.  
If you have Teams enabled, the Outgoing call dialog has a  
tab.  
CCSP UI Help - Agent - 7.4  
51

2.  
If enabled, on the  
tab, you can click Caller Number, and then click the number in the list.  
3.  
Alternatively, do one of the following.  
To select  
Click  
Then  
The search box  
Start typing the number or name.  
A recent outgoing  
or incoming caller  
To filter the list by Outgoing or Incoming callers, click the corresponding  
word. Click the user.  
A logged in user  
To filter the list by state or group, click the corresponding down arrow  
and select the required state or group from the list. Click the user.  
CCSP UI Help - Agent - 7.4  
52

To select  
Click  
Then  
A system endpoint  
Click the endpoint.  
A speed dial  
Click the speed dial.  
The dialpad  
Click the numbers.  
A Teams contact, if  
enabled  
To filter the list by status or group, click the corresponding down arrow  
and select the required status or group from the list. Click the user.  
4.  
Click  
.  
CCSP UI Help - Agent - 7.4  
53

Transfer or conference a call  
Note  
l  
If cold (blind) transfer is enabled, to transfer a call to an external number via speed dial, the dialpad, or  
the recent caller tab without waiting for the line to connect, in the Xfer/Conf dialog, click  
.  
l  
Only use cold transfer if you are certain the transferee will answer the call. If the transferee rejects or  
does not answer the call, the call is cut and the original caller needs to call the contact center again.  
l  
You cannot use cold transfer to another agent/user, system endpoint, or for email.  
1.  
Click  
on the call card or press Win+F7. The Xfer/Conf dialog appears.  
CCSP UI Help - Agent - 7.4  
54

2.  
Do one of the following.  
To select...  
Click...  
Next...  
The search box  
Start typing the number or name.  
A recent  
outgoing or  
incoming caller  
To filter the list by Outgoing or Incoming callers, click the corresponding  
word. Click the user.  
A logged in  
user  
To filter the list by state or group, click the corresponding down arrow and  
select the required state or group from the list. Request a chat consultation  
(if enabled) or click the user.  
A system  
endpoint  
To transfer the call back to script control at a specific named place in the  
system, click the endpoint.  
A speed dial  
Click the speed dial.  
The dialpad  
Click the numbers.  
A Teams  
contact, if  
enabled  
To filter the list by status or group, click the corresponding down arrow and  
select the required status or group from the list. Click the user.  
3.  
Click  
.  
4.  
When the call is connected, do one of the following:  
o  
To transfer the call, click Transfer. Your end of the call will be hung up.  
o  
To conference the call, click Conference. See Hang up from a conference call below.  
o  
For voice calls, to switch between the other user and the caller, click Toggle. When you are connected to  
the other user, the caller is on hold and Transfer, Conference, Toggle, and Cancel are available. When  
you are connected to the caller, only Toggle and Cancel are available.  
CCSP UI Help - Agent - 7.4  
55

Hang up from a conference call  
Do one of the following:  
l  
To hang up the last participant that joined the conference, in the call card, click the hang up conference  
button next to the participant's name. This icon is only visible if you initiated the conference.  
l  
To leave the conference, click  
or press Win+F12.  
CCSP UI Help - Agent - 7.4  
56

Request a chat consultation  
If enabled, during an incoming call or an outbound voice call, you can request a chat consultation using a  
button next to an available agent via the Xfer/Conf dialog.  
Note  
l  
The incoming call must be a voice, chat, or email call. You cannot request a chat consultation during a  
conference call, callback, or voicemail call.  
l  
The other agent must be Available or Available in call.  
l  
You and the other agent can only have one chat consultation at a time. If the other agent is already in a  
chat consultation or still has the Consultation dialog open, the request is automatically rejected.  
l  
An unanswered request times out after 2 minutes.  
1.  
In the Xfer/Conf dialog, on the  
Users tab, point to an Available or Available in Call agent and click  
in the same row.  
2.  
The Consultation dialog appears:  
o  
If the other agent is already in a chat consultation or still has the Consultation dialog open, the request  
is automatically rejected and you receive the User cannot answer right now notification.  
o  
Otherwise, the other agent can accept or decline the request.  
CCSP UI Help - Agent - 7.4  
57

Your view  
Other agent's view  
3.  
If the other agent clicks Accept, you can exchange messages until either of you clicks End session or  
closes the Consultation dialog, or the initial call is hung up.  
CCSP UI Help - Agent - 7.4  
58

4.  
If the other agent clicks Decline, you receive the User cannot answer right now notification and can only  
close the dialog.  
5.  
After the session ends, you receive a termination notification. The Consultation dialog remains open until  
you close it.  
If you missed any consultation requests while you were in the chat consultation, you receive the You missed  
one or more requests for consultation. Click here to view notification at the bottom of the dialog. Click the  
notification to view the list of attempts.  
CCSP UI Help - Agent - 7.4  
59

Adjust the volume  
Note  
For SIP, you can also test and adjust the volume in Preferences. See Set your preferences.  
1.  
On the call card, hover over  
. The headset volume control appears. Adjust the volume.  
2.  
On the call card, hover over  
. The microphone volume control appears. Adjust the volume.  
Mute a call  
To mute a call, on the Active call card, click  
.  
To unmute the call, on the Active call card, click  
.  
CCSP UI Help - Agent - 7.4  
60

Schedule a callback  
1.  
On the gadgetory, on a call card in the Call Menu, or in the Wrap Up dialog, click  
. The Callback dialog  
appears.  
2.  
In Name, type the name of the calling party.  
3.  
If the number does not already appear in Number, type the telephone number.  
4.  
Under Date, click the appropriate button or in Specific, select the date from the calendar.  
5.  
Under Time, click the appropriate button or in Specific, select the hour and minute.  
6.  
To personally receive the callback, select the Call Me check box.  
CCSP UI Help - Agent - 7.4  
61

7.  
If immediate dialing is not enabled by default, to automatically initiate dialing of the callback number,  
select the Immediate Dialing check box.  
8.  
If necessary, scroll down to view the Message and URL text boxes.  
9.  
In Message, type the message to display when the callback arrives.  
10.  
In URL, type the caller's URL.  
11.  
Click Save.  
CCSP UI Help - Agent - 7.4  
62

Use canned phrases  
Note  
In Firefox and Edge, CCSP UI inserts canned phrases at the bottom of the email.  
1.  
On the right of the CCSP UI page, click  
Canned Phrases. The Canned Phrases dialog appears.  
2.  
In the Search box, start typing the word to search for. CCSP UI looks for this word in the phrase titles and  
text.  
3.  
Click the phrase or expand a category and click the required phrase.  
4.  
Click Use.  
CCSP UI Help - Agent - 7.4  
63

Co-browse (Web collaboration)  
If enabled, you can start a co-browsing session with a caller when you are in a chat call. The same web pages  
then automatically appear on both the caller's and the agent's screens. Both people in the call can navigate  
web pages, complete online forms, and click links to other pages. See Co-browse example.  
Note  
l  
The features available to you in the co-browsing session depend on the co-browsing product configured  
by the system administrator.  
l  
Whether you can open a co-browsing session for each chat call or only a specific number of co-browsing  
instances, is configured by the system administrator.  
l  
For Edge, you cannot pop out the Co-Browsing tab.  
1.  
In the chat dialog, click  
and then click  
. The Co-browsing tab appears. If configured to do so, the  
current chat automatically docks in the tab.  
2.  
To manually dock and undock the current chat, click  
.  
3.  
Navigate the web pages and change presenter, as required.  
4.  
To end the co-browsing session, click Exit session.  
CCSP UI Help - Agent - 7.4  
64

Co-browse example  
The co-browsing product that is provided with CCSP UI is togetherJS. This tool enables you and the caller to  
browse the same website and see the same pages and content. You are also both able to edit the same forms  
on a page.  
Note  
TogetherJS co-browsing can only occur when both parties are on a togetherJS-enabled website. If either  
you or the caller navigates to a page where this tool is not enabled, the other party can’t follow.  
1.  
Depending on the configuration, you are asked by the caller to either start a new session or to join an  
existing session that the co-browser started.  
2.  
In the chat dialog, click  
and then click  
. The Co-browsing tab appears. If configured to do so, the  
current chat automatically docks in the tab.  
3.  
To start the togetherJS session, click Start session.  
4.  
When the website appears in the Co-browsing tab, the togetherJS toolbar appears. A togetherJS welcome  
dialog also appears, which you can close.  
CCSP UI Help - Agent - 7.4  
65

5.  
If you started a new togetherJS session, wait for the caller to join the session. The togetherJS UI displays a  
message when the caller has joined.  
CCSP UI Help - Agent - 7.4  
66

6.  
A pointer appears in your window, indicating the position of the caller’s mouse pointer on the page. The  
caller sees a corresponding pointer for your cursor in their window.  
7.  
When either of you click on the window, an expanding circle is drawn around the position clicked.  
CCSP UI Help - Agent - 7.4  
67

8.  
When the caller navigates to another page on the website, a message appears stating the URL of the page  
that the caller has gone to and asking if you want to join them. To go to the same page, click Join them.  
CCSP UI Help - Agent - 7.4  
68

Note  
This message is also displayed when you and the caller are both on the same page and you then  
navigate to a different page. In this case, ignore the message and wait for the caller to follow you.  
9.  
When the caller is typing, an ellipsis (…) appears next to the caller’s name to indicate this.  
10.  
To automatically follow the caller, click the caller’s icon in the togetherJS toolbar and select the Follow this  
participant check box in the popup window. Close the window. The session is now controlled by the caller  
and your window follows the caller across any page changes.  
CCSP UI Help - Agent - 7.4  
69

11.  
To stop following the caller, click the caller’s icon in the togetherJS toolbar and clear the Follow this  
participant check box.  
12.  
To end the co-browsing, do one of the following:  
o  
Click Exit session.  
o  
Click your icon in the togetherJS toolbar and then click End Co-Browsing.  
Note  
You and the caller must both actively end co-browsing.  
CCSP UI Help - Agent - 7.4  
70

Wrap up a call  
You can wrap up a call after hanging up or while in conversation.  
When you hang up a call, the Wrap Up dialog automatically appears. Continue at step 2.  
To wrap up the call while in conversation:  
1.  
On the call card in the Calls Menu or in the chat dialog, click  
and then click  
. The Wrap Up dialog  
appears.  
2.  
Click the wrap up code.  
3.  
Optionally, click Add Note and type a comment.  
4.  
Optionally, click Call Back and schedule a callback. See Schedule a callback.  
5.  
Click Submit or Save.  
CCSP UI Help - Agent - 7.4  
71

4: Messages  
Note  
Only supervisors can restore deleted messages.  
This section contains the following topics:  
l  
View messages  
l  
Filter messages  
l  
Pull messages  
l  
Remove messages  
l  
Access archived messages  
CCSP UI Help - Agent - 7.4  
72

View messages  
Regular agent view  
View subfolders  
Page navigation  
Note  
You can access Deferred Emails and Deferred Voicemails from the Folders page and from inside the  
Folders > Email and Folders > Voicemail pages respectively.  
1.  
On the gadgetory, click  
Message View. The Message View tab appears.  
2.  
Click the folder to view, for example, Email. The Folders > Email view appears.  
CCSP UI Help - Agent - 7.4  
73

Regular agent view  
Generally, regular agents can view only the subfolders containing their items.  
The subfolders are:  
Email:  
l  
<Public folders> — contain email items awaiting handling. Regular agents can view only items that were  
sent to their group by themselves or by other group members.  
l  
Deferred Emails — contains email items deferred by this agent or by a group member. Deferred items can  
be resumed either by agents pulling them or by the system using the ageing policy.  
l  
Replied — contains incoming ACD email items that were handled and replied to by this agent or by a group  
member. The reply message must be sent for the original item to be considered replied to.  
l  
Not Replied — contains incoming ACD email items that were handled but not replied to by this agent or by  
a group member. This folder contains items that were archived after the call-cycle ended or the email call  
ended without an agent replying to it.  
l  
Sent Items — contains sent email items that were sent by this agent: reply, forward, and new.  
l  
Outbox — contains outbound email items that were created by this agent but have not yet been sent.  
l  
Rejected Emails — contains email items that were sent to this agent's group and rejected by the system  
due to invalid item or call characteristics.  
l  
Drafts — contains email items that were composed by this agent and disconnected, accidentally or  
deliberately, by clicking Discard & Keep draft in the Email Pending Hangup dialog.  
CCSP UI Help - Agent - 7.4  
74

The draft of the discarded item in the Drafts folder retains all the relevant fields (From, To, Body) and can  
be pulled, edited, and sent at a later time. See Pulling messages.  
Callback:  
l  
Callbacks — contains callback items waiting for their scheduling time marked with 'call-to-me' by this  
agent, or created by other group members (and not marked with 'call-to-me').  
l  
Rejected Callbacks — contains callback items that were sent to this agent's group and rejected by the  
system due to invalid item or call characteristics.  
Voicemail:  
l  
Voicemail — contains voicemail items waiting for their scheduling time, that were sent to this agent's  
group.  
l  
Rejected Voicemails — contains voicemail items that were sent to this agent's group and rejected by the  
system due to invalid item or call characteristics.  
l  
Archived Voicemails — contains handled voicemail items that were sent to this agent's group.  
l  
Deferred Voicemails — contains voicemail items deferred by this agent or by a group member.  
View subfolders  
1.  
Click the subfolder to view, for example, Deferred Emails. The Folders > Email > Deferred Emails view  
appears. See Filter messages and Page navigation.  
CCSP UI Help - Agent - 7.4  
75

2.  
To read the complete email, click the email's row. To hear a voicemail or callback, you need to pull it. See  
Pull messages.  
3.  
To read the previous or next email in the list, click the up or down arrow at the top of the email.  
Page navigation  
The total number of items and the current set of items within that number are given in the address bar.  
If filtering criteria are currently being applied to the folder, the total number of items is the number of items  
that meet that criteria.  
To go to the:  
l  
Next page in the list, click  
.  
l  
Previous page in the list, click  
.  
l  
Last page in the list, click  
.  
l  
First page in the list, click  
.  
CCSP UI Help - Agent - 7.4  
76

Filter messages  
Note  
l  
The view is refreshed only when you navigate to another page or click  
Refresh.  
l  
For most items, the time is the arrival time. For deferred items, the time is the arrival and the deferred  
time.  
Groups filters  
Defer type filter  
Filter by text contained in the items  
Filter by date and time  
Sort messages  
Reset folder to default filter and sort settings  
Groups filters  
Emails in the public folders are already filtered by group.  
Defer type filter  
Items in the Deferred Emails and Deferred Voicemails folders can be filtered by the defer type.  
The filter bar below the address bar presents the filtering and sorting options available for the selected folder.  
The options presented depend on the user, media, and the folder itself.  
1.  
Click the folder and subfolders until you reach the list page.  
2.  
Select either Personal or Shared from the drop down list. To see all items, select All Deferred.  
CCSP UI Help - Agent - 7.4  
77

Filter by text contained in the items  
Note  
On the Message View tab, searching for punctuation marks may not return the expected result.  
In the search box, start typing the word to search for. You can also enter a list of words, and a whole phrase,  
using quotation marks (""). CCSP UI searches different fields for each media as follows:  
l  
Email — From, To, Subject, and the body of the message.  
l  
Callbacks — Callback Number and Message  
l  
Voicemail — Call To (DNIS) and Call From (ANI)  
Filter by date and time  
Note  
For most items, the time used for the filter is the arrival time of the item. For deferred items, the time is the  
arrival and the deferred time. For deleted items, the time is the arrival/sent time depending on the item  
and the deleted time.  
1.  
Click  
and select the Start Date and End Date.  
2.  
After selecting the date, you can change the specific time range to search in.  
CCSP UI Help - Agent - 7.4  
78

Sort messages  
Click the Sort By down arrow and then click the field to sort by. The available fields to sort on are dependent on  
the media and the folder. In addition, the folders containing deleted items also offer the option to sort by  
Deleted time.  
Incoming Email items  
Deferred Emails  
Sent Emails  
Callbacks  
Voicemail  
Deferred Voicemail  
To change the sorting direction, click again on the selected field in the Sort By menu.  
CCSP UI Help - Agent - 7.4  
79

Reset folder to default filter and sort settings  
Note  
Any filters and sort options that you select within a folder are remembered by that folder for the next time  
you visit it, until you log out of your CCSP UI session.  
Click  
.  
This clears the date filter and the search box, and resets the:  
l  
Displayed page to the first item  
l  
Selected Defer Type to All Deferred (Deferred folders only)  
l  
Sort option to the default for the folder  
CCSP UI Help - Agent - 7.4  
80

Pull messages  
Note  
When pulling a partial call-part item from the Drafts folder:  
l  
The call-parts fields contain the original data saved at the point the call part item moved to the Drafts  
folder.  
l  
If the parent item of the call-part can be located, it is opened as part of the new incoming ACD email call,  
together with all the call-parts belonging to the parent item.  
l  
If the parent item of the call-part cannot be located, only the pulled call-part item is opened as a new  
outgoing email call.  
l  
Bulk pulling is not currently supported.  
Messages can be pulled from incoming folders (Public, Deferred) and outgoing folders (Sent Items, Outbox and  
Drafts). When pulling from outgoing folders, you can edit and (re)send the original item.  
You can only pull messages to yourself.  
1.  
Click the folder and subfolders until you reach the list page.  
2.  
If necessary, filter the items in the list. See Filter messages.  
3.  
Do one of the following:  
o  
Point to the item and click  
.  
o  
Click the email's title to view the complete email and then click  
Pull.  
4.  
Answer the call. See Answer a call.  
CCSP UI Help - Agent - 7.4  
81

Remove messages  
Note  
l  
Agents can remove only scheduled callbacks that are visible to them.  
l  
Supervisors can remove emails, callbacks, and voicemail.  
l  
The removed item moves to the Deleted Emails, Deleted Callbacks, or Deleted Voicemail folder, as  
appropriate. Removing the item from these folders permanently removes the item.  
1.  
Click the folder and subfolders until you reach the list page.  
2.  
If necessary, filter the items in the list. See Filter messages.  
3.  
Do one of the following:  
o  
Point to the item and click  
.  
o  
Click the email's title to view the complete email and then click  
Remove.  
CCSP UI Help - Agent - 7.4  
82

Access archived messages  
Note  
l  
The online archive is only available for non-Exchange data stores and must be set up by your system  
administrator.  
l  
Messages in the online archive cannot be deleted by any user.  
View messages from a folder in the online archive  
When configured by your system administrator, the following folders may be available in the online archive:  
l  
Not Replied  
l  
Rejected Emails  
l  
Replied  
l  
Sent Items  
l  
Rejected Callbacks  
l  
Archived Voicemails  
l  
Rejected Voicemails  
1.  
Click on the folder from the relevant media parent folder.  
2.  
If the folder is available in the online archive, the  
button appears next to the search box. Click this  
button to see the contents of the folder in the online archive. You can return to the active folder contents  
by clicking  
again.  
3.  
To view an email in any of the email folders in the online archive, click the email preview in the folder.  
Filter messages from a folder in the online archive  
See the filtering and sorting options described in Filter messages. Any selected filters and sort options are  
applied to both the active and online archive content of the folder.  
Pull messages from a folder in the online archive  
See the steps in Pull messages.  
Note  
Pulling a message from the online archive creates a copy of the message in the destination folder in the  
Active content data store. The original message remains in the online archive.  
CCSP UI Help - Agent - 7.4  
83

5: Personal voicemail  
If you do not have a direct extension, you should skip this section.  
This section contains the following topics:  
l  
Retrieve voicemail  
l  
Set up your mailbox  
l  
Change your PIN  
l  
Personal voicemail quick reference  
Some CCSP UI users are granted a direct extension that enables them to receive voice calls directly and not  
through an agent group. Generally, a direct extension is provided with a personal voice mailbox and a PIN to  
control access to the mailbox.  
Your administrator or supervisor should provide your PIN. The PIN must be a number.  
Retrieve personal voicemail  
When you have personal voicemail, a notification appears in the gadgetory  
1.  
Click  
. An audio message asks for your ID.  
2.  
Click/press your ID numbers on the CCSP UI dialpad. Your ID is typically your direct extension telephone  
number.  
3.  
To send your ID, click/press #. An audio message asks for your PIN.  
4.  
Click/press your PIN numbers on the CCSP UI dialpad.  
5.  
To send your PIN, click/press #. An audio message states the number of read messages and the number of  
new messages.  
6.  
Do one or more of the following:  
o  
To listen to your messages, click/press 1.  
o  
To delete the message, click/press 7.  
o  
To save the message, click/press 9.  
o  
For more options, click/press 0.  
o  
To replay the current message, click/press 4.  
o  
To hear the message envelope, click/press 5.  
o  
To return to the main menu, click/press \*.  
o  
To repeat the menu, click/press #.  
o  
To replay a message from the start, click/press 1 while the message is still playing.  
CCSP UI Help - Agent - 7.4  
84

Set up your personal mailbox  
After dialing into your mailbox and entering your ID and PIN, set up your mailbox as follows.  
Set up or edit your mailbox preferences  
1.  
Click/press 8.  
2.  
To change your password, click/press 1. See Change the password.  
3.  
To change your personal greeting, click/press 2. See Record a greeting/name.  
4.  
To change the recording of your name, click/press 3. See Record a greeting/name.  
5.  
To return to the main menu, click/press \*.  
6.  
To hear the menu again, click/press #.  
Change the password  
An audio message asks you to enter a new password, followed by the pound sign.  
1.  
Enter the password and click/press #. The password must be numeric.  
2.  
Do one of the following:  
o  
To keep the password, click/press 1.  
o  
To enter a different password click/press 2.  
Record a greeting/name  
An audio message plays your current greeting/name.  
1.  
To record a new greeting/name, click/press 1.  
2.  
To finish recording, click/press #.  
3.  
Do one of the following:  
o  
To save the new greeting/name, click/press 1.  
o  
To record a different greeting/name, click/press 2.  
o  
To delete the greeting and use the standard greeting, click/press 7.  
CCSP UI Help - Agent - 7.4  
85

Change your PIN  
Your Personal Identification Number (PIN) can be changed through CCSP UI and through your voice device.  
1.  
On the gadgetory, click  
Settings and then click Preferences. The Preferences dialog appears.  
2.  
In Mailbox Pin, type the four digits of your new PIN.  
3.  
Click Save.  
CCSP UI Help - Agent - 7.4  
86

Personal voicemail quick reference  
Log on  
1.  
Through your own phone or another phone, dial your voicemail telephone number.  
2.  
Dial your PIN.  
3.  
Press  
.  
Play Messages menu  
Press the appropriate key.  
Key  
Option  
Play your messages.  
Replay this message.  
Play the envelope of this message.  
Delete this message.  
Change mailbox preferences. Continue at Mailbox preferences below.  
Save this message.  
To hear more options.  
Repeat this menu.  
Return to the main menu.  
Note  
If you press  
 in the main menu, you hang up the  
call.  
CCSP UI Help - Agent - 7.4  
87

Mailbox preferences  
Press the appropriate key.  
Key  
Option  
Change password. Continue at Password menu below.  
Change greeting. Continue at Greeting menu below.  
Change the recording of your name.  
Return to the main menu.  
Repeat this menu.  
Password menu  
Press the appropriate key.  
Key  
Option  
Keep new password.  
Select a different password.  
Greeting menu  
Press the appropriate key.  
Key  
Option  
Record new greeting. Continue at Greeting Confirmation menu below.  
Erase greeting and use standard greeting.  
Greeting Confirmation menu  
Press the appropriate key.  
Key  
Option  
Save new greeting.  
Replace greeting.  
CCSP UI Help - Agent - 7.4  
88

6: Contact History  
This section contains the following topics:  
l  
Open Contact History  
l  
History dashboard  
l  
Search for an interaction  
l  
Manage interactions  
l  
Contacts dashboard  
l  
Search for a contact  
l  
Manage contacts  
l  
Send direct message  
l  
View errors  
Contact History is a CCSP add-on for managing contacts and contact account details.  
Contact History consists of the following parts that are deployed together:  
l  
Social Connector Helper API — API to link between database and client and provide data for drop down lists.  
l  
Contact History Application — user interface to display and manage account data.  
l  
Functional gadget — customizes the CCSP UI client side and allows interaction with Contact History.  
l  
Social Connector — optional component used to interact with social media channels, such as WhatsApp and  
Facebook.  
Contact History language  
By default, Contact History uses the language you select when you log in to CCSP UI.  
If you want to use a different language for Contact History, your Administrator can set that language in the  
(Contact History Application—Language Configuration) section of the Contact History configuration file.  
CCSP UI Help - Agent - 7.4  
89

Open Contact History  
Note  
If you cannot see the Contact History gadget in the gadgetory, ask your Administrator to check that in CCSP  
UI Admin, the gadget is enabled, the Contact History URL is set, and the gadget is assigned to your group.  
1.  
On the gadgetory, click  
Contact History. The History dashboard appears.  
2.  
To view and manage contact information, click  
Contacts to open the Contacts dashboard.  
CCSP UI Help - Agent - 7.4  
90

History dashboard  
Search and filter panel  
On the History page, you can view a grid of calls and Social Connector interactions made during the default  
configured time period.  
You can perform a search on previous interactions using the search field and advanced filters. For details, see  
Search for an interaction.  
You can also set the My Calls filter to automatically display only your interactions. For details, see Search using  
filters.  
Note  
l  
The default time period for the displayed search results is configurable and is set by the SQL Job.  
l  
By default, the process that retrieves the latest updates of call interactions for Contact History runs every  
60 minutes. If you require more frequent updates, speak to you Administrator.  
From the interaction history search results, you can do the following.  
CCSP UI Help - Agent - 7.4  
91

Contact actions  
In the Contact Name column:  
To...  
Click...  
Next...  
Open a contact's page  
The name link  
See Contacts dashboard.  
Add an unregistered contact  
See Add an unregistered contact.  
Interaction actions  
In the Actions column:  
To...  
Click...  
Next...  
View interaction details  
Interaction  
See View interaction details.  
Play a recording of an interaction  
Play Recording  
See Play a recording of an interaction.  
Make an outbound voice or email call  
Call or  
Email  
See Make an outbound voice or email  
call.  
Send an outbound message to a  
social channel  
Chat history  
See Send an outbound message.  
CCSP UI Help - Agent - 7.4  
92

Search for an interaction  
On the History page, you can search on all the fields or specify filters.  
You can use any combination of these search types:  
l  
Single search only – returns records where the requested text exists in one of the single-search text fields  
(itemized below). It also uses the default values in the Filters, such as dates.  
l  
Filters only – when no single-search is supplied, all data is returned that matches the configured filters.  
l  
Both – returns records where the requested text exists in one of the single-search text fields (itemized  
below) and these matched records are additionally filtered by the values you specify in the Filters section. It  
is the same as single-search but with additional non-default filters applied.  
Search all fields  
To search for text contained in the interaction record:  
1.  
Open the History page.  
2.  
Type the search text in the Search all fields box and click  
.  
3.  
To clear the search criteria, click  
.  
The single-search looks for the requested text in any of these fields.  
Properties  
Search values (example)  
Call Type  
Chat Call, Voice Call  
Caller ANI  
The caller number  
Caller DNIS  
The called number  
Caller Name  
The name of the caller  
CCSP UI Help - Agent - 7.4  
93

Properties  
Search values (example)  
Queue Name  
The name of the queue where the call is delivered and handled  
Agent Name  
The name of the agent who handle the interaction  
WrapUp Name  
The name of the assigned wrap-up  
WrapUp Data  
The wrap-up note data/information  
Email Subject  
The subject of the email interaction  
Channel Name  
The channel name (such as WhatsApp/Facebook) from Social Connector  
interactions  
AccountDetails  
Identifier  
The user identifier used to match to accounts via Social Connector  
SCFromID  
The user identifier (ID) from Social Connector  
SCToID  
The bot identifier (ID) from Social Connector  
Search using filters  
If you need to perform a search based on more complex criteria:  
1.  
Open the History page.  
2.  
Click  
to expand the search attributes and properties.  
CCSP UI Help - Agent - 7.4  
94

3.  
Select or enter the advanced filter criteria. You can search for an existing interaction record based on the  
following attributes:  
o  
Start Date and End Date  
o  
Account name  
o  
Contact details identifier  
o  
From Party and/or To Party  
o  
Direction  
o  
Interaction Type  
o  
Wrap-up Name  
o  
Queue  
o  
Call ID  
o  
Caller Message  
o  
Channel Name  
o  
Agent Name  
4.  
If you always want to filter for calls that you have handled, select the My Calls check box. When this option  
is enabled, each time you log in to CCSP UI, your name automatically appears in the Agent Name filter.  
CCSP UI Help - Agent - 7.4  
95

In the following example, calls are filtered by the caller's number, entered in From Party.  
CCSP UI Help - Agent - 7.4  
96

Manage interactions  
This section contains the following topics:  
l  
Add an unregistered contact  
l  
View interaction details  
l  
Play a recording of an interaction  
l  
Make an outbound voice or email call  
l  
Send an outbound message to a social channel  
See also History dashboard.  
CCSP UI Help - Agent - 7.4  
97

Add an unregistered contact  
1.  
Open the History page.  
2.  
In the Contact Name column, click  
. The Add Contact dialog appears.  
CCSP UI Help - Agent - 7.4  
98

3.  
To add the information to a new contact, following the instructions for Add a new contact.  
4.  
To add the information to an existing contact, click the Add to existing contact tab.  
5.  
Search for the existing contact. The matching contacts appear below the search field.  
6.  
Click  
. The message Record was successfully added appears at the bottom of the screen.  
CCSP UI Help - Agent - 7.4  
99

View interaction details  
1.  
Open the History page.  
2.  
Search for the interaction.  
3.  
In the Actions column, click  
Interaction. The Interaction Details dialog appears.  
4.  
For chat calls, you can view the Conversation log.  
5.  
To close the dialog, click X.  
CCSP UI Help - Agent - 7.4  
100

Play a recording of an interaction  
Note  
Your Administrator can configure Replay or Agent Evaluation to review interactions.  
1.  
Open the History page.  
2.  
Search for the interaction.  
3.  
In the Actions column, click  
Play Recording. CCSP Replay opens in a new tab.  
Make an outbound voice or email call  
Note  
If you cannot make outgoing calls or send emails from the Contact History gadget, ask your Administrator  
to check that the touchPointType of channel is set to email or call.  
1.  
Open the History page.  
2.  
Search for the interaction.  
3.  
In the Actions column, click  
Call or  
Email. The CCSP UI call card appears.  
CCSP UI Help - Agent - 7.4  
101

Send an outbound message to a social channel  
Note  
l  
If you cannot make outgoing calls or send emails from the Contact History gadget, ask your  
Administrator to check that the touchPointType of channel is set to email or call.  
l  
If you want to view more than 5 conversations at a time, ask your Administrator to change this setting.  
1.  
Open the History page.  
2.  
Search for the interaction.  
3.  
In the Actions column, click  
Chat history. The conversation history dialog appears.  
4.  
If there are more conversations than the configured limit, click Show 5 more conversations to view pre-  
vious conversations.  
5.  
Type your message and click Send.  
CCSP UI Help - Agent - 7.4  
102

Contacts dashboard  
Contacts panel  
On the Contacts page, you can perform a search on a user account from the search field. See Search for a  
contact.  
If you have permission to modify contacts, you can also Add a new contact.  
Details tab  
When a contact's information is retrieved, their identifiers are presented on the Details tab.   
On this tab, you can initiate a voice call or chat via the interactive icon (if it is available and depending on the  
configuration of Contact History and Social Connector).  
If you have permission to modify contacts, you can also perform the following actions.  
Contact actions  
To...  
Click...  
Next...  
Edit a contact  
See Edit a contact  
Delete a contact  
See Delete a contact  
CCSP UI Help - Agent - 7.4  
103

To...  
Click...  
Next...  
Add a contact identifier  
See Add a contact identifier  
Edit a contact identifier  
See Edit a contact identifier  
Delete a contact identifier  
See Delete a contact identifier  
Import or export contacts  
See Import contacts or Export contacts  
History tab  
On the History tab, you and/or your supervisor can view all past call interactions between the contact and the  
call center.  
You and/or your supervisor can view details of each interaction, including conversation history (available only  
for Social Connector channel calls), call arrival and termination time, queue information, call type, ANI, and  
DNIS details.  
You can also perform the following actions.  
CCSP UI Help - Agent - 7.4  
104

Interaction actions  
In the Actions column:  
To...  
Click...  
Next...  
View interaction details  
Interaction  
See View interaction details.  
Play a recording of an interaction  
Play Recording  
See Play a recording of an  
interaction.  
Make an outbound voice or email call  
Call or  
Email  
See Make an outbound voice or  
email call.  
Send an outbound message to a  
social channel  
Chat history  
See Send direct message.  
Dashboard tab  
On the Dashboard tab, in the Interactions panel, you can view the contact's previous interaction stats by  
interaction type, the total number of previous interactions, and the last interaction date and time.  
The Survey and Evaluation statistics panels are optional and require specific features to be deployed.  
CCSP UI Help - Agent - 7.4  
105

Interactions  
Stat name  
Description  
Total  
Total number of interactions  
Chat non-social  
Total records for Non-social Connector chat  
Chat social  
Total records for Social Connector chat  
Phone  
Total records for Phone interaction  
Email  
Total records for email interaction  
Voice  
Total records for Voice call interaction  
Callback  
Total records for created callback(s)  
Voicemail  
Total records for created voicemail  
Surveys  
Stat name  
Description  
Last Survey Result  
Total number of survey records that was completed by the caller  
CCSP UI Help - Agent - 7.4  
106

Stat name  
Description  
CES  
Customer effort score displayed as a percentage  
CAST  
Customer sanctification score displayed as a percentage  
Evaluation  
Evaluation displayed as a percentage  
NPS  
The Net Promoter Score for the surveys made, displayed as a percentage  
Sentiments (evaluation)  
Stat name  
Description  
Sentiment Positive  
Total number of submitted positive sentiments  
Last Sentiment Positive  
The last date where a positive sentiment was submitted  
Sentiment Negative  
Total number of submitted negative sentiments  
Last Sentiment Negative  
The last date where a negative sentiment was submitted  
Sentiment Neutral  
Total number of submitted neutral sentiments  
Last Sentiment Neutral  
The last date where a neutral sentiment was submitted  
CCSP UI Help - Agent - 7.4  
107

Direct Message history  
The Direct Message history tab is a feature of Social Connector call detail record (CDR) chat conversation  
recording.  
On this tab, you and/or your supervisor can view the existing conversation records between the contact and an  
agent and also send a direct message from this session. See Send direct message.  
Fields tab  
The Fields tab shows information such as the region and job title records that were assigned to the contact.  
CCSP UI Help - Agent - 7.4  
108

These fields serve as keywords for Contact Search. When the field name matches, the search shows all contacts  
assigned to that field name in the results.  
These fields in the database are defined by the Administrator. While you can assign these fields from Contact  
History, you cannot create or edit them within Contact History. The same restrictions apply to the Organization  
field.  
Note  
For the Administrator:  
The Organization and custom fields are related to accounts and do not pertain to call history. You can  
create a list of organizations in the database, or have a custom external import process manage this.  
Custom fields consist of small lists of data that can be linked to an account, while organizations can be  
represented in a larger list. Examples of custom fields include job titles, sites, or any other arbitrary  
categories.  
Related Accounts tab  
The same name can be assigned to multiple contact accounts within different organizations with externally  
imported data. Use the Related Accounts tab to view all the contact accounts with the same name as the  
selected contact.  
CCSP UI Help - Agent - 7.4  
109

Search for a contact  
On the Contacts page, you can search on the following fields:  
l  
First name  
l  
Last name  
l  
External CRM ID  
l  
Organization  
l  
Custom fields, such as Region (see Fields tab)  
1.  
Open the Contacts page.  
2.  
In the Contacts panel search bar, do one of the following:  
o  
Start typing in the search box to automatically filter the list.  
o  
Type a keyword. See Search on organization or field below.  
3.  
Select the contact in the list to access their account information and historical interactions.  
CCSP UI Help - Agent - 7.4  
110

Search on organization or fields  
For example, this contact has the following fields assigned to values:  
l  
Region  
l  
Job Title  
l  
Organisation  
If we examine this user's account details, we can find the following information that was assigned.  
CCSP UI Help - Agent - 7.4  
111

When searching for a keyword in one of the fields, the related results are automatically included. For example,  
if I search for the Organisation name or part of it, I will receive relevant matches.  
The same applies if I search for the Region name or part of it.  
CCSP UI Help - Agent - 7.4  
112

CCSP UI Help - Agent - 7.4  
113

Manage contacts  
This section contains the following topics:  
l  
Add a new contact  
l  
Edit a contact  
l  
Delete a contact  
l  
Add a contact identifier  
l  
Edit a contact identifier  
l  
Delete a contact identifier  
l  
Import contacts  
l  
Export contacts  
See also Contacts dashboard.  
Note  
Actions related to modifying data (edit/add/delete) may be limited to specific roles. If you should be able to  
manage contacts but cannot, contact your Administrator.  
CCSP UI Help - Agent - 7.4  
114

Add a new contact  
Note  
You can create a contact with identical first and last names.  
However, you cannot create a duplicate identifier for any voice or social channels, such as duplicate phone,  
SMS, or WhatsApp numbers. Each identifier is unique and can only be registered under one contact  
account.  
1.  
Open the Contacts page.  
2.  
Click  
. The Add Contact dialog appears.  
CCSP UI Help - Agent - 7.4  
115

3.  
Enter the following information:  
o  
First name (mandatory)  
o  
Last name (mandatory)  
o  
External CRM ID — a unique identifier for the contact, helping to distinguish between the same names.  
o  
Channel (mandatory) — select the channel, for example, Facebook.  
o  
Contact identity (mandatory) — for example, 123456789@whatsapp.com, a phone number, or email.  
o  
Comment — notes about the contact.  
o  
Alias — the alias for the channel, such as a user name in Facebook.  
o  
Region — select from the list. To filter the list, start typing the region.  
o  
Job title — select from the list. To filter the list, start typing the job title.  
o  
Organization — select from the list. To filter the list, start typing the organization.  
4.  
Click Add Contact. The new [contact's name] page appears.  
CCSP UI Help - Agent - 7.4  
116

Edit a contact  
1.  
Open the Contacts page.  
2.  
Search for the contact.  
3.  
On the [contact name's] page, on the Details tab, click  
. The Edit Contact [contact's name]  
dialog appears.  
4.  
Edit the information. For details, see Add a new contact.  
5.  
Click Save. The edited information appears on the [contact name's] page.  
CCSP UI Help - Agent - 7.4  
117

Delete a contact  
Caution  
Deleting a contact permanently erases the contact details. It cannot be undone.  
1.  
Open the Contacts page.  
2.  
Search for the contact.  
3.  
On the [contact's name] page, on the Details tab, click  
. The Delete Contact dialog  
appears.  
4.  
Type DELETE (uppercase) in the field. The Delete button is now active.  
5.  
Click Delete.  
CCSP UI Help - Agent - 7.4  
118

Add a contact identifier  
1.  
Open the Contacts page.  
2.  
Search for the contact.  
3.  
On the [contact's name] page, on the Details tab, click  
. The Add [contact's name] dialog  
appears.  
4.  
Enter the following information:  
o  
Channel (mandatory) — select the channel, for example, Facebook.  
o  
Contact identity (mandatory) — for example, 123456789@whatsapp.com, a phone number, or email.  
o  
Alias — the alias for the channel, such as a user name in Facebook.  
o  
Description — for example, Home phone or Work contact.  
5.  
Click Add. The new contact identifier appears on the [contact's name] page.  
CCSP UI Help - Agent - 7.4  
119

Edit a contact identifier  
1.  
Open the Contacts page.  
2.  
Search for the contact.  
3.  
On the [contact's name] page, on the Details tab, click  
Edit in the row of the contact identifier to  
edit. The Edit [contact's name] dialog appears.  
4.  
Edit the contact identifier information. For details, see Add a contact identifier.  
5.  
Click Save. The edited contact identifier information appears on the [contact's name] page.  
CCSP UI Help - Agent - 7.4  
120

Delete a contact identifier  
1.  
Open the Contacts page.  
2.  
Search for the contact.  
3.  
On the [contact name's] page, on the Details tab, click  
Delete in the row of the contact identifier to  
delete.  
4.  
In the Delete Contact Details dialog, click Delete.  
CCSP UI Help - Agent - 7.4  
121

Import contacts  
Note  
l  
The import file must be an Excel file with the two tabs and columns specified below.  
l  
The OfficialID field appears on both tabs and links the account with the account details.  
l  
You can only import new accounts. You cannot edit existing ones.  
Excel file  
Prepare the Excel file to import:  
1.  
Create an Excel file with two tabs: Accounts and AccountDetails.  
2.  
On the Accounts tab, add the following columns:  
3.  
On the AccountDetails tab, add the following columns:  
4.  
Add your data to this file.  
Import process  
Import into Contact History:  
1.  
Open the Contacts page.  
2.  
In the Contacts panel, click  
and then click Import contacts.  
3.  
In the Open dialog, select the Excel file and click Open.  
CCSP UI Help - Agent - 7.4  
122

Export contacts  
Note  
The contacts are exported as an Excel file, with two tabs: Accounts and AccountDetails. For more  
information, see Import contacts.  
1.  
Open the Contacts page.  
2.  
In the Contacts panel, click  
and then click Export contacts.  
3.  
The Excel file is exported to your Downloads folder.  
CCSP UI Help - Agent - 7.4  
123

Send direct message  
Send Direct Message enables an agent to use Social Connector to proactively send messages ad-hoc without a  
corresponding contact-center chat conversation.  
For example, Twitter/X Tweets and Facebook posts can be queued for a CCSP agent to collect and respond to  
without waiting for a subsequent response from the recipient.  
From the perspective of Contact History, Send Direct Message enable agents to visually see the conversation  
history, when the channel’s CDR is enabled in Social Connector.  
Agents can see the conversation history from the History and the Contacts dashboards.  
On the History dashboard, click the  
Chat history icon to view all the historical conversations for the  
identifier.  
On the Contacts dashboard, on the Details tab, click the  
Chat icon for the same view of all the historical  
conversations for the identifier  
CCSP UI Help - Agent - 7.4  
124

For more information, see Send an outbound message.  
If there are more conversations than the configured limit a button will be presented at the top to allow the user  
to get some previous conversations (the initial quantity loaded, and the amount subsequently loaded each time  
are configurable – if users prefer to have fewer/more than currently they may contact their Administrators to  
review changing):  
The editor at the bottom may allow more text formatting than the channel permits, and most channels require  
that the text is reformatted (for example, putting \* either side to signify bold text) and this reformatted text is  
displayed in this view when the conversation is revisited.  This is also performed by Social Connector status  
messages, for example, where the text has \_ either side for displaying in WhatsApp as italics.  
Send Direct Message as a standalone gadget  
Administrators can create and assign a standalone gadget in CCSP UI that enables the agent to send direct  
messages through a specific media channel, such as SMS, WhatsApp, or Instagram messaging.  
This customized gadget allows the agent to send a message directly without having to navigate through the  
Contact History or Interactions pages.  
A preview of SMS gadget (as a standalone gadget).  
CCSP UI Help - Agent - 7.4  
125

The agent can send a direct SMS to the recipient using the standalone gadget.  
CCSP UI Help - Agent - 7.4  
126

View errors  
Error notifications  
When an error occurs, a red error notification appears at the bottom of the screen, providing brief error  
information.  
To learn more about the error, open the Errors dashboard for additional information.  
CCSP UI Help - Agent - 7.4  
127

Errors dashboard  
The Errors dashboard provides additional information about issues or errors related to the Contact History.  
The standard errors are caused by:  
l  
Adding a duplicated identifier to a contact  
l  
Misconfiguration of Contact History  
l  
Misconfiguration of Social Connector Helper API  
l  
Networking issues, firewall, or SSL configuration  
l  
Missing required configuration files  
Common errors  
The following errors can be caused due to misconfiguration:  
l  
Contact History not being updated – Contact History list is occasionally cached to reduce the load on the  
database. By default, {HISTORICAL\_DB\_NAME}\_CHH\_Cache SQL Job runs once per hour. Running intervals  
can be changed in job’s properties by Administrators.  
l  
Account is not found in Contact History list – when you have a related account in Contact History but Contact  
History still displays the Add button – make sure, that the AccountDetailsIdentifier is exactly the same as in  
ANI for example, +123@whatsapp.com and 123@whatsapp.com are different identifiers.  
l  
A duplicated account identifier is found  – when you attempt to add an identifier to an account that exists in  
the system, an error occurs and the information is displayed on the Errors dashboard for more information.  
CCSP UI Help - Agent - 7.4  
128

l  
Add, delete and edit functionality is not available – your role may not have permissions to perform account  
actions. Contact your Administrator to ensure you are configured for account management.  
l  
OPTIONS {Service API URL} net::ERR\_CONNECTION\_REFUSED – means the connection to Social Connector is  
either disabled or unreachable from your desktop. Contact your Administrator to resolve the connection  
issues.  
CCSP UI Help - Agent - 7.4  
129

# Contact Center: Service Provider CCSP UI Help - Supervisor Version 7.4

Contents  
1: Intro  
6  
What's new  
6  
Legal disclaimer  
6  
Support  
6  
2: Overview  
7  
Application overview  
7  
Types of connections  
8  
Types of CCSP UI users  
8  
Supervisor tools  
8  
Configure your browser  
10  
Clear your browser's cache  
10  
Log in  
11  
Password expired message  
12  
Additional login messages  
12  
Main page  
13  
Agent view  
13  
Supervisor view  
13  
Gadgetory  
14  
Calls Menu  
15  
Supervisor view  
16  
Supervisor tab  
16  
Dashboard  
16  
Historical Reports  
17  
Replay tab  
17  
Gadgets and buttons  
18  
Keyboard shortcuts  
21  
Notification area icons and menu  
21  
CCSP UI icon colors  
21  
Screen recording upload service icons  
21  
Menu of available actions  
22  
Use an alternative device  
23  
Log in and be available  
24  
Answer a call on an alternative device  
25  
Hold a call on an alternative device  
25  
Make a call on an alternative device  
26  
Transfer or conference a call on an alternative device  
26  
End a call on an alternative device  
26  
Retrieve voicemail on an alternative device  
26  
Use an alternative device remotely  
27  
View the Mini Wallboard  
28  
View reports  
30  
Real Time Reports  
30  
CCSP UI Help - Supervisor - 7.4  
2

Historical Reports  
30  
Set your preferences  
31  
Change your password  
34  
Expired password  
34  
Report a problem  
35  
Send an outgoing email  
36  
Request supervisor assistance  
37  
Receive broadcast messages  
37  
Log out  
38  
Additional logout messages  
38  
Session expiry  
38  
Remote logout  
39  
3: Calls  
40  
Become available or released  
41  
Become available  
41  
Become released  
42  
Answer a call  
43  
Refuse a call  
44  
Defer an email or a voicemail call  
45  
Process a voice call  
46  
Process a chat call  
47  
Process an email call  
49  
Process a callback call  
51  
Process a voicemail call  
52  
View call information  
53  
Control media recording  
54  
Control screen recording  
55  
Hold a call  
56  
Make an outgoing call  
57  
Transfer or conference a call  
60  
Hang up from a conference call  
62  
Request a chat consultation  
63  
Adjust the volume  
66  
Mute a call  
66  
Schedule a callback  
67  
Use canned phrases  
69  
Co-browse (Web collaboration)  
70  
Co-browse example  
71  
Wrap up a call  
77  
4: Messages  
78  
View messages  
79  
Regular agent view  
80  
Supervisor view  
81  
CCSP UI Help - Supervisor - 7.4  
3

View subfolders  
82  
Page navigation  
83  
Messaging items visibility by role  
85  
Callbacks  
85  
Email  
86  
Voicemail  
89  
Filter messages  
91  
Groups and teams filters  
91  
Defer type filter  
92  
Filter by group, team, and defer type  
92  
Filter by text contained in the items  
93  
Filter by date and time  
94  
Sort messages  
94  
Reset folder to default filter and sort settings  
96  
Pull messages  
97  
Remove messages  
98  
Restore messages  
99  
Source folders  
99  
Destination folders  
100  
Access archived messages  
101  
5: Personal voicemail  
103  
Retrieve personal voicemail  
103  
Set up your personal mailbox  
104  
Set up or edit your mailbox preferences  
104  
Change the password  
104  
Record a greeting/name  
104  
Change your PIN  
105  
Personal voicemail quick reference  
106  
Log on  
106  
Play Messages menu  
106  
Mailbox preferences  
107  
6: Contact History  
108  
Contact History language  
108  
Open Contact History  
109  
History dashboard  
110  
Search and filter panel  
110  
Contact actions  
111  
Interaction actions  
111  
Search for an interaction  
112  
Search all fields  
112  
Search using filters  
113  
Manage interactions  
116  
Add an unregistered contact  
117  
CCSP UI Help - Supervisor - 7.4  
4

View interaction details  
119  
Play a recording of an interaction  
120  
Make an outbound voice or email call  
120  
Send an outbound message to a social channel  
121  
Contacts dashboard  
122  
Contacts panel  
122  
Details tab  
122  
History tab  
123  
Dashboard tab  
124  
Direct Message history  
127  
Fields tab  
127  
Related Accounts tab  
128  
Search for a contact  
129  
Manage contacts  
133  
Add a new contact  
134  
Edit a contact  
136  
Delete a contact  
137  
Add a contact identifier  
138  
Edit a contact identifier  
139  
Delete a contact identifier  
140  
Import contacts  
141  
Export contacts  
142  
Send direct message  
143  
Send Direct Message as a standalone gadget  
144  
View errors  
146  
Error notifications  
146  
Errors dashboard  
147  
Common errors  
147  
7: Supervisor  
149  
Filter agents  
149  
View agent interactions  
150  
Monitor agents  
151  
Use camp on monitoring  
152  
Answer an assistance request  
153  
Initiate agent assistance  
154  
Broadcast messages  
154  
Remotely log out agents  
155  
CCSP UI Help - Supervisor - 7.4  
5

1: Intro  
The CCSP user interface (UI) is a browser-based application that enables users to receive voice, chat, email,  
callback, and voicemail calls that arrive at a contact center. Calls are automatically directed to particular users  
because they are available and have the expertise to handle the calls.  
CCSP UI was previously called TouchPoint.  
Supervisor Help is for CCSP supervisors and managers of a contact center.  
What's new  
Version 7.4  
Contact History — added Contact History, a CCSP add-on for managing contacts and contact account details.  
Request a chat consultation — an agent can request a chat consultation during a call (if enabled). Requires  
HF74-66036.  
Set your preferences — if the selected SIP input and/or output device becomes unavailable during your CCSP  
UI login session (for example, it is unplugged from your computer), a warning appears in both CCSP UI and the  
notification area of your task bar. Requires HF74-64991.  
Refuse a call — when you refuse a call, you may automatically become released if this has been configured by  
your system administrator. Requires HF74-64991.  
Log in and Log out — added process for CCSP single sign-on. Requires HF74-61362.  
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.  
CCSP UI Help - Supervisor - 7.4  
6

2: Overview  
This section contains the following topics and subsection:  
l  
Quick reference cards  
l  
Application overview  
l  
Configure your browser  
l  
Clear your browser's cache  
l  
Log in  
l  
Main page  
l  
Supervisor view  
l  
Gadgets and buttons  
l  
Keyboard shortcuts  
l  
Notification area icons and menu  
l  
Use an alternative device  
l  
View the Mini Wallboard  
l  
View reports  
l  
Set your preferences  
l  
Change your password  
l  
Report a problem  
l  
Send an outgoing email  
l  
Request supervisor assistance  
l  
Receive broadcast messages  
l  
Log out  
l  
Remote logout  
Application overview  
Types of connections  
Types of CCSP UI users  
Supervisor tools  
CCSP UI is a browser-based application that enables users to receive calls that arrive at contact center phone  
numbers. Calls are automatically directed to particular users because they are available and have the expertise  
to handle the calls. Users can also receive chat calls.  
If the system administrator provides email service, emails are also directed to the user from email coming into  
the business. Users may also be able to receive voice calls on a direct extension.  
CCSP UI Help - Supervisor - 7.4  
7

CCSP can also direct outbound calls to users who participate in outbound dialing campaigns, a capability  
provided by the Dialer component of CCSP. As with email, the system administrator must provide this  
capability.  
Types of connections  
CCSP provides Automatic Call Distribution (ACD) and direct inward dialing (DID) voice connections to users, in  
addition to other media such as email and chat.  
ACD automatically queues outside calls and then routes them to those users best able to handle the calls. ACD  
also works with CCSP Dialer to route calls for outbound dialing campaigns to users.  
DID enables outside callers to dial a user’s specific phone number.  
Types of CCSP UI users  
Various types of users of the CCSP system can use CCSP UI. The available features differ for each user type and  
some features may not apply to your environment.  
The types of CCSP UI users are:  
l  
Business users — standard office workers  
l  
ReadyReps — knowledge workers and others who sometimes receive ACD-routed calls  
l  
Agents — contact center customer service representatives who spend the majority of their day handling  
ACD-routed customer contacts  
l  
Supervisors — agents with supervisory duties over other agents. Agents are assigned to groups according  
to their expertise and, if the contact center needs an administrative collection of agents different from  
groups, to teams. Supervisors have supervision responsibilities and CCSP capabilities only for the agents  
that are in their groups or teams. A supervisor not restricted to specific groups or teams has supervisory  
capabilities over all agents in the contact center.  
The Business User software license provides basic unified communication capabilities. A business user can only  
be reached by direct dial or direct transfer. A business user cannot be a destination of CCSP ACD or participate  
in outbound dialing campaigns. A business user without a CCSP UI license is not a CCSP UI user and cannot log  
on to CCSP UI.  
Agents and ReadyReps can be used as ACD destinations. Agent licenses are for traditional, formal contact  
center agents. ReadyRep licenses are designed for use by people who are not solely or even primarily  
dedicated to a formal contact center, but who need to be part of the contact center infrastructure in most other  
ways. They are sometimes known as informal agents or experts, and their ACD use is part-time. Supervisors can  
monitor and remotely manage agents and ReadyReps.  
The instructions in this Help are for all users of CCSP UI. If a specific instruction is for an agent or a supervisor,  
the instruction addresses agents or supervisors.  
Supervisor tools  
The tools available to supervisors include:  
CCSP UI Help - Supervisor - 7.4  
8

l  
Monitor agents. The monitoring tools include:  
o  
Silent — use to listen to conversations between agents and customer contacts.  
o  
Whisper — use to speak to an agent on a channel separate from the channel between an agent and a  
customer contact.  
o  
Barge in — use to enter into a conversation between an agent and customer contact.  
o  
Stop — use to stop monitoring a selected agent.  
l  
Remote logout — use to log off selected agents.  
l  
Reports:  
o  
Real Time Reports — use to view performance measures for your contact center and individual agents  
during the last 24 hours or less. Use Dashboard to create and display reports. You can also display  
reports created by other supervisors.  
o  
Historical Reports — use to view contact center and individual agent statistics.  
l  
Replay — use to view call details and play back recordings of agent-caller interactions.  
See Supervisor view.  
CCSP UI Help - Supervisor - 7.4  
9

Configure your browser  
Before running CCSP UI from your browser, some initial preparation is required:  
l  
For Microsoft Edge,you must enable localhost access on your computer by running the following command  
from an administrator command prompt:  
CheckNetIsolation LoopbackExempt -a -n=Microsoft.MicrosoftEdge\_8wekyb3d8bbwe.  
l  
For Ubuntu client, to play voicemail, you need to install a codec for MP4.  
l  
To start CCSP UI in Toolbar mode, you must allow pop-ups.  
l  
Do not disable Auto Complete in your browser. CCSP UI uses this option to allow redials and to display your  
call history.  
l  
Do not disable cookies in your browser. CCSP UI uses this option to save your audio preferences.  
l  
For Chrome and Firefox, no specific settings are required.  
Note  
CCSP UI supports TLS 1.0.  
Clear your browser's cache  
After you update CCSP UI by running the CCSP Client Services Installer, you must clear your browser's cache to  
ensure that you run the updated version of CCSP UI.  
If you use Edge:  
1.  
On the Edge menu, click Settings.  
2.  
Click Privacy, search, & services.  
3.  
Under Clear browsing data, click Choose what to clear every time you close the browser.  
4.  
Select the Cached images and files option.  
If you use Chrome:  
1.  
On the Chrome menu, click Settings.  
2.  
Click Show advanced settings.  
3.  
Under Privacy, click Clear browsing data.  
4.  
In the Clear browsing data dialog, select the time period, select the Cached images and files check box,  
and then click Clear browsing data.  
If you use Firefox:  
1.  
On the Firefox menu, click Settings and then click History.  
2.  
Click Clear recent history.  
3.  
In the Clear Recent History dialog, select the time range to clear.  
4.  
Expand Details and then select the Cache check box.  
5.  
Click Clear Now.  
CCSP UI Help - Supervisor - 7.4  
10

Log in  
The log in process is slightly different depending on the identity provider configured by your administrator.  
Agents and supervisors log on the same way. You need a supervisor user name and password to use the  
supervisor interface.  
Note  
l  
If Teams is enabled, it must be running before you log on to CCSP UI.  
l  
To start CCSP UI in Toolbar mode, you must allow pop-ups.  
l  
If you use a voice device with Keep Connection Open (KCO), and you are on a call when your browser  
crashes or your computer loses its network connection, you must complete the active call before you log  
on to CCSP UI again.  
l  
If your permission profile is changed, you must log off and log on again.  
1.  
In your browser, in the Address bar, type the URL supplied by your supervisor or administrator, for  
example, https://<touchpoint host/TouchPoint/  
2.  
Click Go or press Enter. The Welcome to CCSP UI page appears.  
3.  
Click Log in. If you are already logged in to CCSP, the main CCSP UI page appears.  
4.  
If you are not already logged in to CCSP, the Welcome to CCSP page appears.  
5.  
Type your User name in User Principal Name (UPN) format, for example, someone@domainname.com and  
click Continue.  
6.  
The next step depends on the identity provider configured by your administrator:  
o  
For Auth Server, on the Welcome to CCSP page, type your Password and click Log in. Alternatively, to  
change the username, click Back to return to the previous page. If necessary, see Password expired  
message below.  
o  
For ADFS, you are redirected to the ADFS login page to authenticate. Please consult your local  
procedures for how to proceed from this point.  
o  
For Enghouse Identity Server (EIS), you are redirected to the EIS login page. Type your Password and  
click Login.  
7.  
If necessary, see Additional login messages below.  
8.  
If your administrator has configured the Client Services options to be presented, the option to continue  
with or skip Client Services appears:  
o  
To run CCSP UI with Client services, click Continue with Client Services. The Client Services are  
installed and activated, depending on the configuration by your administrator.  
o  
To run CCSP UI without Client services, click Skip Client Services.  
The main CCSP UI page appears.  
CCSP UI Help - Supervisor - 7.4  
11

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 Auth Server login 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:  
l  
Auth Server is not available.  
l  
Failed to get authorization for TouchPoint use. This can occur because:  
o  
Your user profile is not permitted to use the CCSP UI application. Only agents, supervisors and profiles  
with supervisor permissions are permitted to use the CCSP UI application.  
o  
Your user has not been assigned to a group. Users must be allocated to a group to use CCSP UI.  
o  
Your user has no skills assigned. Users must have at least one skill assigned to use CCSP UI.  
o  
Your user has been assigned to a group with no allocated queue. A user must be belong to a group that  
has at least one queue assigned to it.  
l  
Failed to login to AIS. If one of the following messages appears, contact your administrator:  
o  
License Control Server Warning: The License Control Server is not working. Soon you will not be able to login.  
o  
Login Failed: Unable to login. The maximum number of concurrent agents exceeded the license definition,  
please try again later.  
o  
Login Failed: Unable to login. The maximum number of user logins have been exceeded, please try again later.  
l  
You are not authorized to view this page. This can occur because:  
o  
You are logged in to an external identity provider with a different username than the one you entered  
on the Auth Server Login Selector page to log in to the CCSP client application.  
o  
You entered a username on the Auth Server Login Selector page and were redirected to an external  
identity provider, where you log on with a different username.  
CCSP UI Help - Supervisor - 7.4  
12

Main page  
The CCSP UI page provides access to all of the capabilities of CCSP UI.  
Agent view  
Supervisor view  
CCSP UI Help - Supervisor - 7.4  
13

For details, see Supervisor view.  
Gadgetory  
In CCSP UI, the toolbar is called the gadgetory. Notification appear on top of the gadget icons in the gadgetory,  
for example, how many active calls you have.  
Depending on your CCSP UI configuration, you can change the applications state to one of the following:  
l  
Tab — use for applications that need maximum screen space.  
l  
Float — use for applications that you want to view side-by-side.  
l  
Pop out — use for applications that you want to view on a separate screen.  
l  
Minimize  
To change the application state, click the down arrow on the tab and then click the state.  
For information about the individual gadgets, see Gadgets and buttons.  
CCSP UI Help - Supervisor - 7.4  
14

Calls Menu  
The Calls Menu shows active interactions and opens when you click  
or, if you selected Answer Calls  
Automatically in your preferences, when a new call arrives.  
The Calls Menu contains the following call types:  
l  
Voice  
l  
Chat  
l  
Email  
l  
Callback  
l  
Voicemail  
To close the Calls Menu, click the left arrow at the top of the menu or click  
again.  
CCSP UI Help - Supervisor - 7.4  
15

Supervisor view  
Supervisors have the tabs available to agents and the following:  
l  
Supervisor tab — lists all agents and their statuses and assists monitoring functions.  
l  
Dashboard — provides a variety of real-time statistical reports you can view regarding your contact center.  
l  
Historical Reports — provides a variety of longer term statistical reports you can view regarding your contact  
center.  
l  
Replay — enables you to view call details and play back recordings of agent-caller interactions.  
Dashboard and Historical Reports open in separate browser tabs.  
Supervisor tab  
To view the Supervisor tab, click  
Supervisor.  
To communicate with a specific agent, double-click the agent's name to open the agent card.  
Dashboard  
To view the Dashboard, in the gadgetory, click  
Real Time Reports. For more information, see the CCSP  
Dashboard Help.  
CCSP UI Help - Supervisor - 7.4  
16

Historical Reports  
To view Historical Reports, click  
Historical Reports. For more information, see the CCSP Historical Reports  
User Guide.  
Replay tab  
To view Replay, click  
Replay. For more information, see the CCSP Replay Help.  
CCSP UI Help - Supervisor - 7.4  
17

Gadgets and buttons  
The specific gadgets and buttons that you see in CCSP UI depend on your permissions.  
Gadget  
Description  
Available — click to become released.  
Released — click to become available.  
Calls Menu - click to view active interactions.  
Outgoing Call — click to make an outgoing call.  
Outgoing Email — click to send an outgoing email.  
Supervisor — click to view the Supervisor tab.  
Message View — click to handle messages.  
Callback — click to schedule a callback.  
Assistance — click to request supervisor assistance.  
Voicemail — click to retrieve personal voicemail.  
Historical Reports — click to view Historical Reports on a separate browser  
tab.  
Replay — click to view CCSP Replay.  
Real Time Reports — click to view Dashboard on a separate browser tab.  
Contact History — click to manage interactions and contacts.  
Mini Wallboard — click to view queue statistics.  
Settings — click to set your preferences, change your password, report a  
problem, or open the Help.  
Screen Recording — click to manually control screen recording.  
Logout — click to log out.  
CCSP UI Help - Supervisor - 7.4  
18

Button  
Description  
Answer — click to answer a call.  
Broadcast — click to send a message to selected agents.  
Call — click to call the selected user or endpoint.  
Call information — click to view call details.  
Callback — click to schedule a callback.  
Canned phrases — click to use canned phrases in email and chat  
calls.  
Defer or pull call— click to defer or pull an email or voicemail call.  
Dial pad — click to open the dial pad in voice calls.  
Hang up — click to refuse a call.  
Headset volume — hover over to adjust the volume. Click to mute.  
Hold — click to put a call on hold.  
Microphone volume — hover over to adjust the microphone  
volume. Click to mute.  
Monitor — click to view the monitoring-type list.  
More options — click to view more options.  
Record — click to manually control media recording.  
Remote logout — click to remotely log off selected agents.  
Web collaboration — click to co-browse web pages with the caller in  
chat calls.  
Wrap up — click to wrap up a call.  
CCSP UI Help - Supervisor - 7.4  
19

Button  
Description  
Xfer/Conf — click to transfer or conference a call.  
CCSP UI Help - Supervisor - 7.4  
20

Keyboard shortcuts  
l  
Ctrl+Backspace — to make a new call  
l  
Win+F2 — to hold or release hold  
l  
Win+F3 — to toggle availability  
l  
Win+F6 — to answer  
l  
Win+F7 — to transfer or conference  
l  
Win+F9 — to maximize CCSP UI  
l  
Win+F12 — to hang up  
Notification area icons and menu  
Note  
You can shut down your computer while the upload service is working. The next time you log on to  
Windows, the upload service reprocesses any incomplete screen recording uploading tasks.  
When you run CCSP UI, icons appears in the notification area of your task bar. These icons change, depending  
on whether you are in call, in an unavailable, released, or available state, if you have lost the connection to the  
contact center, and if screen recordings are still uploading in the background.  
CCSP UI icon colors  
Color  
Meaning  
CCSP UI is connected and available for calls.  
CCSP UI is unavailable (released).  
CCSP UI is in call.  
CCSP UI is not connected.  
Screen recording upload service icons  
Note  
l  
After you log on to Windows, the upload icon appears.  
l  
After you log on to CCSP UI, the upload icon gets the status of the upload service, whether it is idle or  
currently uploading screen recordings.  
CCSP UI Help - Supervisor - 7.4  
21

Icon  
Meaning  
Idle — the upload service is not currently working.  
Uploading — the upload service is working.  
Menu of available actions  
To view a menu of available actions, right-click the icon that appears in the notification area of your task bar.  
CCSP UI Help - Supervisor - 7.4  
22

Use an alternative device  
This subsection contains the following topics:  
l  
Log in and be available  
l  
Answer a call  
l  
Hold a call  
l  
Make a call  
l  
Transfer or conference a call  
l  
End a call  
l  
Retrieve voicemail  
l  
Use an alternative device remotely  
If you have an alternative device managed directly by the CCSP system, you can use the alternative device  
stand-alone without CCSP UI or alongside CCSP UI. When you use the alternative device with CCSP UI, you can  
place and manage calls from both the phone and CCSP UI, and CCSP synchronizes call status on both the  
phone and CCSP UI.  
For instructions for your IP phone, see the user guide for your specific phone model. For Teams instructions,  
see the Microsoft Teams Help.  
CCSP UI Help - Supervisor - 7.4  
23

Log in and be available  
If you have Teams enabled, your CCSP UI agent status and Teams presence are synchronized. See Become  
available and released.  
If you have an IP phone, it is set up as one of the following:  
l  
Permanent extension — is always assigned to a single telephone number.  
l  
Session-based extension — can be assigned different telephone numbers and therefore to different users. A  
user begins a session by logging on to the phone.  
The extension type determines how you log on and become available.  
Permanent extension  
If your phone is set up as a permanent extension, the number is automatically assigned when it is first started  
so that you can receive direct calls. Optionally, you can also log on to CCSP UI to control your IP phone.  
You cannot receive Automatic Call Distribution (ACD)-routed calls without using CCSP UI. If your position  
requires you to receive ACD-routed calls, log on to CCSP UI and make yourself available there. You always  
become available or unavailable through CCSP UI. See Become available and released.  
When you exit CCSP UI, the ACD considers you logged on but unavailable for reporting purposes.  
If you are not set up to receive ACD calls, you may still control your availability in CCSP UIfor reporting  
purposes, but the availability setting will not affect your ability to receive direct inward dialed calls to your  
phone number.  
Session-based extension  
If your phone is set up as a session-based extension, the phone can only be used when you are logged on. If  
you try to place a call when the phone is not logged on, you receive an error message, such as Call Failed, and  
an audible fast busy signal.  
You have a numeric user ID, that is usually your phone extension number, as well as a numeric PIN. After you  
log on to the phone, your phone number is assigned to the phone. If your administrator permits it, you can  
receive direct inward dialed calls to your phone number.  
Optionally, you can use the phone with CCSP UI. To do so, log on to CCSP UI with your alphanumeric user name  
and alphanumeric password, that are usually different from the ID and PIN used for the IP phone.   
To make yourself available to receive ACD-routed calls, do one of the following:  
l  
Make yourself available from the IP phone.  
l  
Make yourself available from CCSP UI. See Become available and released.  
When you change the status from one device, the status is automatically updated in the other device.  
The mechanisms for logging on and becoming available vary from phone to phone.  
Aastra phone  
If you have an Aastra phone, it should have an ACD key that you use to manage the phone-based ACD features:  
CCSP UI Help - Supervisor - 7.4  
24

l  
Logging in  
l  
Logging out  
l  
Becoming available  
l  
Becoming unavailable  
To log on to the phone:  
1.  
Press the ACD key. A request for your user ID appears.  
2.  
Enter your user ID and select Log In. A request for your PIN appears.  
3.  
Enter your PIN and select Log In.  
When you are logged on, you are placed in the unavailable status. To make yourself available from the IP  
phone:  
1.  
Press the ACD key.  
2.  
Select Avail.  
To make yourself unavailable from the IP phone:  
1.  
Press the ACD key.  
2.  
Select Unavail.  
The Aastra ACD key provides a visual indicator of the ACD state, as follows.  
Visual indicator  
Indication  
Unlit  
Logged out  
Blinking  
Logged in and unavailable  
Lit  
Available  
Answer a call on an alternative device  
When you receive a call, the alternative device and CCSP UI both ring. Answer the call by:  
l  
IP phone — picking up the phone’s handset or pressing the IP phone's Speaker button.  
l  
Teams — clicking Answer in Teams.  
If you selected Answer Calls Automatically in the CCSP UI preferences, then the phone also goes immediately  
off-hook on the speakerphone, headset, or in Teams.  
Hold a call on an alternative device  
You can put a call on hold through CCSP UI or an alternative device.  
You must release the call from hold with the same device that you used to put the call on hold.  
For CCSP UI, see Hold a call. For an alternative device, see the instructions for your phone.  
CCSP UI Help - Supervisor - 7.4  
25

Make a call on an alternative device  
When you make a call from CCSP UI, your alternative device rings and you must answer it before the call can be  
connected at the destination.  
If the alternative device is configured to automatically pick up, the call is connected immediately.  
Transfer or conference a call on an alternative device  
Note  
To transfer a call to the system, perform all the steps from CCSP UI.  
You can transfer or conference a call through CCSP UI or an alternative device.  
You must leave a conference or complete a transfer from the same device. For CCSP UI, see Transfer or  
conference a call. For an alternative device, see the instructions for your phone.  
End a call on an alternative device  
You can end the call from either CCSP UI or an alternative device.  
If using CCSP UI and you can display wrap-up codes, then select the wrap-up code from CCSP UI.  
Retrieve voicemail on an alternative device  
To receive voicemail, your administrator or supervisor must give you a voice mailbox and a PIN. The PIN must  
be a number.  
When you have voicemail, the message waiting indicator appears in CCSP UI.  
1.  
Dial the voice mailbox. An audio message asks for your PIN.  
2.  
Click/press your PIN numbers on the device's dialpad.  
3.  
To send your PIN, click/press #. An audio message states the number of read messages and the number of  
new messages.  
4.  
Do one or more of the following:  
o  
To listen to your messages, click/press 1.  
o  
To delete the message, click/press 7.  
o  
To save the message, click/press 9.  
o  
For more options, click/press 0.  
o  
To replay the current message, click/press 4.  
o  
To hear the message envelope, click/press 5.  
o  
To return to the main menu, click/press \*.  
o  
To repeat the menu, click/press #.  
o  
To replay a message from the start, click/press 1 while the message is still playing.  
CCSP UI Help - Supervisor - 7.4  
26

Use an alternative device remotely  
You may be able to take your alternative device with you when you travel. Check with your administrator to see  
if this option is available to you. If it is available, then your device should work in any location where you have a  
public Internet connection.  
Alternatively, you may want to travel without your alternative device, and use either the built-in CCSP softphone  
feature or the Use Voice Device option to deliver your phone calls to any telephone number. If you use a  
permanent extension, this option is not available to you. If you use a session-based extension, this option is  
available to you only when you are not logged on to an alternative device.  
If you need to remotely log off from an alternative device:  
1.  
Log on to CCSP UI.  
2.  
Log off from CCSP UI. This message appears: IP phone still logged in. Do you want to log it out?  
3.  
Click Yes.  
4.  
Log on to CCSP UI again. CCSP UI will now deliver voice according to your CCSP UI preference settings.  
CCSP UI Help - Supervisor - 7.4  
27

View the Mini Wallboard  
If you have the Mini Wallboard enabled, you can view the queue statistics of the queues associated with your  
group.  
Note  
For detailed information about the measures, refer to the CCSP UI Admin Help.  
1.  
On the gadgetory, click  
Mini Wallboard. The Mini Wallboard appears.  
You can view the name of the primary measure in the tooltip.  
2.  
To expand the Mini Wallboard, click  
. The secondary measures appear.  
3.  
To view additional information, in the bottom right corner, click  
.  
CCSP UI Help - Supervisor - 7.4  
28

4.  
To minimize the Mini Wallboard, click  
.  
5.  
To view the Mini Wallboard on a separate screen, click  
.  
CCSP UI Help - Supervisor - 7.4  
29

View reports  
Real Time Reports  
Historical Reports  
Note  
Use Dashboard to create a customized wallboard for your contact center. Use large fonts so the agents  
working in the same location can view the wallboard.  
Real Time Reports  
1.  
 Do one of the following:  
o  
On the CCSP UI page, in the gadgetory, click  
Real Time Reports.  
o  
In your browser, type the Dashboard URL in the address bar.  
The logon dialog appears.  
2.  
Type your user ID and your password.  
3.  
Click Log on. The Dashboard workspace appears. For more information, see the CCSP Dashboard Help.  
Historical Reports  
1.  
Do one of the following:  
o  
On the CCSP UI page, in the gadgetory, click  
Historical Reports.  
o  
In your browser, type the Historical Reports URL in the address bar.  
2.  
Type your administrator user ID and your password.  
3.  
Click Log on. The main Historical Reports page appears. For more information, see the CCSP Historical  
Reports User Guide.  
CCSP UI Help - Supervisor - 7.4  
30

Set your preferences  
Note  
In Remote Desktop Services environments, SIP does not work as expected.  
The first time you start CCSP UI, you may have to set up your preferences to ensure that you can receive and  
make calls.  
1.  
On the gadgetory, click  
Settings and then click Preferences. The Preferences dialog appears.  
If you select SIP, the Preferences dialog appears as follows.  
CCSP UI Help - Supervisor - 7.4  
31

2.  
In Country Code, type your country code. This code can be a maximum of three numeric digits and can  
have a leading plus sign (+).  
3.  
In Area Code, type your area code. This code can be a maximum of 12 numeric digits.  
Note  
To make calls, you must enter the Country Code and Area Code information.  
4.  
Under Communication Type, do one of the following:  
o  
If you use SIP for call delivery and origination, click SIP. In Input Device and Output Device, select the  
device from the list. Test and adjust the microphone volume and headset volume.  
Note  
If the selected SIP input and/or output device becomes unavailable during your CCSP UI login  
session (for example, it is unplugged from your computer), a warning appears in both CCSP UI and  
the notification area of your task bar. You cannot receive or make voice calls using SIP until either  
the device is available again and you have re-selected it in Preferences, or you have selected  
another input/output device.  
o  
If you use a voice device for call delivery and origination, click Voice Device. In Voice Device Address,  
type your phone IP or extension. If Keep Connection Open (KCO) is enabled for your company, select  
the Keep Voice Device Connection Open check box.  
Note  
CCSP UI Help - Supervisor - 7.4  
32

If you select Keep Voice Device Connection Open, CCSP automatically logs you on again. If you  
enabled Auto Available, you will automatically become available.  
5.  
Under Answer Options:  
o  
If you want to automatically become available when you log on to CCSP UI, select the Auto Available  
check box.  
o  
If you want CCSP UI to automatically answer calls when they are delivered, select the Answer Calls  
Automatically check box.  
6.  
In External Email Address, type your email address.  
7.  
In Mailbox Pin, type the four digits of your new PIN.  
8.  
If you want to use a chat greeting, select the Use greeting check box and type the greeting in the text box,  
for example, Hello, my name is Harry, how may I assist you?  
9.  
Click Save.  
CCSP UI Help - Supervisor - 7.4  
33

Change your password  
1.  
On the gadgetory, click  
Settings and then 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.  
Expired password  
If you are already logged on and receive a message that your password has expired and must be changed, log  
off and log on again. CCSP UI prompts you to change your password.  
CCSP UI Help - Supervisor - 7.4  
34

Report a problem  
1.  
On the gadgetory, click  
Settings and then click Report a Problem. The Report a Problem dialog  
appears.  
2.  
Type a description of the problem.  
3.  
Click Send.  
CCSP UI Help - Supervisor - 7.4  
35

Send an outgoing email  
Note  
l  
Separate email addresses with a semi-colon (;). CCSP UI does not support email addresses that contain  
double quotation marks (") or are separated by a comma (,).  
l  
CCSP UI auto-saves the draft every 30 seconds.  
l  
Email address autocomplete is only available if it is enabled in CCSP UI Admin.  
l  
If you insert two or more images using drag and drop, CCSP UI only inserts the last image.  
l  
In Firefox and Edge, CCSP UI inserts canned phrases at the bottom of the email.  
l  
Background text color may not work as expected.  
l  
You cannot include a digital signature in an outbound email.  
1.  
On the gadgetory, click  
Outgoing Email. The Outgoing Email tab appears.  
2.  
In To, and optionally in CC and BCC, start typing the email address. If email address autocomplete is  
enabled, an alphabetical list of matching email addresses appears as you type. If autocomplete is not  
enabled, click  
Address lookup to see this list after you have started typing.  
By default, 20 email addresses appear in the list. If you cannot see the complete email address you want to  
use, continue typing until it appears in the list. To use an address from the list, click on it.  
3.  
Write the email. To include canned phrases, click  
. See Use canned phrases.  
4.  
Click Send.  
CCSP UI Help - Supervisor - 7.4  
36

Request supervisor assistance  
Note  
If you are a business user, supervisor assistance is not available.  
1.  
On the gadgetory, click  
. The Assistance dialog appears.  
2.  
Type the request in the text box.  
3.  
Click Send.  
4.  
Optionally type another questions in the text box and click Send.  
5.  
To end the assistance session, click End session.  
Receive broadcast messages  
Messages broadcast from your supervisor appear in the Assistance dialog.  
CCSP UI Help - Supervisor - 7.4  
37

Log out  
Note  
If you are using an external identity provider such as ADFS or EIS, when you click Log out of your CCSP  
session, you are also logged out of your external identity provider.  
1.  
On the gadgetory, click  
Logout.  
2.  
If the Logout Confirmation dialog appears, click OK.  
3.  
Do one of the following:  
o  
If you want to return to CCSP UI, click Return to Login page.  
o  
If you are logged on with SSO and have finished working with CCSP, click Log out of your CCSP  
session.  
Additional logout messages  
When you click Log out of your CCSP session, the following message may appear:  
l  
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 CCSP session again.  
The application will return to the login page. Depending on the problem resolution, this may take slightly  
longer than normal.  
Session expiry  
Your CCSP UI session will not become invalid when the login session time exceeds the maximum login session  
lifetime specified by your administrator. If you are using Teams however, your CCSP UI status will no longer be  
synchronized with your Teams client presence and you will no longer be able to search or display Teams  
contacts.  
CCSP UI Help - Supervisor - 7.4  
38

Remote logout  
If your supervisor logs you out of the contact center, a message similar to the following example appears.  
To close the message, click OK.  
CCSP UI Help - Supervisor - 7.4  
39

3: Calls  
Note  
l  
For Elsbeth campaign calls, the caller name only appears on the call card. For Sytel campaign calls, the  
caller name and number do not appear on the call card. The campaign server provides this information  
in the call information, which is automatically opened for campaign calls.  
l  
The call duration timer on the call card counts the time from when you pick up the call until you hang up.  
It does not count the time the call spent in the ringing state, pickup-pending state, and in-call  
transferred state.  
l  
If you are on one or more calls when you are unexpectedly disconnected from the network, before trying  
to log on again, complete and hang up the active calls.  
l  
The maximum outgoing call number or SIP URI is 370 characters.  
This section contains the following topics:  
l  
Become available or released  
l  
Answer a call  
l  
Refuse a call  
l  
Defer an email or a voicemail call  
l  
Process a voice call  
l  
Process a chat call  
l  
Process an email call  
l  
Process a callback call  
l  
Process a voicemail call  
l  
View call information  
l  
Control media recording  
l  
Control screen recording  
l  
Hold a call  
l  
Make an outgoing call  
l  
Transfer or conference a call  
l  
Request a chat consultation  
l  
Adjust the volume  
l  
Mute a call  
l  
Schedule a callback  
l  
Use canned phrases  
l  
Co-browse (Web collaboration)  
l  
Co-browse example  
l  
Wrap up a call  
CCSP UI Help - Supervisor - 7.4  
40

Become available or released  
Note  
If you have Teams enabled:  
l  
Your CCSP UI agent status and Teams presence are synchronized. When you log on to CCSP UI, it  
synchronizes with your Teams presence. If your Teams status is Available, CCSP UI automatically sets  
your status to Available.  
l  
If you have Teams enabled and you select Auto Available, Teams automatically becomes available.  
l  
If you select a CCSP UI release code that does not have a corresponding presence mapping in Teams, or  
the other way around, then CCSP UI shows Released and Teams shows Away.  
l  
Call-initiated status changes only apply to voice calls.  
To see for how long you have been in the current state, point to Available  
or Released  
.  
Become available  
Do one of the following:  
l  
To automatically become available when you log on, select Auto Available in preferences or the status  
menu.  
l  
To manually become available, in the gadgetory, click  
. Your status changes to Available  
.  
l  
Press Win+F3.  
l  
In the notification area, right-click the CCSP UI icon and then click Go Available.  
CCSP UI Help - Supervisor - 7.4  
41

Become released  
Note  
l  
When you become released, CCSP UI does not deliver queued interactions.  
l  
During an outbound dialing campaign, when you try to become unavailable, your state changes to  
Pending Unavailable  
. You can continue working in your company's Customer Relationship  
Management (CRM) program or other applications connected to CCSP UI.  
If you are not in-call, the system will try to route new calls to other agents who are available. If other  
agents are not available, a call may arrive during this time. If you are in-call or you receive another call  
after trying to become unavailable, you could remain in the Pending Unavailable state for the whole  
duration of the call. Your state could change to Unavailable only after you finish the wrap up.  
Do one of the following:  
l  
In the gadgetory, click  
and then click the release code. Your status changes to Released  
.  
l  
Press Win+F3.  
l  
In the notification area, right-click the CCSP UI icon and then click Go Released.  
CCSP UI Help - Supervisor - 7.4  
42

Answer a call  
To automatically answer calls, select Answer calls automatically in preferences or the status menu.  
To manually answer a call, click the green button on the Ringing call card or press Win+F6.  
Call type  
Click...  
Next...  
Voice  
See Process a voice call.  
Chat  
See Process a chat call.  
CCSP UI Help - Supervisor - 7.4  
43

Call type  
Click...  
Next...  
Email  
See Process an email call.  
Callback  
See Process a callback call.  
Voicemail  
See Process a voicemail call.  
Refuse a call  
When you refuse a call, you may automatically become released if this has been configured by your system  
administrator. Depending on this configuration, you may:  
l  
Remain released until you either make yourself available or are automatically made available after a set  
time.  
l  
See a warning that you have been made released, along with the number of seconds before you are  
automatically made available again.  
To refuse a call, click the red button in the Ringing call card or press Win+F12.  
For examples of Ringing call card types, see Answer a call.  
CCSP UI Help - Supervisor - 7.4  
44

Defer an email or a voicemail call  
Note  
Supervisors can see personal and shared deferred calls under the deferred folder.  
1.  
Answer the call. See Answer a call.  
2.  
On the call card, click  
and then click  
. The Defer Call dialog appears.  
3.  
Do one of the following:  
o  
If you do not want other agents to see the deferred call under the deferred folder, click Personal.  
o  
If you want other agents in the same group to see the deferred call, under the deferred folder, click  
Shared.  
4.  
Click Save.  
For information about handling deferred calls, see Pull messages.  
CCSP UI Help - Supervisor - 7.4  
45

Process a voice call  
After you answer a voice call, the call card changes from Ringing to Active and the call duration timer starts.  
If CCSP is automatically recording the call, a red dot appears.  
To process a voice call, on the call card in the Calls Menu, do one or more of the following.  
To...  
Click...  
Next...  
Hold the call  
See Hold a call.  
Transfer or conference the call  
See Transfer or  
conference a call.  
Record the call, if CCSP is not  
automatically recording it  
and then  
See Record a call.  
Wrap up the call before hanging  
up  
and then  
See Wrap up a call.  
Schedule a callback  
and then  
See Schedule a callback.  
Open the dial pad  
and then  
See Make an outgoing  
call.  
Adjust the volume, hover over...  
See Adjust the volume.  
Mute the call  
See Mute a call.  
View call information  
See View call information.  
Hang up the call  
See Wrap up a call.  
CCSP UI Help - Supervisor - 7.4  
46

Process a chat call  
Note  
l  
The chat box turns orange in one corner when there are pending changes or after the maximum  
inactivity timeout.  
l  
When you close the chat box, the call is not hung up. To reopen the chat box, click the chat call in the  
Calls Menu.  
l  
You cannot conference a chat call.  
After you answer a chat call, the call card changes from Ringing to Active, the call duration timer starts, and the  
chat dialog appears. A pencil indicates that the caller is typing a message.  
To process a chat call, on the call card in the Calls Menu or in the chat dialog, do one or more of the following.  
CCSP UI Help - Supervisor - 7.4  
47

To...  
Click...  
Next...  
Hold the call  
See Hold a call.  
Transfer the call  
See Transfer or conference a call.  
Record the call, if CCSP is not  
automatically recording it  
and then  
See Record a call.  
Wrap up the call before hanging up  
and then  
See Wrap up a call.  
Schedule a callback  
and then  
See Schedule a callback.  
Co-browse web pages with the caller  
and then  
See Co-browse (Web collaboration).  
View call information  
See View call information.  
Resize a floating chat box  
Use canned phrases  
See Use canned phrases.  
Reopen a closed chat box  
The chat call in the Calls Menu.  
Hang up the call  
See Wrap up a call.  
CCSP UI Help - Supervisor - 7.4  
48

Process an email call  
After you answer an email call, the call card changes from Ringing to Active, the call duration timer starts, and  
the Email tab appears.  
To process an email call, on the call card in the Calls Menu do one of the following.  
To...  
Click...  
Next...  
Transfer the call  
See Transfer or conference  
a call.  
Record the call, if CCSP is not  
automatically recording it  
and then  
See Record a call.  
Wrap up the call before hanging  
up  
and then  
See Wrap up a call.  
Schedule a callback  
and then  
See Schedule a callback.  
Defer the call  
and then  
See Defer an email or a  
voicemail call.  
View call information  
See View call information.  
Hang up the call  
See Wrap up a call.  
CCSP UI Help - Supervisor - 7.4  
49

Note  
l  
Separate email addresses with a semi-colon (;). CCSP UI does not support email addresses that contain  
double quotation marks (") or are separated by a comma (,).  
l  
Background text color may not work as expected.  
l  
If you insert two or more images using drag and drop, CCSP UI only inserts the last image.  
l  
In Firefox and Edge, CCSP UI inserts canned phrases at the bottom of the email.  
l  
You can receive a digitally signed inbound email. However, you cannot include a digital signature in an  
outbound email (reply, forward, or new).  
To reply to or forward the email:  
1.  
On the Email tab, at the bottom, click Reply, Reply All or Forward.  
2.  
Write the email. CCSP UI auto-saves the draft every 30 seconds.  
3.  
Optionally, to include canned phrases, click  
Canned Phrases. See Use canned phrases.  
4.  
Click Send.  
5.  
On the call card, click  
.  
6.  
Wrap up the email call. See Wrap up a call.  
CCSP UI Help - Supervisor - 7.4  
50

Process a callback call  
To process a callback call, on the call card in the Calls Menu or in the Outgoing callback call dialog, do one or  
more of the following.  
To...  
Click...  
Next...  
Make the call  
Reschedule an unsuccessful callback  
and then  
See Schedule a callback.  
View call information  
See View call information.  
Hang up the call  
See Wrap up a call.  
After you answer a callback call, another instance of the Outgoing call dialog appears, the call card closes  
automatically, and your CCSP UI agent status changes to Released.  
1.  
On the Outgoing call dialog, click  
.  
2.  
When you hang up, the Wrap Up dialog appears. See Wrap up a call.  
CCSP UI Help - Supervisor - 7.4  
51

Process a voicemail call  
Note  
For Ubuntu client, to play voicemail, you need to install a codec for MP4. Ask your administrator for the  
installation instructions.  
After you answer a voicemail call, the call card changes from Ringing to Active, the call duration timer starts,  
and the message plays.  
To process a voicemail call, on the call card in the Calls Menu do one or more of the following.  
To...  
Click...  
Next...  
Hold the call  
See Hold a call.  
Stop playing the message  
Transfer or conference the call  
See Transfer or conference a  
call.  
Wrap up the call before hanging  
up  
and then  
See Wrap up a call.  
Schedule a callback  
and then  
See Schedule a callback.  
Defer the call  
and then  
See Defer an email or a  
voicemail call.  
Adjust the volume, hover over  
See Adjust the volume.  
View call information  
See View call information.  
Hang up the call  
See Wrap up a call.  
CCSP UI Help - Supervisor - 7.4  
52

View call information  
Depending on your CCSP UI configuration, call information may always open when a new call arrives.  
To view call information, on the call card in the Calls Menu, click  
. The information appears underneath the  
call card.  
CCSP UI Help - Supervisor - 7.4  
53

Control media recording  
Note  
l  
You cannot start media recording until the call is connected.  
l  
If you click record in the middle of a voice call, CCSP records only the part of the voice call after the click.  
l  
If you click hold in the middle of a voice call and only you and the caller are present on the call, when you  
take the call off hold, a separate media recording starts and a new media file is created in the database.  
l  
For email recordings, CCSP captures the recording at the call termination point, so you cannot partially  
record email.  
l  
For chat calls, you can simultaneously record the text and screens of each chat.  
CCSP may automatically record all your CCSP UI calls, depending on how your system administrator configured  
recording.  
When CCSP is recording a call's media, a red dot appears on the call card.  
To manually record a call's media and its screens, after answering the call, on the call card, click  
and then  
click  
.  
To manually record only a call's screens, see Control screen recording.  
CCSP UI Help - Supervisor - 7.4  
54

Control screen recording  
Note  
l  
You can start screen recording by clicking the Screen Recording gadget while the call is ringing.  
However, you cannot start screen recording when there are no calls.  
l  
Clicking the Screen Recording gadget starts screen recording but not media recording.  
l  
The maximum size for screen recording is 2048 x 1536. If you use multiple screens with the Extend  
option, the screen recording includes all the screens.  
l  
In Remote Desktop Services environments, screen recording does not work as expected.  
l  
If there is a problem with the screen recording service, the following warning appears: Screen Recording  
has been selected, but is not available. Ensure that CCSP Screen Recording service is properly installed.  
This warning indicates that the background client services required for screen recording were  
unexpectedly terminated on your computer and the screen recording process is no longer active for you.  
To restart the screen recording process, log out of CCSP UI and then log back in to CCSP UI.  
When there are multiple calls, CCSP automatically records screen only if all the calls should be recorded. Screen  
recording stops if a call starts that should not be recorded or if you stop screen recording to avoid recording  
sensitive data such as credit cards.  
Screen recording also stops or starts if another agent participating in a call with you stops or starts the  
recording.  
When CCSP is recording a call's screens, the Screen Recording gadget is red.  
To manually stop recording screens, on the gadgetory, click  
Screen Recording.  
To manually start recording screens, on the gadgetory, click  
Screen Recording.  
CCSP UI Help - Supervisor - 7.4  
55

Hold a call  
To hold a call, click  
on the call card or press Win+F2.  
To take a call off hold, click  
 again on the call card or press Win+F2 again.  
CCSP UI Help - Supervisor - 7.4  
56

Make an outgoing call  
Note  
l  
When you make a call to a system endpoint, Call to the System appears on the call card.  
l  
Received campaign calls appear in the Incoming calls list of the  
tab as either Predictive Call or  
Preview Call. You cannot dial a campaign call from the incoming calls list.  
1.  
On the gadgetory, click  
or press Ctrl+Backspace. The Outgoing Call dialog appears.  
If you have Teams enabled, the Outgoing call dialog has a  
tab.  
CCSP UI Help - Supervisor - 7.4  
57

2.  
If enabled, on the  
tab, you can click Caller Number, and then click the number in the list.  
3.  
Alternatively, do one of the following.  
To select  
Click  
Then  
The search box  
Start typing the number or name.  
A recent outgoing  
or incoming caller  
To filter the list by Outgoing or Incoming callers, click the corresponding  
word. Click the user.  
A logged in user  
To filter the list by state or group, click the corresponding down arrow  
and select the required state or group from the list. Click the user.  
CCSP UI Help - Supervisor - 7.4  
58

To select  
Click  
Then  
A system endpoint  
Click the endpoint.  
A speed dial  
Click the speed dial.  
The dialpad  
Click the numbers.  
A Teams contact, if  
enabled  
To filter the list by status or group, click the corresponding down arrow  
and select the required status or group from the list. Click the user.  
4.  
Click  
.  
CCSP UI Help - Supervisor - 7.4  
59

Transfer or conference a call  
Note  
l  
If cold (blind) transfer is enabled, to transfer a call to an external number via speed dial, the dialpad, or  
the recent caller tab without waiting for the line to connect, in the Xfer/Conf dialog, click  
.  
l  
Only use cold transfer if you are certain the transferee will answer the call. If the transferee rejects or  
does not answer the call, the call is cut and the original caller needs to call the contact center again.  
l  
You cannot use cold transfer to another agent/user, system endpoint, or for email.  
1.  
Click  
on the call card or press Win+F7. The Xfer/Conf dialog appears.  
CCSP UI Help - Supervisor - 7.4  
60

2.  
Do one of the following.  
To select...  
Click...  
Next...  
The search box  
Start typing the number or name.  
A recent  
outgoing or  
incoming caller  
To filter the list by Outgoing or Incoming callers, click the corresponding  
word. Click the user.  
A logged in  
user  
To filter the list by state or group, click the corresponding down arrow and  
select the required state or group from the list. Request a chat consultation  
(if enabled) or click the user.  
A system  
endpoint  
To transfer the call back to script control at a specific named place in the  
system, click the endpoint.  
A speed dial  
Click the speed dial.  
The dialpad  
Click the numbers.  
A Teams  
contact, if  
enabled  
To filter the list by status or group, click the corresponding down arrow and  
select the required status or group from the list. Click the user.  
3.  
Click  
.  
4.  
When the call is connected, do one of the following:  
o  
To transfer the call, click Transfer. Your end of the call will be hung up.  
o  
To conference the call, click Conference. See Hang up from a conference call below.  
o  
For voice calls, to switch between the other user and the caller, click Toggle. When you are connected to  
the other user, the caller is on hold and Transfer, Conference, Toggle, and Cancel are available. When  
you are connected to the caller, only Toggle and Cancel are available.  
CCSP UI Help - Supervisor - 7.4  
61

Hang up from a conference call  
Do one of the following:  
l  
To hang up the last participant that joined the conference, in the call card, click the hang up conference  
button next to the participant's name. This icon is only visible if you initiated the conference.  
l  
To leave the conference, click  
or press Win+F12.  
CCSP UI Help - Supervisor - 7.4  
62

Request a chat consultation  
If enabled, during an incoming call or an outbound voice call, you can request a chat consultation using a  
button next to an available agent via the Xfer/Conf dialog.  
Note  
l  
The incoming call must be a voice, chat, or email call. You cannot request a chat consultation during a  
conference call, callback, or voicemail call.  
l  
The other agent must be Available or Available in call.  
l  
You and the other agent can only have one chat consultation at a time. If the other agent is already in a  
chat consultation or still has the Consultation dialog open, the request is automatically rejected.  
l  
An unanswered request times out after 2 minutes.  
1.  
In the Xfer/Conf dialog, on the  
Users tab, point to an Available or Available in Call agent and click  
in the same row.  
2.  
The Consultation dialog appears:  
o  
If the other agent is already in a chat consultation or still has the Consultation dialog open, the request  
is automatically rejected and you receive the User cannot answer right now notification.  
o  
Otherwise, the other agent can accept or decline the request.  
CCSP UI Help - Supervisor - 7.4  
63

Your view  
Other agent's view  
3.  
If the other agent clicks Accept, you can exchange messages until either of you clicks End session or  
closes the Consultation dialog, or the initial call is hung up.  
CCSP UI Help - Supervisor - 7.4  
64

4.  
If the other agent clicks Decline, you receive the User cannot answer right now notification and can only  
close the dialog.  
5.  
After the session ends, you receive a termination notification. The Consultation dialog remains open until  
you close it.  
If you missed any consultation requests while you were in the chat consultation, you receive the You missed  
one or more requests for consultation. Click here to view notification at the bottom of the dialog. Click the  
notification to view the list of attempts.  
CCSP UI Help - Supervisor - 7.4  
65

Adjust the volume  
Note  
For SIP, you can also test and adjust the volume in Preferences. See Set your preferences.  
1.  
On the call card, hover over  
. The headset volume control appears. Adjust the volume.  
2.  
On the call card, hover over  
. The microphone volume control appears. Adjust the volume.  
Mute a call  
To mute a call, on the Active call card, click  
.  
To unmute the call, on the Active call card, click  
.  
CCSP UI Help - Supervisor - 7.4  
66

Schedule a callback  
1.  
On the gadgetory, on a call card in the Call Menu, or in the Wrap Up dialog, click  
. The Callback dialog  
appears.  
2.  
In Name, type the name of the calling party.  
3.  
If the number does not already appear in Number, type the telephone number.  
4.  
Under Date, click the appropriate button or in Specific, select the date from the calendar.  
5.  
Under Time, click the appropriate button or in Specific, select the hour and minute.  
6.  
To personally receive the callback, select the Call Me check box.  
CCSP UI Help - Supervisor - 7.4  
67

7.  
If immediate dialing is not enabled by default, to automatically initiate dialing of the callback number,  
select the Immediate Dialing check box.  
8.  
If necessary, scroll down to view the Message and URL text boxes.  
9.  
In Message, type the message to display when the callback arrives.  
10.  
In URL, type the caller's URL.  
11.  
Click Save.  
CCSP UI Help - Supervisor - 7.4  
68

Use canned phrases  
Note  
In Firefox and Edge, CCSP UI inserts canned phrases at the bottom of the email.  
1.  
On the right of the CCSP UI page, click  
Canned Phrases. The Canned Phrases dialog appears.  
2.  
In the Search box, start typing the word to search for. CCSP UI looks for this word in the phrase titles and  
text.  
3.  
Click the phrase or expand a category and click the required phrase.  
4.  
Click Use.  
CCSP UI Help - Supervisor - 7.4  
69

Co-browse (Web collaboration)  
If enabled, you can start a co-browsing session with a caller when you are in a chat call. The same web pages  
then automatically appear on both the caller's and the agent's screens. Both people in the call can navigate  
web pages, complete online forms, and click links to other pages. See Co-browse example.  
Note  
l  
The features available to you in the co-browsing session depend on the co-browsing product configured  
by the system administrator.  
l  
Whether you can open a co-browsing session for each chat call or only a specific number of co-browsing  
instances, is configured by the system administrator.  
l  
For Edge, you cannot pop out the Co-Browsing tab.  
1.  
In the chat dialog, click  
and then click  
. The Co-browsing tab appears. If configured to do so, the  
current chat automatically docks in the tab.  
2.  
To manually dock and undock the current chat, click  
.  
3.  
Navigate the web pages and change presenter, as required.  
4.  
To end the co-browsing session, click Exit session.  
CCSP UI Help - Supervisor - 7.4  
70

Co-browse example  
The co-browsing product that is provided with CCSP UI is togetherJS. This tool enables you and the caller to  
browse the same website and see the same pages and content. You are also both able to edit the same forms  
on a page.  
Note  
TogetherJS co-browsing can only occur when both parties are on a togetherJS-enabled website. If either  
you or the caller navigates to a page where this tool is not enabled, the other party can’t follow.  
1.  
Depending on the configuration, you are asked by the caller to either start a new session or to join an  
existing session that the co-browser started.  
2.  
In the chat dialog, click  
and then click  
. The Co-browsing tab appears. If configured to do so, the  
current chat automatically docks in the tab.  
3.  
To start the togetherJS session, click Start session.  
4.  
When the website appears in the Co-browsing tab, the togetherJS toolbar appears. A togetherJS welcome  
dialog also appears, which you can close.  
CCSP UI Help - Supervisor - 7.4  
71

5.  
If you started a new togetherJS session, wait for the caller to join the session. The togetherJS UI displays a  
message when the caller has joined.  
CCSP UI Help - Supervisor - 7.4  
72

6.  
A pointer appears in your window, indicating the position of the caller’s mouse pointer on the page. The  
caller sees a corresponding pointer for your cursor in their window.  
7.  
When either of you click on the window, an expanding circle is drawn around the position clicked.  
CCSP UI Help - Supervisor - 7.4  
73

8.  
When the caller navigates to another page on the website, a message appears stating the URL of the page  
that the caller has gone to and asking if you want to join them. To go to the same page, click Join them.  
CCSP UI Help - Supervisor - 7.4  
74

Note  
This message is also displayed when you and the caller are both on the same page and you then  
navigate to a different page. In this case, ignore the message and wait for the caller to follow you.  
9.  
When the caller is typing, an ellipsis (…) appears next to the caller’s name to indicate this.  
10.  
To automatically follow the caller, click the caller’s icon in the togetherJS toolbar and select the Follow this  
participant check box in the popup window. Close the window. The session is now controlled by the caller  
and your window follows the caller across any page changes.  
CCSP UI Help - Supervisor - 7.4  
75

11.  
To stop following the caller, click the caller’s icon in the togetherJS toolbar and clear the Follow this  
participant check box.  
12.  
To end the co-browsing, do one of the following:  
o  
Click Exit session.  
o  
Click your icon in the togetherJS toolbar and then click End Co-Browsing.  
Note  
You and the caller must both actively end co-browsing.  
CCSP UI Help - Supervisor - 7.4  
76

Wrap up a call  
You can wrap up a call after hanging up or while in conversation.  
When you hang up a call, the Wrap Up dialog automatically appears. Continue at step 2.  
To wrap up the call while in conversation:  
1.  
On the call card in the Calls Menu or in the chat dialog, click  
and then click  
. The Wrap Up dialog  
appears.  
2.  
Click the wrap up code.  
3.  
Optionally, click Add Note and type a comment.  
4.  
Optionally, click Call Back and schedule a callback. See Schedule a callback.  
5.  
Click Submit or Save.  
CCSP UI Help - Supervisor - 7.4  
77

4: Messages  
Note  
Only supervisors can restore deleted messages.  
This section contains the following topics:  
l  
View messages  
l  
Messaging items visibility by role  
l  
Filter messages  
l  
Pull messages  
l  
Remove messages  
l  
Restore messages  
l  
Access archived messages  
CCSP UI Help - Supervisor - 7.4  
78

View messages  
Regular agent view  
Supervisor view  
View subfolders  
Page navigation  
See also Messaging items visibility by role.  
Note  
You can access Deferred Emails and Deferred Voicemails from the Folders page and from inside the  
Folders > Email and Folders > Voicemail pages respectively.  
1.  
On the gadgetory, click  
Message View. The Message View tab appears.  
2.  
Click the folder to view, for example, Email. The Folders > Email view appears.  
CCSP UI Help - Supervisor - 7.4  
79

Regular agent view  
Generally, regular agents can view only the subfolders containing their items.  
The subfolders are:  
Email:  
l  
<Public folders> — contain email items awaiting handling. Regular agents can view only items that were  
sent to their group by themselves or by other group members.  
l  
Deferred Emails — contains email items deferred by this agent or by a group member. Deferred items can  
be resumed either by agents pulling them or by the system using the ageing policy.  
l  
Replied — contains incoming ACD email items that were handled and replied to by this agent or by a group  
member. The reply message must be sent for the original item to be considered replied to.  
l  
Not Replied — contains incoming ACD email items that were handled but not replied to by this agent or by  
a group member. This folder contains items that were archived after the call-cycle ended or the email call  
ended without an agent replying to it.  
l  
Sent Items — contains sent email items that were sent by this agent: reply, forward, and new.  
l  
Outbox — contains outbound email items that were created by this agent but have not yet been sent.  
l  
Rejected Emails — contains email items that were sent to this agent's group and rejected by the system  
due to invalid item or call characteristics.  
l  
Drafts — contains email items that were composed by this agent and disconnected, accidentally or  
deliberately, by clicking Discard & Keep draft in the Email Pending Hangup dialog.  
CCSP UI Help - Supervisor - 7.4  
80

The draft of the discarded item in the Drafts folder retains all the relevant fields (From, To, Body) and can  
be pulled, edited, and sent at a later time. See Pulling messages.  
Callback:  
l  
Callbacks — contains callback items waiting for their scheduling time marked with 'call-to-me' by this  
agent, or created by other group members (and not marked with 'call-to-me').  
l  
Rejected Callbacks — contains callback items that were sent to this agent's group and rejected by the  
system due to invalid item or call characteristics.  
Voicemail:  
l  
Voicemail — contains voicemail items waiting for their scheduling time, that were sent to this agent's  
group.  
l  
Rejected Voicemails — contains voicemail items that were sent to this agent's group and rejected by the  
system due to invalid item or call characteristics.  
l  
Archived Voicemails — contains handled voicemail items that were sent to this agent's group.  
l  
Deferred Voicemails — contains voicemail items deferred by this agent or by a group member.  
Supervisor view  
The Supervisor view contains the same folders as the regular agent. In addition, Supervisors see:  
CCSP UI Help - Supervisor - 7.4  
81

l  
Folders for Deleted Emails, Deleted Callbacks, and Deleted Voicemail that contain items that were  
deleted by agents, and the Deleted Time and Deleted Reason columns.  
l  
In the Rejected Emails, Rejected Callbacks, and Rejected Voicemails folders — the Rejected Time and  
Rejected Reason columns.  
There are 3 types of supervisors:  
l  
Regular supervisor (has a customized permission profile that contains specific groups and teams) — has  
access to all the items that were sent to or from the specific groups and team members in their permission  
profile.  
l  
Team supervisor (Team Supervisor default permission profile) — has access to all items that were sent to or  
from the members of their teams.  
l  
Tenant-wide supervisor (Supervisor or Administrator and Supervisor default permission profiles) — has  
access to all items in all groups and teams.  
View subfolders  
1.  
Click the subfolder to view, for example, Deferred Emails. The Folders > Email > Deferred Emails view  
appears. See Filter messages and Page navigation.  
Agent view  
Supervisor view  
2.  
To read the complete email, click the email's row. To hear a voicemail or callback, you need to pull it. See  
Pull messages.  
CCSP UI Help - Supervisor - 7.4  
82

Agent view  
Supervisor view  
3.  
To read the previous or next email in the list, click the up or down arrow at the top of the email.  
Page navigation  
The total number of items and the current set of items within that number are given in the address bar.  
CCSP UI Help - Supervisor - 7.4  
83

If filtering criteria are currently being applied to the folder, the total number of items is the number of items  
that meet that criteria.  
To go to the:  
l  
Next page in the list, click  
.  
l  
Previous page in the list, click  
.  
l  
Last page in the list, click  
.  
l  
First page in the list, click  
.  
CCSP UI Help - Supervisor - 7.4  
84

Messaging items visibility by role  
Note  
l  
If the team or group is not specified on items in Archive, Voicemail or Callback folders, these items are visible only by tenant-wide supervisors.  
l  
If the team or group is not specified on items in Outgoing and Sent Item folders, these items are visible only by the composer and tenant-wide  
supervisors.  
l  
If the allowed teams or allowed groups is not specified for a regular supervisor or team supervisor, they cannot view Archive folders.  
Callbacks  
Folder  
role  
Folder  
group  
Agent  
-includes assigned group  
Regular supervisor  
- includes assigned group  
- also includes allowed groups and allowed  
teams  
Team supervisor  
- includes assigned group  
- also includes allowed teams  
Tenant-wide  
supervisor  
- all groups  
and teams  
Callback  
ACD  
Can see:  
l  
Items marked with "call-to-me" for  
this agent  
l  
Items tagged with agent's group that  
aren't marked with "call-to-me" for  
another agent  
Can see:  
l  
Items marked with "call-to-me" for this  
supervisor  
l  
Items marked with "call-to-me" for agents  
belonging to this supervisor's allowed  
groups or teams  
l  
Items tagged with group that exist in  
allowed groups list  
l  
Items tagged with team that exist in  
allowed teams list  
Can see:  
l  
Items marked with "call-to-me" for  
this supervisor  
l  
Items marked with "call-to-me" for  
agents belonging to supervisor's  
allowed teams  
l  
Items tagged with team that exist  
in allowed teams list  
Can see all  
items  
Rejected  
Callback  
Archive  
Deleted  
Callback  
Archive  
Folder isn't visible to agent  
CCSP UI Help - Supervisor - 7.4  
85

Email  
Folder role  
Folder  
group  
Agent  
-includes assigned group  
Regular supervisor  
- includes assigned group  
- also includes allowed groups  
and allowed teams  
Team supervisor  
- includes assigned group  
- also includes allowed teams  
Tenant-wide  
supervisor  
- all groups and teams  
Public  
ACD  
Can see all items in visible folders  
Can see all items in visible folders  
Can see all items in visible folders  
Can see all folders and  
items  
Replied  
Archive  
Can see items tagged with agent's  
group  
Can see:  
l  
Items tagged with group that  
exist in allowed groups list  
l  
Items tagged with team that  
exist in allowed teams list  
Can see items tagged with team  
that exist in allowed teams list  
Can see all items  
Not-Replied  
Archive  
Deferred  
Emails  
Deferred  
Can see:  
l  
Personal deferred items  
(deferred by this agent)  
l  
Shared deferred items (deferred  
by this agent's group members)  
Can see:  
l  
Personal deferred items  
(deferred by this supervisor)  
l  
Personal deferred items  
(deferred by any of allowed  
group members)  
l  
Personal deferred items  
(deferred by any of allowed  
team members)  
l  
Shared deferred items  
(deferred by this supervisor's  
group members)  
l  
Shared deferred items  
Can see:  
l  
Personal deferred items  
(deferred by this supervisor)  
l  
Personal deferred items  
(deferred by any of allowed  
team members)  
l  
Shared deferred items  
(deferred by this supervisor's  
group members)  
l  
Shared deferred items  
(deferred by any of allowed  
team members)  
Can see all deferred  
items  
CCSP UI Help - Supervisor - 7.4  
86

Folder role  
Folder  
group  
Agent  
-includes assigned group  
Regular supervisor  
- includes assigned group  
- also includes allowed groups  
and allowed teams  
Team supervisor  
- includes assigned group  
- also includes allowed teams  
Tenant-wide  
supervisor  
- all groups and teams  
(deferred by any of allowed  
group members)  
l  
Shared deferred items  
(deferred by any of allowed  
team members)  
Rejected  
Emails  
Archive  
Can see items tagged with agent's  
group  
Can see:  
l  
Items tagged with group that  
exist in allowed groups list  
l  
Items tagged with team that  
exist in allowed teams list  
Can see items tagged with team  
that exist in allowed teams list  
Can see all items  
Deleted  
Emails  
Archive  
Folder isn't visible to agent  
Can see:  
l  
All items deleted by this  
supervisor  
l  
All items deleted by users in  
allowed groups list  
l  
All items deleted by users in  
allowed teams list  
Can see:  
l  
All items deleted by this  
supervisor  
l  
All items deleted by users in  
allowed teams list  
Can see all items  
CCSP UI Help - Supervisor - 7.4  
87

Folder role  
Folder  
group  
Agent  
-includes assigned group  
Regular supervisor  
- includes assigned group  
- also includes allowed groups  
and allowed teams  
Team supervisor  
- includes assigned group  
- also includes allowed teams  
Tenant-wide  
supervisor  
- all groups and teams  
Sent Items  
Archive  
Can see items sent by this agent  
Can see:  
l  
All items sent by this supervisor  
l  
All items sent by users in  
allowed groups list  
l  
All items sent by users in  
allowed teams list  
Can see:  
l  
All items sent by this supervisor  
l  
All items sent by users in  
allowed teams list  
Can see all items  
Outbox  
Outgoing  
Drafts  
Outgoing  
Can see:  
l  
Disconnected draft1 items  
composed by this agent  
l  
Free draft2 items composed by  
this agent  
Can see:  
l  
Disconnected draft1 items  
composed by this supervisor  
l  
Free draft2 items composed by  
this supervisor  
l  
Disconnected draft items  
composed by users in allowed  
groups list  
l  
Free draft items composed by  
users in allowed groups list  
l  
Disconnected draft items  
composed by users in allowed  
teams list  
l  
Free draft items composed by  
users in allowed teams list  
Can see:  
l  
Disconnected draft1 items  
composed by this supervisor  
l  
Free draft2 items composed by  
this supervisor  
l  
Disconnected draft items  
composed by users in allowed  
teams list  
l  
Free draft items composed by  
users in allowed teams list  
Can see:  
l  
All disconnected  
draft1 items  
l  
All free draft2 items  
1.  
Disconnected draft — the item is not paired with an active call or call-part.  
2.  
Free draft — the item has no active call or call-part held by the agent or supervisor.  
CCSP UI Help - Supervisor - 7.4  
88

Voicemail  
Folder role  
Folder  
group  
Agent  
-includes assigned group  
Regular supervisor  
- includes assigned group  
- also includes allowed groups  
and allowed teams  
Team supervisor  
- includes assigned group  
- also includes allowed teams  
Tenant-wide  
supervisor  
- all groups and teams  
Voicemail  
ACD  
Can see items tagged with agent's  
group  
Can see:  
l  
Items tagged with group that  
exist in allowed groups list  
l  
Items tagged with team that  
exist in allowed teams list  
Can see items tagged with team  
that exist in allowed teams list  
Can see all items  
Rejected  
Voicemail  
Archive  
Archived  
Voicemail  
Archive  
Deferred  
Voicemails  
Deferred  
Can see:  
l  
Personal deferred items  
(deferred by this agent)  
l  
Shared deferred items (deferred  
by this agent's group members)  
Can see:  
l  
Personal deferred items  
(deferred by this supervisor)  
l  
Personal deferred items  
(deferred by any of allowed  
group members)  
l  
Personal deferred items  
(deferred by any of allowed  
team members)  
l  
Shared deferred items  
(deferred by this supervisor's  
group members)  
l  
Shared deferred items  
(deferred by any of allowed  
group members)  
Can see:  
l  
Personal deferred items  
(deferred by this supervisor)  
l  
Personal deferred items  
(deferred by any of allowed  
team members)  
l  
Shared deferred items  
(deferred by this supervisor's  
group members)  
l  
Shared deferred items  
(deferred by any of allowed  
team members)  
Can see all deferred  
items  
CCSP UI Help - Supervisor - 7.4  
89

Folder role  
Folder  
group  
Agent  
-includes assigned group  
Regular supervisor  
- includes assigned group  
- also includes allowed groups  
and allowed teams  
Team supervisor  
- includes assigned group  
- also includes allowed teams  
Tenant-wide  
supervisor  
- all groups and teams  
l  
Shared deferred items  
(deferred by any of allowed  
team members)  
Deleted  
Voicemails  
Archive  
Folder isn't visible to agent  
Can see:  
l  
All items deleted by this  
supervisor  
l  
Items deleted by users in  
allowed groups list  
l  
Items deleted by users in  
allowed teams list  
Can see:  
l  
All items deleted by this  
supervisor  
l  
Items deleted by users in  
allowed teams list  
Can see all items  
CCSP UI Help - Supervisor - 7.4  
90

Filter messages  
Note  
l  
The view is refreshed only when you navigate to another page or click  
Refresh.  
l  
For most items, the time is the arrival time. For deferred items, the time is the arrival and the deferred  
time. For deleted items, the time is the arrival/sent time depending on the item and the deleted time.  
Groups and teams filters  
Defer type filter  
Filter by group, team, and defer type  
Filter by text contained in the items  
Filter by date and time  
Sort messages  
Reset folder to default filter and sort settings  
Groups and teams filters  
Groups and teams filters are available to all supervisors in the non-public email folders and all other folders in  
callback and voicemail. Emails in the public folders are already filtered by group. Users can filter by the groups  
and teams that are relevant for their type, for example, a tenant-wide supervisor can filter by all groups and  
teams.  
Email filters for public folders (agent and supervisor view)  
Email filters for non-public folders (supervisor view)  
CCSP UI Help - Supervisor - 7.4  
91

Defer type filter  
Items in the Deferred Emails and Deferred Voicemails folders can be filtered by the defer type. This filter is  
available to both agents and supervisors.  
The filter bar below the address bar presents the filtering and sorting options available for the selected folder.  
The options presented depend on the user, media, and the folder itself.  
Deferred Emails filter (agent view)  
Deferred Emails filters (supervisor view)  
Filter by group, team, and defer type  
You can combine all 3 filters to show only those items that meet all the selected criteria.  
1.  
Click the folder and subfolders until you reach the list page.  
2.  
Do one or more of the following:  
o  
To filter on a specific group, select the required group from the drop down list. To see items from all  
groups, select All Groups.  
CCSP UI Help - Supervisor - 7.4  
92

o  
To filter on a specific team, select the required team from the drop down list.  
To see items from all teams, select All Teams.  
o  
To filter on a specific defer type, select either Personal or Shared from the drop down list.  
To see all items, select All Deferred.  
Filter by text contained in the items  
Note  
On the Message View tab, searching for punctuation marks may not return the expected result.  
In the search box, start typing the word to search for. You can also enter a list of words, and a whole phrase,  
using quotation marks (""). CCSP UI searches different fields for each media as follows:  
CCSP UI Help - Supervisor - 7.4  
93

l  
Email — From, To, Subject, and the body of the message.  
l  
Callbacks — Callback Number and Message  
l  
Voicemail — Call To (DNIS) and Call From (ANI)  
Filter by date and time  
Note  
For most items, the time used for the filter is the arrival time of the item. For deferred items, the time is the  
arrival and the deferred time. For deleted items, the time is the arrival/sent time depending on the item  
and the deleted time.  
1.  
Click  
and select the Start Date and End Date.  
2.  
After selecting the date, you can change the specific time range to search in.  
Sort messages  
Click the Sort By down arrow and then click the field to sort by. The available fields to sort on are dependent on  
the media and the folder. In addition, the folders containing deleted items also offer the option to sort by  
Deleted time.  
Incoming Email items  
Deferred Emails  
CCSP UI Help - Supervisor - 7.4  
94

Sent Emails  
Callbacks  
Voicemail  
Deferred Voicemail  
To change the sorting direction, click again on the selected field in the Sort By menu.  
CCSP UI Help - Supervisor - 7.4  
95

Reset folder to default filter and sort settings  
Note  
Any filters and sort options that you select within a folder are remembered by that folder for the next time  
you visit it, until you log out of your CCSP UI session.  
Click  
.  
This clears the date filter and the search box, and resets the:  
l  
Displayed page to the first item  
l  
Selected Group to All Groups (supervisor only)  
l  
Selected Team to All Teams (supervisors only)  
l  
Selected Defer Type to All Deferred (Deferred folders only)  
l  
Sort option to the default for the folder  
CCSP UI Help - Supervisor - 7.4  
96

Pull messages  
Note  
When pulling a partial call-part item from the Drafts folder:  
l  
The call-parts fields contain the original data saved at the point the call part item moved to the Drafts  
folder.  
l  
If the parent item of the call-part can be located, it is opened as part of the new incoming ACD email call,  
together with all the call-parts belonging to the parent item.  
l  
If the parent item of the call-part cannot be located, only the pulled call-part item is opened as a new  
outgoing email call.  
l  
Bulk pulling is not currently supported.  
Messages can be pulled from incoming folders (Public, Deferred) and outgoing folders (Sent Items, Outbox and  
Drafts). When pulling from outgoing folders, you can edit and (re)send the original item.  
You can only pull messages to yourself.  
1.  
Click the folder and subfolders until you reach the list page.  
2.  
If necessary, filter the items in the list. See Filter messages.  
3.  
Do one of the following:  
o  
Point to the item and click  
.  
o  
Click the email's title to view the complete email and then click  
Pull.  
4.  
Answer the call. See Answer a call.  
CCSP UI Help - Supervisor - 7.4  
97

Remove messages  
Note  
l  
Agents can remove only scheduled callbacks that are visible to them.  
l  
Supervisors can remove emails, callbacks, and voicemail.  
l  
The removed item moves to the Deleted Emails, Deleted Callbacks, or Deleted Voicemail folder, as  
appropriate. Removing the item from these folders permanently removes the item.  
1.  
Click the folder and subfolders until you reach the list page.  
2.  
If necessary, filter the items in the list. See Filter messages.  
3.  
Do one of the following:  
o  
Point to the item and click  
.  
o  
Select the check boxes for the items and click  
Delete.  
o  
Click the email's title to view the complete email and then click  
Remove.  
CCSP UI Help - Supervisor - 7.4  
98

Restore messages  
Note  
l  
Only supervisors can restore messages from deleted item folders.  
l  
Inbound messages can be moved only to inbound messages folders. Outbound messages can be moved  
only to outbound messages folders. For more information, see Source folders and Destination folders  
below.  
l  
Bulk restore is not currently supported.  
1.  
Go to the Deleted Emails, Deleted Callbacks, or Deleted Voicemail folder, as appropriate.  
2.  
If necessary, filter the items in the list. See Filter messages.  
3.  
Point to the item and click  
.  
4.  
Choose a folder from the list. You can also search for a folder.  
Source folders  
The following table shows the supported source folders and source item states.  
Media and item type  
Source folders  
Source item state  
Incoming email  
Deleted Emails  
Archived  
Outgoing email  
Deleted Emails  
Archived  
Callbacks  
Deleted Callbacks  
Archived  
Voicemail  
Deleted Voicemails  
Archived  
CCSP UI Help - Supervisor - 7.4  
99

Destination folders  
The following table shows the supported destination folders and target item states.  
Media and item type  
Destination folders  
Target item state  
Incoming email  
Allowed Public Folders  
Unmapped  
Not Replied  
Replied  
Rejected Emails  
Archived  
Outgoing email  
Drafts  
Unmapped  
Sent Items  
Archived  
Callbacks  
Callbacks  
Unmapped  
Rejected Callbacks  
Archived  
Voicemail  
Voicemail  
Unmapped  
Archived Voicemails  
Rejected Voicemails  
Archived  
CCSP UI Help - Supervisor - 7.4  
100

Access archived messages  
Note  
l  
The online archive is only available for non-Exchange data stores and must be set up by your system  
administrator.  
l  
Messages in the online archive cannot be deleted by any user.  
l  
For detailed information, see Messaging items visibility by role.  
View messages from a folder in the online archive  
When configured by your system administrator, the following folders may be available in the online archive:  
l  
Deleted Emails  
l  
Not Replied  
l  
Rejected Emails  
l  
Replied  
l  
Sent Items  
l  
Deleted Callbacks  
l  
Rejected Callbacks  
l  
Archived Voicemails  
l  
Deleted Voicemails  
l  
Rejected Voicemails  
1.  
Click on the folder from the relevant media parent folder.  
2.  
If the folder is available in the online archive, the  
button appears next to the search box. Click this  
button to see the contents of the folder in the online archive. You can return to the active folder contents  
by clicking  
again.  
3.  
To view an email in any of the email folders in the online archive, click the email preview in the folder.  
Filter messages from a folder in the online archive  
See the filtering and sorting options described in Filter messages. Any selected filters and sort options are  
applied to both the active and online archive content of the folder.  
Pull messages from a folder in the online archive  
See the steps in Pull messages.  
Note  
Pulling a message from the online archive creates a copy of the message in the destination folder in the  
Active content data store. The original message remains in the online archive.  
CCSP UI Help - Supervisor - 7.4  
101

Restore deleted messages from a folder in the online archive  
Note  
Restoring a message from a Deleted Items folder in the Archive creates a copy of the message in the  
destination folder. The original message remains in the online archive.  
1.  
Go to the Deleted Emails, Deleted Callbacks, or Deleted Voicemail folder, as appropriate and click  
.  
2.  
If necessary, filter the items in the list. See Filter messages.  
3.  
Point to the item and click the that appears.  
4.  
Choose a folder from the list. You can also search for a folder.  
CCSP UI Help - Supervisor - 7.4  
102

5: Personal voicemail  
If you do not have a direct extension, you should skip this section.  
This section contains the following topics:  
l  
Retrieve voicemail  
l  
Set up your mailbox  
l  
Change your PIN  
l  
Personal voicemail quick reference  
Some CCSP UI users are granted a direct extension that enables them to receive voice calls directly and not  
through an agent group. Generally, a direct extension is provided with a personal voice mailbox and a PIN to  
control access to the mailbox.  
Your administrator or supervisor should provide your PIN. The PIN must be a number.  
Retrieve personal voicemail  
When you have personal voicemail, a notification appears in the gadgetory  
1.  
Click  
. An audio message asks for your ID.  
2.  
Click/press your ID numbers on the CCSP UI dialpad. Your ID is typically your direct extension telephone  
number.  
3.  
To send your ID, click/press #. An audio message asks for your PIN.  
4.  
Click/press your PIN numbers on the CCSP UI dialpad.  
5.  
To send your PIN, click/press #. An audio message states the number of read messages and the number of  
new messages.  
6.  
Do one or more of the following:  
o  
To listen to your messages, click/press 1.  
o  
To delete the message, click/press 7.  
o  
To save the message, click/press 9.  
o  
For more options, click/press 0.  
o  
To replay the current message, click/press 4.  
o  
To hear the message envelope, click/press 5.  
o  
To return to the main menu, click/press \*.  
o  
To repeat the menu, click/press #.  
o  
To replay a message from the start, click/press 1 while the message is still playing.  
CCSP UI Help - Supervisor - 7.4  
103

Set up your personal mailbox  
After dialing into your mailbox and entering your ID and PIN, set up your mailbox as follows.  
Set up or edit your mailbox preferences  
1.  
Click/press 8.  
2.  
To change your password, click/press 1. See Change the password.  
3.  
To change your personal greeting, click/press 2. See Record a greeting/name.  
4.  
To change the recording of your name, click/press 3. See Record a greeting/name.  
5.  
To return to the main menu, click/press \*.  
6.  
To hear the menu again, click/press #.  
Change the password  
An audio message asks you to enter a new password, followed by the pound sign.  
1.  
Enter the password and click/press #. The password must be numeric.  
2.  
Do one of the following:  
o  
To keep the password, click/press 1.  
o  
To enter a different password click/press 2.  
Record a greeting/name  
An audio message plays your current greeting/name.  
1.  
To record a new greeting/name, click/press 1.  
2.  
To finish recording, click/press #.  
3.  
Do one of the following:  
o  
To save the new greeting/name, click/press 1.  
o  
To record a different greeting/name, click/press 2.  
o  
To delete the greeting and use the standard greeting, click/press 7.  
CCSP UI Help - Supervisor - 7.4  
104

Change your PIN  
Your Personal Identification Number (PIN) can be changed through CCSP UI and through your voice device.  
1.  
On the gadgetory, click  
Settings and then click Preferences. The Preferences dialog appears.  
2.  
In Mailbox Pin, type the four digits of your new PIN.  
3.  
Click Save.  
CCSP UI Help - Supervisor - 7.4  
105

Personal voicemail quick reference  
Log on  
1.  
Through your own phone or another phone, dial your voicemail telephone number.  
2.  
Dial your PIN.  
3.  
Press  
.  
Play Messages menu  
Press the appropriate key.  
Key  
Option  
Play your messages.  
Replay this message.  
Play the envelope of this message.  
Delete this message.  
Change mailbox preferences. Continue at Mailbox preferences below.  
Save this message.  
To hear more options.  
Repeat this menu.  
Return to the main menu.  
Note  
If you press  
 in the main menu, you hang up the  
call.  
CCSP UI Help - Supervisor - 7.4  
106

Mailbox preferences  
Press the appropriate key.  
Key  
Option  
Change password. Continue at Password menu below.  
Change greeting. Continue at Greeting menu below.  
Change the recording of your name.  
Return to the main menu.  
Repeat this menu.  
Password menu  
Press the appropriate key.  
Key  
Option  
Keep new password.  
Select a different password.  
Greeting menu  
Press the appropriate key.  
Key  
Option  
Record new greeting. Continue at Greeting Confirmation menu below.  
Erase greeting and use standard greeting.  
Greeting Confirmation menu  
Press the appropriate key.  
Key  
Option  
Save new greeting.  
Replace greeting.  
CCSP UI Help - Supervisor - 7.4  
107

6: Contact History  
This section contains the following topics:  
l  
Open Contact History  
l  
History dashboard  
l  
Search for an interaction  
l  
Manage interactions  
l  
Contacts dashboard  
l  
Search for a contact  
l  
Manage contacts  
l  
Send direct message  
l  
View errors  
Contact History is a CCSP add-on for managing contacts and contact account details.  
Contact History consists of the following parts that are deployed together:  
l  
Social Connector Helper API — API to link between database and client and provide data for drop down lists.  
l  
Contact History Application — user interface to display and manage account data.  
l  
Functional gadget — customizes the CCSP UI client side and allows interaction with Contact History.  
l  
Social Connector — optional component used to interact with social media channels, such as WhatsApp and  
Facebook.  
Contact History language  
By default, Contact History uses the language you select when you log in to CCSP UI.  
If you want to use a different language for Contact History, your Administrator can set that language in the  
(Contact History Application—Language Configuration) section of the Contact History configuration file.  
CCSP UI Help - Supervisor - 7.4  
108

Open Contact History  
Note  
If you cannot see the Contact History gadget in the gadgetory, ask your Administrator to check that in CCSP  
UI Admin, the gadget is enabled, the Contact History URL is set, and the gadget is assigned to your group.  
1.  
On the gadgetory, click  
Contact History. The History dashboard appears.  
2.  
To view and manage contact information, click  
Contacts to open the Contacts dashboard.  
CCSP UI Help - Supervisor - 7.4  
109

History dashboard  
Search and filter panel  
On the History page, you can view a grid of calls and Social Connector interactions made during the default  
configured time period.  
You can perform a search on previous interactions using the search field and advanced filters. For details, see  
Search for an interaction.  
You can also set the My Calls filter to automatically display only your interactions. For details, see Search using  
filters.  
Note  
l  
The default time period for the displayed search results is configurable and is set by the SQL Job.  
l  
By default, the process that retrieves the latest updates of call interactions for Contact History runs every  
60 minutes. If you require more frequent updates, speak to you Administrator.  
From the interaction history search results, you can do the following.  
CCSP UI Help - Supervisor - 7.4  
110

Contact actions  
In the Contact Name column:  
To...  
Click...  
Next...  
Open a contact's page  
The name link  
See Contacts dashboard.  
Add an unregistered contact  
See Add an unregistered contact.  
Interaction actions  
In the Actions column:  
To...  
Click...  
Next...  
View interaction details  
Interaction  
See View interaction details.  
Play a recording of an interaction  
Play Recording  
See Play a recording of an interaction.  
Make an outbound voice or email call  
Call or  
Email  
See Make an outbound voice or email  
call.  
Send an outbound message to a  
social channel  
Chat history  
See Send an outbound message.  
CCSP UI Help - Supervisor - 7.4  
111

Search for an interaction  
On the History page, you can search on all the fields or specify filters.  
You can use any combination of these search types:  
l  
Single search only – returns records where the requested text exists in one of the single-search text fields  
(itemized below). It also uses the default values in the Filters, such as dates.  
l  
Filters only – when no single-search is supplied, all data is returned that matches the configured filters.  
l  
Both – returns records where the requested text exists in one of the single-search text fields (itemized  
below) and these matched records are additionally filtered by the values you specify in the Filters section. It  
is the same as single-search but with additional non-default filters applied.  
Search all fields  
To search for text contained in the interaction record:  
1.  
Open the History page.  
2.  
Type the search text in the Search all fields box and click  
.  
3.  
To clear the search criteria, click  
.  
The single-search looks for the requested text in any of these fields.  
Properties  
Search values (example)  
Call Type  
Chat Call, Voice Call  
Caller ANI  
The caller number  
Caller DNIS  
The called number  
Caller Name  
The name of the caller  
CCSP UI Help - Supervisor - 7.4  
112

Properties  
Search values (example)  
Queue Name  
The name of the queue where the call is delivered and handled  
Agent Name  
The name of the agent who handle the interaction  
WrapUp Name  
The name of the assigned wrap-up  
WrapUp Data  
The wrap-up note data/information  
Email Subject  
The subject of the email interaction  
Channel Name  
The channel name (such as WhatsApp/Facebook) from Social Connector  
interactions  
AccountDetails  
Identifier  
The user identifier used to match to accounts via Social Connector  
SCFromID  
The user identifier (ID) from Social Connector  
SCToID  
The bot identifier (ID) from Social Connector  
Search using filters  
If you need to perform a search based on more complex criteria:  
1.  
Open the History page.  
2.  
Click  
to expand the search attributes and properties.  
CCSP UI Help - Supervisor - 7.4  
113

3.  
Select or enter the advanced filter criteria. You can search for an existing interaction record based on the  
following attributes:  
o  
Start Date and End Date  
o  
Account name  
o  
Contact details identifier  
o  
From Party and/or To Party  
o  
Direction  
o  
Interaction Type  
o  
Wrap-up Name  
o  
Queue  
o  
Call ID  
o  
Caller Message  
o  
Channel Name  
o  
Agent Name  
4.  
If you always want to filter for calls that you have handled, select the My Calls check box. When this option  
is enabled, each time you log in to CCSP UI, your name automatically appears in the Agent Name filter.  
CCSP UI Help - Supervisor - 7.4  
114

In the following example, calls are filtered by the caller's number, entered in From Party.  
CCSP UI Help - Supervisor - 7.4  
115

Manage interactions  
This section contains the following topics:  
l  
Add an unregistered contact  
l  
View interaction details  
l  
Play a recording of an interaction  
l  
Make an outbound voice or email call  
l  
Send an outbound message to a social channel  
See also History dashboard.  
CCSP UI Help - Supervisor - 7.4  
116

Add an unregistered contact  
1.  
Open the History page.  
2.  
In the Contact Name column, click  
. The Add Contact dialog appears.  
CCSP UI Help - Supervisor - 7.4  
117

3.  
To add the information to a new contact, following the instructions for Add a new contact.  
4.  
To add the information to an existing contact, click the Add to existing contact tab.  
5.  
Search for the existing contact. The matching contacts appear below the search field.  
6.  
Click  
. The message Record was successfully added appears at the bottom of the screen.  
CCSP UI Help - Supervisor - 7.4  
118

View interaction details  
1.  
Open the History page.  
2.  
Search for the interaction.  
3.  
In the Actions column, click  
Interaction. The Interaction Details dialog appears.  
4.  
For chat calls, you can view the Conversation log.  
5.  
To close the dialog, click X.  
CCSP UI Help - Supervisor - 7.4  
119

Play a recording of an interaction  
Note  
Your Administrator can configure Replay or Agent Evaluation to review interactions.  
1.  
Open the History page.  
2.  
Search for the interaction.  
3.  
In the Actions column, click  
Play Recording. CCSP Replay opens in a new tab.  
Make an outbound voice or email call  
Note  
If you cannot make outgoing calls or send emails from the Contact History gadget, ask your Administrator  
to check that the touchPointType of channel is set to email or call.  
1.  
Open the History page.  
2.  
Search for the interaction.  
3.  
In the Actions column, click  
Call or  
Email. The CCSP UI call card appears.  
CCSP UI Help - Supervisor - 7.4  
120

Send an outbound message to a social channel  
Note  
l  
If you cannot make outgoing calls or send emails from the Contact History gadget, ask your  
Administrator to check that the touchPointType of channel is set to email or call.  
l  
If you want to view more than 5 conversations at a time, ask your Administrator to change this setting.  
1.  
Open the History page.  
2.  
Search for the interaction.  
3.  
In the Actions column, click  
Chat history. The conversation history dialog appears.  
4.  
If there are more conversations than the configured limit, click Show 5 more conversations to view pre-  
vious conversations.  
5.  
Type your message and click Send.  
CCSP UI Help - Supervisor - 7.4  
121

Contacts dashboard  
Contacts panel  
On the Contacts page, you can perform a search on a user account from the search field. See Search for a  
contact.  
If you have permission to modify contacts, you can also Add a new contact.  
Details tab  
When a contact's information is retrieved, their identifiers are presented on the Details tab.   
On this tab, you can initiate a voice call or chat via the interactive icon (if it is available and depending on the  
configuration of Contact History and Social Connector).  
If you have permission to modify contacts, you can also perform the following actions.  
Contact actions  
To...  
Click...  
Next...  
Edit a contact  
See Edit a contact  
Delete a contact  
See Delete a contact  
CCSP UI Help - Supervisor - 7.4  
122

To...  
Click...  
Next...  
Add a contact identifier  
See Add a contact identifier  
Edit a contact identifier  
See Edit a contact identifier  
Delete a contact identifier  
See Delete a contact identifier  
Import or export contacts  
See Import contacts or Export contacts  
History tab  
On the History tab, you and/or your supervisor can view all past call interactions between the contact and the  
call center.  
You and/or your supervisor can view details of each interaction, including conversation history (available only  
for Social Connector channel calls), call arrival and termination time, queue information, call type, ANI, and  
DNIS details.  
You can also perform the following actions.  
CCSP UI Help - Supervisor - 7.4  
123

Interaction actions  
In the Actions column:  
To...  
Click...  
Next...  
View interaction details  
Interaction  
See View interaction details.  
Play a recording of an interaction  
Play Recording  
See Play a recording of an  
interaction.  
Make an outbound voice or email call  
Call or  
Email  
See Make an outbound voice or  
email call.  
Send an outbound message to a  
social channel  
Chat history  
See Send direct message.  
Dashboard tab  
On the Dashboard tab, in the Interactions panel, you can view the contact's previous interaction stats by  
interaction type, the total number of previous interactions, and the last interaction date and time.  
The Survey and Evaluation statistics panels are optional and require specific features to be deployed.  
CCSP UI Help - Supervisor - 7.4  
124

Interactions  
Stat name  
Description  
Total  
Total number of interactions  
Chat non-social  
Total records for Non-social Connector chat  
Chat social  
Total records for Social Connector chat  
Phone  
Total records for Phone interaction  
Email  
Total records for email interaction  
Voice  
Total records for Voice call interaction  
Callback  
Total records for created callback(s)  
Voicemail  
Total records for created voicemail  
Surveys  
Stat name  
Description  
Last Survey Result  
Total number of survey records that was completed by the caller  
CCSP UI Help - Supervisor - 7.4  
125

Stat name  
Description  
CES  
Customer effort score displayed as a percentage  
CAST  
Customer sanctification score displayed as a percentage  
Evaluation  
Evaluation displayed as a percentage  
NPS  
The Net Promoter Score for the surveys made, displayed as a percentage  
Sentiments (evaluation)  
Stat name  
Description  
Sentiment Positive  
Total number of submitted positive sentiments  
Last Sentiment Positive  
The last date where a positive sentiment was submitted  
Sentiment Negative  
Total number of submitted negative sentiments  
Last Sentiment Negative  
The last date where a negative sentiment was submitted  
Sentiment Neutral  
Total number of submitted neutral sentiments  
Last Sentiment Neutral  
The last date where a neutral sentiment was submitted  
CCSP UI Help - Supervisor - 7.4  
126

Direct Message history  
The Direct Message history tab is a feature of Social Connector call detail record (CDR) chat conversation  
recording.  
On this tab, you and/or your supervisor can view the existing conversation records between the contact and an  
agent and also send a direct message from this session. See Send direct message.  
Fields tab  
The Fields tab shows information such as the region and job title records that were assigned to the contact.  
CCSP UI Help - Supervisor - 7.4  
127

These fields serve as keywords for Contact Search. When the field name matches, the search shows all contacts  
assigned to that field name in the results.  
These fields in the database are defined by the Administrator. While you can assign these fields from Contact  
History, you cannot create or edit them within Contact History. The same restrictions apply to the Organization  
field.  
Note  
For the Administrator:  
The Organization and custom fields are related to accounts and do not pertain to call history. You can  
create a list of organizations in the database, or have a custom external import process manage this.  
Custom fields consist of small lists of data that can be linked to an account, while organizations can be  
represented in a larger list. Examples of custom fields include job titles, sites, or any other arbitrary  
categories.  
Related Accounts tab  
The same name can be assigned to multiple contact accounts within different organizations with externally  
imported data. Use the Related Accounts tab to view all the contact accounts with the same name as the  
selected contact.  
CCSP UI Help - Supervisor - 7.4  
128

Search for a contact  
On the Contacts page, you can search on the following fields:  
l  
First name  
l  
Last name  
l  
External CRM ID  
l  
Organization  
l  
Custom fields, such as Region (see Fields tab)  
1.  
Open the Contacts page.  
2.  
In the Contacts panel search bar, do one of the following:  
o  
Start typing in the search box to automatically filter the list.  
o  
Type a keyword. See Search on organization or field below.  
3.  
Select the contact in the list to access their account information and historical interactions.  
CCSP UI Help - Supervisor - 7.4  
129

Search on organization or fields  
For example, this contact has the following fields assigned to values:  
l  
Region  
l  
Job Title  
l  
Organisation  
If we examine this user's account details, we can find the following information that was assigned.  
CCSP UI Help - Supervisor - 7.4  
130

When searching for a keyword in one of the fields, the related results are automatically included. For example,  
if I search for the Organisation name or part of it, I will receive relevant matches.  
The same applies if I search for the Region name or part of it.  
CCSP UI Help - Supervisor - 7.4  
131

CCSP UI Help - Supervisor - 7.4  
132

Manage contacts  
This section contains the following topics:  
l  
Add a new contact  
l  
Edit a contact  
l  
Delete a contact  
l  
Add a contact identifier  
l  
Edit a contact identifier  
l  
Delete a contact identifier  
l  
Import contacts  
l  
Export contacts  
See also Contacts dashboard.  
Note  
Actions related to modifying data (edit/add/delete) may be limited to specific roles. If you should be able to  
manage contacts but cannot, contact your Administrator.  
CCSP UI Help - Supervisor - 7.4  
133

Add a new contact  
Note  
You can create a contact with identical first and last names.  
However, you cannot create a duplicate identifier for any voice or social channels, such as duplicate phone,  
SMS, or WhatsApp numbers. Each identifier is unique and can only be registered under one contact  
account.  
1.  
Open the Contacts page.  
2.  
Click  
. The Add Contact dialog appears.  
CCSP UI Help - Supervisor - 7.4  
134

3.  
Enter the following information:  
o  
First name (mandatory)  
o  
Last name (mandatory)  
o  
External CRM ID — a unique identifier for the contact, helping to distinguish between the same names.  
o  
Channel (mandatory) — select the channel, for example, Facebook.  
o  
Contact identity (mandatory) — for example, 123456789@whatsapp.com, a phone number, or email.  
o  
Comment — notes about the contact.  
o  
Alias — the alias for the channel, such as a user name in Facebook.  
o  
Region — select from the list. To filter the list, start typing the region.  
o  
Job title — select from the list. To filter the list, start typing the job title.  
o  
Organization — select from the list. To filter the list, start typing the organization.  
4.  
Click Add Contact. The new [contact's name] page appears.  
CCSP UI Help - Supervisor - 7.4  
135

Edit a contact  
1.  
Open the Contacts page.  
2.  
Search for the contact.  
3.  
On the [contact name's] page, on the Details tab, click  
. The Edit Contact [contact's name]  
dialog appears.  
4.  
Edit the information. For details, see Add a new contact.  
5.  
Click Save. The edited information appears on the [contact name's] page.  
CCSP UI Help - Supervisor - 7.4  
136

Delete a contact  
Caution  
Deleting a contact permanently erases the contact details. It cannot be undone.  
1.  
Open the Contacts page.  
2.  
Search for the contact.  
3.  
On the [contact's name] page, on the Details tab, click  
. The Delete Contact dialog  
appears.  
4.  
Type DELETE (uppercase) in the field. The Delete button is now active.  
5.  
Click Delete.  
CCSP UI Help - Supervisor - 7.4  
137

Add a contact identifier  
1.  
Open the Contacts page.  
2.  
Search for the contact.  
3.  
On the [contact's name] page, on the Details tab, click  
. The Add [contact's name] dialog  
appears.  
4.  
Enter the following information:  
o  
Channel (mandatory) — select the channel, for example, Facebook.  
o  
Contact identity (mandatory) — for example, 123456789@whatsapp.com, a phone number, or email.  
o  
Alias — the alias for the channel, such as a user name in Facebook.  
o  
Description — for example, Home phone or Work contact.  
5.  
Click Add. The new contact identifier appears on the [contact's name] page.  
CCSP UI Help - Supervisor - 7.4  
138

Edit a contact identifier  
1.  
Open the Contacts page.  
2.  
Search for the contact.  
3.  
On the [contact's name] page, on the Details tab, click  
Edit in the row of the contact identifier to  
edit. The Edit [contact's name] dialog appears.  
4.  
Edit the contact identifier information. For details, see Add a contact identifier.  
5.  
Click Save. The edited contact identifier information appears on the [contact's name] page.  
CCSP UI Help - Supervisor - 7.4  
139

Delete a contact identifier  
1.  
Open the Contacts page.  
2.  
Search for the contact.  
3.  
On the [contact name's] page, on the Details tab, click  
Delete in the row of the contact identifier to  
delete.  
4.  
In the Delete Contact Details dialog, click Delete.  
CCSP UI Help - Supervisor - 7.4  
140

Import contacts  
Note  
l  
The import file must be an Excel file with the two tabs and columns specified below.  
l  
The OfficialID field appears on both tabs and links the account with the account details.  
l  
You can only import new accounts. You cannot edit existing ones.  
Excel file  
Prepare the Excel file to import:  
1.  
Create an Excel file with two tabs: Accounts and AccountDetails.  
2.  
On the Accounts tab, add the following columns:  
3.  
On the AccountDetails tab, add the following columns:  
4.  
Add your data to this file.  
Import process  
Import into Contact History:  
1.  
Open the Contacts page.  
2.  
In the Contacts panel, click  
and then click Import contacts.  
3.  
In the Open dialog, select the Excel file and click Open.  
CCSP UI Help - Supervisor - 7.4  
141

Export contacts  
Note  
The contacts are exported as an Excel file, with two tabs: Accounts and AccountDetails. For more  
information, see Import contacts.  
1.  
Open the Contacts page.  
2.  
In the Contacts panel, click  
and then click Export contacts.  
3.  
The Excel file is exported to your Downloads folder.  
CCSP UI Help - Supervisor - 7.4  
142

Send direct message  
Send Direct Message enables an agent to use Social Connector to proactively send messages ad-hoc without a  
corresponding contact-center chat conversation.  
For example, Twitter/X Tweets and Facebook posts can be queued for a CCSP agent to collect and respond to  
without waiting for a subsequent response from the recipient.  
From the perspective of Contact History, Send Direct Message enable agents to visually see the conversation  
history, when the channel’s CDR is enabled in Social Connector.  
Agents can see the conversation history from the History and the Contacts dashboards.  
On the History dashboard, click the  
Chat history icon to view all the historical conversations for the  
identifier.  
On the Contacts dashboard, on the Details tab, click the  
Chat icon for the same view of all the historical  
conversations for the identifier  
CCSP UI Help - Supervisor - 7.4  
143

For more information, see Send an outbound message.  
If there are more conversations than the configured limit a button will be presented at the top to allow the user  
to get some previous conversations (the initial quantity loaded, and the amount subsequently loaded each time  
are configurable – if users prefer to have fewer/more than currently they may contact their Administrators to  
review changing):  
The editor at the bottom may allow more text formatting than the channel permits, and most channels require  
that the text is reformatted (for example, putting \* either side to signify bold text) and this reformatted text is  
displayed in this view when the conversation is revisited.  This is also performed by Social Connector status  
messages, for example, where the text has \_ either side for displaying in WhatsApp as italics.  
Send Direct Message as a standalone gadget  
Administrators can create and assign a standalone gadget in CCSP UI that enables the agent to send direct  
messages through a specific media channel, such as SMS, WhatsApp, or Instagram messaging.  
This customized gadget allows the agent to send a message directly without having to navigate through the  
Contact History or Interactions pages.  
A preview of SMS gadget (as a standalone gadget).  
CCSP UI Help - Supervisor - 7.4  
144

The agent can send a direct SMS to the recipient using the standalone gadget.  
CCSP UI Help - Supervisor - 7.4  
145

View errors  
Error notifications  
When an error occurs, a red error notification appears at the bottom of the screen, providing brief error  
information.  
To learn more about the error, open the Errors dashboard for additional information.  
CCSP UI Help - Supervisor - 7.4  
146

Errors dashboard  
The Errors dashboard provides additional information about issues or errors related to the Contact History.  
The standard errors are caused by:  
l  
Adding a duplicated identifier to a contact  
l  
Misconfiguration of Contact History  
l  
Misconfiguration of Social Connector Helper API  
l  
Networking issues, firewall, or SSL configuration  
l  
Missing required configuration files  
Common errors  
The following errors can be caused due to misconfiguration:  
l  
Contact History not being updated – Contact History list is occasionally cached to reduce the load on the  
database. By default, {HISTORICAL\_DB\_NAME}\_CHH\_Cache SQL Job runs once per hour. Running intervals  
can be changed in job’s properties by Administrators.  
l  
Account is not found in Contact History list – when you have a related account in Contact History but Contact  
History still displays the Add button – make sure, that the AccountDetailsIdentifier is exactly the same as in  
ANI for example, +123@whatsapp.com and 123@whatsapp.com are different identifiers.  
l  
A duplicated account identifier is found  – when you attempt to add an identifier to an account that exists in  
the system, an error occurs and the information is displayed on the Errors dashboard for more information.  
CCSP UI Help - Supervisor - 7.4  
147

l  
Add, delete and edit functionality is not available – your role may not have permissions to perform account  
actions. Contact your Administrator to ensure you are configured for account management.  
l  
OPTIONS {Service API URL} net::ERR\_CONNECTION\_REFUSED – means the connection to Social Connector is  
either disabled or unreachable from your desktop. Contact your Administrator to resolve the connection  
issues.  
CCSP UI Help - Supervisor - 7.4  
148

7: Supervisor  
This section contains the following topics:  
l  
Filter agents  
l  
View agent interactions  
l  
Monitor agents  
l  
Answer an assistance request  
l  
Initiate agent assistance  
l  
Broadcast messages  
l  
Remotely log out agents  
Filter agents  
Note  
l  
Team supervisors are associated with teams through their user properties.  
l  
Regular supervisors are associated with teams and groups through their user profile  
definition.  
To filter agents by state, group, or team, click the arrow to the right of category.  
The groups and teams that appear in the lists are as follows.  
Supervisor type  
Groups  
Teams  
Tenant-wide  
supervisor  
All provisioned groups and All Groups  
option  
All provisioned teams and All Teams  
option  
Team supervisor  
All Groups option  
All associated teams and All Teams  
option  
Regular supervisor  
All associated groups and All Groups  
option  
All associated teams and All Teams  
option  
Tenant-wide  
supervisor  
All provisioned groups and All Groups  
option  
All provisioned teams and All Teams  
option  
CCSP UI Help - Supervisor - 7.4  
149

View agent interactions  
The number of interactions each agent is handling appears on the Supervisor tab.  
To view details about the type, duration, and status of each interaction, hover over the number.  
CCSP UI Help - Supervisor - 7.4  
150

Monitor agents  
Note  
l  
When two supervisors whisper monitor the same agent, the supervisors hear each other, but the caller  
cannot hear the supervisors.  
l  
An agent cannot transfer a call to a supervisor that is monitoring that agent.  
l  
You cannot monitor email or callback calls.  
l  
ReadyReps and business users are not listed on the Supervisor tab.  
Supervisors can monitor an agent's voice and chat calls.  
The monitoring modes are as follows:  
l  
Silent — click to hear or observe live dialog.  
l  
Whisper — click to hear or observe live dialog and whisper in to the representative's ear piece or listening  
device.  
l  
Barge in — click to hear or observe live dialog and barge into the session.  
1.  
If necessary, search for the agent on the Supervisor tab. To filter agents, click the arrow to the right of the  
category. See Filter agents.  
2.  
Click the agent.  
3.  
In the agent card, click  
Monitor and then click the monitoring mode.  
4.  
To change monitoring mode, click a different monitoring mode.  
5.  
To stop monitoring, click Monitor and then click Stop.  
CCSP UI Help - Supervisor - 7.4  
151

Use camp on monitoring  
In CCSP, the type of monitoring that is performed when supervisors elect to monitor agents is referred to as  
camp on monitoring. In camp on monitoring, supervisors can click to monitor any agent. If the agent is in call,  
monitoring begins immediately. If the agent is not in call, for example unavailable or in wrap-up, then the  
supervisor is camped on to the agent's CCSP UI and begins monitoring the agent as soon as the agent goes in  
call, thus enabling the supervisor to hear the beginning of the call. The supervisor continues to monitor that  
agent until the supervisor manually ends the monitoring session.  
If the agent ends a call that is being monitored, the supervisor continues to be camped on to that agent until  
the next call, at which point, that call is monitored. In the interim, the supervisor cannot hear the camped-on  
agent.  
If the agent logs off, the camp-on session ends.  
If the supervisor is camped on to an agent waiting for a call, a short zip tone is played to the supervisor at the  
start of the call, when the agent answers it. If the agent is already in call when the supervisor starts monitoring,  
the zip tone is not heard.  
CCSP UI Help - Supervisor - 7.4  
152

Answer an assistance request  
Agents with a pending assistance request appear at the top of the supervisor list, irrespective of the filter. The  
number of pending requests appears in the gadgetory. To view the assistance request and time lapsed since it  
was sent, hover over the corresponding  
.  
1.  
Do one of the following:  
o  
In the gadgetory, click  
. The Assistance dialog appears.  
o  
If you previously closed the agent card, click the corresponding  
. The agent card appears.  
2.  
Click Assist.  
3.  
Type the response.  
4.  
Click Send.  
5.  
To end the session, click End session.  
CCSP UI Help - Supervisor - 7.4  
153

Initiate agent assistance  
Note  
Supervisor assistance is not available for ReadyReps or business users.  
To request assistance from another supervisor, in the gadgetory, click  
Assistance.  
1.  
Double-click the agent. The agent card appears.  
2.  
Click Assist.  
3.  
Type the message in the text box.  
4.  
Click Assist.  
5.  
To end the session, click End Session.  
Broadcast messages  
1.  
Select the agents that will receive the message. To select all the agents, select the check box to the left of  
the Name column. The total selected agents appears at the top of the Supervisor tab.  
2.  
Click  
Broadcast. The Broadcast dialogue box appears.   
3.  
Type the message.  
4.  
Click Send.  
The message appears with a time stamp together with all previous messages in the agent's Assistance dialog.  
The message does not appear in the supervisor's Assistance dialog.  
CCSP UI Help - Supervisor - 7.4  
154

Remotely log out agents  
1.  
Select the agents. The total selected agents appears at the top of the Supervisor tab.  
2.  
Click  
Remote Logout. The Remote Logout dialog appears.  
3.  
Click OK. The Logout Confirmation dialog appears on each agent's screen. See Remote logout.  
CCSP UI Help - Supervisor - 7.4  
155

# Contact Center: Service Provider Security Guide Version 7.3 and 7.4

CCSP-SG-73/74-R19-06/2025   
Contents   
1: Intro .............................................................................................................................................5   
What’s new ................................................................................................................................................. 5   
Definitions .................................................................................................................................................. 5   
Legal disclaimer ......................................................................................................................................... 6   
Support ....................................................................................................................................................... 6   
2: CCSP configuration best practice .............................................................................................7   
Protect against Microsoft IIS tilde directory enumeration ................................................................. 7   
Protect against server version disclosure .............................................................................................. 9   
Set the customErrors configuration to RemoteOnly ......................................................................... 10   
Protect against clickjacking .................................................................................................................... 10   
Set the HttpOnly flag on cookies .......................................................................................................... 12   
Remove obsolete provisioning applications ....................................................................................... 12   
CCSP UI Client Services folder must be under the TouchPoint folder .............................................. 13   
3: CCSP component security settings ........................................................................................ 14   
Overview ................................................................................................................................................... 14   
svccosmocall user ........................................................................................................................... 14   
Dedicated users .............................................................................................................................. 15   
System services security change mapping ................................................................................. 15   
System web services security change mapping ......................................................................... 17   
System web virtual directories ...................................................................................................... 18   
Elsbeth web services identity ........................................................................................................ 18   
File location changes ...................................................................................................................... 18   
Registry changes ............................................................................................................................. 19   
Configure the alsUser ............................................................................................................................. 20   
Configure the ccspAdminUser .............................................................................................................. 22   
Configure the ccspUIUser ...................................................................................................................... 26   
Configure the ccspWebUser .................................................................................................................. 29   
Configure the cfmUser ........................................................................................................................... 31   
Configure the dtrUser ............................................................................................................................. 35   
Configure the ecsUser ............................................................................................................................ 37   
Configure the lsUser ............................................................................................................................... 41   
Configure the msUser............................................................................................................................. 43   
Configure the msUser to work with Exchange as MSR ............................................................. 43   
Configure the msUser to work with MSR DB .............................................................................. 49   
Configure the pdsUser ........................................................................................................................... 50   
Configure the privateVMUser ................................................................................................................ 52   
Configure the reducedSVC user ............................................................................................................ 54

CCSP Security Guide - 7.3 and 7.4   
3   
Configure the replayUser ....................................................................................................................... 55   
Configure CCSP services to log on with reduced users ..................................................................... 57   
Configure CCSP web services to log on with reduced users ............................................................ 59   
Configure SMTP server to relay outbound emails ............................................................................. 60   
4: CCSP recording encryption deployment ............................................................................... 62   
Overview ................................................................................................................................................... 62   
Key Server installation and configuration ........................................................................................... 63   
Prerequisites .................................................................................................................................... 63   
Install the CCSP Key Server ........................................................................................................... 65   
Configure the CCSP Key Server ..................................................................................................... 65   
Manually install the codec ............................................................................................................. 66   
Certificate generation ............................................................................................................................. 67   
Generate the signing certificate ................................................................................................... 67   
Install the signing certificate on CosmoCorder servers............................................................ 67   
Export the verification certificate ................................................................................................. 68   
Import the verification certificate to the Key Server ................................................................. 68   
Generate a tenant’s encryption certificate ................................................................................. 69   
Install a tenant’s encryption certificate ....................................................................................... 69   
Client configuration ................................................................................................................................ 69   
Set up access to the Key Server via HTTPS .................................................................................. 69   
Admin settings ......................................................................................................................................... 70   
Verification................................................................................................................................................ 71   
Encryption ........................................................................................................................................ 71   
Playback ........................................................................................................................................... 71   
5: Authentication process ........................................................................................................... 72   
6: CCSP HTTPS configuration ..................................................................................................... 73   
Prerequisites ............................................................................................................................................ 73   
Configuration and installation ............................................................................................................... 73   
Summary .......................................................................................................................................... 73   
Before starting ................................................................................................................................ 74   
Configure DC as Root Certification Authority ............................................................................. 74   
Configure the front-end server .................................................................................................... 91   
Configure the clients .................................................................................................................... 102   
CCSP components installation ............................................................................................................. 106   
Install server components ........................................................................................................... 106   
Upgrade to CCSP 7.2 and then CCSP 7.3 ................................................................................... 108   
Post-installation configuration ................................................................................................... 108   
Configure Campaign Manager to run as HTTPS ...................................................................... 114   
Install CCSP UI (7.3) on agent computers ................................................................................. 115

CCSP Security Guide - 7.3 and 7.4   
4   
Appendix A: TLS 1.2 and TLS1.3 configuration ....................................................................... 117   
Overview ................................................................................................................................................. 117   
Access the front-end server after TLS configuration ....................................................................... 117   
Use Remote Desktop Connection............................................................................................... 118   
Use VNC .......................................................................................................................................... 118   
Configure servers and clients for TLS ................................................................................................. 118   
Enable inheritance of TLS settings for .NET 4.x ................................................................................ 120   
Appendix B: Use self-signed certificates ................................................................................. 121   
Generate self-signed certificates ........................................................................................................ 121   
Import certificates to the client ........................................................................................................... 122   
Appendix C: Use HTTPS in Firefox ........................................................................................... 123   
Configure Firefox to use the Windows certificate store .................................................................. 123   
Add exceptions in Firefox for localhost on the ports used by CCSP UI ......................................... 123

CCSP Security Guide - 7.3 and 7.4   
5   
1: Intro   
This document is for the system administrator responsible for setting up a Contact Center: Service   
Provider (CCSP) contact center. This document assumes familiarity with the network architecture,   
the Microsoft Windows 2012 R2 operating system, Microsoft Exchange 2010, and Domain Controller   
2012 R2, and the system configuration and installed software that comprise the CCSP system based   
at a typical ASP site.   
What’s new   
Version 7.4   
•   
3: CCSP component security settings   
o   
Added the dedicated ccspUIUser, for use by the CCSP UI and CCSP UI Admin web services.   
o   
The new security setting for Syslog is reducedSVC.   
o   
Updated the cfmUser, dtrUser, lsUser, and privateVMUser permissions in SQL Server, by   
removing redundant server roles.   
o   
Removed the user mapping from the cfmUser to the msdb database and fixed the   
cfmUser user mapping to the master database.   
•   
Appendix A: TLS 1.2 and TLS1.3 configuration — CCSP supports TLS 1.3 on the client side.   
Definitions   
Terms   
Definitions   
AdminIS   
Admin Interaction Server   
ADP   
Advanced Dial Plan   
AgentWS   
Agent web service   
AIS   
Agent Interaction Server   
APS   
Agent Proxy Server   
CMS   
Call Management Server   
CFM   
Configuration Manager   
DTR   
Data Transport Server   
DPS   
Device Proxy Server   
IMS   
Interactive Media Server

CCSP Security Guide - 7.3 and 7.4   
6   
Terms   
Definitions   
ICS   
Internet Connection Server   
IPS   
Internet Chat Proxy Server   
LCS   
License Control Server   
LS   
Logging Server   
MCS   
Message Connection Server   
MS   
Messaging Server   
PDS   
Predictive Dialing Server   
RtpRelay   
RTP Relay Server   
RTRweb   
Real Time Reports web service   
STAT   
Statistics Server   
VCS   
VoIP Connection Server   
VMS   
Voicemail Server   
voicemailWS   
Voicemail web service   
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 content in this document, please open a case with   
Enghouse Support.

CCSP Security Guide - 7.3 and 7.4   
7   
2: CCSP configuration best practice   
This chapter contains the following information:   
•   
Protect against Microsoft IIS tilde directory enumeration   
•   
Protect against server version disclosure   
•   
Set the customErrors configuration to RemoteOnly   
•   
Protect against clickjacking   
•   
Set the HttpOnly flag on cookies   
•   
Remove obsolete provisioning applications   
•   
CCSP UI Client Services folder must be under the TouchPoint folder   
Protect against Microsoft IIS tilde directory enumeration   
The tilde character (~) with the old DOS 8.3 name convention (SFN) in a HTTP request causes a   
vulnerability that you must protect against. For details, refer to   
https://support.detectify.com/support/solutions/articles/48001048944-microsoft-iis-tilde-  
vulnerability.   
To protect against Microsoft IIS tilde directory enumeration:   
1.   
In the registry, add or change the following key and set the value to 1.   
NtfsDisable8dot3NameCreation to HKLM\SYSTEM\CurrentControlSet\Control\FileSystem   
2.   
In IIS Manager:   
a.   
Click Sites > Default Web Site-> Request Filtering.

CCSP Security Guide - 7.3 and 7.4   
8   
b.   
On the URL tab, right-click and then click Deny Sequence.   
c.   
In the Deny Sequence dialog box, type ~ (tilde).   
   
Changes in the root application apply to all sub applications: TouchPoint, Replay, Dashboard, and   
AgentWS.   
If you cannot change the parent Default Web Site, you must set the following configuration in the   
web.config file of each of the required applications.   
<requestFiltering removeServerHeader="true">   
   
<denyUrlSequences>   
 <add sequence="~" />   
   
 <add sequence="\*" />   
   
 <add sequence=".." />-   
 </denyUrlSequences>   
 </requestFiltering>

CCSP Security Guide - 7.3 and 7.4   
9   
Protect against server version disclosure   
Disclosing IIS version information can help an attacker gain a greater understanding of the systems   
in use and potentially develop further attacks targeted at the specific version of IIS.   
To protect against server version disclosure:   
1.   
In IIS Manager, click the CCSP Website->HTTP Response Headers module and remove X-  
Powered-By.   
   
2.   
Navigate to Server Components\TouchPoint and open the web.config file in a text editor:   
a.   
In the <system.web> section, find httpRuntime and add the following attribute:   
enableVersionHeader="false"   
b.   
In the <system.webServer> section, add the following <security> section:   
<security>   
 <requestFiltering removeServerHeader="true" />   
</security>   
   
The response should now not mention the IIS or ASP version, as in the following example.

CCSP Security Guide - 7.3 and 7.4   
10   
Set the customErrors configuration to RemoteOnly   
To comply with security recommendations, the customErrors configuration has been set to   
RemoteOnly. We recommend that you do not change this setting.   
Protect against clickjacking   
If possible, disable all sites from being displayed in Iframe.   
If you do need to display some sites in Iframe, instead of enabling all, create a white-list of allowed   
domains by setting the Content-Security-Policy property of IIS.   
Note   
•   
For further information on how to configure IIS and the available options, see   
https://www.owasp.org/index.php/Clickjacking\_Defense\_Cheat\_Sheet.   
•   
The white list feature does not work in IE.   
   
The Content-Security-Policy custom response header can be defined on the server, web site, or   
application level of IIS, using the following procedure.   
To allow hosting an application inside Iframes from a specific domain:   
1.   
On all servers that host IIS, in Server Manager, on the Tools menu, click IIS Manager.   
2.   
In the Connections pane, click the server, site or application name.   
3.   
Under IIS, double-click HTTP Response Header.

CCSP Security Guide - 7.3 and 7.4   
11   
4.   
Under Actions, click Add.   
   
5.   
In the Edit Custom HTTP Response Header dialog:   
c.   
Under Name, type Content-Security-Policy.   
d.   
Under Value, enter frame-src and one or more white-listed domains. Separate all values   
in the text field by one space.   
   
6.   
Click OK.

CCSP Security Guide - 7.3 and 7.4   
12   
Set the HttpOnly flag on cookies   
Note   
Before upgrading to CCSP 7.2.9, you must install the HTTP rewrite module on all servers hosting   
IIS, otherwise CCSP 7.2.9 will not function as expected.   
   
To make it impossible to steal cookies via an XSS, you should set the HttpOnly flag on cookies that   
contain sensitive data such as session IDs. JavaScript then returns an empty string when trying to   
read it.   
To set the HttpOnly flag on cookies:   
1.   
On all servers that host IIS, install the URL Rewrite 2.1 component. You can download   
URL Rewrite from https://www.iis.net/downloads/microsoft/url-rewrite   
2.   
Set the cookies with the HttpOnly attribute.   
Remove obsolete provisioning applications   
Note   
Before removing these applications, you must upgrade the Provisioning Web Service to 7.2.9.   
   
The Tenant Provisioning, Agent Provisioning, and ProvisionerWSX applications are no longer used.   
After you upgrade the Provisioning Web Service to 7.2.9, we strongly recommend that you remove   
these applications from the system because they introduce security issues.   
   
You now access both the Tenant and Agent provisioning directly via the Provisioning Web Service   
application, using a URL in the following format: http(s)://<localhost>/ProvisionerWS   
For example, http://ccsp12.pj12.loc/ProvisionerWS

CCSP Security Guide - 7.3 and 7.4   
13   
The landing page has two links:   
•   
Enter Tenant Provisioning   
•   
Enter Agent Provisioning   
   
After you click the link, the functionality remains the same.   
CCSP UI Client Services folder must be under the   
TouchPoint folder   
For security reasons, the Client Services folder can no longer be placed on a remote server. This   
folder must be in its default location, which is under the TouchPoint folder. The CCSP UI server takes   
the configuration only from the default location.   
The ClientServicesDirUrl is now kept in ClientServicesConfig.js, to inform the ClientInstallationService   
about the location of Client Services. It must point to the same CCSP UI server to correlate with the   
retrieved CCSP UI server configuration.   
If ClientServicesDirUrl does not point to the same CCSP UI server, it may result in inconsistent   
configuration. For example, if the current CCSP UI server is configured to work with HTTP while the   
one that the ClientServicesDirUrl points to is configured for HTTPS, CCSP UI and the   
ClientInstallation Service will not be able to communicate.   
The internalClientServicesDirUrl has been removed as it was used only for CCSP UI server   
configuration retrieval and is consequently no longer required.

CCSP Security Guide - 7.3 and 7.4   
14   
3: CCSP component security settings   
This chapter contains the following information:   
•   
Overview   
•   
Configure the alsUser   
•   
Configure the ccspAdminUser   
•   
Configure the ccspUIUser   
•   
Configure the ccspWebUser   
•   
Configure the cfmUser   
•   
Configure the dtrUser   
•   
Configure the ecsUser   
•   
Configure the lsUser   
•   
Configure the msUser   
•   
Configure the pdsUser   
•   
Configure the privateVMUser   
•   
Configure the reducedSVC user   
•   
Configure the replayUser   
•   
Configure CCSP services to log on with reduced users   
•   
Configure CCSP web services to log on with reduced users   
•   
Configure SMTP server to relay outbound emails   
Overview   
To increase the security of the CCSP system environment in the field, the svccosmocall user’s   
domain-wide capabilities have been reduced and service-specific users and accounts have been   
created or adopted for different service classifications.   
After deploying the CCSP 7.x system, databases, services, and web services, you manually configure   
the service-specific users for each service.   
svccosmocall user   
CCSP system deployment was previously based on a superuser named svccosmocall.   
This user has strong system-wide rights and admin rights for every task that a service or operator   
of the system requires, including installation, maintenance, and runtime tasks.   
You still use the svccosmocall account for new deployments and installations. However, after   
completing these activities, you should disable this account until further such activities are   
required.

CCSP Security Guide - 7.3 and 7.4   
15   
If you re-enable svccosmocall for additional deployment tasks, it also requires access to the   
database, so you must also enable its database rights.   
Dedicated users   
The following users and accounts were created or adopted for different service classifications:   
•   
AlsUser   
•   
ccspAdminUser   
•   
ccspUIUser   
•   
ccspWebUser   
•   
cfmUser   
•   
dtrUser   
•   
ecsUser   
•   
lsUser   
•   
msUser   
•   
pdsUser   
•   
privateVMUser   
•   
reducedSVC   
•   
replayUser   
•   
localSystem (existing)   
•   
localService (existing)   
   
The above names represent services. You can use your own naming convention but each renamed   
account must have the exact permissions set out in this document. Each user type can be shared   
among different services, that is, the same user for the same class of service throughout CCSP.   
System services security change mapping   
The following table shows the new security setting for each system service.   
System service   
Current security   
setting   
New security setting   
AdminIS   
svccosmocall   
localService   
Agent List   
svccosmocall   
alsUser   
AIS   
svccosmocall   
reducedSVC   
APS   
svccosmocall   
localSystem   
CFM   
svccosmocall   
cfmUser   
CMS   
svccosmocall   
localSystem

CCSP Security Guide - 7.3 and 7.4   
16   
System service   
Current security   
setting   
New security setting   
DPS   
svccosmocall   
reducedSVC   
DTR   
svccosmocall   
dtrUser   
ECS   
svccosmocall   
ecsUser   
Elsbeth Gate (Adapter)1   
svccosmocall   
localSystem   
ICS   
svccosmocall   
localSystem   
IMS2   
svccosmocall   
localService   
IPS   
svccosmocall   
localSystem   
LCS3   
svccosmocall   
reducedSVC   
LoggingServer   
svccosmocall   
lsUser   
MCS   
svccosmocall   
reducedSVC   
MS   
svccosmocall   
msUser   
PDS   
svccosmocall   
pdsUser   
PerformanceCountersRelayCollector   
svccosmocall   
reducedSVC   
PerformanceCountersRelayTransmitter   
svccosmocall   
reducedSVC   
RtpRelay   
svccosmocall   
localService   
STAT   
svccosmocall   
localSystem   
Syslog   
svccosmocall   
reducedSVC   
VCS   
svccosmocall   
reducedSVC   
VMS   
svccosmocall   
reducedSVC   
1. By default, ECM customer installations and the Elsbeth Gate Service that hosts the CCSP   
Adapter are installed with the local system account. To access network paths, some ECM   
services (for example, Elsbeth Data Manager) can be configured to run with a dedicated user   
account. If there are no special requirements, then we recommend that you run the Elsbeth   
Gate Service with the local system account.   
2. To hear the beep.wav file, on each VCS server you must grant full access to the localService   
user in the security of the folder C:\ProgramData\CosmoCom\CCU\AFM\Landlord.   
3. To enable the LCS to send email notifications, the configured SMTP server must be set to   
allow the reducedSVC account. See Configure SMTP server to relay outbound emails.

CCSP Security Guide - 7.3 and 7.4   
17   
System web services security change mapping   
The following table shows the new security setting for each system web service.   
Web service   
Current security setting   
New security setting   
ADP   
svccosmocall   
ccspWebUser   
AgentWs   
svccosmocall   
ccspAdminUser   
AuthServer   
svccosmocall   
ccspWebUser   
CCSP Admin   
svccosmocall   
ccspAdminUser   
CCSP UI (previously called   
TouchPoint)   
svccosmocall   
ccspUIUser   
CCSP UI Admin (previously   
called AdminPoint)   
svccosmocall   
ccspUIUser   
Chat API (Scripts)   
AppPoolIdentity   
(DefaultAppPool)   
AppPoolIdentity (DefaultAppPool)   
CosmoCorder   
svccosmocall   
replayUser   
CosmoWebServices   
svccosmocall   
localService   
Dashboard   
svccosmocall   
ccspWebUser   
Designer   
svccosmocall   
ccspWebUser   
DialingPlan   
svccosmocall   
localSystem   
PhoneAgent   
svccosmocall   
localSystem   
ProvisionerWS   
svccosmocall   
localSystem   
Replay   
svccosmocall   
replayUser   
ReportedProblems   
svccosmocall   
localSystem   
RTRweb (RTRDll)   
svccosmocall   
localService   
ScreenRecordingServer   
svccosmocall   
localSystem   
voicemailWS   
svccosmocall   
privateVMUser   
WebAdmin API   
svccosmocall   
ccspWebUser   
WebCollaboration   
svccosmocall   
localSystem

CCSP Security Guide - 7.3 and 7.4   
18   
System web virtual directories   
The following CCSP web virtual directories do not need a dedicated user:   
•   
CosmoDesigner   
•   
WebAdministrator   
•   
IVR (optionally defined for tenant's applications scripts)   
Elsbeth web services identity   
Applicable only from Elsbeth 9.x integrations.   
We recommend that you use the same identity for the EPC Agent Gadget application pool and the   
CCSP UI.   
File location changes   
The security changes and creation of different users requires the following changes to file   
locations.   
Logging server cached files   
The LS cached files are usually placed under %temp%, under a folder named LS.<server ID>. For   
example, if the server id of the VCS instance is 7, its folder is LS.7   
Previously, this folder was located in the svccosmocall account area, for example,   
C:\Users\svccosmocall\AppData\Local\Temp\LS.7   
Now, with the different user accounts, the file locations are:   
•   
For built-in accounts like localService or localSystem, the path is C:\Windows\Temp\LS.7   
•   
For newly created users such as dtrUser, the path is C:\Users\dtrUser\AppData\Local\Temp\LS.7   
DTR server cached files   
Previously, the DTR server cached files were located under   
C:\Users\svccosmocall\AppData\Local\Temp\   
   
Now, with the dtrUser, the files are placed in folders under   
 C:\Users\dtrUser\AppData\Local\Temp\

CCSP Security Guide - 7.3 and 7.4   
19   
Registry changes   
To use the G729 codec, on the VCS server, the localService user needs to have full access to the   
following registry key:   
HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows   
NT\CurrentVersion\Drivers32   
   
To create short folder names to access voice files, on the VCS server, the localService user needs to   
have full access to the following registry key:   
HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem   
   
To be able to post messages to the EventViewer, on the database servers, the owner of the CCSP   
databases or the user that runs the Database Services needs to have full access to the following   
registry key:   
HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog

CCSP Security Guide - 7.3 and 7.4   
20   
Configure the alsUser   
After installing the Agent List service, manually add the alsUser.   
To configure the alsUser:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the alsUser and give it domain membership of Domain Users.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the alsUser to the   
list.   
d.   
Double-click Log on as a service. Click Add User or Group and add the alsUser to the   
list.   
e.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each ALS server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the alsUser and give it local administrator rights.   
4.   
Launch SQL Server Management Studio.   
e.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\alsUser.   
f.   
Give the alsUser a database account with the following Server roles enabled:

CCSP Security Guide - 7.3 and 7.4   
21   
g.   
Give the alsUser the following user mappings and permissions on the configuration   
database:   
   
5.   
Activate the alsUser profile in SQL:   
a.   
 Log on to SQL Server with the alsUser and then start SQL Server Management Studio   
(SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the alsUser.   
6.   
Configure each Agent List service to log on as the alsUser user and restart the service. See   
Configure CCSP services to log on with reduced users.

CCSP Security Guide - 7.3 and 7.4   
22   
Configure the ccspAdminUser   
After installing CCSP Admin, manually add the ccspAdminUser.   
To configure the ccspAdminUser:   
1.   
In CCSP Admin, add the CCSPAdminUser to the System Resources > Personnel list.   
2.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Find the ccspAdminUser and give it domain membership of Account   
Operators.   
3.   
On each CCSP server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the ccspAdminUser and give it local administrator   
rights.   
4.   
Add the ccspAdminUser and grant it full permission on CCSPRoot OU and its child objects, as   
follows:   
a.   
Open ADSI Edit.   
b.   
Connect to the Default naming context.   
c.   
Right-click the CCSPRoot node and then click Properties.   
d.   
In the OU=CCSPRoot Properties dialog box, click the Security tab. Add the   
ccspAdminUser, select Full control, and then click Advanced.   
e.   
In the Advanced Security Settings for CCSPRoot dialog box, select the ccspAdminUser   
and then click Edit.

CCSP Security Guide - 7.3 and 7.4   
23   
f.   
In the Permission Entry for CCSPRoot dialog box, in the Applies to list, select This object   
and all descendant objects. Keep all the permissions selected and click OK to apply the   
changes.   
   
5.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the   
ccspAdminUser to the list.   
d.   
Double-click Log on as a service. Click Add User or Group and add the ccspAdminUser   
to the list.   
e.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
6.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\ccspAdminUser.   
b.   
Give the ccspAdminUser a database account with the following Server roles enabled:

CCSP Security Guide - 7.3 and 7.4   
24   
   
c.   
Give the ccspAdminUser the following user mappings on the configuration database   
(CosmoCall):   
   
7.   
Activate the ccspAdminUser profile in SQL:   
a.   
 Log on to SQL Server with the ccspAdminUser and then start SQL Server Management   
Studio (SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the ccspAdminUser.   
8.   
Configure Admin Application Pool (on IIS Management) to run under the ccspAdminUser user   
(by setting the Application Pool Identity) and restart the Application Pool.

CCSP Security Guide - 7.3 and 7.4   
25   
   
9.   
Run the following query against the configuration database to set CCSPAdminUser as   
internal:   
UPDATE [CosmoCall].[dbo].[crPersonnel]   
SET [Internal] = 1   
WHERE Name LIKE 'CCSPAdminUser'

CCSP Security Guide - 7.3 and 7.4   
26   
Configure the ccspUIUser   
After installing the CCSP UI and CCSP UI Admin web services, manually add the ccspUIUser.   
To configure the ccspUIUser:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the ccspUIUser and give it domain membership of Domain Users.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the ccspUIUser to   
the list.   
d.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each CCSP UI or CCSP UI Admin server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the ccspUIUser and give it local administrator rights.

CCSP Security Guide - 7.3 and 7.4   
27   
4.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Databases, expand the configuration database followed   
by Security and then Roles.   
b.   
Right-click Database Roles, and select New Database Role to create a new named   
db\_executor.   
i.   
Select Securables, then search for objects of the type Databases.   
ii.   
Click Grant on the Execute permission for the Configuration Database:

CCSP Security Guide - 7.3 and 7.4   
28   
c.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\ccspUIUser.   
d.   
The ccspUIUser should only be a member of the public SQL Server fixed server role:   
   
e.   
Map the ccspUIUser user to the Configuration Database, and give the user   
db\_datareader, db\_executor, and public database role membership:   
   
5.   
Configure each AdminPoint and TouchPoint Application Pool (on IIS Management) to run   
under ccspUIUser user, by selecting Advanced Settings…   
a.   
Set the Identity to <domain>\ccspUIUser.   
b.   
Set Load User Profile to True.   
c.   
Restart the Application Pool. See Configure CCSP web services to log on with reduced   
users.

CCSP Security Guide - 7.3 and 7.4   
29   
Configure the ccspWebUser   
After installing the Auth, ADP, and WebAdminAPI web services, manually add the ccspWebUser.   
To configure the ccspWebUser:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the ccspWebUser and give it domain membership of Domain Users.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the ccspWebUser   
to the list.   
d.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each Auth, CCSP Admin, ADP, or WebAdminAPI server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the ccspWebUser and give it local administrator rights.   
4.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\ccspWebUser.   
b.   
Give the ccspWebUser a database account with the following Server roles enabled:

CCSP Security Guide - 7.3 and 7.4   
30   
c.   
Give the ccspWebUser the following user mappings and permissions on the   
Configuration database:   
   
5.   
Activate the ccspWebUser profile in SQL:   
a.   
 Log on to SQL Server with the ccspWebUser and then start SQL Server Management   
Studio (SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the ccspWebUser.   
6.   
Configure each Auth, ADP, or WebAdminAPI Application Pool (on IIS Management) to run   
under ccspWebUser user (by setting the Application Pool Identity) and restart the Application   
Pool. See Configure CCSP web services to log on with reduced users.

CCSP Security Guide - 7.3 and 7.4   
31   
Configure the cfmUser   
After installing the Configuration Manager, use CCSP Admin to enable Root CCSP OU mode in the   
CCU Universe parameters. Then verify that the CCSPRoot OU was created in Active Directory. For   
details, see the CCSP Operation Guide.   
Next manually add the cfmUser.   
To configure the cfmUser:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the cfmUser and give it domain membership of Domain Users and   
Account Operators.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the cfmUser to the   
list.   
d.   
Double-click Log on as a service. Click Add User or Group and add the cfmUser to the   
list.   
e.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each CFM server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the cfmUser and give it local administrator rights.   
4.   
Add the cfmUser and grant it full permission on CCSPRoot OU and its child objects, as follows:   
a.   
Open ADSI Edit.   
b.   
Connect to the Default naming context.   
c.   
Right-click the CCSPRoot node and then click Properties.   
d.   
In the OU=CCSPRoot Properties dialog box, click the Security tab. Add the cfmUser ,   
select Full control, and then click Advanced.

CCSP Security Guide - 7.3 and 7.4   
32   
   
e.   
In the Advanced Security Settings for CCSPRoot dialog box, select the cfmUser and   
then click Edit.

CCSP Security Guide - 7.3 and 7.4   
33   
f.   
In the Permission Entry for CCSPRoot dialog box, in the Applies to list, select This object   
and all descendant objects. Keep all the permissions selected and click OK to apply the   
changes.   
   
5.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\cfmUser.   
b.   
The cfmUser should only be a member of the public SQL Server fixed server role.   
   
For RSS, also enable the sysadmin Server role.

CCSP Security Guide - 7.3 and 7.4   
34   
c.   
Give the cfmUser the following user mappings on:   
▪   
Configuration database (CosmoCall):   
   
▪   
System Database - master:   
   
Note   
As an alternative to assigning db\_owner against the master database, you can   
further reduce the user permissions by granting execute permission on the stored   
procedure xp\_logevent in the master database instead.   
To do this, run the following SQL:   
USE [master];   
GRANT EXECUTE ON [master].dbo.xp\_logevent TO [<domain>\cfmUser];   
   
6.   
Activate the cfmUser profile in SQL:   
a.   
 Log on to SQL Server with the cfmUser and then start SQL Server Management Studio   
(SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the cfmUser.

CCSP Security Guide - 7.3 and 7.4   
35   
Configure the dtrUser   
After installing the Data Transport Server, manually add the dtrUser.   
To configure the dtrUser:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the dtrUser and give it domain membership of Domain Users.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the dtrUser to the   
list.   
d.   
Double-click Log on as a service. Click Add User or Group and add the dtrUser to the   
list.   
e.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each DTR server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the dtrUser and give it local administrator rights.   
4.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\dtrUser.   
b.   
The dtrUser should only be a member of the public SQL Server fixed server role:

CCSP Security Guide - 7.3 and 7.4   
36   
c.   
Give the dtrUser the following user mappings on the media database (CosmoMedia):   
   
5.   
Activate the dtrUser profile in SQL:   
a.   
 Log on to SQL Server with the dtrUser and then start SQL Server Management Studio   
(SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the dtrUser.   
6.   
Ensure that the dtrUser has access to C:\Users\dtrUser\AppData\Local\Temp and set the folder   
rights to Full Control for the dtrUser. See System web virtual directories.   
7.   
Configure each Data Transport Server service to log on as the dtrUser user and restart the   
service. See Configure CCSP services to log on with reduced users.

CCSP Security Guide - 7.3 and 7.4   
37   
Configure the ecsUser   
After installing the Configuration Manager, use CCSP Admin to enable Root CCSP OU mode in the   
CCU Universe parameters. Then verify that the CCSPRoot OU was created in Active Directory. For   
details, see the CCSP Operation Guide.   
Next manually add the ecsUser.   
To configure the ecsUser:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the ecsUser and give it domain membership of Domain Users and   
Account Operators.   
2.   
Add the ecsUser and grant it full permission on CCSPRoot OU and its child objects, as follows:   
a.   
Open ADSI Edit.   
b.   
Connect to the Default naming context.   
c.   
Right-click the CCSPRoot node and then click Properties.   
d.   
In the OU=CCSPRoot Properties dialog box, click the Security tab. Add the ecsUser,   
select Full control, and then click Advanced.

CCSP Security Guide - 7.3 and 7.4   
38   
e.   
In the Advanced Security Settings for CCSPRoot dialog box, select the ecsUser and then   
click Edit.   
   
f.   
In the Permission Entry for CCSPRoot dialog box, in the Applies to list, select This object   
and all descendant objects. Keep all the permissions selected and click OK to apply the   
changes.

CCSP Security Guide - 7.3 and 7.4   
39   
   
3.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the ecsUser to the   
list.   
d.   
Double-click Log on as a service. Click Add User or Group and add the ecsUser to the   
list.   
e.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
4.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\ecsUser.   
b.   
Give the ecsUser a database account with the following Server roles enabled:

CCSP Security Guide - 7.3 and 7.4   
40   
c.   
Give the ecsUser the following user mappings on the configuration database   
(CosmoCall):   
   
5.   
Activate the ecsUser profile in SQL:   
a.   
 Log on to SQL Server with the ecsUser and then start SQL Server Management Studio   
(SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the ecsUser.

CCSP Security Guide - 7.3 and 7.4   
41   
Configure the lsUser   
After installing the Logging Server, manually add the lsUser.   
To configure the lsUser:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the lsUser and give it domain membership of Domain Users.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the lsUser to the   
list.   
d.   
Double-click Log on as a service. Click Add User or Group and add the lsUser to the list.   
e.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each LS server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the lsUser and give it local administrator rights.   
4.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\lsUser.   
b.   
The lsUser should only be a member of the public SQL Server fixed server role:

CCSP Security Guide - 7.3 and 7.4   
42   
c.   
Give the lsUser the following user mappings on the system administrator and tenant   
historical databases:   
   
5.   
Activate the lsUser profile in SQL:   
a.   
 Log on to SQL Server with the lsUser and then start SQL Server Management Studio   
(SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the lsUser.   
6.   
Ensure that the lsUser has access to C:\Users\lsUser\AppData\Local\Temp and set the folder   
rights to Full Control for the lsUser. See System web virtual directories.   
7.   
Configure each Logging Server service to log on as the lsUser user and restart the service. See   
Configure CCSP services to log on with reduced users.

CCSP Security Guide - 7.3 and 7.4   
43   
Configure the msUser   
After installing the MS, manually add the msUser user.   
This user is similar to the reducedSVC but is also configured for Exchange accessibility.   
To configure the msUser user:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the msUser and give it domain membership of Domain Users.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the msUser to the   
list.   
d.   
Double-click Log on as a service. Click Add User or Group and add the msUser to the   
list.   
e.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each MS server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the msUser and give it local administrator rights.   
4.   
On each MS server, open an elevated PowerShell console and set the Service Principal Name   
as follows:   
setspn -U -S EngCCSPMS/<FQDN> msUser   
5.   
Replace <FQDN> with the MS Network address defined in the MS Profile.   
6.   
Follow the appropriate step below according to the MSR type used for storing and managing   
CCSP Messaging:   
a.   
For Exchange-MSR: Configure the msUser to work with Exchange as MSR   
b.   
For MSR-DB: Configure the msUser to work with MSR DB   
Configure the msUser to work with Exchange as MSR   
1.   
Create a mailbox for that user and access that mailbox, as explained in Appendix F: Install   
Microsoft Exchange Server 2010 SP3 of the CCSP Prerequisites Guide.   
2.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Microsoft Exchange Security Groups. Make the msUser user a member of the   
following exchange domain groups:   
o   
Public Folder Management   
o   
Server Management   
o   
Organization Management

CCSP Security Guide - 7.3 and 7.4   
44   
3.   
Make sure Exchange EWS certification is installed in Explorer on each machine where MS is   
installed, by accessing https://<Exchange Address>/EWS/Exchange.asmx and installing the   
certifications for it, as follows:   
a.   
In the Certificate dialog box, click Install Certificate.   
b.   
On the Certificate Import Wizard – Welcome page, under Store Location, click Local   
Machine.   
   
Note   
We recommend that you install the certificate on Local Machine rather than Current User   
to ease user replacement and future support in multiple MSR access.   
   
c.   
On the Certificate Import Wizard – Certificate Store page, click Place all certificates in   
the following store, and then click Browse.

CCSP Security Guide - 7.3 and 7.4   
45   
d.   
In the Select Certificate Store dialog box, click Trusted Root Certificate Authorities   
and then click OK.   
   
4.   
Make this user owner in the root of CCSP PublicFolder.   
Note   
Perform this step only in the following cases:   
•   
If CCSP Root folder already exists, for example, after installing and running MS with   
svccosmocall assigned to DefaultTenant.   
•   
After creating the CCSP Root folder with the msUser user.   
5.   
Ensure that each msUser user can access the MSR PublicFolder. On the Exchange machine,   
open Exchange Management Shell and execute the following:   
Get-PublicFolder -Identity "\CCSP 7.1" -Recurse | Add-  
PublicFolderClientPermission -User "msUser" -AccessRights Owner   
6.   
Assign the CCSP Throttling policy for the msUser:   
a.   
Perform the steps described in CCSP Exchange MSR throttling configuration in the CCSP   
Operation Guide.   
b.   
On the Exchange machine, open Exchange Management Shell and execute the following:   
Set-ThrottlingPolicyAssociation -Identity msUser -ThrottlingPolicy   
CCSPThrottlingPolicy

CCSP Security Guide - 7.3 and 7.4   
46   
7.   
Grant the msUser user Send as permission for Exchange Public folders, as follows:   
a.   
Open AD Users and Computers Console. On the View menu, verify that Advanced   
Features is selected.   
   
b.   
Right-click Microsoft Exchange System Objects and then click Properties.   
c.   
In the Microsoft Exchange System Objects Properties dialog box, click the Security tab   
and then click Advanced.

CCSP Security Guide - 7.3 and 7.4   
47   
d.   
In the Advanced Security Settings for Microsoft Exchange System Objects dialog box,   
click Add.   
   
e.   
In the Permission Entry for Microsoft Exchange System Objects dialog box:   
i.   
In Principal, select the msUser user.   
ii.   
In Applies to, select Descendant publicFolder objects.   
iii.   
Clear all the checkboxes except Send as.

CCSP Security Guide - 7.3 and 7.4   
48   
f.   
Ignore the Permission Warning and click Yes.   
   
8.   
Configure the MS service’s properties to log on as the msUser user and restart the service.   
See Configure CCSP services to log on with reduced users.

CCSP Security Guide - 7.3 and 7.4   
49   
Configure the msUser to work with MSR DB   
1.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\msUser.   
b.   
Give the msUser a database account with the following Server roles enabled:   
   
c.   
Give the msUser the following user mappings and permissions on the appropriate tenant   
messaging database (MSR DB):   
   
2.   
Activate the msUser profile in SQL:   
a.   
Log on to SQL Server with the msUser and then start SQL Server Management Studio   
(SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the msUser.   
3.   
Configure the MS service’s properties to log on as the msUser user and restart the service.   
See Configure CCSP services to log on with reduced users.

CCSP Security Guide - 7.3 and 7.4   
50   
Configure the pdsUser   
After installing the Predictive Dialing Server, manually add the pdsUser.   
To configure the pdsUser:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the pdsUser and give it domain membership of Domain Users.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the pdsUser to the   
list.   
d.   
Double-click Log on as a service. Click Add User or Group and add the pdsUser to the   
list.   
e.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each PDS server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the pdsUser and give it local administrator rights.   
4.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\pdsUser.   
b.   
Give the pdsUser a database account with the following Server roles enabled:

CCSP Security Guide - 7.3 and 7.4   
51   
c.   
Give the pdsUser the following user mappings on the sytel Database:   
   
5.   
Activate the pdsUser profile in SQL:   
a.   
 Log on to SQL Server with the pdsUser and then start SQL Server Management Studio   
(SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the pdsUser.   
6.   
Configure the PDS service’s properties to log on as the pdsUser user and restart the service.   
See Configure CCSP services to log on with reduced users.

CCSP Security Guide - 7.3 and 7.4   
52   
Configure the privateVMUser   
After installing the voicemail web service, manually add the privateVMUser.   
To configure the privateVMUser:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the privateVMUser and give it domain membership of Domain   
Users.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the   
privateVMUser to the list.   
d.   
Double-click Log on as a service. Click Add User or Group and add the privateVMUser   
to the list.   
e.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each VoiceMail WS server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the privateVMUser and give it local administrator   
rights.   
4.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\privateVMUser.   
b.   
The privateVMUser should only be a member of the public SQL Server fixed server role:

CCSP Security Guide - 7.3 and 7.4   
53   
c.   
Give the privateVMUser the following user mappings and permissions on the media   
database (CosmoMedia):   
   
5.   
Activate the privateVMUser profile in SQL:   
a.   
 Log on to SQL Server with the privateVMUser and then start SQL Server Management   
Studio (SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the privateVMUser.#   
6.   
Configure each Voicemail Web service to log on as the privateVMUser user and restart the   
service. See Configure CCSP web services to log on with reduced users.

CCSP Security Guide - 7.3 and 7.4   
54   
Configure the reducedSVC user   
After installing the AIS, LCS, VCS, MCS, DPS, PerformanceCountersRelayCollector,   
PerformanceCountersRelayTransmitter, Syslog, and VMS, manually add the reducedSVC user.   
To configure the reducedSVC user:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the reducedSVC and give it domain membership of Domain Users.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the reducedSVC   
user to the list.   
d.   
Double-click Log on as a service. Click Add User or Group and add the reducedSVC user   
to the list.   
e.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each AIS, LCS, VCS, MCS, DPS, PerformanceCountersRelayCollector,   
PerformanceCountersRelayTransmitter, Syslog, and VMS server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the reducedSVC and give it local administrator rights.   
4.   
Configure each service’s properties to log on as the reducedSVC user and restart the service.   
See Configure CCSP services to log on with reduced users.

CCSP Security Guide - 7.3 and 7.4   
55   
Configure the replayUser   
After installing Replay, manually add the replayUser.   
To configure the replayUser:   
1.   
In Active Directory, open Active Directory Users and Computers. Expand your domain and   
then click Users. Create the replayUser and give it domain membership of Domain Users.   
2.   
On your domain controller, open Group Policy Management by running the command   
gpmc.msc.   
a.   
Under your domain, right-click Default Domain Policy and then click Edit.   
b.   
Expand Policies > Windows Settings > Security Settings > Local Policies, and then click   
User Rights Assignment.   
c.   
Double-click Log on as a batch job. Click Add User or Group and add the replayUser to   
the list.   
d.   
Run the command gpupdate /force on the domain controller and any server that will use   
this user.   
3.   
On each Replay server:   
a.   
Log in with the existing administrator account.   
b.   
In the Control Panel, click User Accounts and then click Manage User Accounts.   
c.   
In the User Accounts dialog, add the replayUser and give it local administrator rights.   
4.   
Launch SQL Server Management Studio.   
a.   
Under Object Browser, expand Security, right-click Logins, and create a new login   
named <domain>\replayUser.   
b.   
Give the replayUser a database account with the following Server roles enabled

CCSP Security Guide - 7.3 and 7.4   
56   
c.   
Give the replayUser the following user mappings and permissions on the configuration   
database and the media database:   
▪   
Media database (CosmoMedia):   
   
▪   
Configuration database (CosmoCall):   
   
5.   
Activate the replayUser profile in SQL:   
a.   
 Log on to SQL Server with the replayUser and then start SQL Server Management Studio   
(SSMS).   
b.   
After SSMS starts, connect to the database using the Windows Authentication option,   
which is now under the replayUser.   
6.   
Configure each Replay Application Pool (on IIS Management) to run under replayUser user   
(by setting Application Pool Identity) and restart the Application Pool. See Configure CCSP web   
services to log on with reduced users.

CCSP Security Guide - 7.3 and 7.4   
57   
Configure CCSP services to log on with reduced users   
After adding and configuring the reduced users, you need to change each CCSP service to   
automatically log on with the appropriate reduced user. For the list of services and their reduced   
users, see System services security change mapping.   
To change a service to log on with a reduced user:   
1.   
In Windows Services, right-click the service and then click Properties.   
2.   
Click the Log On tab.   
   
3.   
For localSystem, click Local System account.

CCSP Security Guide - 7.3 and 7.4   
58   
4.   
For the other reduced users:   
a.   
Click This account and then click Browse.   
b.   
In the Select User dialog, click Advanced.   
   
c.   
Click Find Now.   
d.   
Select the appropriate reduced user.   
5.   
Restart the service.

CCSP Security Guide - 7.3 and 7.4   
59   
Configure CCSP web services to log on with reduced users   
After adding and configuring the reduced users, you need to change each CCSP web service in IIS   
to automatically log on with the appropriate reduced user. For the list of web services and their   
reduced users, see System web services security change mapping.   
To change a CCSP web service to log on with a reduced user:   
1.   
In IIS Manager, click Application Pools.   
2.   
Right-click the web service and then click Advanced Settings.   
   
3.   
Select Identity and then click   
.   
4.   
In the Application Pool Identity dialog box:   
o   
For localSystem or localService, click Built-in account and select the user from the list.   
o   
For the other reduced users, click Custom account and then click Set. In the Set   
Credentials dialog box, type the reduced user name and password.   
5.   
Restart the application pool.

CCSP Security Guide - 7.3 and 7.4   
60   
Configure SMTP server to relay outbound emails   
The following CCSP applications need to send emails and require the SMTP relay service to be   
available on their local host:   
•   
Chat Server (ICS)   
•   
Dashboard   
•   
License Control Server (LCS)   
•   
Message Connection Server (MCS)   
•   
Voice Connection Server (VCS)   
   
You must configure the SMTP server to allow inbound communication coming from the above   
applications (or anonymous) and to authenticate to the mail server in use to allow emails to be   
relayed to that mail server.   
The user account or service account used to run the above applications is different from the user   
account used by the SMTP server to authenticate to the mail server.   
The user account used by the SMTP server to authenticate to the mail server must have sufficient   
permissions to send emails as required by the applications. These permissions include:   
•   
“Send As” permission   
•   
“Send on Behalf” permission   
•   
Any other configuration or rule on the mail server that allows the user to send email using   
another email address in the FROM field   
   
Do not configure these applications to directly communicate with the mail server. Instead,   
configure them to communicate with the SMTP server running on localhost, port 25.   
   
To change the SMTP server to authenticate to the mail server using a dedicated user:   
1.   
In IIS Manager, in the navigation pane, expand the local computer name.   
   
2.   
Right-click [SMTP Virtual Server #], and then click Properties.

CCSP Security Guide - 7.3 and 7.4   
61   
3.   
In the [SMTP Virtual Server #] Properties dialog box, click the Delivery tab, and then click   
Outbound Security.   
   
4.   
Select Integrated Windows Authentication.   
5.   
In Account and Password, enter the username and password for an account configured to   
authenticate against the mail server.   
6.   
Click OK.   
   
For example:   
1. The CCSP LCS application runs with reducedSVC user account and needs to send email using   
lcs@enghouse.loc as the FROM email address.   
2. The SMTP relay server is configured to user Integrated Windows Authentication, using the user   
named an\_email\_user (see the above screenshot).   
3. The an\_email\_user has permissions to send emails as lcs@enghouse.loc.   
4. When the LCS applications sends the email to localhost, port 25, the SMTP relay server   
authenticates to the mail server as an\_email\_user and relays the email there.   
5. The mail server accepts and processes the email because an\_email\_user is permitted to   
perform that operation.

CCSP Security Guide - 7.3 and 7.4   
62   
4: CCSP recording encryption   
deployment   
This chapter outlines the procedures and actions required to deploy recording encryption on CCSP.   
It contains the following information:   
•   
Overview   
•   
Key Server installation and configuration   
•   
Certificate generation   
•   
Client configuration   
•   
Admin settings   
•   
Verification   
Overview   
The encryption system uses two types of certificate pairs:   
•   
Encryption/Decryption   
•   
Signing/Verification   
   
An Encryption/Decryption certificate pair is used for each tenant. The public key is used by the DTR   
and performs the encryption. The private key is used by the Key Server and handles the decryption.   
There is a single Signing/Verification certificate pair on a system. The private key is used by the   
Replay server to digitally sign recordings that are passed to the client. The public key is used by the   
KeyServer to validate that the user details in the recording are correct and have not been tampered   
with.   
A Key Server can be hosted and used by multiple tenants or a tenant may host their own private   
Key Server. With a private Key Server, the platform provider never has access to the decryption   
private keys, which makes it impossible for the provider to decrypt the tenant’s recordings after   
they have been processed by the DTR.

CCSP Security Guide - 7.3 and 7.4   
63   
   
Key Server installation and configuration   
The Key Server consists of two parts: the Key Server service and the Key Server web application.   
This design provides a layer of security that keeps the tenant’s private keys secure. Even if the web   
server is hacked, the keys are stored in a separate user account, which is not accessible from the   
account that the web server runs as.   
Prerequisites   
The prerequisites for installing the Key Server are:   
•   
Windows Server 2012 R2   
•   
User with Logon as a service right.   
•   
IIS with ASP.NET   
•   
Application pool defaults set to Enable 32-Bit Applications = False   
•   
HTTPS enabled   
Creating the user   
When creating a user we recommend using SvcKeyServer as the username, but you may choose   
any username. To assign the Logon as a service right, see:   
https://technet.microsoft.com/en-us/library/dn221981.aspx

CCSP Security Guide - 7.3 and 7.4   
64   
Install IIS and ASP.NET   
Note   
If you want to cut and paste commands from this PDF, you must download it and open it in   
Adobe Acrobat.   
   
To install the IIS and ASP.NET use the following PowerShell commands:-   
Import-Module Servermanager   
Add-WindowsFeature Web-Server, Web-Asp-Net45, Net-Framework-45-ASPNET,   
Web-Mgmt-Console   
Configure Application Pool Defaults   
After installing IIS:   
1.   
In IIS Manager, in the Connections pane, click Application Pools.   
2.   
In the Actions pane, click Set Application Pool Defaults.   
3.   
Ensure that Enable 32-Bit Applications = False.   
Enabling HTTPS   
The Key Server rejects any requests that do not come over a secure connection. You must enable   
HTTPS for IIS. The HTTPS connection from the clients must have a full certificate chain from the   
client to the server so that the browser on the clients does not display any certificate warnings.   
The following steps demonstrate how to set up a self-signed certificate, but this method is not   
recommended for production environments.   
To enable HTTPS:   
1.   
Using IIS Manager, select the root node (the server name), then select Server Certificates.   
2.   
On the Actions pane, select Create Self-Signed Certificate.   
3.   
Type a friendly name of your choice, for example, KeyServerHttps.   
4.   
Leave the Personal certificate store selected and click OK.   
5.   
Select the Default Web Site node.   
6.   
On the Actions pane, select Bindings.   
7.   
Add a binding with the following values:   
o   
Type: https   
o   
IP address: All Unassigned   
o   
Port: 443   
o   
Host name: The name that the clients will be using to resolve this server.   
o   
SSL Certificate: Select KeyServerHttp (or the name entered in step 3).

CCSP Security Guide - 7.3 and 7.4   
65   
Install the CCSP Key Server   
To install the CCSP Key Server:   
1.   
Copy the CCSPKeyServer.msi from the 7.3.0 hotfix folder to the server on which it is to be   
installed.   
2.   
Run the CCSPKeyServer.msi.   
3.   
Accept all the defaults.   
4.   
Enter the SvcKeyServer user credentials when prompted.   
Configure the CCSP Key Server   
To configure the CCSP Key Server:   
1.   
Navigate to C:\Program Files (x86)\Enghouse\KeyServer\Web\ and open Web.config in a text   
editor.   
2.   
Edit the settings as required.   
Setting   
Description   
Default value   
MaxContentLength   
The maximum size of the post   
data that is accepted.   
The default does not typically   
need changing.   
1048576   
LogName   
   
The name of the Windows event   
log that events are written to.   
CCSP Key Server   
SignatureAge   
For files that have been   
downloaded for playback, this   
specifies the maximum age (in   
seconds) that a signature is   
considered valid.   
3600 (1 hour)   
SignatureAgeSaved   
For files that have been   
downloaded for saving to disk,   
this specifies the maximum age   
(in seconds) that a signature is   
considered valid.   
7776000 (90 days)   
RateLimitSeconds   
The rate limiter prevents a client   
from attempting brute force   
operations against the server or   
causing a denial of service attack.   
This is the period (in seconds) that   
the RateLimitCount is relative to.   
1

CCSP Security Guide - 7.3 and 7.4   
66   
Setting   
Description   
Default value   
RateLimitCount   
The maximum number of client   
requests allowed in the period   
defined by 'RateLimitSeconds'.   
20   
VerifyExemptKeyHash   
For testing purposes only, this   
allows a particular hash key to   
bypass verification.   
   
Is empty   
VerifyUsers   
Specifies which requests require   
verification:   
•   
0 — bypass all verification   
(only to be used for testing   
purposes)   
•   
1 — allow bypass of   
verification for authenticated   
requests (not recommended)   
•   
4 — all requests must be   
signed (default option)   
4   
CsvAuditFolder   
The directory to write decryption   
audit logs to.   
If the folder does not exist the key   
server attempts to create it.   
%ProgramData%\CCSP Key   
Server   
This typically expands to   
C:\ProgramData\CCSP Key   
Server.   
CsvAuditFilePattern   
The pattern to use for the CSV   
decryption audit log files.   
A new file is created each day at   
the time of the first request of the   
day.   
'Audit' yyyy'-'MM'-'dd'T'HH'-  
'mm'-'ss'.csv'   
This expands to a file name   
such as: Audit 2016-08-01T15-  
45-49.csv   
AuditPlugins   
A list of audit plugins to enable.   
Built in plugins available are:   
•   
KeyServer.CsvFileAudit   
•   
KeyServer.EventLogAudit   
   
Manually install the codec   
The codec required for decrypting the encrypted recordings is automatically installed by the client   
installer of CCSP UI. If CCSP UI is not installed on the client machine, you need to manually install   
the codec.   
To manually install the codec:   
1.   
On a client machine with CCSP UI installed, navigate to C:\Program Files   
(x86)\Enghouse\7\TouchPoint\Client Services and find AudioCryptTransform.dll.

CCSP Security Guide - 7.3 and 7.4   
67   
2.   
Copy the file to the new client machine.   
3.   
On the new client machine, open a command window as administrator and change to the   
directory that contains the file.   
4.   
Run the following command:   
regsvr32.exe AudioCryptTransform.dll   
5.   
Restart the machine.   
Certificate generation   
Caution   
•   
You should keep the keys safe. Whoever has access to the signing key has the potential   
ability to decrypt recordings if they also have access to the Key Server web service and   
recording database.   
•   
Losing the signing key is not a problem because a new key pair can be created and installed   
on all servers.   
Generate the signing certificate   
To generate the signing certificate:   
1.   
Run KeyManagement.exe from the install location. The default is C:\Program Files   
(x86)\Enghouse\KeyServer.   
2.   
Click the Signing Certificate tab.   
3.   
Enter a suitable country code and the name of the platform provider. These can be any value   
and are only used for human identification of the certificates.   
4.   
Optionally enter a password for the certificate. This step adds a small amount of additional   
security.   
5.   
Optionally select when the certificate should expire. The default is 10 years.   
6.   
Click Generate. You are prompted for where to save the .pfx file.   
Install the signing certificate on CosmoCorder servers   
You need to install the signing certificate in the Personal certificate store of the svccosmocall user   
on each CosmoCorder web server. You then need to enter the certificate thumbprint in CCSP   
Admin.   
On each CosmoCorder web server, to install the signing certificate:   
1.   
Log on to the CosmoCorder web server as the CCSP system user, typically svccosmocall.   
2.   
Double-click the signing certificate .pfx file to open the Certificate Import Wizard.   
3.   
Continue through the wizard. When prompted, enter the password for the certificate.   
4.   
When prompted where to install the certificate, choose to place the certificate in the Personal   
certificate store.

CCSP Security Guide - 7.3 and 7.4   
68   
On any CosmoCorder server, to install the signing certificate:   
1.   
Run certmgr.msc.   
2.   
Expand Personal and then click Certificates.   
3.   
Double-click the CosmoCorder signing key.   
4.   
Click the Details tab.   
5.   
Scroll to the bottom of the list and select the thumbprint.   
6.   
Copy the thumbprint value from the frame at the bottom of the window.   
7.   
Log on to CCSP Admin, right-click the root CC Universe node and then click Properties.   
8.   
Paste the value into the Signing Key Thumbprint parameter.   
9.   
If the value starts with a question mark, you must remove the question mark. An example   
value is d4 e0 b9 2a 12 5f fe 3f 52 02 51 bf 25 c4 e2 bd 6b 48 47 57   
Export the verification certificate   
After the signing certificate has been imported to a server, you can export the public key from this   
certificate. You need the public key so it can be imported to the Key Servers. We call this key the   
verification certificate.   
To export the verification certificate:   
1.   
Run certmgr.msc.   
2.   
Expand Personal and then click Certificates.   
3.   
Right-click the CosmoCorder signing key, click All Tasks, and then click Export.   
4.   
In the Certificate Export Wizard, leave the following defaults selected:   
a.   
No, do not export the private key   
b.   
DER encoded binary X.509 (.CER)   
5.   
Select a filename and complete the wizard.   
Import the verification certificate to the Key Server   
To import the verification certificate to the Key Server:   
1.   
Copy the public key (.cer) to each Key Server.   
2.   
Log on to the Key Server as SvcKeyServer.   
3.   
Double-click the .cer verification certificate file.   
4.   
Click Install Certificate.   
5.   
When prompted, select to install the certificate in the Personal certificate store.

CCSP Security Guide - 7.3 and 7.4   
69   
Generate a tenant’s encryption certificate   
Note   
It is important to keep the encryption certificate safe. After recordings have been encrypted, it is   
impossible to play them back without this certificate. Anyone who has access to the private key   
has the potential to decrypt all recordings in the tenant database.   
   
To generate a tenant’s encryption certificate:   
1.   
Run KeyManagement.exe.   
2.   
Click the Encryption Certificate tab.   
3.   
Enter a suitable country code and the name of the platform provider. These can be any value   
and are only used for human identification of the certificates.   
4.   
Enter the tenant’s name.   
5.   
Optionally enter a password for the certificate. This step adds a small amount of additional   
security.   
6.   
Optionally select when the certificate should expire. The default is 10 years.   
7.   
Click Generate. You are prompted for where to save the .pfx file.   
Install a tenant’s encryption certificate   
To install a tenant’s encryption certificate:   
1.   
On the Key Server, install the certificate in the Personal certificate store of the SvcKeyServer   
user.   
2.   
Export the public key (.cer file).   
3.   
On each DTR server, import the public key to the Personal certificate store of the svccosmocall   
user.   
Client configuration   
Note   
To listen to encrypted recordings, the client requires at least Windows 10 with IE 11.   
Set up access to the Key Server via HTTPS   
For security, the Key Server rejects any decryption requests that come on an unencrypted   
connection such as http. You must therefore only use https.   
Test the connection from the client to the Key Server using the following URL:   
https://server/KeyServer/getkeys.aspx   
If you use a self-signed certificate as described earlier in the document, the following instructions   
show how to install the self-signed certificate on each client. We recommend that you use a proper   
certificate chain, rather than installing the self-signed certificate on every client.

CCSP Security Guide - 7.3 and 7.4   
70   
You will receive a security certificate warning. For security reasons, the encryption codec will refuse   
to work with a certificate that the machine does not trust, so the certificate must be installed.   
The following instructions show how to install the certificate using IE 11.   
To install the certificate:   
1.   
Click Continue to this website (Not recommended).   
2.   
In the address bar, click the red shield and then click View Certificates.   
3.   
Click Install Certificate, and then click Next.   
4.   
Click Place all certificates in the following store.   
5.   
Click Trusted Root Certification Authorities.   
6.   
Click Next, and then click Finish.   
7.   
When asked Do you want to install this certificate, click Yes.   
8.   
Close the browser, re-open IE, and navigate back to the KeyServer URL.   
   
The page should now load without any security warning, and show the following message:   
Secure connection from: xx.xx.xx.xx   
Admin settings   
For each tenant set the following values in the Tenant Properties.   
Parameter   
Description   
Example   
Recording Encryption   
Public Key   
The thumbprint of the   
tenant’s public encryption   
key, the value will accept   
spaces or dashes   
included in the value   
98 61 d1 49 21 6d 87 c3 f8 97 b5 55 96 97   
9a 72 87 e9 e0 ce   
Recording Encryption key   
server URL   
The HTTPS URL of the Key   
Server web service   
https://de-  
keyserv/keyserver/getkeys.aspx   
Encryption enabled   
Toggle to enable or   
disable encryption of   
recordings by the DTR   
servers   
Enabled   
Encrypted recording   
restrictions   
Restrictions that apply to   
supervisors when playing   
back encrypted   
recordings   
Playback and save

CCSP Security Guide - 7.3 and 7.4   
71   
Verification   
Encryption   
After making the changes in CCSP Admin, wait for a couple of minutes for the changes to   
propagate to all servers. When the DTR server has received the new configuration, a message such   
as the following will be logged:   
TenantDB::loadConfiguration() TenantId 1013 recordingFormat="2"   
recordingStorage="E:\StereoRecordings" encryptionPublicKey="98 61 d1 49 21 6d   
87 c3 f8 97 b5 55 96 97 9a 72 87 e9 e0 ce" encryptionEnabled\_="true"   
   
When a recording has been processed, a successful encryption includes just two log messages:   
FileEncryptor::Encrypt()src="C:\Users\SVCCOS~1\AppData\Local\Temp\DTR.14\rcd\_2  
056d3040013000d\_2e82e18f.tmp.wav",dst="C:\Users\SVCCOS~1\AppData\Local\Temp\DT  
R.14\rcd\_2056d3040013000d\_2e82e18f.tmp.wav.enc", key="98 61 d1 49 21 6d 87 c3   
f8 97 b5 55 96 97 9a 72 87 e9 e0 ce"   
FileEncryptor::Encrypt() - AudioCrypt\_FileEncrypt success.   
Playback   
After testing if the file is encrypted and calling SetRoles, and before streaming the file to the   
browser, the following success message appears in the log:   
"Writing encrypted file to response"   
   
When the client codec accesses the Key Server to decrypt the wave header, the Key Server event log   
shows a message such as:   
10.116.101.224: Decryption success for call ID 0x21977edf0018000d   
File downloaded by user 3051 from tenant 3008 at 8/11/2016 2:23:19 PM   
Download for: Playback   
   
The information logged includes:   
•   
Call ID of the recording.   
•   
Tenant ID and Supervisor ID of the user that downloaded the file.   
•   
Time that the file was downloaded.   
•   
Time that the playback operation occurred. A downloaded file can be played back multiple   
times.   
•   
The IP address of the machine that attempted to play back the file.   
•   
Whether the decryption was successful along with any failure reason.

CCSP Security Guide - 7.3 and 7.4   
72   
5: Authentication process   
From CCSP 7.4 onwards, most of the client applications (CCSP Admin, Dashboard, Designer, Replay,   
CCSP UI and CCSP UI Admin) use tokens to authenticate users. Tokens are created by the   
Authentication server, and signed using a secret key.   
For details of how to generate and apply the secret key, see the CCSP SSO Secret Key Guide.

CCSP Security Guide - 7.3 and 7.4   
73   
6: CCSP HTTPS configuration   
This chapter describes the procedure to install and configure a CCSP 7.3 system on a Windows   
Server 2012 or 2016 system as HTTPS. It contains the following information:   
•   
Prerequisites   
•   
Configuration and installation   
•   
CCSP components installation   
Prerequisites   
To configure CCSP for HTTPS, you need Publisher installed on a remote server. The latest version of   
Publisher is available from the FTP server.   
You also need a system installed with CCSP 7.3, with the following minimum configuration:   
•   
Windows Server 2012 R2 or 2016 with Domain Controller installed   
•   
Windows Server 2012 R2 or 2016 with Exchange Server installed   
•   
Windows Server 2012 or 2016 CCSP Server hosting Back End components, for example,   
Configuration Database, CFM, AIS, MCS, and MS   
•   
Windows Server 2012 or 2016 machine to be installed with Front End components (for   
example, CCSP UI and Agent Proxy Server)   
o   
IIS web server role – Certificate Mapping Authentication – to be able to create Domain   
certificates on the front-end server IIS   
Configuration and installation   
This section contains the following information:   
•   
Summary   
•   
Before starting   
•   
Configure DC as Root Certification Authority   
•   
Configure the front-end server   
•   
Configure the clients   
Summary   
The configuration steps are as follows:   
1.   
Add Active Directory Certificate Services to DC.   
2.   
Configure Certificate Services on DC for it to act as a Root Certification Authority.   
3.   
In the front-end server IIS, create a Domain certificate with the Root CA on the DC as the   
issuing authority.

CCSP Security Guide - 7.3 and 7.4   
74   
4.   
Create https binding with the above certificate and remove the http binding from Default   
Website.   
5.   
Install the required Front End services from the CCSP 7.1.2.111 installation package.   
6.   
Upgrade the services to CCSP 7.2 using Publisher.   
7.   
Upgrade the services to CCSP 7.3 using Publisher.   
8.   
Install CCSP UI Client Services on the agent machine.   
Before starting   
Ensure that all servers have the latest Windows updates applied.   
If using a Virtual Machine system, take snapshots of the existing DC and back-end server before   
starting the configuration.   
Configure DC as Root Certification Authority   
Adding Active Directory Certificate Services to the DC   
Note   
Development/QA Note: Root Certificates are used instead of just creating a self-signed certificate   
from the server as this is believed to be more representative of customer systems.   
   
To add Active Directory Certificate Services to the DC:   
1.   
Log on to the DC as a domain administrator, typically the svccosmocall user.   
2.   
Open Server Manager, on the Manage menu, click Add Roles and Features.   
   
3.   
If the Before you begin page appears, click Next.   
4.   
On the Select installation type page, ensure that Role-based or feature-based installation   
is selected and click Next.   
5.   
On the Select destination server page, to use this server, click Next.

CCSP Security Guide - 7.3 and 7.4   
75   
6.   
On the Select Server roles page, under Roles, select Active Directory Certificate Services.   
   
7.   
When the Add Roles and Features Wizard appears, click Add Features.   
8.   
On the Select Server roles page, click Next.   
9.   
On the Select Features page, click Next to accept the selected features.   
10. On the Active Directory Certificate Services page, click Next.   
11. On the Select role services page:   
a.   
Ensure that Certification Authority is selected.   
b.   
Select Certificate Authority Web Enrollment.

CCSP Security Guide - 7.3 and 7.4   
76   
12. When the Add Roles and Features Wizard appears, click Add Features.   
13. On the Select role services page, click Next.   
14. On the Web Server (IIS) page, click Next.   
15. On the Select role services page, click Next.   
16. On the Confirm installation selections page:   
17. Ensure that the Restart the destination server automatically if required check box is not   
selected.   
18. Check that the installation parameters are correct.   
   
19. Click Install to start the installation process.   
20. The Installation progress page appears. At the end of the installation, click Configure Active   
Directory Certificate Services on the destination server.

CCSP Security Guide - 7.3 and 7.4   
77   
Configure DC as a Root Certification Authority   
To configure the DC as a Root Certification Authority:   
1.   
In the AD CS Configuration wizard, on the Credentials page, click Next to use the selected   
credentials.   
2.   
On the Role Services page, ensure that Certification Authority and Certification Authority   
Web Enrollment are selected.

CCSP Security Guide - 7.3 and 7.4   
78   
3.   
On the Setup Type page, click Enterprise CA.   
   
4.   
On the CA Type page, click Root CA.

CCSP Security Guide - 7.3 and 7.4   
79   
5.   
On the Private Key page, ensure that Create a new private key is selected.   
   
6.   
On the Cryptography for CA page:   
a.   
Keep the default selection for cryptographic provider.   
b.   
In the Key length list, click 4096.   
c.   
In the Select the hash algorithm for signing certificates issues by this CA list, click   
SHA256.

CCSP Security Guide - 7.3 and 7.4   
80   
7.   
On the CA Name page, ensure that the name of the DC is selected as the Common name.   
   
8.   
On the Validity Period page, change the validity period if required (default is 5 years).

CCSP Security Guide - 7.3 and 7.4   
81   
9.   
On the CA Database page, click Next to use the default locations.   
   
10. On the Confirmation page, check the values are correct, and then click Configure to start the   
configuration.   
   
11. On the Results page, verify that the configuration succeeded, and then click Close.

CCSP Security Guide - 7.3 and 7.4   
82   
Configure the Certificate Revocation List   
To configure the Certificate Revocation List:   
1.   
Open File Explorer and navigate to the folder C:\inetpub\wwwroot.   
2.   
Create a new folder and rename it CertData.   
   
3.   
In Server Manager, on the Tools menu, click Certification Authority. Alternatively, select   
Certification Authority in Administrative tools using the Start Menu or Control Panel.

CCSP Security Guide - 7.3 and 7.4   
83   
4.   
In the certsrv window, right-click the CA name, and then click Properties.   
   
5.   
Click the Extensions tab. Ensure that CRL Distribution Point is the selected extension and   
then click Add.

CCSP Security Guide - 7.3 and 7.4   
84   
6.   
In the Add Location dialog box:   
a.   
In Location, type the following (substitute the FQDN of the DC for <dcname>):   
http://<dcname>/CertData/<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl   
b.   
Click OK.   
   
7.   
On the Extensions tab, select the following check boxes:   
o   
Include in CRLs. Clients use this to find Delta CRL locations   
o   
Include in the CDP extensions of issued certificates

CCSP Security Guide - 7.3 and 7.4   
85   
8.   
Click Apply. In the Certification Authority pop-up dialog restart service request, click No.   
9.   
On the Extensions tab, in the Selected extension list, click Authority Information Access   
(AIA), then click Add.   
   
10. In the Add Location dialog box:   
a.   
In Location, type the following (substitute the FQDN of the DC for <dcname>):   
http://<dcname>/CertData/<ServerDNSName>\_<CaName><CertificateName>.crl   
b.   
Click OK.

CCSP Security Guide - 7.3 and 7.4   
86   
11. On the Extensions tab, select the Include in the AIA extension of issued certificates check   
box.   
   
12. Click Apply. In Certification Authority pop-up dialog restart service request, click Yes. In the   
Properties dialog box, click OK.

CCSP Security Guide - 7.3 and 7.4   
87   
13. In the certsrv window, expand the current server and right-click Revoked Certificates, click   
All Tasks, and then click Publish.   
   
14. In the Publish CRL dialog box, ensure that New CRL is selected and then click OK.

CCSP Security Guide - 7.3 and 7.4   
88   
Exporting the root certificate   
To export the root certificate:   
1.   
In the certsrv window, right-click the CA name and then click Properties.   
2.   
On the General tab, click View Certificate.   
   
3.   
Click the Details tab and then click Copy to File.

CCSP Security Guide - 7.3 and 7.4   
89   
4.   
In the Certificate Export Wizard, click Next.   
5.   
On the Export File Format page, ensure that the DER format is selected.   
   
6.   
On the File to Export page, select an appropriate location on the server and filename.

CCSP Security Guide - 7.3 and 7.4   
90   
7.   
Click Finish to complete the export.

CCSP Security Guide - 7.3 and 7.4   
91   
Configure the front-end server   
Caution   
Certificates for front end servers that host Web Services such as CCSP UI and AgentWS should no   
longer be generated using the IIS facility to request a Domain Certificate. Certificates generated   
in this manner do not contain a Subject Alternative Name field and are therefore not accepted by   
the latest versions of the Chrome and Firefox web browsers.   
Prerequisites   
Before continuing, ensure that the Client Certificate Mapping Authentication role is installed on   
the front-end server.   
   
Also ensure that you have permissions to request certificates from the front-end server:   
1.   
On the CA computer, open an administrator Command Prompt and type the command   
certtmpl.msc.   
2.   
In the Certificate Templates Console dialog box, double click the Web Server template to   
view its properties.

CCSP Security Guide - 7.3 and 7.4   
92   
   
3.   
In the Web Server Properties dialog box, click the Security tab. If Permissions for   
Authenticated Users does not have Allow Enroll ticked, select the check box.   
   
   
4.   
Click OK.

CCSP Security Guide - 7.3 and 7.4   
93   
Requesting a domain certificate   
To request a new certificate for the front-end server from the CA:   
1.   
Log on to the front-end server using a domain administrator, typically the svccosmocall user.   
2.   
Run the Microsoft Management Console from an administrator Command Prompt using the   
command mmc.   
3.   
On the File menu, click Add/Remove Snap-in.   
   
4.   
In the Add or Remove Snap-ins dialog box, click Certificates and then click Add.

CCSP Security Guide - 7.3 and 7.4   
94   
5.   
In the Certificates snap-in dialog box, click Computer account and then click Next.   
   
6.   
In the Select Computer dialog box, ensure that Local computer is selected and then click   
Finish. Click OK in the main dialog box.

CCSP Security Guide - 7.3 and 7.4   
95   
7.   
Expand the Certificates tree, right-click Personal, click All Tasks, and then click Request New   
Certificate.   
   
8.   
In the Certificate Enrollment Wizard, click Next.

CCSP Security Guide - 7.3 and 7.4   
96   
9.   
Ensure that the Active Directory Enrolment Policy is available and selected, and then click   
Next.   
   
10. Select the Web Server template and then click More information is required to enroll for   
this certificate. Click here to configure these settings.   
If the Web Server template is not displayed, select the Show all templates check box. If the   
template is still listed as Unavailable return to the Prerequisites section and enable the Enroll   
permission for your current user.

CCSP Security Guide - 7.3 and 7.4   
97   
   
11. In the Certificate Properties dialog box, under Subject name, select Common name.   
   
12. In Value, type the fully qualified domain name of the server and click Add.

CCSP Security Guide - 7.3 and 7.4   
98   
   
13. Under Alternative name, select DNS in the list.   
   
14. In Value, type the fully qualified domain name of the server and click Add.

CCSP Security Guide - 7.3 and 7.4   
99   
   
15. If further alternative names are required, repeat the previous step.   
16. It is optional and advisable to also add a Friendly name and Description for the certificate. To   
do this, select the General tab and enter the appropriate values.

CCSP Security Guide - 7.3 and 7.4   
100   
17. Click OK.   
18. In the Certificate Enrollment dialog box, click Enroll.   
   
19. Click Finish.

CCSP Security Guide - 7.3 and 7.4   
101   
Enabling HTTPS   
To enable the HTTPS protocol:   
1.   
In Internet Information Services (IIS) Manager, expand the server entry in the tree and   
expand Sites.   
2.   
Right-click Default Web Site and then click Edit Bindings.   
3.   
In Site Bindings, click Add.   
4.   
In the Add Site Binding dialog box:   
a.   
In the Type list, click https.   
b.   
For SSL Certificate, click Select and select the certificate just created. Click OK to close   
the selection dialog.   
c.   
Click OK.   
   
5.   
Restart the IIS server.   
   
Note   
•   
Development/QA note: On a machine cloned from an integration template, open Windows   
Services and find the SQL services. Ensure that the SQL Server Reporting Services is stopped   
and disabled as this uses port 80.   
•   
The root certificate for the CA is automatically distributed to the Trusted Root Certification   
Authority store of the front-end server, so there is no need to import it manually on that   
server.

CCSP Security Guide - 7.3 and 7.4   
102   
Configure the clients   
Install the certificates   
To access services on the front-end server after it has been configured for HTTPS, clients need only   
install the root certificate from the CA server. This is essential when using Remote Desktop   
Connection applications to access the front-end server as it may not be accessible without the   
installed certificate.   
To install the certificates:   
1.   
Copy the certificate file created in Exporting the root certificate to a location on the client   
machine.   
2.   
Run the Microsoft Management Console from an administrator Command Prompt using the   
command mmc.   
3.   
In the File menu, click Add/Remove Snap-in.   
   
4.   
In the Add or Remove Snap-ins dialog box, click Certificates and then click Add.

CCSP Security Guide - 7.3 and 7.4   
103   
5.   
In the Certificates snap-in dialog box, click Computer account and then click Next.   
   
6.   
In the Select Computer dialog box, ensure that Local computer is selected and then click   
Finish. Click OK in the main dialog.

CCSP Security Guide - 7.3 and 7.4   
104   
7.   
Expand the Certificates tree, right-click Trusted Root Certification Authorities, click All   
Tasks, and then click Import.   
   
8.   
In the Certificate Import Wizard, click Next and then browse to the root certificate location on   
the machine.

CCSP Security Guide - 7.3 and 7.4   
105   
9.   
Select the certificate and click Next.   
10. Click Next again to place the certificate in the default location.   
   
11. Click Finish to perform the import.

CCSP Security Guide - 7.3 and 7.4   
106   
CCSP components installation   
This section contains the following information:   
•   
Install server components   
•   
Upgrade to CCSP 7.2 and then CCSP 7.3   
•   
Post-installation configuration   
•   
Configure Campaign Manager to run as HTTPS   
•   
Install CCSP UI (7.3) on agent   
Install server components   
To install the required CCSP 7.1.2 components on the front-end server, follow the relevant section   
for each component in the CCSP Installation Guide -712 to 72x with the following important   
additional instructions.   
Note   
When performing the installation, all URLs that are entered must specify https as the protocol   
unless they refer to components hosted on the back-end server.   
Standalone install components   
HTTPS is currently supported for the following standalone install components:   
•   
ADP – install using the Advanced Dial Plan.msi and select the https binding   
•   
Agent List Server - install using the Agent List Server.msi and select the https binding   
•   
Auth Server – install using the Auth Server.msi and select the https binding   
•   
CCSP Admin – install using the Admin.msi and select the https binding   
•   
CCSP UI – install using TouchPoint.msi   
•   
Replay – install using the Replay.msi and select the https binding   
   
The installation procedure for these components is described in the CCSP Standalone Component   
Installation Guide.   
HTTPS is currently also supported for the following components:   
•   
Agent Proxy Server   
•   
Dashboard   
•   
Historical Reports   
•   
Recording Web Playback   
•   
Screen Recording   
   
The installation procedure for these components is as described in the CCSP Installation Guide -712   
to 72x.

CCSP Security Guide - 7.3 and 7.4   
107   
Install Agent List Server as HTTPS   
Before installing the Agent List Server, you must bind the port that will be used in its Connector URL   
parameter to the certificate for the server. You can do this by running the netsh http add sslcert   
command from an administrator command prompt. Specify the binding as:   
netsh http add sslcert ipport=0.0.0.0:<Connector port number>   
certhash=<certificate hash> appid=<GUID>   
   
The certificate hash is the thumbprint of the certificate in hash form. The appid parameter can be   
any valid GUID that you generate online (for example, using www.guidgenerator.com) because it is   
only used to identify the binding.   
For example, when using the default port of 59071 and a certificate whose thumbprint is 1a f9 8d 89   
9a 22 12 80 90 7c e3 8e 1c 2b 7d 63 43 3b 58 b3, the command is :   
netsh http add sslcert ipport=0.0.0.0:59071   
certhash=1af98d899a221280907ce38e1c2b7d63433b58b3 appid={777CD7A2-2E24-4F98-  
A2D0-AD304FAB0D3E}   
   
When the binding is set, you can install the Agent List Server.msi according to the procedure in the   
CCSP Standalone Component Installation Guide.   
Install CCSP UI   
Install CCSP UI from the TouchPoint.msi. For details, refer to the CCSP Standalone Component   
Installation Guide.   
   
Note   
An error may occur that causes IIS to create the CCSP UI application under a new site called   
Default Web Site\_2 (this may not be visible until the server has been restarted). This website must   
be deleted and the CCSP UI application created in the Default Web Site using   
TouchPointAppPool.

CCSP Security Guide - 7.3 and 7.4   
108   
Upgrade to CCSP 7.2 and then CCSP 7.3   
After completing installation of the required components, follow the instructions for applying the   
CCSP 7.2 Upgrade and then the CCSP 7.3 Upgrade.   
Post-installation configuration   
If an existing CCSP installation is being reconfigured for HTTPS, only the front-end components   
previously listed should be modified.   
CCUniverse URLs   
All the URLS on the CCUniverse tab need to be updated to https.

CCSP Security Guide - 7.3 and 7.4   
109   
Tenant properties URLs   
The following URLs in the Tenant properties need to be updated to https:   
•   
Historical Reports URL   
•   
RTR URL   
•   
CCSP Designer URL   
•   
Replay URL   
•   
Voicemail URL   
CCSP Admin configuration   
To edit the configuration for CCSP Admin to use https:   
1.   
Edit the file Server Components\Admin\configuration.js as administrator.   
1.   
Change the prefix on all URLs from http to https.   
2.   
Save the file.   
APS configuration   
To configure the Agent Proxy Service to use the correct protocol for its ping file:   
1.   
Do one of the following to open the registry editor:   
o   
In an administrator Command Prompt, type regedit   
o   
Right-click the Start Menu and click Run.   
2.   
Navigate to the key   
HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\CCUAgentProxyServer\Parameters.   
3.   
Edit the APSWebPath value and change the prefix to https. If the server name is currently   
localhost, change this to the FQDN of the server to match the name on its certificate.   
4.   
Restart the Agent Proxy Service.   
   
To configure the Agent Proxy Service Event Mechanism to use https:   
1.   
Precede the connector url with https, for example:   
<add key="AgentWS Connector Url" value="https://aps1.pj12.loc:54632" />   
2.   
Specify this port as an https binding in the site bindings configured with the site certificate.

CCSP Security Guide - 7.3 and 7.4   
110   
   
Dashboard configuration   
To configure the Dashboard to use https:   
1.   
Edit the file Server Components\DashboardService\DashboardWS\appsettings.json as   
administrator.   
2.   
Change the prefix on all URLs from http to https.   
3.   
Save the file.   
4.   
Edit the file Server Components\DashboardService\Dashboard\assets\config\app-settings.json as   
administrator.   
5.   
Change the prefix on all URLs from http to https.   
6.   
Save the file.   
   
To configure CosmoDashboard (legacy) to use https:   
1.   
Edit the file Server Components\CosmoDashboard\Source\DashboardEmailService \web.config as   
administrator.   
2.   
Navigate to the section <basicHttpBinding>, and change the security mode to   
"Transport", as follows:   
 <basicHttpBinding>   
 <binding name="bindingMode">   
 <security mode="Transport">   
 <transport clientCredentialType="None"/>   
 </security>   
 </binding>   
 </basicHttpBinding>   
3.   
Edit the file \Server Components\CosmoDashboard\Source\Dashboard.exe.config as administrator.   
4.   
Navigate to the section <basicHttpBinding>, and change the security mode to   
"Transport", as follows:   
 <basicHttpBinding>

CCSP Security Guide - 7.3 and 7.4   
111   
 <binding name="BasicHttpBinding\_IEmail" ... >   
 <readerQuotas maxDepth="32" .../>   
 <security mode="Transport">   
 ...   
 </security>   
 </binding>   
4.   
Edit the file \Server Components\CosmoDashboard\Source\Web.config as administrator.   
5.   
Search for and uncomment the following line:   
<rule name="Add Secure">   
5.   
Republish Dashboard by running \Server Components\CosmoDashboard\   
PublishCosmoDashboard.bat as administrator.   
Historical Reports configuration   
To configure Historical Reports to use https:   
1.   
Open Reporting Services Configuration Manager and connect to your server.   
2.   
On Web Service URL, select your HTTPS Certificate and HTTPS Port.   
   
3.   
Apply the change.

CCSP Security Guide - 7.3 and 7.4   
112   
4.   
On Web Portal URL , click Advanced.   
   
5.   
In Multiple HTTPS identities, click Add.   
   
6.   
Select your HTTPS Port and HTTPS Certificate and save the changes.

CCSP Security Guide - 7.3 and 7.4   
113   
Phone Agent configuration   
To edit the configuration for Phone Agent to use https:   
1.   
Edit the file Server Components\PhoneAgent\web.config as administrator.   
2.   
Change the prefix on all URLs from http to https.   
3.   
Save the file.   
Replay configuration   
To edit the configuration for Replay to use https:   
1.   
Edit the file Server Components\Replay\configuration.js as administrator.   
2.   
Change the prefix on all URLs from http to https.   
3.   
Save the file.   
CCSP UI configuration   
If an existing installation of CCSP UI is present on the front-end server before the server is   
reconfigured for https, the configuration files must be edited for CCSP UI to function.   
To edit the configuration for CCSP UI to use https:   
1.   
Edit the file Server Components\TouchPoint\Ajax\Config.js as administrator.   
2.   
Change the prefix on all URLs from http to https.   
3.   
Save the file.   
4.   
Edit the file Server Components\TouchPoint\ClientServices\ClientServicesConfig.js as administrator.   
5.   
Change the prefix on all URLs from http to https.   
6.   
Save the file.   
7.   
Edit the file Server Components\TouchPoint\ClientServices\ClientServices.xml as administrator.   
8.   
Change the prefix on all URLs from http to https.   
9.   
Save the file.   
10. Edit the file   
Server Components\TouchPoint\ClientServices\ClientInstallationService\   
ClientInstallationService.xml as administrator.   
11. Change the value of the usingHttps element from "False" to "True".   
12. If required due to another application on the target client machines already using port 8001,   
change the value of the wcfHttpsPort element to a new port number.   
13. Change the prefix on the URL from http to https.   
14. Save the file.

CCSP Security Guide - 7.3 and 7.4   
114   
Configure Campaign Manager to run as HTTPS   
To configure Campaign Manager web to run using the secure HTTPS protocol:   
1.   
Obtain a valid SSL certificate. You can create one in IIS 7 by following the steps 1 to 5 described   
at http://www.sslshopper.com/article-how-to-create-a-self-signed-certificate-in-iis-7.html.   
2.   
Export the certificate and the private key into files with names server.crt and server.key. You   
can use any third-party utility, for example, the DigiCert utility from   
https://www.digicert.com/util/.   
3.   
Put the server certificate into WebServer\usr\local\apache2\conf\ssl.crt\server.crt on the CM   
computer.   
4.   
Put the server key into WebServer\usr\local\apache2\conf\ssl.key\server.key on the CM computer.   
5.   
Navigate to WebServer\usr\local\apache2\conf\.   
6.   
In a text editor, open ssl.conf.   
7.   
Find the line 34, and then change DocumentRoot to "C:/Softdial/WebServer/www".   
For Softdial 10.6.936, do the following:   
a.   
Find the line 22, and then change SSLMutex default to Mutex default.   
b.   
Find the line 46, and then change SSLProtocol -all +TLSv1 +SSLv3 to   
SSLProtocol all -SSLv3.   
If you have IIS on the same server that is configured to run using HTTPS, change the listening   
port 443 to another port, for example, 1443.   
8.   
Save the file.   
9.   
Navigate to WebServer\usr\local\apache2\conf\.   
10. In a text editor, open httpd.conf.   
11. Find and uncomment (remove "#" character from) the following line:   
#LoadModule ssl\_module modules/mod\_ssl.so   
For Softdial 10.6.936, in line 202, add the following line:   
LoadModule socache\_shmcb\_module modules/mod\_socache\_shmcb.so   
12. Save the file.   
13. Restart the Apache web server. The Softdial URL is now https://Server:1443/softdial.

CCSP Security Guide - 7.3 and 7.4   
115   
Install CCSP UI (7.3) on agent computers   
This section describes the procedure to follow when using Client Services with CCSP UI (required   
for SIP and Screen Recording) on the agent computer.   
If CCSP UI can only be accessed via HTTPS, the Client Services must also operate under HTTPS   
otherwise the browser will either generate mixed-content warnings or prevent CCSP UI from   
interacting with Client Services altogether.   
   
Note   
You cannot automatically upgrade an HTTP installation of the Client Services to an HTTPS one. If   
Client Services has already been installed on the agent computer as HTTP, it must all be manually   
uninstalled first. This can be done by running the following commands as administrator:   
wmic service where "PathName like '%%Enghouse.CCSP%%'" call StopService   
wmic product where "caption like 'CCSP%%'" call uninstall   
You must also clear the browser cache before attempting to use CCSP UI (see Clearing your   
Browser's Cache in CCSP UI Help).   
   
To install CCSP UI on agent computers:   
1.   
Navigate to the CCSP UI URL.   
2.   
Wait until CCSP UI downloads the CCSPClientInstallationService.msi file and prompts for first   
time installation.   
   
3.   
Open an administrator Command Prompt and navigate to the location of the downloaded file.

CCSP Security Guide - 7.3 and 7.4   
116   
4.   
Run the command as given in the CCSP Client Services installer window but add /l\*v   
ccspinst.log to provide a log file.   
5.   
In the installation wizard, click Next, accept the Licence Agreement and then click Next.   
6.   
On the WCF and Connector URLs page:   
a.   
Ensure that the Use https for WCF and Connector URLs check box is selected.   
b.   
Ensure that the Installation Service WCF Port matches the wcfHttpsPort value in the   
ClientInstallationService.xml file on the server.   
   
7.   
Click Next and then click Install. The Client Services are installed on the agent computer and   
the CCSP UI login screen appears.   
   
Note   
CCSP UI may jump to the login screen without fully installing all the Client Services. If this   
happens, close the browser without logging on, and then restart the   
CCSPClientInstallationService using the following commands from an administrator Command   
Prompt:   
net stop CCSPClientInstallationService   
net start CCSPClientInstallationService   
Open a new browser and navigate to the CCSP UI URL. The Client Services should now be fully   
installed before the login screen appears.

CCSP Security Guide - 7.3 and 7.4   
117   
Appendix A: TLS 1.2 and TLS1.3   
configuration   
This appendix contains the following information:   
•   
Overview   
•   
Access the front-end server after TLS configuration   
•   
Configure servers and clients for TLS   
•   
Enable inheritance of TLS settings for .NET 4.x   
   
Caution   
•   
Exchange 2010 does not support TLS 1.2 or TLS 1.3 for POP/IMAP. This is a limitation of   
Exchange 2010 and is in line with Microsoft’s recommendations at Exchange Server TLS   
guidance Part 3: Turning Off TLS 1.0/1.1.   
•   
If the usage of TLS 1.2 or TLS 1.3 is mandatory, customers may need to add servers   
specifically configured to support TLS 1.0 only for POP/IMAP in their environment.   
Overview   
This appendix describes the procedure to disable the SSL1/2/3 protocols on the front-end server.   
These protocols must be explicitly disabled for the front-end server to be accessible using only the   
TLS 1.2 or TLS 1.3 protocol.   
For increased security, we recommend also disabling older and weaker protocols and cyphers on   
client computers. CCSP Client services are supported when using TLS 1.2 or TLS 1.3 as the only   
enabled protocol.   
For the proper functioning of various client services features (for example, screen recording and   
client tray) additional configuration might be required on client computers to ensure that .NET   
Framework 4.x uses the system wide settings for TLS. See Enable inheritance of TLS settings for   
.NET 4.x.   
However other applications installed on client computers might not work properly when using only   
TLS 1.2 or TLS 1.3. Consult with your security provider and fully test before any system-wide   
changes.   
Access the front-end server after TLS configuration   
This section applies to earlier versions of Windows. Remote Desktop Protocol (RDP) 8.1 was   
introduced in Windows Server 2012 R2 and Windows 8.1 and was later made available to Windows   
Server 2008 R2 and Windows 7 via updates.   
If you’re running RDP versions older than 8.1 you will need to keep TLS 1.0 enabled.

CCSP Security Guide - 7.3 and 7.4   
118   
   
Use Remote Desktop Connection   
If TLS 1.0 is disabled, connection to the server will not be possible from a Windows 7 client using   
RDP unless the RDP Client uses RDP Protocol 8.1 or above. Earlier versions require TLS 1.0 to be   
enabled on the server to connect. Information regarding the installation of this update is available   
at https://support.microsoft.com/en-us/kb/2830477.   
If TLS 1.0 is not disabled, then any version of RDP can be used.   
Use VNC   
If for any reason, you cannot use the latest RDP Protocol on a connecting client, before configuring   
the front-end server for TLS 1.2 or TLS 1.3, we highly recommended that you install a VNC server on   
the server such as RealVNC (https://www.realvnc.com/) or TightVNC (http://www.tightvnc.com/).   
For TightVNC, the Server component is installed on the front-end server and the Viewer component   
is installed on the connecting client.   
Configure servers and clients for TLS   
Configuring a server or client for TLS can be done by changing the registry settings and restarting   
the machine. However, there is a freely available tool that can perform this task called IIS Crypto,   
which is available at https://www.nartac.com/Products/IISCrypto. It does not require installation.   
There is also a CLI version of the tool for reconfiguring multiple computers.   
To configure the servers and clients for TLS:   
1.   
Download the .GUI version of the IIS Crypto Tool to a location on the computer you want to   
reconfigure..   
2.   
Open the IIS Crypto Tool and click Yes at the UAC prompt if requested.   
3.   
Accept the licensing conditions.   
4.   
When IIS Crypto appears, it shows the default settings, as shown below.

CCSP Security Guide - 7.3 and 7.4   
119   
   
5.   
Click Best Practices and clear the TLS 1.0 and TLS 1.1 check boxes from both the Server   
Protocols and Client Protocols sections, as shown below.   
   
6.   
Click OK to acknowledge the restart required warning, and close IIS Crypto.   
7.   
To complete the configuration, restart the server.

CCSP Security Guide - 7.3 and 7.4   
120   
Enable inheritance of TLS settings for .NET 4.x   
For the proper functioning of various client services features (for example, screen recording and   
client tray), it’s necessary to ensure that .NET Framework 4.x uses the system wide settings for TLS.   
This can be achieved by modifying the registry.   
Details about these settings and other security and cryptography related information can be found   
in the following articles: How to enable Transport Layer Security (TLS) 1.2 on clients - Configuration   
Manager | Microsoft Learn and Transport Layer Security (TLS) best practices with the .NET   
Framework - .NET Framework | Microsoft Learn.   
The SystemDefaultTlsVersions registry value defines which security protocol version defaults are   
used by .NET Framework 4.x. If the value is set to 1, then .NET Framework 4.x inherits its defaults   
from the Windows channel. The SchUseStrongCrypto registry setting disables the RC4 stream   
cipher and requires a restart.   
To create a .reg file that can be inserted in Windows Registry:   
1.   
In Notepad, create a text file named NET4X-UseSchannelDefaults.reg.   
2.   
Copy, and then paste the following text into the file:   
Windows Registry Editor Version 5.00   
[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]   
"SystemDefaultTlsVersions"=dword:00000001   
"SchUseStrongCrypto" = dword:00000001   
[HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v4.0.3031  
9]   
"SystemDefaultTlsVersions"=dword:00000001   
"SchUseStrongCrypto" = dword:00000001   
3.   
Save the NET4X-UseSchannelDefaults.reg file.   
4.   
Double-click the NET4X-UseSchannelDefaults.reg file.   
5.   
Click Yes to update your Windows Registry with these changes.   
6.   
Restart your computer for the change to take effect.

CCSP Security Guide - 7.3 and 7.4   
121   
Appendix B: Use self-signed certificates   
We recommend the use of certificates issued by a Certification Authority as only the root certificate   
then needs to be installed on client machines.   
If this is not possible, you can use self-signed certificates for the front-end servers instead.   
Caution   
Do not use the Internet Information Service manager to generate the self-signed certificates as   
they are not accepted by the latest versions of Chrome and Firefox.   
Generate self-signed certificates   
To generate self-signed certificates:   
1.   
On the font-end server, open an admin PowerShell and enter the following command:   
New-SelfSignedCertificate -DnsName <FQDN of server> -CertStoreLocation   
"Cert:\LocalMachine\My"   
Note   
You can specify more than one value for DnsName. Use commas to separate the list, for   
example, DnsName serverID1.com,serverID2.com   
   
2.   
Run the Microsoft Management Console from an administrator Command Prompt using the   
command mmc.   
3.   
On the File menu, click Add/Remove Snap-in and load the snap-in for Local Computer.   
4.   
Go to the Personal store, Certificates folder, right-click on the new certificate and then click   
Export.   
5.   
In the Certificate Export Wizard, click Next.   
6.   
Ensure that No, do not export the private key is selected then click Next. Click Next again to   
use the default encryption.   
7.   
Enter or select a suitable location for the file, click Next, and then click Finish to export the   
certificate.   
8.   
The certificate also needs to be imported to the Trusted Root Certification Authorities Store on   
the front-end server. To do this, copy the certificate from the Personal Certificates folder and   
paste it into the Certificates folder under Trusted Root Certification Authorities.

CCSP Security Guide - 7.3 and 7.4   
122   
Import certificates to the client   
To import certificates to the client:   
1.   
Copy the exported certificate from the front-end server to the client machine.   
2.   
Do one of the following:   
o   
Import the certificate into the Local Computer Trusted Root Certification Authorities store,   
following the procedure for Install the certificates.   
o   
Double-click the certificate in File Explorer and then click Install Certificate. Ensure that   
the Local Machine option is selected on the first screen of the Certificate Import Wizard.

CCSP Security Guide - 7.3 and 7.4   
123   
Appendix C: Use HTTPS in Firefox   
This appendix contains the following information:   
•   
Configure Firefox to use the Windows certificate store   
•   
Add exceptions in Firefox for localhost on the ports used by CCSP UI   
   
Firefox has an issue with the self-signed certificates used in CCSP UI. To get round this, you need to   
configure Firefox to use the Windows certificate store (recommended) or add exceptions in Firefox   
for localhost on the ports used by CCSP UI.   
Configure Firefox to use the Windows certificate store   
To configure Firefox to use the Windows certificate store:   
1.   
Open Firefox.   
2.   
Go to the URL about:config.   
3.   
Acknowledge the warning.   
4.   
Search for the setting security.enterprise\_roots.enabled.   
5.   
Set it to true.   
Add exceptions in Firefox for localhost on the ports used   
by CCSP UI   
Note   
These instructions assume that the customer is using the default port numbers that CCSP UI is   
installed with. If not, please change accordingly. Although you could add a single exception for   
localhost, we recommend that you use the specific port numbers.   
   
To add exceptions in Firefox for localhost on the ports used by CCSP UI:   
1.   
On the Firefox menu, click Options.

CCSP Security Guide - 7.3 and 7.4   
124   
2.   
Click the Advanced tab and then click Certificates.   
   
3.   
Click View Certificates and then click the Servers tab.

CCSP Security Guide - 7.3 and 7.4   
125   
4.   
On the Servers tab:   
a.   
Click Add Exception.   
b.   
In Location, type https://localhost:8001   
   
   
c.   
Click Get Certificate.   
d.   
Click Confirm Security Exception.   
5.   
Repeat step 4 to add the other three localhost ports:   
o   
https://localhost:49071   
o   
https://localhost:49073   
o   
https://localhost:49074   
6.   
In the Certificate Manager dialog box, click OK.   
7.   
Restart Firefox and navigate to the CCSP UI URL.   
8.   
If you get a message prompting you to confirm the use of ClientApplicationStarter, select the   
check box to remember the action and then start the system tray application.   
9.   
If you see a long running script error when attempting to log on to CCSP UI, close the browser   
window and try again. If CCSP UI says that you are already logged on, click OK to re-logon. It   
should work the second time round.

# Contact Center: Service Provider Replay Help Version 7.4

Contents  
1: Intro  
4  
What's new  
4  
Legal disclaimer  
4  
Support  
4  
2: Overview  
5  
Configure your browser  
5  
Log in  
6  
Password expired message  
6  
Additional login messages  
7  
Replay page  
8  
Matching segments logic  
9  
Matching segments  
9  
Matching segments and permissions  
10  
Screen recording playback expected result  
11  
One screen playback segment  
11  
Two screen playback segments  
14  
Three screen playback segments  
16  
Log out  
17  
Additional logout messages  
17  
Session expiry  
17  
3: Recordings  
18  
Permissions  
18  
Call list  
18  
Download location  
19  
Search for calls  
20  
Filter calls by agent name and/or call type  
20  
Filter calls by date and time  
21  
Filter calls by ANI, DNIS, and/or message content  
21  
View only agents with calls in the specified time range  
21  
Media player fields  
22  
Play audio recordings  
24  
Play encrypted audio recordings  
25  
Display chat transcripts  
27  
Display email previews  
28  
Play video recordings  
29  
Play voicemail recordings  
30  
4: Call scenarios  
31  
Call scenarios overview  
32  
Recorded with screen  
34  
CCSP Replay Help - 7.4  
2

Inbound voice call recorded with screen  
34  
Inbound voice call recorded with screen with 2 call holds  
35  
Inbound voice call recorded with screen manually recorded by agent  
36  
Direct consultation with 2nd agent (A2A)  
37  
Inbound voice call, direct consultation with 2nd agent, return to caller  
37  
Inbound voice call, direct consultation with 2nd agent, transfer to 2nd agent  
39  
Inbound voice call, direct consultation with 2nd agent (A2A), conference with 2nd agent and caller,  
one agent drops from conference  
41  
Consultation via entry point (TTS)  
44  
Inbound voice call, consultation with agent via entry point (TTS), return to caller  
44  
Inbound voice call, consultation with 2nd agent via entry point (TTS), transfer to 2nd agent  
46  
Direct consultation and conference  
48  
Inbound voice call, direct consultation with 2nd agent (A2A), conference with 2nd agent and caller,  
one agent drops from conference  
49  
Transfer to a non-attendable entry point  
52  
Inbound voice call, transfer to non-attendable entry point, 2nd agent picks up  
52  
5: Archiver Playback  
54  
Off mode  
54  
Online mode  
55  
Offline mode  
56  
On-premise mode  
58  
CCSP Replay Help - 7.4  
3

1: Intro  
This Help is for CCSP supervisors who view call details and play back media and screen recordings of agent-  
caller interactions.  
What's new  
Version 7.4  
Log in and Log out — added process for CCSP single sign-on. Requires HF-61362.  
Recordings — added process to select where to download recordings.  
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.  
CCSP Replay Help - 7.4  
4

2: Overview  
This section contains the following topics:  
l  
Configure your browser  
l  
Log in  
l  
Replay page  
l  
Matching segments logic  
l  
Screen recording playback expected result  
l  
Log out  
Replay is a standalone product that works in conjunction with the CCSP system. It provides a complete record  
of all agent-caller interactions.  
You need a browser to access the call detail record. The call detail record includes the recording, chat history  
for chat call, email, voice mail, agent, caller information, and CCSP UI (previously called TouchPoint) screens.  
A media player is embedded in the web page to play the recording. The HTML5 player starts playing the video  
immediately while downloading in the background.  
Replay enables contact center supervisors to view call details and play back media and screen recordings of  
agent-caller interactions. The calls can be telephone, web chat, VoIP, email, or voice messages and responses.  
Recorded information includes audio, chat, email, and agent screens. Replay stores all sessions automatically in  
a centrally located database.  
The information available for particular calls depends on the media used during the call.  
Agent screens are available only if the system administrator enabled this feature for the contact center and if  
the agents are licensed to use that feature. Even if individual agents are enabled to activate recording, agents  
may not be aware that their screen selections are included. If this feature is enabled, screen captures are  
included together with all other session data in agent recordings. Screen captures are also included in any  
sessions recorded automatically by scripts.  
Configure your browser  
Before running Replay from your browser, some initial preparation is required:  
l  
For Microsoft Edge, the tenant administrator must enable localhost access on your computer.  
l  
For Ubuntu client, you need to install a codec for MP4.  
l  
For Chrome and Firefox, no specific settings are required.  
CCSP Replay Help - 7.4  
5

Log in  
The log in process is slightly different depending on the identity provider configured by your administrator.  
Note  
l  
To log in to Replay, you need the Supervisor permission.  
l  
Before starting Replay, see Configure your browser.  
l  
If your administrator has enabled single sign-on and you are logged in to CCSP UI (previously called  
TouchPoint), to start Replay, click  
Replay in the gadgetory. You do not need to type your user ID  
and password again.  
l  
If you log in again after your CCSP 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.  
2.  
Click Go or press Enter. The Welcome to Replay page appears.  
3.  
Click Log in. If you are already logged in, the main Replay page appears.  
4.  
If you are not already logged in, the Auth Server Login Selector page appears.  
5.  
Type your User name in User Principal Name (UPN) format, for example, someone@domainname.com,  
and click Continue.  
6.  
The next step depends on the identity provider configured by your administrator:  
o  
For Auth Server, the AuthServer login page appears. Type your Password and click Log in.  
Alternatively, to change the username, click Back to return to the previous page. The main Replay page  
appears. If necessary, see Password expired message below.  
o  
For ADFS, you are redirected to the ADFS login page to authenticate. Please consult your local  
procedures for how to proceed from this point.  
o  
For Enghouse Identity Server (EIS), you are redirected to the EIS login page. Type your Password and  
click Login. The main Replay page appears.  
7.  
If necessary, see Additional login messages below.  
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 Auth Server login page, the Password  
expired notification appears.  
1.  
In the notification, click Change Password. The Change password dialog box 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.  
CCSP Replay Help - 7.4  
6

Additional login messages  
The following messages may appear when you try to log in:  
l  
Failed to get authorization for Replay use. Your user profile is not permitted to use the Replay application. Only  
supervisors are permitted to use the Replay application.  
l  
You are not authorized to view this page. This can occur because:  
o  
You are logged in to an external identity provider with a different username than the one you entered  
on the Auth Server Login Selector page to log in to the CCSP client application.  
o  
You entered a username on the Auth Server Login Selector page and were redirected to an external  
identity provider, where you log on with a different username.  
l  
Auth Server is not available or there is a misconfiguration. You must contact your administrator. You will not be  
able to log in to Replay until the issue is resolved.  
CCSP Replay Help - 7.4  
7

Replay page  
On the Replay page, you can search for calls and play or display their recordings.  
The buttons that appear on the right of the call list indicate the recording types that you can view.  
Button  
Action  
Description  
Play video recording  
A video of the agent screens during the call is available.  
Play audio recording or  
play voicemail recording  
A recording of the voice call or voicemail is available.  
Display email preview or  
display chat transcript  
A recording of the email preview or chat transcript is  
available.  
CCSP Replay Help - 7.4  
8

Matching segments logic  
Matching segments  
Matching segments and permissions  
Screens and media are recorded separately because screens are recorded by the agent machines and media  
are recorded by the servers. The length of their individual segments is arbitrary, depending on scripts and  
when agents start and stop the recordings.  
Matching segments  
To find screen recording segments and their corresponding audio recording segments, Replay uses an  
algorithm that takes the timestamps of the screen recording and searches for the corresponding audio  
timestamps with the following condition:  
Screen.Start <= Audio.End AND Screen.End >= Audio.Start  
Note  
Screen recordings made from version 7.3 onwards include the time difference between the agent machine  
and the TouchPoint server at the time that the recording was started. This is known as the client time offset  
and is the difference of the client time relative to the server time. For example, if the client machine is  
running 20 seconds behind the TouchPoint server, the client time offset is -20. If it is ahead by 14 seconds, it  
is 14. This is applied to the Screen.Start time and Screen.End time in the calculation above, to account for  
the machine clocks not being synchronized.  
In the following example, the audio segments that correspond with Screen1 are Audio1 and Audio2.  
CCSP Replay Help - 7.4  
9

In the following example, the audio segments that correspond with Screen2 are Audio2 and Audio3.  
Matching segments and permissions  
You can view recordings only for the agents in the groups and teams that you have permission to see.  
However, when an agent transfers a call to another agent or conferences a call with another agent, you may  
not have permission to view recordings for the other agents.  
Transferred and conferenced calls  
For transferred and conferenced calls, if you have permission for Group 1 but not Group 2, and an agent (Agent  
1) in Group 1 transfers or conferences a call to an agent (Agent 2) in Group 2, you can only see the segments  
that the agent in Group 1 participated in.  
CCSP Replay Help - 7.4  
10

Screen recording playback expected result  
One screen playback segment  
Two screen playback segments  
Three screen playback segments  
Merging takes the start of the screen segment and the end of the last corresponding audio segment, even if  
the screen segment has already finished.  
The following examples show the expected result when displaying a screen recording merged with the  
corresponding audio segments.  
One screen playback segment  
In the following example, the screen recording starts before the audio recording.  
CCSP Replay Help - 7.4  
11

In the following example, the recording starts after the call start.  
CCSP Replay Help - 7.4  
12

In the following example, the audio recording starts before the screen recording.  
In the following example, there is one screen recording with two audio recordings.  
CCSP Replay Help - 7.4  
13

Two screen playback segments  
In the following example, there are two screen recordings with one audio recording.  
The Playback Segment line shows the two segments to expect when watching the screen recording.  
CCSP Replay Help - 7.4  
14

In the Media Player, Playback Segment 1 and 2 appear individually as indicated below.  
In the following example, there are two screen recordings with two audio recordings.  
CCSP Replay Help - 7.4  
15

Three screen playback segments  
In the following example, there are three screen recordings with one audio recording.  
The Playback Segment line shows the three segments to expect when watching the screen recording.  
CCSP Replay Help - 7.4  
16

Log out  
Note  
If you are using an external identity provider such as ADFS or EIS, when you click Log out of your CCSP  
session, you are also logged out of your external identity provider.  
1.  
At the top right of main Replay page, click Logout.  
2.  
Do one of the following:  
o  
If you want to return to Replay, click Return to Login page.  
o  
If you have finished working with CCSP, click Log out of your CCSP session.  
Additional logout messages  
When you click Log out of your CCSP session, the following message may appear:  
l  
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 CCSP session again.  
The application will return to the login page. Depending on the problem resolution, this may take slightly  
longer than normal.  
Session expiry  
Your CCSP application session will become invalid when:  
l  
The login session time exceeds the maximum CCSP login session lifetime specified by your administrator.  
l  
You log out of your CCSP session using another application.  
l  
You clear all cookies in the browser.  
When your CCSP 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 CCSP login session lifetime, you only have to enter your username on the Auth Server Login  
Selector page. You will then automatically be authenticated against your external identity provider.  
CCSP Replay Help - 7.4  
17

3: Recordings  
This section contains the following topics:  
l  
Search for calls  
l  
Media player fields  
l  
Play audio recordings  
l  
Play encrypted audio recordings  
l  
Display chat transcripts  
l  
Display email previews  
l  
Play video recordings  
l  
Play voicemail recordings  
Permissions  
You can view recordings for agents that you have permission to see. For example, if you have permission to see  
recordings of the agents from only Group 1, you cannot view recordings of calls after they are transferred to  
agents in Group 2. See Matching segments and permissions.  
Likewise, you can copy the URL of a recording from your browser and send it to another supervisor to check,  
but to view the recording, the other supervisor must have the permission to view it when logged on.  
Call list  
The call list is a list of media recordings by call type and agent.  
When you click on a media recording in the row, Replay plays the first media segment for that agent and call  
type for that specific call ID.  
In the media player, the segment numbers correspond to the row in the call list, unless there are extra  
segments created either by call hold/unhold or the agent manually stopping and starting recording of the call  
and/or the screen.  
Each media row may have zero, one or more screen segments associated with it. When you click on a screen  
recording in the row, Replay plays the first screen segment for the agent that is associated with that media for  
that specific call ID.  
For preview/predictive campaigns, n/a appears in the Queue column.  
The times are presented in local time.  
CCSP Replay Help - 7.4  
18

Download location  
To select where to download recordings, you need to configure your browser as follows:  
l  
Chrome — on the top right, click the three dots, click Settings, click Downloads, and turn on Ask where to  
save each file before downloading.  
l  
Edge — on the top right, click the three dots, click Settings, click Downloads, and turn on Ask me what to  
do with each download.  
l  
Firefox — on the top right, click the three lines, click Settings, and search for Always ask you where to save  
files. In the Search Results, select the Always ask you where to save files check box.  
CCSP Replay Help - 7.4  
19

Search for calls  
Filter calls by agent name and/or call type  
Filter calls by date and time  
Filter calls by ANI, DNIS, and/or message content  
View only agents with calls in the specified time range  
On the Replay page, you can filter calls by agent name, call type, date, and time. You can also expand the  
search pane and filter calls by ANI, DNIS, or message content.  
In the call list, you can click a column heading to resort the list. Each time you sort by a column heading, you  
update the list of calls.  
The call list shows 50 agent names per page, sorted by call start date and time. It shows all agents that are  
visible to the logged on supervisor, even if they have no recordings.  
Tenant-wide supervisors can select to view only agents with calls in the specified time range.  
Filter calls by agent name and/or call type  
To filter calls by agent name and/or call type:  
l  
In Agent Name and/or Call Type, click the down arrow and do one of the following:  
o  
Type letters contained in the item. The list shows items that start with the letters and contain them.  
o  
Click one or more items in the list. The number of selected items appears in the filter.  
CCSP Replay Help - 7.4  
20

Filter calls by date and time  
Note  
l  
The times are presented in local time.  
l  
The recording may start slightly after the call start time, for example, when an agent clicks record after  
starting the call.  
To filter calls by date and time:  
l  
In Start Date and/or End Date, click in the field and do one or more of the following:  
o  
Click the date.  
o  
To add the time, click  
Select Time.  
To leave the date filter blank, click in the field and then click  
Clear Selection.  
Filter calls by ANI, DNIS, and/or message content  
To filter calls by ANI, DNIS, or message content:  
1.  
Click  
to expand the search pane.  
2.  
In ANI and/or DNIS, type numbers contained in the item. The list shows items that start with the numbers  
and contain them.  
3.  
In Message Contains, type letters contained in the item. The list shows items that start with the letters and  
contain them.  
View only agents with calls in the specified time range  
Tenant-wide supervisors only.  
To select to view only agents with calls in the specified time range:  
1.  
Click  
to expand the search pane.  
2.  
Select the Only Agents With Calls in specified time range check box. The list shows only agents that  
made calls in the date range, even if they have since been deleted from the system.  
CCSP Replay Help - 7.4  
21

Media player fields  
To open the appropriate media player, click on either the screen or media icon of a row in the list.  
Note  
If the message The media playback was stopped due to a corruption issue or unsupported features appears,  
please contact your administrator.  
All media players show the same fields. The recording and call types determine which fields have a value.  
Field  
Description  
Call ID  
Present for all calls. Serial number of the recording segment’s owning call in hexadecimal  
format.  
Agent Name  
Present for all calls. Full name of the agent to which the recording segment belongs.  
Agent ID  
Present for all calls. Login ID of the agent to which the recording segment belongs.  
ANI  
Depending upon the call type, this can be:  
l  
Blank for chat calls  
l  
The number or email address of the external caller for inbound voice, and message calls  
l  
The name of the tenant for outbound or agent to agent voice  
DNIS  
Depending upon the call type, this can be:  
l  
Blank for chat and outbound voice  
l  
The call center number for inbound voice  
l  
The call center folder address for message  
l  
The destination agent for agent to agent  
l  
"Transfer to system" for the inbound (receiving agent) voice leg  
Caller Name  
Always defined for chat calls.  
Defined for other types of call according to the scripts.  
Call Type  
Type of call recording. One of the following:  
l  
Agent to agent  
l  
Campaign  
l  
Chat  
l  
Conference  
l  
Inbound  
l  
IVR recording  
l  
Message  
CCSP Replay Help - 7.4  
22

Field  
Description  
l  
Outbound  
l  
Voicemail  
l  
Transfer from script  
l  
Screen only  
Queue  
ACD queue of the inbound call. Not defined for agent to agent transferred or conferenced  
calls.  
Skill  
Skill attached to the inbound call. Not defined for agent to agent transferred or conferenced  
calls.  
Message  
The caller message field of the call if defined.  
Call Start  
Start time of the recording segment. If multiple segments have been merged, this is the  
start time of the first segment in that sequence.  
Call End  
End time of the recording segment. If multiple segments have been merged, this is the end  
time of the last segment in that sequence.  
Note  
l  
The call start and end times are the start and end times of the recording segments, not the call itself. For  
recordings made with version 7.3+ components, the offset between the agent machine and the server is  
considered when presenting this time and performing the merge. However, on the agent machine itself,  
there may be a delay between CCSP UI receiving the call and instructing the local Screen Recording  
Service to start the recording and the recording being started. This may be especially true for the first  
recording after logging into CCSP UI.  
l  
CCSP UI was previously called TouchPoint.  
CCSP Replay Help - 7.4  
23

Play audio recordings  
Note  
Encrypted recordings can only be downloaded and then viewed. See Play encrypted audio recordings.  
1.  
In the search result list, click  
. The audio recording starts playing in the Media Player.  
2.  
Use the Media Player controls to pause and replay the audio recording and control the associated audio.  
3.  
If the call was transferred, was part of a conference, included wave playback during the agent-customer  
conversation, or if recording was stopped or started by the agent or a script, multiple segments appear.  
Replay goes directly to the appropriate one. You can click the other segments .To play all the segments,  
select the Play All check box.  
Note  
The order of segments is dependent upon several factors, including the versions of the services that  
made and inserted the recordings, and the timing of the recordings. The segments may therefore not  
appear in ascending numerical sequence.  
4.  
If required, to download the mp3 file, click  
Download.  
CCSP Replay Help - 7.4  
24

Play encrypted audio recordings  
Note  
l  
Encrypted recordings can only be downloaded and then viewed. If the downloaded file is encrypted,  
you can only view it in the Windows Media Player with the appropriate codec.  
l  
The available save options depend on your permissions.  
l  
The downloaded files may be available for more or less than the default time period if your system  
administrator has changed the default values.  
1.  
In the search result list, click  
. The encrypted audio recording appears in the Media Player.  
2.  
Do one of the following:  
o  
To download and save the encrypted recording:  
n  
For immediate playback, select Save Encryption For Playback. The downloaded file is encrypted. It is  
available for one hour (by default).  
n  
For later playback, select Save As Encrypted. The saved file is encrypted and must be played back in  
the Windows Media Player. It is available for 90 days (by default).  
CCSP Replay Help - 7.4  
25

o  
To download and save the unencrypted recording, select Save As Unencrypted. The saved file is  
unencrypted and can be played back in the Windows Media Player or another media player. It is  
available for 90 days (by default).  
3.  
If the call was transferred, part of a conference, or if recording was stopped or started by the agent or a  
script, multiple segments appear. To download all the segments, select the Prepare All check box.  
4.  
Click Prepare.  
5.  
When the download icon appears, click it to download and save the wave file.  
CCSP Replay Help - 7.4  
26

Display chat transcripts  
1.  
In the search result list, click  
. The chat recording appears in the Media Player.  
2.  
To view the other segments, click the links.  
3.  
If required, to download the text file, click  
Download.  
CCSP Replay Help - 7.4  
27

Display email previews  
1.  
In the search result list, click  
. The email recording appears in the Media Player.  
2.  
To view the other segments, click the links.  
3.  
To view the email parts, such as Reply and Forward, click the appropriate tab.  
4.  
If required, to download the email file, click  
Download.  
CCSP Replay Help - 7.4  
28

Play video recordings  
Note  
The call type Screen does not have associated audio.  
1.  
In the search result list, click  
. The screen recording starts playing in the Media Player.  
2.  
Use the Media Player controls to pause and replay the video recording and control the associated audio.  
3.  
If required, to download the mp4 file, click  
Download.  
CCSP Replay Help - 7.4  
29

Play voicemail recordings  
1.  
In the search result list, click  
. The voicemail recording starts playing in the Media Player.  
2.  
Use the Media Player controls to pause and replay the video recording and control the associated audio.  
3.  
If required, to download the mp4 file, click  
Download.  
CCSP Replay Help - 7.4  
30

4: Call scenarios  
This section contains the following topics:  
l  
Overview  
l  
Recorded with screen  
l  
Direct consultation with 2nd agent (A2A)  
l  
Consultation via entry point (TTS)  
l  
Direct consultation and conference  
l  
Transfer to a non-attendable entry point  
CCSP Replay Help - 7.4  
31

Call scenarios overview  
The following basic call scenarios show the segments and rows displayed in Replay. You can use these scenarios to extrapolate the expected output for different  
combinations of call holds, transfers, and so on.  
The basic scenarios presented here show:  
l  
The expected display in Replay  
l  
The timeline of the recording segments against the call timeline  
l  
The rows that are inserted to the recording database  
In all scenarios the following is assumed, unless stated otherwise:  
l  
Media and screen recording is enabled in the IVR Recording block  
l  
Default Screen Recording Implementation rules are being used by the Screen Recording gadget  
CCSP Replay Help - 7.4  
32

Note  
l  
Each Replay row contains audio and/or screen segments that have matching:  
l  
Call ID  
l  
Agent Name  
l  
Call Type  
l  
The table line numbers are for illustrative purposes only. In practice, the ordering of the lines may be different, especially when transfers and conferences have been  
performed.  
l  
In the Video and Media segments columns of the tables:  
l  
A1 indicates a segment associated with Agent 1  
l  
A2 indicates a segment associated with Agent 2  
l  
The segment call type is the call type that appears in the Media Player.  
CCSP Replay Help - 7.4  
33

Recorded with screen  
The following scenarios show an inbound voice call:  
l  
Recorded with screen  
l  
Recorded with screen with 2 call holds  
l  
Recorded with screen manually recorded by agent  
Inbound voice call recorded with screen  
Segments  
Replay rows  
Line  
Call ID  
Agent Name  
ANI  
DNIS  
Queue  
Call Type  
Screen  
Media  
Video Player segments  
Media Player segments  
1  
1  
Agent 1  
Caller No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
1 segment:  
l  
A1—Inbound  
1 segment:  
l  
A1—Inbound  
CCSP Replay Help - 7.4  
34

Inbound voice call recorded with screen with 2 call holds  
Segments  
Replay rows  
Line  
Call ID  
Agent Name  
ANI  
DNIS  
Queue  
Call Type  
Screen  
Media  
Video Player segments  
Media Player segments  
1  
1  
Agent 1  
Customer No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
3 segments:  
l  
A1—Inbound  
l  
A1—Inbound  
l  
A1—Inbound  
3 segments:  
l  
A1—Inbound  
l  
A1—Inbound  
l  
A1—Inbound  
CCSP Replay Help - 7.4  
35

Inbound voice call recorded with screen manually recorded by agent  
IVR Recording has Screen Recording set to false  
Agent starts screen recording after call start  
Segments  
Replay rows  
Line  
Call ID  
Agent Name  
ANI  
DNIS  
Queue  
Call Type  
Screen  
Media  
Video Player segments  
Media Player segments  
1  
1  
Agent 1  
Customer No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
1 segment:  
l  
A1—Inbound  
1 segment:  
l  
A1—Inbound  
CCSP Replay Help - 7.4  
36

Direct consultation with 2nd agent (A2A)  
The following scenarios show an inbound voice call with:  
l  
Direct consultation with 2nd agent, return to caller  
l  
Direct consultation with 2nd agent, transfer to 2nd agent  
l  
Direct consultation with 2nd agent, conference with 2nd agent and caller, one agent drops from conference  
Inbound voice call, direct consultation with 2nd agent, return to caller  
Segments  
CCSP Replay Help - 7.4  
37

Replay rows  
Line  
Call ID  
Agent Name  
ANI  
DNIS  
Queue  
Call Type  
Screen  
Media  
Video Player segments  
Media Player segments  
1  
1  
Agent 1  
Caller No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
3 segments:  
l  
A1—Inbound  
l  
A2—Agent to Agent  
l  
A1—Inbound  
4 segments:  
l  
A1—Inbound  
l  
A1—Agent to Agent  
l  
A2—Agent to Agent  
l  
A1—Inbound  
2  
1  
Agent 1  
Caller No.  
N/A  
AGENT TO AGENT  
Y  
Y  
As for Line 1  
As for Line 1  
2  
1  
Agent 2  
Caller No.  
N/A  
AGENT TO AGENT  
Y  
Y  
As for Line 1  
As for Line 1  
CCSP Replay Help - 7.4  
38

Inbound voice call, direct consultation with 2nd agent, transfer to 2nd agent  
Segments  
CCSP Replay Help - 7.4  
39

Replay rows  
Line  
Call ID  
Agent Name  
ANI  
DNIS  
Queue  
Call Type  
Screen  
Media  
Video Player segments  
Media Player segments  
1  
1  
Agent 1  
Caller No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
3 segments:  
l  
A1—Inbound  
l  
A2—Agent to Agent  
l  
A2—Inbound  
4 segments:  
l  
A1—Inbound  
l  
A1—Agent to Agent  
l  
A2—Agent to Agent  
l  
A2—Inbound  
2  
1  
Agent 1  
Caller No.  
N/A  
AGENT TO AGENT  
Y  
Y  
As for Line 1  
As for Line 1  
3  
1  
Agent 2  
Caller No.  
N/A  
AGENT TO AGENT  
Y  
Y  
As for Line 1  
As for Line 1  
4  
1  
Agent 2  
Caller No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
As for Line 1  
As for Line 1  
CCSP Replay Help - 7.4  
40

Inbound voice call, direct consultation with 2nd agent (A2A), conference with 2nd agent and caller,  
one agent drops from conference  
Segments  
CCSP Replay Help - 7.4  
41

CCSP Replay Help - 7.4  
42

Replay rows  
Line  
Call ID  
Agent Name  
ANI  
DNIS  
Queue  
Call Type  
Screen  
Media  
Video Player segments  
Media Player segments  
1  
1  
Agent 1  
Caller No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
4 segments:  
l  
A1—Inbound  
l  
A2—Agent to Agent  
l  
A1—Conference  
l  
A2—Conference  
5 segments:  
l  
A1—Inbound  
l  
A1—Agent to Agent  
l  
A2—Agent to Agent  
l  
A1—Conference  
l  
A2—Conference  
2  
1  
Agent 1  
Caller No.  
N/A  
AGENT TO AGENT  
Y  
Y  
As for Line 1  
As for Line 1  
3  
1  
Agent 2  
Caller No.  
N/A  
AGENT TO AGENT  
Y  
Y  
As for Line 1  
As for Line 1  
4  
1  
Agent 1  
Caller No.  
Call Center No.  
N/A  
CONFERENCE  
Y  
Y  
As for Line 1  
As for Line 1  
5  
1  
Agent 2  
Caller No.  
Call Center No.  
N/A  
CONFERENCE  
Y  
Y  
As for Line 1  
As for Line 1  
CCSP Replay Help - 7.4  
43

Consultation via entry point (TTS)  
The following scenarios show an inbound voice call with:  
l  
Consultation with agent via entry point (TTS), return to caller  
l  
Consultation with 2nd agent via entry point (TTS), transfer to 2nd agent  
Inbound voice call, consultation with agent via entry point (TTS), return to caller  
Segments  
CCSP Replay Help - 7.4  
44

Replay rows  
Line  
Call ID  
Agent  
Name  
ANI  
DNIS  
Queue  
Call Type  
Screen  
Media  
Video Player  
segments  
Media Player  
segments  
1  
1  
Agent 1  
Caller No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
2 segments:  
l  
A1—Inbound  
l  
A1—Inbound  
2 segments:  
l  
A1—Inbound  
l  
A1—Inbound  
2  
2  
Agent 1  
Tenant  
name  
n/a  
OUTBOUND  
N\*  
Y\*\*  
N/A  
1 segment:  
l  
A1—Outbound  
3  
3  
Agent 2  
Caller No.  
Transfer to  
System  
Q2  
INBOUND  
Y  
Y  
1 segment:  
l  
A2—Inbound  
1 segment:  
l  
A2—Inbound  
\* Not present because Outbound TTS screen recording is not enabled in CCSP UI (previously called TouchPoint) Screen Recording gadget.  
\*\* If Agent 1 presses the Transfer or Conference button when dialing to an attendable entry point, before the call is answered by an agent, there will still be a voice recording  
of the Outbound leg, albeit only a few seconds long.  
CCSP Replay Help - 7.4  
45

Inbound voice call, consultation with 2nd agent via entry point (TTS), transfer to 2nd agent  
Segments  
CCSP Replay Help - 7.4  
46

Replay rows  
Line  
Call ID  
Agent  
Name  
ANI  
DNIS  
Queue  
Call Type  
Screen  
Media  
Video Player  
segments  
Media Player  
segments  
1  
1  
Agent 1  
Caller No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
2 segments:  
l  
A1—Inbound  
l  
A2—Inbound  
2 segments:  
l  
A1—Inbound  
l  
A2—Inbound  
2  
2  
Agent 1  
Tenant name  
n/a  
OUTBOUND  
Y\*  
Y\*\*  
1 segment:  
l  
A1—Outbound  
1 segment:  
l  
A1—Outbound  
3  
3  
Agent 2  
Caller No.  
Transfer to  
System  
Q2  
INBOUND  
Y  
Y  
1 segment:  
l  
A2—Inbound  
1 segment:  
l  
A2—Inbound  
4  
1  
Agent 2  
Customer  
No.  
Call Center No.  
Q2  
INBOUND  
Y  
Y  
As for Line 1  
As for Line 1  
\* Only present if Outbound TTS screen recording is enabled in CCSP UI (previously called TouchPoint) Screen Recording gadget.  
\*\* If Agent 1 presses the Transfer button when dialing to an attendable entry point, before the call is answered by an agent, there will still be a voice recording of the  
Outbound leg, albeit only a few seconds long.  
CCSP Replay Help - 7.4  
47

Direct consultation and conference  
The following scenario shows a call with a direct consultation and conference.  
CCSP Replay Help - 7.4  
48

Inbound voice call, direct consultation with 2nd agent (A2A), conference with 2nd agent and caller,  
one agent drops from conference  
Segments  
CCSP Replay Help - 7.4  
49

CCSP Replay Help - 7.4  
50

Replay rows  
Line  
Call ID  
Agent  
Name  
ANI  
DNIS  
Queue  
Call Type  
Screen  
Media  
Video Player  
segments  
Media Player  
segments  
1  
1  
Agent 1  
Caller No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
3 segments:  
l  
A1—Inbound  
l  
A1—Conference  
l  
A2—Conference  
3 segments:  
l  
A1—Inbound  
l  
A1—Conference  
l  
A2—Conference  
2  
2  
Agent 1  
Tenant  
name  
N/A  
OUTBOUND  
Y\*  
Y\*\*  
1 segment:  
l  
A1—Outbound  
1 segment:  
l  
A1—Outbound  
3  
3  
Agent 2  
Caller No.  
Transfer to  
System  
Q2  
INBOUND  
Y  
Y  
1 segment:  
l  
A2—Inbound  
1 segment:  
l  
A2—Inbound  
4  
1  
Agent 1  
Caller No.  
Call Center No.  
N/A  
CONFERENCE  
Y  
Y  
As for Line 1  
As for Line 1  
5  
1  
Agent 2  
Caller No.  
Call Center No.  
N/A  
CONFERENCE  
Y  
Y  
As for Line 1  
As for Line 1  
\* Only present if Outbound TTS screen recording is enabled in CCSP UI (previously called TouchPoint) Screen Recording gadget.  
\*\* If Agent 1 presses the Conference button when dialing to an attendable entry point, before the call is answered by an agent, there will still be a voice recording of the  
Outbound leg, albeit only a few seconds long.  
CCSP Replay Help - 7.4  
51

Transfer to a non-attendable entry point  
The following scenario shows a call with a transfer to a non-attendable entry point.  
Inbound voice call, transfer to non-attendable entry point, 2nd agent picks up  
Segments  
CCSP Replay Help - 7.4  
52

Replay rows  
Line  
Call ID  
Agent Name  
ANI  
DNIS  
Queue  
Call Type  
Screen  
Media  
Video Player segments  
Media Player segments  
1  
1  
Agent 1  
Caller No.  
Call Center No.  
Q1  
INBOUND  
Y  
Y  
2 segments:  
l  
A1—Inbound  
l  
A2—Inbound  
2 segments:  
l  
A1—Inbound  
l  
A2—Inbound  
3  
1  
Agent 2  
Caller No.  
Call Center No.  
Q2  
INBOUND  
Y  
Y  
As for Line 1  
As for Line 1  
CCSP Replay Help - 7.4  
53

5: Archiver Playback  
Archiver Playback is a feature that enables you to search and view recordings from archived recording  
databases in addition to or instead of the tenant’s live production recording database.  
The PS Recording Archiver application creates and manages the archive recording databases. For information  
about this application, refer to the Recording Archiver User Guide.  
The PS-supplied Config Portal database provides access to the archiver configuration and archived recording  
databases. You must set up the Config Portal according to the instructions in the Config Portal Deployment Guide  
v.3.0.0.1.  
Your system administrator configures Replay to work in the appropriate mode. The modes are:  
l  
Off mode  
l  
Online mode  
l  
Offline mode  
l  
On-premise mode  
Note  
When your system administrator changes the Archiver Playback mode, to ensure that the browser picks up  
the configuration changes, you must always:  
1.  
Log out  
2.  
Refresh your browser  
3.  
Log in again with the appropriate credentials  
Off mode  
In off mode, Replay looks and functions exactly as before and you only have access to the single live recording  
database.  
CCSP Replay Help - 7.4  
54

Online mode  
In online mode, you have access to both the live and archive recording databases. The currently selected  
database is shown as a button in the heading bar. When logging on to Replay, the live database is initially  
selected as shown below.  
To see the list of available databases, click the [database name] button. The list of available live and archive  
recording databases appears.  
To select another database, click the name in the list, and then click OK in the dialog.  
You can search, play, and download recordings in the selected database in the same way as in the live  
database.  
Note  
l  
When changing databases, the start and end times of the current search are maintained. The other  
search parameters and results are cleared.  
l  
If there are no archive databases available for the tenant, the [database name] button is not displayed.  
l  
If an archive database is created while you are logged into Replay, you will not see it in the list until you  
log out and log back in again.  
l  
When an archive database is selected, the list of available Agent Names is taken directly from that  
database. Therefore, when viewing an archive database, the permission profile of the logged in user is  
not applied to the search and recordings from all agents in the archive database are available to view.  
CCSP Replay Help - 7.4  
55

Offline mode  
In offline mode, you only have access to archive recording databases.  
The Replay application is configured by your system administrator to work in offline mode with a specific set of  
archive recording databases.  
The Replay host is isolated from the rest of the CCSP system and has no access to the domain or the  
configuration database.  
Because of this, you are typically required to log in to Replay with a user local to the machine that is hosting the  
Replay application. For example, if the local machine user is tempReplayUser, you need to log in using this  
user's name and password, instead of your usual domain login.  
When you log in to Replay, no database is selected and the controls are disabled.  
To select a database, click the No database selected button. The list of available archive recording databases  
appears.  
CCSP Replay Help - 7.4  
56

To select a database, click the name in the list, and then click OK in the dialog.  
The controls become active and you can search, play, and download recordings in the selected database in the  
same way as in the off and online modes.  
CCSP Replay Help - 7.4  
57

On-premise mode  
In on-premise mode, you only have access to archive recording databases.  
The Replay application is configured by your system administrator to work with an Archiving Master database  
that provides the list of archive recording databases to which you have access.  
The Replay host is isolated from the rest of the CCSP system and has no access to the domain or the  
configuration database.  
Because of this, you are typically required to log in to Replay with a user local to the machine that is hosting the  
Replay application. For example, if the local machine user is tempReplayUser, you need to log in using this  
user's name and password, instead of your usual domain login.  
When you log in to Replay, no database is selected and the controls are disabled.  
To select a database, click the No database selected button. The list of available archive recording databases  
appears.  
CCSP Replay Help - 7.4  
58

To select a database, click the name in the list, and then click OK in the dialog.  
The controls become active and you can search, play, and download recordings in the selected database in the  
same way as in the off and online modes.  
CCSP Replay Help - 7.4  
59

# Contact Center: Service Provider Dashboard Help Version 7.4

Contents  
1: Intro  
4  
What's new  
4  
Legal disclaimer  
4  
Support  
4  
2: Overview  
5  
Log in  
5  
Password expired message  
6  
Additional login messages  
6  
Dashboard workspace  
7  
Workspace menus  
9  
Workspace  
9  
View  
9  
Report  
10  
Window  
10  
Reports  
11  
Multichat implications  
11  
Filters  
11  
Gadget types  
13  
Agent and call states  
14  
Interpret reports  
15  
Log out  
17  
Additional logout messages  
17  
Session expiry  
17  
3: Design reports  
19  
Add a view  
20  
Share a view  
20  
Add a report  
21  
Add a gadget  
21  
Add a bar chart  
22  
Add a browser window  
25  
Javascript functions for Web applications  
26  
Add a clock  
27  
Add a pie chart  
29  
Add a filter  
32  
Configure a filter  
32  
Add a table  
33  
Configure table thresholds  
35  
Free form value field  
37  
Configure a complex expression  
37  
Filter a table column  
39  
CCSP Dashboard Help - 7.4  
2

Basic filtering  
39  
Detailed filtering  
39  
Examples  
40  
Add a trend chart  
42  
Copy, edit, or delete a gadget  
44  
Move table columns  
44  
Export a table  
44  
4: Report statistics  
45  
Agent report statistics  
45  
Call report statistics  
49  
Campaign report statistics  
51  
Group report statistics  
54  
Queue report statistics  
56  
CCSP Dashboard Help - 7.4  
3

1: Intro  
Dashboard provides supervisors and administrators with a variety of real-time statistical reports about the  
contact center performance statistics during the most recent 24 hour or shorter period.  
This Help is for CCSP supervisors and CCSP tenant administrators who view or create and edit real-time contact  
center performance reports  
What's new  
Version 7.4  
Log in and Log out — added process for CCSP single sign-on. Requires HF74-63537.  
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.  
CCSP Dashboard Help - 7.4  
4

2: Overview  
This section contains the following topics:  
l  
Log in  
l  
Dashboard workspace  
l  
Workspace menus  
l  
Reports  
l  
Gadget types  
l  
Agent and call states  
l  
Log out  
Dashboard provides supervisors and administrators with a variety of real-time statistical reports about the  
contact center performance statistics during the most recent 24 hour or shorter period.  
The reports display statistics about these categories of data:  
l  
Agents  
l  
Calls  
l  
Campaigns  
l  
Groups  
l  
Queues  
You can set thresholds for warnings and alarms in tables you create in your reports. When threshold values are  
reached, table cells display the warnings shaded in the color configured in the gadget builder.  
You can use Dashboard to create a customized wallboard that your agents can view.  
Log in  
The log in process is slightly different depending on the identity provider configured by your administrator.  
Note  
l  
To log in to Dashboard, you need the Supervisor permission.  
l  
If your administrator has enabled single sign-on and you are logged in to CCSP UI (previously called  
TouchPoint), to start Dashboard, click  
Real Time Reports in the gadgetory. You do not need to  
type your user ID and password again.  
l  
If you log in again after your CCSP 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.  
2.  
Click Go or press Enter. The Welcome to Dashboard page appears.  
CCSP Dashboard Help - 7.4  
5

3.  
Click Log in. If you are already logged in, the Dashboard workspace appears.  
4.  
If you are not already logged in, the Auth Server Login Selector page appears.  
5.  
Type your User name in User Principal Name (UPN) format, for example, someone@domainname.com,  
and click Continue.  
6.  
The next step depends on the identity provider configured by your administrator:  
o  
For Auth Server, the AuthServer login page appears. Type your Password and click Log in.  
Alternatively, to change the username, click Back to return to the previous page. The Dashboard  
workspace appears. If necessary, see Password expired message below.  
o  
For ADFS, you are redirected to the ADFS login page to authenticate. Please consult your local  
procedures for how to proceed from this point.  
o  
For Enghouse Identity Server (EIS), you are redirected to the EIS login page. Type your Password and  
click Login. The Dashboard workspace appears.  
7.  
If necessary, see Additional login messages below.  
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 Auth Server login page, the Password  
expired notification appears.  
1.  
In the notification, click Change Password. The Change password dialog box 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:  
l  
Failed to get authorization for Dashboard use. Your user profile is not permitted to use the Dashboard  
application. Only supervisors are permitted to use the Dashboard application.  
l  
You are not authorized to view this page. This can occur because:  
o  
You are logged in to an external identity provider with a different username than the one you entered  
on the Auth Server Login Selector page to log in to the CCSP client application.  
o  
You entered a username on the Auth Server Login Selector page and were redirected to an external  
identity provider, where you log on with a different username.  
l  
Auth Server is not available or there is a misconfiguration. You must contact your administrator. You will not be  
able to log in to Dashboard until the issue is resolved.  
CCSP Dashboard Help - 7.4  
6

Dashboard workspace  
When you start Dashboard, the Workspace appears.  
The workspace presents views of reports. Views are collections of reports that appear together in a single  
screen. Each view in the workspace has its own tab.  
When you create reports, this workspace is a place where you work using graphical tools. You can edit views  
that you created. You cannot edit views that you did not create, but you can load them into your workspace to  
display them. Such views are shared views. You can also copy parts of views created by others in your contact  
center to your own views, where you can edit and save them.  
Dashboard provides a number of tools to create reports. Some of the tools are available from the main menu.  
The rest of the tools are available through shortcut menus. See Workspace menus.  
See also Reports and Gadget types.  
If you minimize the Dashboard, after the configured amount of time, a dialog box appears with the message  
Workspace must be reactivated.  
CCSP Dashboard Help - 7.4  
7

CCSP Dashboard Help - 7.4  
8

Workspace menus  
The Workspace has the following main menus:  
l  
Workspace  
l  
View  
l  
Report  
l  
Window  
To view a menu of additional options for a particular view, right-click its tab. See Additional view options.  
To view a menu of additional options for a particular report, right-click its title bar. See Additional report  
options.  
Workspace  
Use the Workspace menu to:  
l  
Save workspace — save a workspace you are creating.  
l  
Load workspace — load an existing workspace.  
l  
Arrange workspace — rename a view, move parts of the views in your current workspace, and resize them.  
View  
Use the View menu to:  
l  
Add view — create a new view in your workspace.  
l  
Add shared view — add a shared view to your workspace. This view can be created by a different user.  
l  
Delete view — delete the current view. The message Do you want to remove this view and all its reports?  
appears. Click OK to continue.  
Additional view options  
To display additional options for a particular view, right-click its tab. Use the shortcut-menu to:  
l  
Delete view — delete the current view from the workspace.  
l  
View shared — share a view that you created.  
l  
Add report — add a report to the current view.  
l  
Paste report — add a report that you copied to the Clipboard to the current view.  
CCSP Dashboard Help - 7.4  
9

Report  
Use the Report menu to:  
l  
Add report — add a report to the current view. When you add a report, an option to add a gadget appears.  
See Gadget types.  
l  
Paste report — add a report that you copied to the Clipboard to the current view.  
Additional report options  
To display additional options for a particular report, right-click its title bar. Use the shortcut-menu to:  
l  
Cut report — delete a report from the current view and add it to the Clipboard.  
l  
Copy report — copy a report to the Clipboard from the current view. This can be a report from a shared  
view from another user.  
l  
Delete report — delete a selected report from the current view. The message Do you want to remove this  
report and all its gadgets? appears. Click Yes to continue.  
l  
Add gadget — add a gadget to the current report. See Gadget types.  
l  
Paste gadget — paste a gadget from the Clipboard into the current report. Use to copy gadgets from  
another user's report into your reports.  
Window  
When you save a workspace, the message Workspace saved and a timestamp appear. The message box  
disappears automatically after a few seconds.  
If there are problems getting statistics from the reporting server, the message Unable to connect to RTR Server  
and a timestamp appear. The status icon changes to  
. The message box disappears automatically after a  
few seconds.  
When the connection is re-established, the message Reconnected to server appears for a few seconds and the  
status icon changes back to  
.  
To display all the messages, on the Windows menu, click Show Messages. The messages appear on the right  
of your browser.  
CCSP Dashboard Help - 7.4  
10

Reports  
Reports are grouped together in views.  
A report contains one or more gadgets, which are the basic tools in the toolkit. When you create a report, you  
select one or more gadgets to include in the report. See Gadget types.  
When completely defined for your report, each of these gadgets, except for Report Filter, displays statistics  
about one of these categories:  
l  
Agents  
l  
Calls  
l  
Campaigns  
l  
Groups  
l  
Queues  
Multichat implications  
If multichat is enabled, Dashboard shows simultaneous calls handled by an agent, which results in reports  
showing more agents in the Agent-In-Call state (comprising Wrapup, Hold, Ringing, and InCall) than are logged  
on.  
Agent tables and trend charts show current simultaneous calls and current load. Group tables and trend charts  
show average simultaneous calls per agent and average load per agent.  
Filters  
A report filter assigns filters to multiple tables and charts in the same report.  
Suppose you have three tables describing campaigns and you want to assign filters to all of them, for example,  
campaign selection by name. You can do that with the filter gadget.  
You select a category and the columns that are the basis of your filter.  
Dashboard applies the filter only to:  
CCSP Dashboard Help - 7.4  
11

l  
Tables in the report that are of the same category and contain the column used as the filter criterion.  
l  
Charts in the report that are of the same category and match a category name.  
CCSP Dashboard Help - 7.4  
12

Gadget types  
Gadgets create commonly used statistical representations of data describing entities belonging to your contact  
center.  
You can add the following gadgets to a report:  
l  
Bar chart — creates a chart with rectangular bars with lengths proportional to the values that they  
represent.  
l  
Browser — retrieves and displays Real Time Reports data.  
l  
Table — presents the data describing your selected contact center entities as text and numbers.  
l  
Trend chart — creates a line graph that tracks one performance metric over a period of time.  
l  
Report filter — assigns filters to multiple tables and charts within the same report. See Report filters.  
l  
Pie chart — creates a pie chart that represents percentages and fractions of a total.  
l  
Time — creates a time display. You select the time zone suitable for your geographic region.  
When completely defined for your report, each of these gadgets, except for report filter, displays statistics  
about one of these categories:  
l  
Agents  
l  
Queues  
l  
Groups  
l  
Campaigns  
l  
Calls  
Note  
To copy a report's contents as a .csv file into another document, such as an Excel spreadsheet, use a Table  
gadget.  
CCSP Dashboard Help - 7.4  
13

Agent and call states  
The agent states and corresponding call states are as follows. If an agent is simultaneously handling multiple  
calls, there are multiple call states for that agent.  
Agent state  
Call state  
Available  
Idle (no call)  
Ringing  
Available in Call  
In Call  
On Hold  
Wrap Up  
Preview  
Multiple (more than one call handled)  
Released  
Idle (no call)  
Ringing  
Released in Call  
In Call  
On Hold  
Wrap Up  
Preview  
Multiple (more than one call handled)  
CCSP Dashboard Help - 7.4  
14

Interpret reports  
In the following example, agent HV is Available In Call. In the report, the same agent is allocated to Busy [1], In  
Call [1], and Available In Call [1].  
CCSP Dashboard Help - 7.4  
15

In the following example, agent HV is still Available In Call. In the report, the same agent is now allocated to  
Busy [1], On Hold [1], and Available In Call [1].  
In the following example, agent HV is still Available In Call. In the report, the same agent is now allocated to  
Busy [1], In Wrap-up[1], and Available In Call [1].  
CCSP Dashboard Help - 7.4  
16

Log out  
Note  
If you are using an external identity provider such as ADFS or EIS, when you click Log out of your CCSP  
session, you are also logged out of your external identity provider.  
1.  
At the top right of the Dashboard workspace, click the user icon and then click Log out.  
2.  
Do one of the following:  
o  
If you want to return to Dashboard, click Return to Login page.  
o  
If you have finished working with CCSP, click Log out of your CCSP session.  
Additional logout messages  
When you click Log out of your CCSP session, the following message may appear:  
l  
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 CCSP session again.  
The application will return to the login page. Depending on the problem resolution, this may take slightly  
longer than normal.  
Session expiry  
Your CCSP application session will become invalid when:  
l  
The login session time exceeds the maximum CCSP login session lifetime specified by your administrator.  
l  
You log out of your CCSP session using another application.  
l  
You clear all cookies in the browser.  
When your CCSP application session becomes invalid, you must log in again. See Log in.  
CCSP Dashboard Help - 7.4  
17

If you are logged in with an external identity provider (ADFS or EIS) and that provider’s login session lifetime  
exceeds the CCSP login session lifetime, you only have to enter your username on the Auth Server Login  
Selector page. You will then automatically be authenticated against your external identity provider.  
CCSP Dashboard Help - 7.4  
18

3: Design reports  
This section contains the following topics:  
l  
Add a view  
l  
Share a view  
l  
Add a report  
l  
Add a gadget  
l  
Add a bar chart  
l  
Add a browser window  
l  
Javascript functions for Web applications  
l  
Add a clock  
l  
Add a pie chart  
l  
Add a filter  
l  
Configure a filter  
l  
Add a table  
l  
Configure table thresholds  
l  
Filter a table column  
l  
Add a trend chart  
l  
Copy, edit, or delete a gadget  
To design reports:  
1.  
Add a view and optionally share it.  
2.  
Add up to six reports to the view.  
3.  
Add up to 10 gadgets to the report.  
CCSP Dashboard Help - 7.4  
19

Add a view  
You can add a maximum of 10 views per workspace.  
To import an existing shared view:  
1.  
On the View menu, click, Add Shared View.  
2.  
Select the view to import and click OK.  
An  
icon appears next to the name of the imported view. You can view the name of the view's owner in a  
tooltip.  
To add a new view:  
1.  
On the View menu, click Add View.  
2.  
Click  
Edit next to New View and type the new name.  
To rename the view, click  
Edit next to the view name and type the new name.  
The next step is to add a report.  
Share a view  
1.  
Right-click in the view. A shortcut menu appears.  
2.  
Click View Shared. An  
icon appears next to the view name.  
To unshare a view:  
1.  
Right-click in the view. A shortcut menu appears.  
2.  
Click View Shared to remove the check mark.  
CCSP Dashboard Help - 7.4  
20

Add a report  
You can add a maximum of six reports to a view.  
1.  
Right-click in the view. A shortcut menu appears.  
2.  
Click Add report.  
3.  
Click New Report and type the new name.  
4.  
To rename the report, click its name and type the new name.  
The next step is to add a gadget.  
Add a gadget  
You can add a maximum of 10 gadgets to a report. This number includes the report filter.  
Add a bar chart  
Add a browser window  
Add a clock  
Add a pie chart  
Add a report filter  
Add a table  
Add a trend chart  
CCSP Dashboard Help - 7.4  
21

Add a bar chart  
1.  
Right-click in the report window and click Add gadget. The Gadget Builder dialog box appears.  
2.  
Click Bar chart. The General tab appears.  
3.  
Under Caption:  
a.  
To display the caption on the chart, select the Visible check box. To hide the caption, clear the Visible  
check box.  
b.  
Optionally, type a new caption in the text box and click Font to format it.  
4.  
Under Opacity sliders, set the Gadget opacity and Background opacity. You increase the opacity by  
making it less transparent.  
5.  
Click the Bar Chart tab.  
6.  
In the Orientation list, click one of the following:  
o  
To display horizontal bars (default), click Horizontal.  
o  
To display vertical bars, click Vertical.  
CCSP Dashboard Help - 7.4  
22

7.  
Under Legend, do one of the following:  
o  
To hide the legend, clear the Show check box.  
o  
To display only the numerical values, select the Show and Value check boxes.  
o  
To display only the percentages, select the Show check box, clear the Value check box, and select the  
Percentage check box.  
o  
To display the value and the percentage, select the Show, Value, and Percentage check boxes.  
8.  
Optionally, change the legend's Font size.  
9.  
Click the Data Connection tab.  
11.  
Under Counter selection, click the category.  
12.  
To include or exclude counters, select the check boxes and click the right or left arrow. To re-order  
counters, use  
to move the counter.  
For details see:  
o  
Agent report statistics  
o  
Call report statistics  
o  
Campaign report statistics  
o  
Group report statistics  
o  
Queue report statistics  
13.  
Under Filter, do one of the following:  
o  
To filter using regular expressions, type the criterion in the text box. For examples of regular  
expressions, see Filter a table column.  
o  
To filter the data by one item, click the filter.  
o  
To select multiple filters, click the filters.  
o  
To avoid filtering, click All (the default).  
Note  
The text box has precedence over the list box. The list box is disabled when the text box is used. If the  
text box is empty and you select from the list, the text box is disabled.  
CCSP Dashboard Help - 7.4  
23

14.  
Click Confirm.  
The bar chart appears in the report window.  
CCSP Dashboard Help - 7.4  
24

Add a browser window  
Note  
To prevent the browser window from hiding the report, before adding the browser window:  
1.  
On the Workspace menu, enable Arrange workspace.  
2.  
Enlarge the report.  
1.  
Right-click in the report window and click Add gadget. The Gadget Builder dialog box appears.  
2.  
Click Browser. The General tab appears.  
3.  
Under Caption:  
a.  
To display the caption on the browser, select the Visible check box. To hide the caption, clear the  
Visible check box.  
b.  
Optionally, type a new caption in the text box and click Font to format it.  
4.  
Under Opacity sliders, set the Gadget opacity and Background opacity. You increase the opacity by  
making it less transparent.  
5.  
Click the Browser tab.  
CCSP Dashboard Help - 7.4  
25

6.  
In the text box, type the URL for the location of Real Time Reports data.  
7.  
To use RTR data, select the Store RTR Data check box.  
8.  
To include the ID of the logged-on supervisor in the browser URL, for example,  
http://www.enghouse.com?loginid=1, select the Include Metadata in URL check box.  
9.  
Click Confirm.  
The browser window appears in the report window.  
The remaining steps depend on the design of your web site or application. For some of the information you  
need for creating the web application used by the Browser gadget, see Javascript functions for Web  
application.  
Caution  
Your browser application may show some limitations and issues known to browser applications. For more  
information, see the Microsoft Developer Network.  
Javascript functions for Web applications  
The Browser gadget provides the following JavaScript functions for your hosted web application to retrieve RTR  
data:  
l  
parent.GetStatisticsXml();  
This function returns a JavaScript Promise containing agent and queue statistics in XML format.  
l  
parent.GetConfigurationXml();  
This function returns a JavaScript Promise containing group, agent, queue, and campaign configurations in  
XML format.  
l  
parent.GetHistoryXml();  
This function returns a JavaScript Promise containing queue history in XML format.  
Ask your Enghouse representative for a sample script to access RTR data.  
CCSP Dashboard Help - 7.4  
26

Add a clock  
1.  
Right-click in the report window and click Add gadget. The Gadget Builder dialog box appears.  
2.  
Click Time. The General tab opens.  
3.  
Under Caption:  
a.  
To display the caption on the clock, select the Visible check box. To hide the caption, clear the Visible  
check box.  
b.  
Optionally, type a new caption in the text box and click Font to format it.  
4.  
Under Opacity sliders, set the Gadget opacity and Background opacity. You increase the opacity by  
making it less transparent.  
5.  
Click the Clock tab.  
6.  
In the Time zone list, click the time zone.  
Note  
The only times available are standard time. Adjustments for summer daylight hours are not available.  
CCSP Dashboard Help - 7.4  
27

7.  
To display seconds, select the Show seconds check box.  
8.  
Under Clock Format, click Analog Clock or Digital Clock.  
9.  
If you clicked Digital Clock, click 12-hour format or 24-hour format.  
10.  
To format the clock, click Font, select the font family, size, and color, and then click OK.  
11.  
Click Confirm.  
The clock appears in the report window.  
CCSP Dashboard Help - 7.4  
28

Add a pie chart  
1.  
Right-click in the report window and click Add gadget. The Gadget Builder dialog box appears.  
2.  
Click Pie chart. The General tab appears.  
3.  
Under Caption:  
a.  
To display the caption on the chart, select the Visible check box. To hide the caption, clear the Visible  
check box.  
b.  
Optionally, type a new caption in the text box and click Font to format it.  
4.  
Under Opacity sliders, set the Gadget opacity and Background opacity. You increase the opacity by  
making it less transparent.  
5.  
Click the Pie chart tab.  
6.  
Set the inner radius of the pie chart.  
7.  
Under Legend, do one of the following:  
o  
To hide the legend, clear the Show check box.  
o  
To display only the numerical values, select the Show and Value check boxes.  
CCSP Dashboard Help - 7.4  
29

o  
To display only the percentages, select the Show check box, clear the Value check box, and select the  
Percentage check box.  
o  
To display the value and the percentage, select the Show, Value, and Percentage check boxes.  
8.  
Optionally, change the legend's Font size.  
9.  
Click the Data Connection tab.  
10.  
Under Counter selection, click the category.  
11.  
To include or exclude counters, select the check boxes and click the right or left arrow. To re-order  
counters, use  
to move the counter.  
For details see:  
o  
Agent report statistics  
o  
Call report statistics  
o  
Campaign report statistics  
o  
Group report statistics  
o  
Queue report statistics  
12.  
Under Filter, do one of the following:  
o  
To filter using regular expressions, type the criterion in the text box. For examples of regular  
expressions, see Filter a table column.  
o  
To filter the data by one item, click the filter.  
o  
To select multiple filters, click the filters.  
o  
To avoid filtering, click All (the default).  
Note  
The text box has precedence over the list box. The list box is disabled when the text box is used. If the  
text box is empty and you select from the list, the text box is disabled.  
13.  
Click Confirm.  
The pie chart appears in the report window.  
CCSP Dashboard Help - 7.4  
30

Note  
Percentage on a pie segment = (The value of the segment) /(sum of the values of all the segments) x 100%  
CCSP Dashboard Help - 7.4  
31

Add a filter  
1.  
Right-click in the report window and click Add gadget. The Gadget Builder dialog box appears.  
2.  
Click Filter. The General tab appears.  
3.  
Under Caption:  
a.  
To display the caption on the report filter, select the Visible check box. To hide the caption, clear the  
Visible check box.  
b.  
Optionally, type a new caption in the text box and click Font to format it.  
4.  
Under Opacity sliders, set the Gadget opacity and Background opacity. You increase the opacity by  
making it less transparent.  
5.  
Click Confirm.  
The report filter now appears in the window.  
Configure the report filter at report run time. See Configure a report filter.  
Configure a filter  
For overview information, see Filters.  
1.  
In the Category list, click the category.  
2.  
In the Column list, click the table column  
3.  
In Filter, type the criterion for the filter. The criterion depends on the selected column.  
4.  
Click Apply.  
CCSP Dashboard Help - 7.4  
32

Add a table  
1.  
Right-click in the report window and click Add Gadget. The Gadget Builder dialog box appears.  
2.  
Click Table. The General tab appears.  
3.  
Under Caption:  
a.  
To display the caption on the table, select the Visible check box. To hide the caption, clear the Visible  
check box.  
b.  
Optionally, type a new caption in the text box and click Font to format it.  
4.  
Under Opacity sliders, set the Gadget opacity and Background opacity. You increase the opacity by  
making it less transparent.  
5.  
Click the Table tab.  
6.  
In Rows per page, type the number of rows you want in your table. The default is 20.  
7.  
If you want to display the number of data rows in the table, select the Footer visible check box. To hide  
the number of rows in the table, clear the Footer visible check box.  
8.  
Optionally, click Font to format the table titles and data.  
CCSP Dashboard Help - 7.4  
33

Note  
l  
Font selection does not affect the table caption.  
l  
Select a data font color that is different from the colors for thresholds and warnings. The colors  
selected for thresholds override colors selected for fonts.  
9.  
Click the Data Connection tab.  
10.  
Under Counter selection list, click the category.  
11.  
To include or exclude counters, select the check boxes and click the right or left arrow. To re-order  
columns, use  
to move the column.  
For details see:  
o  
Agent report statistics  
o  
Call report statistics  
o  
Campaign report statistics  
o  
Group report statistics  
o  
Queue report statistics  
12.  
In the Summary column, for each numerical column, click the summary type. The default is None.  
13.  
Optionally, for Queues:  
o  
Under Filter, do one of the following:  
n  
To display data for a defined number of hours in the past, click Relative and move the slider to the  
right to set the number of hours.  
n  
To display data from a specified time onwards, click Absolute and then click the time in the list.  
o  
In the Media type list, click the medium to include.  
14.  
Configure the thresholds and alerts. See Configure table thresholds.  
15.  
Click Confirm.  
The table appears in the report window.  
Note  
To deselect a row in a table gadget, uncheck the row.  
For information on filtering tables by selecting entries in table columns, see Filter a table column.  
CCSP Dashboard Help - 7.4  
34

Configure table thresholds  
Note  
l  
To configure a complex expression, insert each condition individually in reverse order. Each new  
condition is inserted at the start of the expression. See Configure a complex expression.  
l  
Thresholds only work in the language you define them in.  
l  
If you add or update thresholds, you must save the workspace for the changes to be reflected in the  
email alerts.  
Use equations, inequalities, and boolean conditions to define table threshold.  
1.  
Click the Threshold Manager tab.  
2.  
Click New. The Threshold Builder dialog box appears.  
3.  
Optionally, type a new name in Threshold Name.  
4.  
To set the threshold:  
CCSP Dashboard Help - 7.4  
35

a.  
Click a table column in the list.  
b.  
Click the required symbols.  
c.  
Select or type a value in the free form field to the right of the symbols. See Free form value field.  
d.  
Click Insert. The threshold expression appears in the equation field.  
e.  
Repeat steps a-d to add each additional condition for complex expressions. See Configure a complex  
expression.  
Alternatively, type the entire expression or complete the rest of your expression in the equation field. Use  
only the listed symbols.  
5.  
To set the threshold alarm:  
a.  
In the Target Column list, click the column for the threshold alarm.  
b.  
In the Color list, click the color to indicate the threshold is exceeded.  
6.  
Optionally, to send email alerts:  
a.  
Select the Send Email check box.  
b.  
Type the email addresses. Separate multiple email addresses with a semi-colon (;), for example,  
supervisor1@abc\_corp.com; supervisor2@abc\_corp.com; supervisor3@abc\_corp.com  
c.  
In Send if condition persists for, type the number of seconds for the system to wait before sending  
the alert. The default is 60 seconds.  
Note  
l  
The refresh interval for threshold data is 5 seconds.  
l  
The Dashboard SMTP Server from which to send email alerts is configured by the tenant  
administrator in CCSP Admin, in Tenant Properties, under Parameters.  
l  
To activate the new email alerts, you must save the workspace.  
7.  
Click OK.  
8.  
On the Threshold Manager tab:  
o  
To change the priority of a threshold, use  
to move the threshold.  
o  
To activate or deactivate a threshold, select or clear the check box in front of it.  
9.  
Click Confirm.  
CCSP Dashboard Help - 7.4  
36

Free form value field  
The contents of this free form field depend on the selected column.  
Column type  
Field format  
A column containing time values  
The format of the available data is D.HH:MM:SS. The  
default value is 0.00:00:00  
A column containing text  
Text enclosed in " "  
A column that can contain only a few values,  
like Agent State  
A list where you select values by clicking  
Configure a complex expression  
To configure a complex expression, insert each condition individually, in reverse order. Each new condition is  
inserted at the start of the expression.  
The following example, shows how to add the following complex expression in Threshold Manager:  
[Waiting in Queue]>0 And [Name]="Q01" or [Name]="Q02"  
1.  
Add: [Name] = "Q02"  
Result: [Name] = "Q02"  
2.  
Add: [Name] = "Q01" OR  
Result: [Name] = "Q01" OR [Name] = "Q02"  
3.  
Add: [Waiting in Queue] > 0 AND  
Result: [Waiting in Queue] > 0 AND [Name] = "Q01" OR [Name] = "Q02"  
CCSP Dashboard Help - 7.4  
37

CCSP Dashboard Help - 7.4  
38

Filter a table column  
You can limit the rows that appear in a table by selecting values in one or more columns.  
Note  
l  
Filtering uses regular expressions.  
l  
The following characters are regular expression operational characters: \* ? + $ # ^ [ \ .|  
If you want to use them as plain text, add a backslash (\) before the character.  
Basic filtering  
1.  
Click  
in the column containing the criterion, for example, First Name in an agent table. The Filter field  
appears.  
2.  
In Filter, type the criterion:  
o  
For text data, type text.  
o  
For numeric data, type numbers.  
o  
To combine filter criteria, type the OR symbol |.  
The filter button changes to  
to indicate it is activated.  
3.  
To change your selection, repeat steps 1 and 2. You can delete your previous entry.  
Detailed filtering  
The following table provides examples of commonly used filter techniques.  
Metacharacter  
Description  
.  
Selects any single character, for example, a.c matches abc.  
[a.c] selects only a, ., or c when the dot is part of a bracket expression. See bracket  
expression below.  
[ ]  
A bracket expression. Selects a single character contained within the brackets, for  
example, [abc] selects a, b, or c.  
[a-z] specifies a range which selects any lowercase letter from a to z. These forms can  
be mixed, for example, [abcx-z] selects a, b, c, x, y, or z, as does [a-cx-z].  
The - character is treated as a literal character if it is the last or the first (after the ^)  
character within the brackets, for example, [abc-], [-abc].  
^  
Matches the starting position within a character string.  
[^ ]  
Selects a single character not contained within the brackets, for example, [^abc] selects  
CCSP Dashboard Help - 7.4  
39

Metacharacter  
Description  
any character other than a, b, or c. [^a-z] Selects any single character that is not a  
lowercase letter from a to z. As in bracket expression, literal characters and ranges can  
be mixed.  
\*  
Matches the preceding element zero or more times, for example, ab\*c matches ac, abc,  
and abbbc. [xyz]\* matches , x, ycz, zx, zyx, xyzzy, and so on.  
Examples  
The following examples show how to use the filter techniques.  
Content of a filter  
Syntax  
Explanation  
Select objects containing the  
string  
Sales  
All names containing string Sales are included.  
Exclude objects containing  
string  
^((?!Sales).)\*$  
All names containing string Sales are excluded.  
Exclude objects containing list  
of strings  
^((?!Sales|Support).)\*$  
All names containing string Sales or Support are  
excluded.  
Select objects starting with a  
character  
^o  
All names starting with o are included.  
Match objects starting with a  
list of characters  
^[o,e]  
All names starting with o and e are included.  
The following example shows an unfiltered table.  
CCSP Dashboard Help - 7.4  
40

The following table includes only names that begin with the specified letter.  
The following table includes only names that contain the specified string.  
The following table includes only names that begin with the specified letters.  
CCSP Dashboard Help - 7.4  
41

Add a trend chart  
1.  
Right-click in the report window and click Add Gadget. The Gadget Builder dialog box appears.  
2.  
Click Trend chart. The General tab appears.  
3.  
Under Caption:  
a.  
To display the caption on the chart, select the Visible check box. To hide the caption, clear the Visible  
check box.  
b.  
Optionally, type a new caption in the text box and click Font to format it.  
4.  
Under Opacity sliders, set the Gadget opacity and Background opacity. You increase the opacity by  
making it less transparent.  
5.  
Click the Data Connection tab.  
6.  
Under Counter selection, click the category.  
7.  
Click the counter to include.  
CCSP Dashboard Help - 7.4  
42

For details see:  
o  
Agent report statistics  
o  
Call report statistics  
o  
Campaign report statistics  
o  
Group report statistics  
o  
Queue report statistics  
8.  
Optionally, change the Font size.  
9.  
Under Filter, do one of the following:  
o  
To filter using regular expressions, type the criterion in the text box. For examples of regular  
expressions, see Filter a table column.  
o  
To filter the data by one item, click the filter.  
o  
To select multiple filters, click the filters.  
o  
To avoid filtering, click All (the default).  
Note  
The text box has precedence over the list box. The list box is disabled when the text box is used. If the  
text box is empty and you select from the list, the text box is disabled.  
10.  
Click Confirm.  
The trend chart appears in the report window.  
CCSP Dashboard Help - 7.4  
43

Copy, edit, or delete a gadget  
You can copy a gadget from a shared view, created by a different user, into your report. You can edit, cut, and  
delete gadgets only in your own views.  
1.  
Right-click the gadget. A shortcut menu appears.  
2.  
Do one of the following:  
o  
To edit the gadget, click Edit gadget, and follow the instructions for adding that type of gadget.  
o  
To cut and paste the gadget, click Cut gadget, navigate to the new location, right-click, and then click  
Paste gadget.  
o  
To copy and paste the gadget, click Copy gadget, navigate to the new location, right-click, and then  
click Paste gadget.  
o  
To delete the gadget, click Delete gadget.  
Move table columns  
1.  
On the Workspace menu, confirm that Arrange workspace is enabled.  
2.  
Select the column by its heading and drag it to the new location.  
For information on displaying only a subset of the rows in the table, see Filter a table column.  
Export a table  
You can export the contents of a table into a .csv file.  
1.  
In the table, click  
.  
2.  
A file named Export.csv is download with the contents of the table.  
CCSP Dashboard Help - 7.4  
44

4: Report statistics  
This section contains the following topics:  
l  
Agent report statistics  
l  
Call report statistics  
l  
Campaign report statistics  
l  
Group report statistics  
l  
Queue report statistics  
Note  
A value of –1 for a statistic in a gadget indicates a lack of information.  
Agent report statistics  
You can insert the following statistics into agent report tables.  
Table column  
Description  
% Current Load  
((Current simultaneous calls) / (Max number of simultaneous calls allowed per agent))  
x 100  
Administrator  
Whether the user is an administrator  
Agent  
Whether the user is an agent  
Note  
CCSP also supports ReadyReps and business users.  
Agent State  
Agent's current state  
Available Time  
Available time. How long the agent has been in the Available state. Includes  
Available and Available in call.  
Average Call Duration  
Average duration of all media types  
Average Callback Call  
Duration  
Average duration of callback calls  
Average Chat Call  
Duration  
Average duration of chat calls  
Average Email Call  
Duration  
Average duration of email calls  
Average Talk Time  
(Total talk time for all calls handled by the agent) / (Number of calls handled by the  
agent)  
CCSP Dashboard Help - 7.4  
45

Table column  
Description  
Number of calls handled by the agent = total of inbound, outbound, and agent to  
agent calls.  
Average VoiceMail Call  
Duration  
Average duration of acd voicemail calls  
Average Voip Call  
Duration  
Average duration of VoIP calls  
Block Codes  
If agent is blocked from making outbound calls, the agent’s block codes  
separated by commas  
Block International Calls  
Whether the agent is blocked from making international calls  
Block Local Calls  
Whether the agent is blocked from making local calls  
Block Long Distance Calls  
Whether the agent is blocked from making long distance calls  
Call Direction  
The direction of the call the agent is in: Outbound, Inbound, Agent to Agent,  
Bridge Call, Callback Request, or Callback Response. For multichat scenarios,  
Multiple appears in this column  
Call Queue  
Queue of the call that the agent is handling (also known as the current queue).  
For multichat scenarios, Multiple appears in this column  
Call State  
Call state of the call that the agent is handling. When the agent is handling  
multiple calls, the call states appear with the number of calls in that state, for  
example, Ringing:5,In Call:2,On Hold:1,In Wrap-up:3  
Campaign Name  
The name of the campaign(s) that the agent is logged in to. For Elsbeth  
campaigns, the first campaign in the list is the agent's primary campaign.  
If not logged into a campaign, this column is empty.  
Current Call  
Type of call the agent is handling: Telephone, Internet, message call, or callback  
(also known as call type). For multichat scenarios, Multiple appears in this  
column  
Current Simultaneous  
Calls  
Number of simultaneous calls agents are handling  
Description  
Agent’s description  
First Name  
Agent's first name  
Group  
Name of agent's group  
Id  
Internal CCSP agent ID  
Interaction Profile Name  
Interaction profile name  
CCSP Dashboard Help - 7.4  
46

Table column  
Description  
Last Name  
Agent's last name  
Longest Duration Call  
Duration of the call that is the currently the longest active at the agent. Includes  
on hold time.  
Release Code  
Agent's release code  
Skills  
Agent's skills  
Supervisor  
Whether the agent is a supervisor  
Teams  
Agent’s teams, separated by commas  
Time in Agent State  
Amount of time the agent has been in the current Agent state  
Time in Call State  
Amount of time the agent has been in the current Call State. When the agent is  
handling multiple calls, this column presents 00:00:00.  
Note  
Time In Call State works only for a single call, with the following limitations:  
l  
When an agent reduces the number of calls to a single call, Time In Call State  
is reset and may be incorrect.  
l  
When an agent transfers a call, Time In Call State for the second agent is reset  
after the first agent hangs up the call.  
Time in Release Code  
Time in release code.  
Note  
This attribute only appears when the agent is in the released state and has  
chosen a release code.  
Time Logged-in  
Total time that the agent has been logged on  
User Name  
Agent’s logon name  
CCSP Dashboard Help - 7.4  
47

You can insert the following statistics into agent report charts.  
Chart counter  
Chart type  
Description  
% Current Load  
Trend  
((Current simultaneous calls) / (Max number of simultaneous calls  
allowed per agent)) x 100  
Agents  
All  
Number of agents  
Available  
All  
Number of agents in the available state  
Available In Call  
All  
Number of agents in the available in call state  
Busy  
All  
Number of agents in the busy state  
Current Simultaneous  
Calls  
Trend  
Number of simultaneous calls agents are handling  
In Call  
All  
Number of agents in the in call state  
In Wrap-up  
All  
Number of agents in call wrap-up  
Not Logged In  
All  
Number of agents that belong to the contact center and are not  
logged on  
On Hold  
All  
Number of agents in the on hold state  
Released  
All  
Number of agents in the released state  
Released In Call  
All  
Number of agents in the released in call state  
Ringing  
All  
Number of agents in the ringing state  
CCSP Dashboard Help - 7.4  
48

Call report statistics  
You can insert the following statistics into call report tables.  
Table column  
Description  
Agent First Name  
Agent's first name  
Agent Group Name  
Group that this agent is assigned to  
Agent Id  
Internal CCSP agent ID  
Agent Last Name  
Agent's last name  
Agent User Name  
Agent's Windows username  
Agent Team Names  
Agent's teams. Multiple teams are separated by commas.  
Call Direction  
Call type. Possible values are:  
l  
Outbound  
l  
Inbound  
l  
Agent to agent  
l  
Bridged  
l  
Callback request  
l  
Callback response  
l  
PDS  
l  
Previewing  
Call Duration  
Call duration. Includes all the possible call states.  
Call Id  
Internal CCSP call ID  
Call Media  
Media type. Possible values are:  
l  
Chat  
l  
Email  
l  
Voip  
l  
CallBack  
l  
Voice Mail  
Call Queue  
Call queue  
Call State  
Call state. Possible values are:  
l  
Preview  
CCSP Dashboard Help - 7.4  
49

Table column  
Description  
l  
Ringing  
l  
In Call  
l  
On Hold  
l  
Wrap Up  
Time In Call State  
Time in call state. The number of seconds since the call moved to the 'in-call'  
state. Includes wrap up and on hold time.  
You can insert the following statistics into call report charts.  
Chart counter  
Description  
Call Back  
Total number of callback calls being handled  
Calls  
Total number of calls being handled  
Chat  
Total number of chat calls being handled  
Email  
Total number of email calls being handled  
Voicemail  
Total number of voicemail calls being handled  
Voip  
Total number of VoIP calls being handled  
CCSP Dashboard Help - 7.4  
50

Campaign report statistics  
Note  
These statistics apply only to Sytel campaigns. To view Elsbeth campaign statistics, use the EPC  
Administrator.  
You can insert the following statistics into your campaign report gadgets.  
Table column  
Chart counter  
Description  
% Calls Disposed by  
Agent  
-  
Percent of calls with outcomes determined by agent  
% Calls Disposed by  
System  
-  
Percent of calls with outcomes determined by system  
% Campaign Available  
Time  
-  
((Total available time)) / (Total logon time)) x 100  
% Campaign Talk Time  
-  
((Total talk time) / (Total logon time)) x 100  
% Campaign Wrap-up  
Time  
-  
((Total wrap-up  time)) / (Total logon time)) x 100  
% Conversion  
% Conversion  
Sales as a percent of live calls  
Abandoned  
Abandoned  
Number of abandoned calls  
Agent Calls per Hour  
-  
Number of calls handled by agents per hour  
Agents Logged-in  
Time  
Agents Logged-in  
Time  
Total logged on time of the agents in hh:mm:ss  
Answering Machine  
Agent Detected  
Answering Machine  
Agent Detected  
Number of calls to answering machines detected by  
agents  
Answering Machine  
System Detected  
Answering Machine  
System Detected  
Number of calls to answering machines detected by  
system  
Answering Machine  
Total  
Answering Machine  
Total  
Total number of answering machine calls  
Available Agents  
Available Agents  
Number of agents logged on to campaign and in available  
state  
Average Refusal  
Duration  
-  
Average duration of calls which have a disposition of  
refusal  
Average Reschedule  
Duration  
-  
Average duration of calls which have a disposition of  
reschedule  
CCSP Dashboard Help - 7.4  
51

Table column  
Chart counter  
Description  
Average Sale Duration  
-  
Average duration of calls which have a disposition of sale  
Busy  
Busy  
Number of busy calls  
Calls Dialed  
Calls Dialed  
Total number of dialed calls  
Calls Dialed per Hour  
-  
Number of dialed calls per hour  
Calls Disposed by  
Agent  
Calls Disposed by  
Agent  
Number of calls with outcomes determined by agent  
Calls Disposed by  
System  
Calls Disposed by  
System  
Number of calls with outcomes determined by system  
Cumulative Available  
Time  
Cumulative Available  
Time in Minutes  
Total minutes agents were available  
Cumulative Talk Time  
Cumulative Talk Time  
in Minutes  
Total minutes agents were in call  
Cumulative Wrap-up  
Time  
Cumulative Wrap-up  
Time in Seconds  
Total seconds agents were in wrap up  
Fast Busy  
Fast Busy  
Number of fast busy calls  
Id  
-  
Internal CCSP ID for campaign  
In-call Agents  
In-call Agents  
Number of agents logged on to campaign and in call state  
Logged-in Agents  
Logged-in Agents  
Number of logged-on agents  
Name  
-  
Name of the campaign  
No Answer  
No Answer  
Number of calls with no answer  
Number Out of Order  
Number Out of Order  
Number of calls with number out of order  
On-hold Agents  
On-hold Agents  
Number of agents logged on to the campaign and in on-  
hold state  
Other Unworkable  
Other Unworkable  
Number of calls since the campaign started with  
disposition Unworkable  
Previewing Agents  
Previewing Agents  
Number of agents logged on to the campaign and in call  
preview state  
Queue  
-  
The queue assigned to the campaign  
Refusals  
Refusals  
Number of refusals  
Released Agents  
Released Agents  
Number of agents logged on to the campaign and in the  
released state  
CCSP Dashboard Help - 7.4  
52

Table column  
Chart counter  
Description  
Reschedules  
Reschedules  
Number of rescheduled calls  
Run Time  
Campaign Run Time in  
Minutes  
Duration of the campaign in minutes in hh:mm:ss format  
from the last time it was started  
Sales  
Sales  
Number of sales  
Sales per Hour  
-  
Number of sales per hour  
Start Time  
-  
Start time of the campaign in hh:mm:ss format  
Wrap-up Agents  
Wrap-up Agents  
Number of agents in wrap up  
Wrong Number  
Wrong Number  
Number of calls to wrong number  
Wrong Person  
Wrong Person  
Number of calls to wrong person  
CCSP Dashboard Help - 7.4  
53

Group report statistics  
Note  
In multiple interaction scenarios, the Group statistics do not represent the full agent time in state  
measurements.  
Additional information about individual calls that the agent is handling is presented in Call reports.  
You can insert the following statistics into your group report gadgets.  
Table column  
Chart counter  
Description  
% Agents In Call of  
Logged In Agents  
% Agents In Call of  
Logged In Agents  
((Agents Available In Call + Agents Released In Call) / (Agents  
Logged In)) x 100  
% Agents In Callback  
Call  
% Agents In Callback  
Call  
((Number of agents in callback call) / (Total number of  
agents in call)) x 100  
% Agents In Internet  
Call  
% Agents In Internet  
Call  
((Number of agents in internet call) / (Total number of  
agents in call)) x 100  
% Agents In Message  
Call  
% Agents In Message  
Call  
((Number of agents in message call) / (Total number of  
agents in call)) x 100  
% Agents In Telephone  
Call  
% Agents In Telephone  
Call  
((Number of agents in telephone call) / (Total number of  
agents in call)) x 100  
% Avg Current Load  
% Avg Current Load  
(only in trend charts)  
Average load per agent  
-  
Agents  
Total number of agents in the group  
Available  
Available  
Number of agents in the available state  
-  
Available In call  
Number of agents in the available in call state  
Avg Current  
Simultaneous Calls  
Avg Current  
Simultaneous Calls  
(only in trend charts)  
Average simultaneous calls per agent  
-  
Busy  
Number of agents in the busy state  
Id  
-  
Internal CCSP group ID  
In Call  
In Call  
Number of agents that are in call or pending call transfer  
In Wrap-up  
In Wrap-up  
Number of agents in the wrap-up state  
Logged In  
Logged in  
Number of agents logged on  
Longest Duration Call  
-  
Duration of the call that is the currently the longest  
CCSP Dashboard Help - 7.4  
54

Table column  
Chart counter  
Description  
active in the group. Includes on hold time.  
Name  
-  
Name of group  
Not logged In  
Not logged In  
Number of agents logged off  
On Hold  
On Hold  
Number of agents in the on hold state  
Released  
Released  
Number of agents in the released (inactive) state  
-  
Released In Call  
Number of agents in the released in call state  
Ringing  
Ringing  
Number of agents in the ringing state  
Time Available  
-  
Amount of time that the agents are in the available state  
Time Busy  
-  
Amount of time that the agents are in the busy state  
Time in Call  
-  
Amount of time that the agents are in the in call state  
Time in Logged-in  
-  
Amount of time that the agents are in the logged-in  
state  
Time in On-hold  
-  
Amount of time that the agents are in the on-hold state  
Time in Released  
-  
Amount of time that the agents are in the released state  
Time in Ringing  
-  
Amount of time that the agents are in the ringing state  
Time in Wrap-up  
-  
Amount of time that the agents are in the wrap-up state  
CCSP Dashboard Help - 7.4  
55

Queue report statistics  
Note  
After StatisticServer failover, only the following queue counters are recovered:  
l  
Being Handled  
l  
Handled (only for calls currently in the system)  
l  
Offered (the recovered counter includes the Overflowed In counter values before the failover)  
l  
WaitingInQueue  
The counters are recovered only for the calls currently in the system that are not in their termination  
phase.  
All recovered counters are placed in the current hour slot.  
You can insert the following statistics into your queue report gadgets.  
Table column  
Chart counter  
Description  
Abandoned  
Calls Abandoned  
Number of calls offered to the contact center that were  
abandoned  
Abandoned (TQOS)  
Calls Abandoned within  
TQoS  
Number of calls offered to the contact center that were  
abandoned within the specified target quality of service  
Abandoned in %  
Calls Abandoned in %  
((Number of abandoned calls) / (Total calls handled + calls  
abandoned)) \* 100  
Assigned Service  
Level  
Assigned Service Level in  
%  
Target service level assigned to the queue  
Average Call  
Duration  
Average Call Duration  
(only in trend charts)  
(Total talk time for all calls in a queue) / (Number of calls  
answered)  
Average Speed of  
Answer  
Average Speed of Answer  
(only in trend charts)  
Average amount time it takes an agent to answer a call  
Average Waiting  
Time  
-  
(Sum of all waiting time for all calls in a particular queue for  
the selected media) / (Number of all calls in a particular  
queue for the selected media)  
Note  
Average waiting time is computed only for calls waiting in  
the queue and does not include the time until an overflow  
occurs, if it occurs.  
Being Handled  
Calls Being Handled  
Number of calls being answered by agents  
Calls Outside TQOS  
Calls Outside TQOS Level  
(Calls handled and being handled + Calls abandoned - Calls  
CCSP Dashboard Help - 7.4  
56

Table column  
Chart counter  
Description  
Level  
in %  
handled within TQOS - Calls abandoned within TQOS - Calls  
pulled from defer and non-queued items pulled from public))  
\* 100 / (Calls handled and being handled + Calls abandoned)  
Note  
If there are no calls, the service level is 0%.  
Cumulative Wait  
Time  
-  
Total wait time of answered calls in queue  
Note  
This total excludes abandoned calls.  
Groups  
-  
List of groups for each queue  
Handled  
Calls Handled  
(Number of calls that were answered) + (Calls being answered  
by agents)  
Handled (TQOS )  
Calls Handled Within  
TQOS  
Number of calls answered within the target quality of  
service (TQOS ) threshold  
Id  
-  
Internal CCSP Queue ID  
In-queue Timeout  
-  
The (configured) in-queue timeout value  
Longest Time  
-  
Longest wait time of a call currently in queue, the oldest  
call  
Name  
-  
Name of the queue  
Offered  
Calls Offered  
Number of calls offered to the queue  
Overflowed In  
Calls Overflowed In  
Number of calls re-queued from another queue to this  
queue  
Overflowed Out  
Calls Overflowed Out  
Number of calls re-queued from this queue to another  
queue  
Pulled Non-queued  
item from public  
Pulled Non-queued item  
from public  
Number of unqueued calls pulled from public  
Pulled Queued item  
from public  
Pulled Queued item from  
public  
Number of queued calls pulled from public  
Pick-up Timeout  
-  
Pick-up timeout value configured for the queue  
Rejected  
Calls Rejected  
Number of calls rejected by agents or by the system  
Service Level  
Service Level in %  
(Number of calls handled within TQOS / (Number of calls  
handled and being handled + Number of abandoned calls -  
Number of calls abandoned within TQOS - Number of calls  
pulled from defer and non-queued items pulled from public))\*  
100  
CCSP Dashboard Help - 7.4  
57

Table column  
Chart counter  
Description  
Note  
If there are no calls, the service level is 0%.  
Target TQOS  
-  
The target quality of service assigned to the queue, in  
seconds  
Total Wait Time  
-  
Total wait time of calls in queue  
Note  
This total includes abandoned calls.  
Waiting in Queue  
Calls Waiting in Queue  
Number of calls in the queue waiting to be answered  
Wrap-up Timeout  
-  
Wrap-up timeout value configured for the queue  
CCSP Dashboard Help - 7.4  
58

# Contact Center: Service Provider Designer Help - Tenant Version 7.4

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  
11  
Support  
11  
2: Overview  
12  
Log in  
12  
Password expired message  
13  
Additional login messages  
13  
Reseller view  
14  
Select the tenant to manage  
14  
Return to the Tenants page  
14  
Get started page  
15  
Errors page  
17  
Save current errors to a text file  
17  
Clear current errors  
17  
Navigation pane  
18  
List view  
19  
Call flow  
20  
Predefined events  
21  
Flowchart of events  
23  
How to use audio files  
25  
How to use speech  
26  
How to use video  
27  
Naming format for video files  
27  
How to record calls  
28  
How to use variables and constants  
29  
How to use variables  
29  
How to use the $empty$ constant  
30  
Log out  
31  
Additional logout messages  
31  
Session expiry  
31  
3: Workspaces  
32  
Workspace editor overview  
32  
Workspace tab bar  
33  
Script tabs panel  
35  
Script blocks panel  
36  
Flow editor  
36  
CCSP Designer Help - Tenant - 7.4  
2

Properties panel  
39  
Manage workspace templates  
41  
Select a template  
42  
Delete a custom-tenant template  
42  
Open a workspace  
43  
Import a workspace  
44  
Manage workspaces  
46  
Save a copy of the workspace  
46  
Save a workspace as a tenant template  
47  
Revert a workspace  
48  
Reload the last published workspace  
49  
Delete a workspace  
50  
Publish workspaces  
51  
Location to publish to  
51  
Workspace versions  
51  
Publish a workspace  
52  
Manage tables  
53  
Upload a new table  
54  
Upload an updated table  
55  
Download a table  
55  
Publish a table  
56  
Rename a table  
57  
Delete a table  
57  
Manage workspace files  
58  
Upload a new file  
59  
Upload an updated file  
60  
Download a file  
60  
Publish a file  
61  
Rename a file  
62  
Delete a file  
62  
Manage version history  
63  
Open a snapshot  
64  
Edit a version's description  
65  
Republish a version  
66  
4: Scripts  
68  
Manage scripts  
68  
Add a new script  
70  
Add a system script  
70  
Duplicate a script  
71  
Rename a custom script  
71  
Save a script  
72  
Publish a script  
72  
Validate a script  
73  
Delete a custom script  
73  
How to iterate through scripts  
74  
CCSP Designer Help - Tenant - 7.4  
3

Example scripts  
75  
Ivr script  
76  
Agentless IVR script  
80  
IVR Timeout script  
82  
Call processing scripts  
83  
Elsbeth integration scripts  
99  
Music On Hold script  
101  
Music On Hold In Queue script  
102  
In Queue Timeout script  
105  
Pickup Timeout script  
105  
Voicemail processing scripts  
105  
Web services XML examples  
115  
5: Blocks  
117  
Block availability per CallXML script  
118  
Abort  
123  
Add Skill  
123  
Agent Selection Rule  
124  
Agentless Campaign Outcome  
125  
Answer  
126  
Assign  
127  
Bridge Call  
128  
Call Maturity/Priority  
129  
Caller Message  
130  
Caller Name  
130  
Caller URL  
131  
Check Date  
131  
Configure the Check Date block  
131  
Check Time  
134  
Configure the Check Time block  
134  
Clear Digits  
135  
Clear Digits Events  
135  
Clear User Data  
136  
Condition  
136  
Define Grammar  
137  
Routing  
138  
Backward compatibility  
139  
Edit Email  
140  
Get Agent Statistics  
142  
Get Digits  
145  
Get Phone Information  
147  
Information fields  
147  
Get Queue Statistics  
149  
Get User Data  
151  
GoTo  
152  
GoTo Voicemail  
152  
CCSP Designer Help - Tenant - 7.4  
4

Hangup  
153  
IVR Recording  
154  
Examples  
155  
IVR Transfer Recording  
157  
Leave Voicemail  
158  
Examples  
160  
Log  
161  
Logging Control  
162  
Mask Digits  
163  
MD5 Encryption  
163  
Menu  
164  
Configure the Menu block  
165  
Optional Parameters  
167  
Real-Time Speech Analytics  
167  
CCSP UI CustomRing parameter  
167  
Percent Allocated  
169  
Configure the Percent Allocated block  
169  
Play Audio  
170  
Play Date  
171  
Play Greeting  
173  
Play Money  
174  
Play Number  
175  
Play Text  
176  
Play Video  
178  
Implementation  
178  
Playback Audio  
179  
Queue Name  
180  
Read SIP Header  
181  
Record Audio  
182  
Record Video  
184  
Record Voicemail  
185  
Recorder  
187  
Rules table  
188  
Remove Skill  
189  
ReQueue  
189  
Routing Mode  
190  
Routing Properties  
191  
Run  
191  
Schedule Callback  
192  
Script Entry Point  
194  
Send Callback  
195  
Send Email  
197  
Send User Data  
199  
Set Defer Type  
199  
Set Event  
200  
Set Language  
201  
Set SIP Header  
202  
CCSP Designer Help - Tenant - 7.4  
5

Speech Enabled Menu  
203  
Configure the Speech Enabled Menu block  
205  
Speech Play and Collect  
207  
Configure the Speech Play and Collect block  
209  
Stop Video  
210  
Supervised Transfer  
211  
Configure the Supervised Transfer block  
213  
Switch  
214  
Configure the Switch block  
214  
Table  
216  
Table Lookup  
216  
Terminate  
217  
Transfer by Refer  
218  
Configure the Transfer by Refer block  
219  
Transfer Call  
220  
UnQueue  
222  
VXML (VoiceXML) Transfer Call  
223  
Wait for Events  
225  
Wait  
225  
Web Services  
226  
6: Objects  
227  
New model for messaging calls  
227  
Common media API  
228  
CallbackCall object  
232  
EmailCall object  
233  
OutgoingEmailCall object  
233  
VoicemailCall object  
233  
Comparison with CallData object  
238  
CallData object  
243  
Properties from SIP headers  
245  
Playback-related properties  
247  
Determining the Calldata.HideAniFromAgent value  
248  
CCLEvent object  
249  
CosmoCall object  
249  
CosmoCall.CallbackData object  
250  
CosmoCall.TtsInfo object  
251  
Line object  
251  
NewLine object  
252  
Session object  
252  
7: CallXML  
253  
General elements  
254  
assignvar  
254  
clear var  
254  
CCSP Designer Help - Tenant - 7.4  
6

cleardigits  
254  
getDigits var  
254  
goto value  
254  
run value  
255  
sendEvent value  
255  
Call control elements  
255  
answer  
255  
call value  
255  
conference targetSessions  
255  
hangup  
256  
playAudioformat  
256  
playDateformat  
256  
playMoneyformat  
257  
playNumber  
257  
recordAudioformat  
257  
text format  
258  
waitForConferenceEnd  
258  
Test and debug elements  
258  
CallXML high level elements  
258  
block label  
259  
menu label  
259  
inputAudio label  
259  
inputDigitslabel  
259  
Additional Designer elements  
260  
addSkillvalue  
260  
arrivalTimeBiasvalue  
260  
callerMessagevalue  
260  
callerNamevalue  
260  
callSetNamevalue  
260  
callerURLvalue  
260  
conditionalexpr  
261  
optionalParametervar  
261  
removeSkillvalue  
261  
sendEmail server  
261  
unQueue and reQueue  
262  
CCSP Designer Help - Tenant - 7.4  
7

1: Intro  
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. 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.  
What's new  
Version 7.4  
Publish workspaces — added information about workspace versions. Requires HF74-64557.  
Manage version history — Designer automatically creates up to 10 versions (snapshots) of each published  
workspace. Requires HF74-64557.  
Manage workspaces — you can revert a workspace to a previous version (snapshot) or reload the last  
published workspace from the network. Requires HF74-64557.  
Log in and Log out — added process for CCSP single sign-on. Requires HF74-63812.  
Reseller view, Get started page, and Navigation — added reseller administrator view. Requires HF74-61855.  
Errors page — added page. Requires HF74-62208.  
Manage workspace files — you can upload .txt, .xml, and .wav files to the workspace. Requires HF74-62208.  
Manage scripts — you can add a system script. Requires HF74-62208.  
Post Call script — added script. Requires HF74-61186.  
Blocks — block names must not exceed 256 characters in length. Requires HF74-62208.  
Check date block — the default location of Holiday.xml is <installation directory>/CosmoCom/Server  
Components/Designer. Requires HF74-62208.  
IVR Recording block — added block. Requires HF74-61950.  
Read SIP Header block — added block. Requires HF74-61777.  
Set SIP Header block — added block. Requires HF74-61777.  
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.  
CCSP Designer Help - Tenant - 7.4  
8

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 workspace.xml  
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:  
l  
Custom — created by the user with properties and behavior that can be edited as  
required  
l  
System — installed with the Designer application that has predefined properties and  
behavior, with only the property values editable by the user.  
CCSP Designer Help - Tenant - 7.4  
9

Term  
Definition  
call parameter  
A call is associated with these parameters: TQoS, skills required, queue destination, caller  
name, URL, and any number of optional parameters.  
CCSP UI  
CCSP 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 CCSP.  
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  
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 CCSP.  
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.  
CCSP Designer Help - Tenant - 7.4  
10

Term  
Definition  
workspace  
A set of scripts that relate to a single location that can be associated with an application  
entity in CCSP Admin.  
wrap up code  
Code that represents the result of a call.  
VCS  
Voice over IP (VoIp) Connection Service — CCSP 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 CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
11

2: Overview  
This section contains the following topics:  
l  
Log in  
l  
Reseller view  
l  
Get started page  
l  
Errors page  
l  
Navigation pane  
l  
List view  
l  
Call flow  
l  
Predefined events  
l  
Flowchart of events  
l  
How to use audio files  
l  
How to use speech  
l  
How to use video  
l  
How to record calls  
l  
How to use variables and constants  
l  
Log out  
Log in  
The log in process is slightly different depending on the identity provider configured by your administrator.  
Note  
l  
Reseller administrators can log on to multiple tenants.  
l  
Admin users can open several tabs in the same browser with one login.  
l  
If you log in again after your CCSP 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.  
2.  
Click Go or press Enter. The Welcome to Designer page appears.  
3.  
Click Log in. If you are already logged in, the main Designer page appears.  
4.  
If you are not already logged in, the Auth Server Login Selector page appears.  
5.  
Type your User name in User Principal Name (UPN) format, for example, someone@domainname.com,  
and click Continue.  
6.  
The next step depends on the identity provider configured by your administrator:  
CCSP Designer Help - Tenant - 7.4  
12

o  
For Auth Server, the AuthServer login page appears. Type your Password and click Log in.  
Alternatively, to change the username, click Back to return to the previous page. The main Designer  
page appears. If necessary, see Password expired message below.  
o  
For ADFS, you are redirected to the ADFS login page to authenticate. Please consult your local  
procedures for how to proceed from this point.  
o  
For Enghouse Identity Server (EIS), you are redirected to the EIS login page. Type your Password and  
click Login. The main Designer page appears.  
7.  
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 Auth Server login 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:  
l  
Failed to get authorization for Designer use. Your user profile is not permitted to use the Designer application.  
Only landlord, reseller, and tenant administrators are permitted to use the Designer application.  
l  
You are not authorized to view this page. This can occur because:  
o  
You are logged in to an external identity provider with a different username than the one you entered  
on the Auth Server Login Selector page to log in to the CCSP client application.  
o  
You entered a username on the Auth Server Login Selector page and were redirected to an external  
identity provider, where you log on with a different username.  
l  
Auth Server is not available or there is a misconfiguration. You must contact your administrator. You will not be  
able to log in to Designer until the issue is resolved.  
CCSP Designer Help - Tenant - 7.4  
13

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  
.  
CCSP Designer Help - Tenant - 7.4  
14

Get started page  
Note  
l  
Each tenant can have multiple workspaces.  
l  
Each tenant can have two or more workspaces based on the same template.  
l  
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  
CCSP Designer Help - Tenant - 7.4  
15

Tenant view  
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.  
CCSP Designer Help - Tenant - 7.4  
16

Errors page  
Save current errors to a text file  
Clear current errors  
Note  
l  
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.  
l  
Error List data is kept until the user session ends, unless the user manually clears the list.  
l  
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.  
The possible Import or Validate action error types are:  
l  
Block name too long — the block display name exceeds 256 characters.  
l  
Duplicate name — during import of a workspace, more than 1 script with the same name has been found.  
l  
Invalid Block usage —a block exists on a script that is not valid for that script type.  
l  
Invalid Table format — a table .xml file has invalid or missing elements.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
17

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  
CCSP Designer Help - Tenant - 7.4  
18

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:  
l  
Filtered items appears in front of the total items, for example, 1-2 of 2.  
l  
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:  
.  
CCSP Designer Help - Tenant - 7.4  
19

Call flow  
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 CCSP, call flow is determined by:  
l  
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.  
l  
The CCSP Call Management Service response.  
Using 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 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.  
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 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 CCSP service providers, and their tenants, to quickly edit scripts without the need for VBScript  
programmers or CCSP system administrator intervention.  
CCSP Designer Help - Tenant - 7.4  
20

Predefined events  
In every CCSP 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  
CCSP 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 .wav 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  
CCSP Designer Help - Tenant - 7.4  
21

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 CCSP 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 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.  
CCSP Designer Help - Tenant - 7.4  
22

Flowchart of events  
The following diagram shows the flow of the CCSP call events. For a description of the CCSP events, see  
Predefined events.  
CCSP Designer Help - Tenant - 7.4  
23

CCSP Designer Help - Tenant - 7.4  
24

How to use audio files  
CCSP has the ability to play audio (.wav) files as part of the call flow (scripts). CCSP 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:  
l  
Preferred quality — call quality  
l  
Format — PCM  
l  
Attributes — 8,000 kHz, 16 Bit, Mono  
Note  
l  
The path specifying a .wav file must be 80 characters or less in length. This limitation includes the file  
name.  
l  
The URL can include the http or https protocol.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
25

How to use speech  
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:  
l  
Text-to-Speech (TTS) — an application that synthesizes a spoken sound version of text.  
l  
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:  
l  
Set Language  
l  
Speech Play and Collect  
l  
Speech Enabled Menu  
l  
Define Grammar  
l  
Play text  
Note  
Your system administrator should tell you which languages are available for speech enabled scripts.  
To use ASR, you must first develope a grammar. A grammar specifies the words and patterns of words to be  
listened for by a speech recognizer. CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
26

How to use video  
Note  
IVVR scripts only affect calls that have a voice component.  
Designer provides blocks that you can use to process video calls for Interactive Voice and Video Response  
(IVVR).  
These blocks are:  
l  
Play Video  
l  
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:  
l  
CIF — <filename>cif<bitrate>\_Kbps<framerate>\_fps.avi  
For example, greeting\_cif\_256Kbps\_30fps.avi  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
27

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:  
l  
RecordTransfer  
o  
True — call recording continues on transfer (default)  
o  
False — call recording stops on transfer  
l  
RecordConference  
o  
True — call recording continues on conference (default)  
o  
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:  
l  
setTransferRecordingEx(RecordTransfer [Boolean],RecordConference [Boolean]) — sets each  
flag individually.  
l  
setTransferRecording([Boolean]) — sets both flags to the same setting. This function is maintained  
for backward compatibility.  
For example:  
l  
If you call setTransferRecordingEx(false,true) from a Run block in the OutgoingCall script, only  
conferenced calls are recorded.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
28

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 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 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 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:  
CCSP Designer Help - Tenant - 7.4  
29

"$CallerEntry$" = "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.  
CCSP Designer Help - Tenant - 7.4  
30

Log out  
Note  
If you are using an external identity provider such as ADFS or EIS, when you click Log out of your CCSP  
session, you are also logged out of your external identity provider.  
1.  
At the top right of the main Designer page, click Logout.  
2.  
Do one of the following:  
o  
If you want to return to Designer, click Return to Login page.  
o  
If you have finished working with CCSP, click Log out of your CCSP session.  
Additional logout messages  
When you click Log out of your CCSP session, the following message may appear:  
l  
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 CCSP session again.  
The application will return to the login page. Depending on the problem resolution, this may take slightly  
longer than normal.  
Session expiry  
Your CCSP application session will become invalid when:  
l  
The login session time exceeds the maximum CCSP login session lifetime specified by your administrator.  
l  
You log out of your CCSP session using another application.  
l  
You clear all cookies in the browser.  
When your CCSP 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 CCSP login session lifetime, you only have to enter your username on the Auth Server Login  
Selector page. You will then automatically be authenticated against your external identity provider.  
CCSP Designer Help - Tenant - 7.4  
31

3: Workspaces  
This section contains the following topics:  
l  
Workspace editor overview  
l  
Manage workspace templates  
l  
Open a workspace  
l  
Import a workspace  
l  
Manage workspaces  
l  
Publish workspaces  
l  
Manage tables  
l  
Manage workspace files  
l  
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  
CCSP Designer Help - Tenant - 7.4  
32

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.  
CCSP Designer Help - Tenant - 7.4  
33

Workspace tabs  
To save the currently selected workspace, click Save.  
Workspace menu  
Use the workspace menu  
as follows:  
l  
Rename — to rename the currently selected workspace. See Rename a workspace.  
l  
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.  
l  
Save as Platform Template (landlord only) — to save the currently selected workspace as type Custom-  
platform. See Save a workspace as a Platform Template.  
l  
Save as Tenant Template — to save the currently selected workspace as type Custom-tenant. See Save a  
workspace as a Tenant Template.  
l  
Publish — to publish the currently selected workspace. See Publish a workspace.  
l  
Delete — to delete the currently selected workspace. See Delete a workspace.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
34

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 CCSP 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:  
l  
— indicates a system script.  
l  
— indicates a system script that is currently selected.  
l  
— indicates a custom script added by a user.  
l  
— 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:  
l  
Add script — to add a new script. See Add a new script.  
l  
Add system script — to add a system script. See Add a system script.  
l  
Duplicate — to duplicate the currently selected script.  
l  
Save — to save the currently selected script.  
l  
Publish — to publish the currently selected script. See Publish a script.  
l  
Validate — to validate the currently selected script. See Validate a script.  
l  
Delete — to delete the currently selected script. The Delete option is disabled for system scripts.  
CCSP Designer Help - Tenant - 7.4  
35

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:  
l  
Icons — the icons are arranged in the same order as the legacy Designer blocks palette. When you point to  
a block, its name appears in a tooltip.  
l  
Names — the icons with their names are arranged in the same order as the legacy Designer blocks palette.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
36

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:  
l  
Copy  
l  
Duplicate  
l  
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:  
l  
Undo  
l  
Redo  
l  
Paste  
The actions are individually disabled if there is currently nothing to undo, redo, or paste.  
CCSP Designer Help - Tenant - 7.4  
37

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:  
l  
Block creation (including paste of copied blocks)  
l  
Block deletion  
l  
Block move  
l  
Block rename  
l  
Block property change  
l  
Connection creation  
l  
Connection deletion  
CCSP Designer Help - Tenant - 7.4  
38

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:  
l  
Workspace ID — new unsaved workspaces have a negative ID.  
l  
Name — the name of the workspace.  
l  
Description — the description of the workspace. This information can be changed in the workspace editor.  
l  
Version — the version number of the workspace. This number is incremented each time you save the  
workspace or an individual script.  
l  
Designer version — the version of Designer that the workspace was saved in.  
l  
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.  
l  
Last modified by — the name of the person who last saved the workspace.  
l  
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.  
l  
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:  
l  
Workspace ID  
l  
Workspace snapshot ID  
l  
Workspace snapshot version — the version number of this snapshot.  
l  
Workspace snapshot description — the description of this snapshot. This information can be edited on the  
Version History page. See Edit a snapshot's description.  
CCSP Designer Help - Tenant - 7.4  
39

l  
Name — the name of the workspace.  
l  
Description — the description of the workspace.  
l  
Version — the version number of the workspace.  
l  
Designer version — the version of Designer that the workspace was saved in.  
l  
Last modified time — this is currently the UTC time at which the workspace was saved for this snapshot.  
l  
Last modified by — the name of the person who saved the workspace at this point.  
l  
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.  
l  
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:  
l  
Script ID — new unsaved scripts have a negative ID.  
l  
Name  
l  
Description — this information can be edited.  
l  
Designer version — the version of Designer that the script was saved in.  
l  
Script version — the version of the script. This number is incremented each time you save the workspace or  
an individual script.  
l  
Last modified time — the time is updated when you save the whole workspace or the individual script.  
l  
Last modified by  
l  
Last publish time — the time is updated when you publish the whole workspace or an individual script.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
40

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.  
The columns are:  
o  
Name  
o  
Description  
o  
Type — system, custom-platform, or custom-tenant.  
o  
Last modified time — this is currently the UTC time at which the workspace template was saved in  
the database.  
o  
Last modified by  
CCSP Designer Help - Tenant - 7.4  
41

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.  
CCSP Designer Help - Tenant - 7.4  
42

Open a workspace  
Note  
l  
You can open a recently used workspace directly from the Get started page.  
l  
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.  
The columns are:  
o  
Name — the name of the workspace.  
o  
Description — the description of the workspace. This information can be changed in the workspace  
editor.  
o  
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.  
o  
Last modified by — the name of the person who last saved the workspace.  
o  
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 CCSP Admin. The time is updated when you publish the whole  
workspace but not when you publish an individual script, table, or workspace file.  
o  
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:  
o  
Click the workspace to load. Next see Workspace editor overview.  
o  
Hover your mouse over the workspace and click  
Version history. See Manage version history.  
CCSP Designer Help - Tenant - 7.4  
43

Import a workspace  
Note  
l  
Import of a workspace should only be done as part of the process of migrating a tenant from the legacy  
to the new Designer.  
l  
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.  
l  
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.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
44

7.  
If the import fails, click See details on the Error Page. See Errors page.  
CCSP Designer Help - Tenant - 7.4  
45

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.  
CCSP Designer Help - Tenant - 7.4  
46

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.  
CCSP Designer Help - Tenant - 7.4  
47

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.  
CCSP Designer Help - Tenant - 7.4  
48

Reload the last published workspace  
Note  
l  
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.  
l  
To avoid losing any unpublished changes in the workspace, you can backup this version before  
importing the workspace from the network.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
49

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.  
CCSP Designer Help - Tenant - 7.4  
50

Publish workspaces  
Location to publish to  
Workspace versions  
Publish a workspace  
Location to publish to  
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 CCSP Admin.  
The IVR root path must be a valid network or local filesystem location that is accessible to the user under which  
the 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  
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.  
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.  
CCSP Designer Help - Tenant - 7.4  
51

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.  
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.  
CCSP Designer Help - Tenant - 7.4  
52

Manage tables  
Upload a new table  
Upload an updated table  
Download a table  
Publish a table  
Rename a table  
Delete a table  
Note  
l  
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.  
l  
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.  
l  
You select these tables in the Table Name property of a Table or Table Lookup block.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
53

The columns are:  
o  
Name  
o  
Source file  
o  
Version — the version of the source file.  
o  
Rows  
o  
Last modified time — this is currently the UTC time at which the table was saved in the database.  
o  
Last modified by  
o  
Last publish time — this is currently the UTC time at which the table was published in the tenant’s  
IVR root path, configured in CCSP Admin.  
Upload a new table  
Note  
l  
The table file you upload to the database can be an .xml file or a tab-delimited text file.  
l  
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.  
l  
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.  
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.  
CCSP Designer Help - Tenant - 7.4  
54

Upload an updated table  
1.  
In the Tables list, hover your mouse over the table to upload and click  
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.  
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.  
CCSP Designer Help - Tenant - 7.4  
55

Publish a table  
Note  
l  
If an error message appears saying that the IVR root path cannot be found, or CCSP Admin cannot be  
contacted, click  
Publish again. If there is still a problem, see the configuration information in Publish  
a workspace.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
56

Rename a table  
1.  
In the Tables list, hover your mouse over the table to rename and click  
Rename. The Rename [name]  
? dialog appears.  
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.  
CCSP Designer Help - Tenant - 7.4  
57

Manage workspace files  
Upload a new file  
Upload an updated file  
Download a file  
Publish a file  
Rename a file  
Delete a file  
Note  
l  
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.  
l  
You can upload .xml, .txt, and .wav files to the workspace.  
l  
You select these files in your workflow blocks.  
l  
Table .xml or .txt files must be uploaded on the Tables page. See Manage tables.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
58

The columns are:  
o  
Name  
o  
Source file  
o  
Size  
o  
Last modified time — this is currently the UTC time at which the file was saved in the database.  
o  
Last modified by  
o  
Last publish time — this is currently the UTC time at which the file was published in the tenant’s IVR  
root path, configured in CCSP Admin.  
Upload a new file  
1.  
On the Workspace Files page, click  
Add. The Upload Workspace File dialog appears.  
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.  
CCSP Designer Help - Tenant - 7.4  
59

Upload an updated file  
1.  
In the Workspace Files list, hover your mouse over the file to upload and click  
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.  
2.  
Click Download. The workspace file appears in your Downloads folder.  
CCSP Designer Help - Tenant - 7.4  
60

Publish a file  
Note  
l  
If an error message appears saying that the IVR root path cannot be found, or CCSP Admin cannot be  
contacted, click  
Publish again. If there is still a problem, see the configuration information in Publish  
a workspace.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
61

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.  
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.  
CCSP Designer Help - Tenant - 7.4  
62

Manage version history  
Open a snapshot  
Edit a version's description  
Republish a version  
Note  
l  
Designer automatically creates versions (snapshots) of each published/republished workspace. Designer  
stores a maximum of 10 versions in the database and then replaces the oldest existing version with the  
new one.  
l  
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.  
l  
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:  
o  
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.  
o  
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.  
CCSP Designer Help - Tenant - 7.4  
63

The columns are:  
o  
Version — the version number of this snapshot.  
o  
Description — the description of this snapshot.  
o  
Last modified time — this is currently the UTC time at which the workspace was saved for this  
snapshot.  
o  
Last modified by — the name of the person who saved the workspace at this point.  
o  
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 CCSP Admin.  
o  
Last published by — the name of the person who published the workspace and created this  
snapshot.  
o  
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.  
CCSP Designer Help - Tenant - 7.4  
64

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.  
2.  
Edit the version description.  
3.  
Click OK.  
CCSP Designer Help - Tenant - 7.4  
65

Republish a version  
Note  
l  
Republishing a version overwrites the workspace's current data with the data of the selected version.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
66

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.  
CCSP Designer Help - Tenant - 7.4  
67

4: Scripts  
This section contains the following topics and subsection:  
l  
Manage scripts  
l  
How to iterate through scripts  
l  
Example scripts  
See also:  
l  
Script tabs panel in the Workspace editor overview.  
l  
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  
l  
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.  
l  
You manage scripts through the Workspace editor.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
68

The columns are:  
o  
Name  
o  
Published filename  
o  
Version — the version of the script.  
o  
Last modified time — this is currently the UTC time at which the script was saved in the database.  
o  
Last modified by  
o  
Last publish time — this is currently the UTC time at which the script was published in the tenant’s  
IVR root path, configured in CCSP Admin.  
CCSP Designer Help - Tenant - 7.4  
69

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.  
CCSP Designer Help - Tenant - 7.4  
70

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.  
CCSP Designer Help - Tenant - 7.4  
71

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  
l  
If an error message appears saying that the IVR root path cannot be found, or CCSP Admin cannot be  
contacted, click Publish again. If there is still a problem, see the configuration information in Publish a  
workspace.  
l  
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.  
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.  
CCSP Designer Help - Tenant - 7.4  
72

Validate a script  
Designer provides script validation to check that each individual call event script contains only 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.  
CCSP Designer Help - Tenant - 7.4  
73

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  
CCSP Designer Help - Tenant - 7.4  
74

Example scripts  
In the example scripts:  
l  
The workspace file contains all of the information about the Designer workspace and its collection of XML  
scripts.  
l  
The ..\wavefiles directory contains all of the recorded .wav files used in the scripts.  
l  
The DBLookup.asp script looks up a customer ID in a database.  
This subsection contains the following example scripts:  
l  
Ivr script  
l  
Agentless Ivr script  
l  
Ivr Timeout script  
l  
Call Arrival script  
l  
Call Deferred script  
l  
Call Aged script  
l  
Call Rejected script  
l  
Call Answered script  
l  
Call Termination script  
l  
Call Pre Sent script  
l  
Post Call script  
l  
Elsbeth integration scripts  
l  
Music On Hold script  
l  
Music On Hold In Queue script  
l  
In Queue Timeout script  
l  
Pickup Timeout script  
l  
Leave Voicemail script  
l  
Retrieve Voicemail script  
l  
Voicemail processing prompts  
l  
Web services XML examples  
CCSP Designer Help - Tenant - 7.4  
75

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.  
CCSP Designer Help - Tenant - 7.4  
76

CCSP Designer Help - Tenant - 7.4  
77

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.  
CCSP Designer Help - Tenant - 7.4  
78

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 DBLookup.asp 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.  
CCSP Designer Help - Tenant - 7.4  
79

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.  
CCSP Designer Help - Tenant - 7.4  
80

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 CCSP agent (press 1).  
Agentless  
Campaign  
Outcome  
MoveToAgent  
Routes the call to a CCSP agent, based on the existing routing  
properties.  
Terminate  
Ends the call in IVR.  
CCSP Designer Help - Tenant - 7.4  
81

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.  
CCSP Designer Help - Tenant - 7.4  
82

Call processing scripts  
This subsection contains the following scripts:  
l  
Call Arrival script  
l  
Call Deferred script  
l  
Call Aged script  
l  
Call Rejected script  
l  
Call Answered script  
l  
Call Termination script  
l  
Call Pre Sent script  
l  
Post Call script  
CCSP Designer Help - Tenant - 7.4  
83

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  
CCSP Designer Help - Tenant - 7.4  
84

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.  
CCSP Designer Help - Tenant - 7.4  
85

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.  
CCSP Designer Help - Tenant - 7.4  
86

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  
CCSP Designer Help - Tenant - 7.4  
87

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.  
CCSP Designer Help - Tenant - 7.4  
88

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.  
CCSP Designer Help - Tenant - 7.4  
89

CCSP Designer Help - Tenant - 7.4  
90

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  
DMTF digit to interrupt the announcement.  
Assign  
InitCounter  
Initializes a counter called Counter to one (1).  
Play Audio  
LeaveVoicemail  
Plays the Do you want to leave a voicemail ... 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 LeaveVoiceMail.xml  
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.  
CCSP Designer Help - Tenant - 7.4  
91

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  
CCSP Designer Help - Tenant - 7.4  
92

Block  
Name  
Description  
abandoned calls in the rejected script to be reported as  
abandoned.  
Note  
If cosmocall.MarkAsAbandoned(false), or the Run block is  
missing, sets abandoned calls in the rejected script to be  
reported as rejected.  
CCSP Designer Help - Tenant - 7.4  
93

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  
l  
You cannot use the Recorder block in this script.  
l  
CallAnswered.xml is not supported in preview campaigns.  
CCSP Designer Help - Tenant - 7.4  
94

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.  
CCSP Designer Help - Tenant - 7.4  
95

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.  
CCSP Designer Help - Tenant - 7.4  
96

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:  
$outgoingEmailcall.outgoingEmailData.Subject$  
l  
Subject is valid: use the Assign block to set the From  
recipient value  
l  
Subject is invalid: use the Terminate block to suppress  
the email  
CCSP Designer Help - Tenant - 7.4  
97

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  
Note  
Post Call script does not support campaign calls.  
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:  
l  
If the call was transferred to another agent and the customer is still connected after the transferred agent  
hangs up.  
l  
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.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
98

Elsbeth integration scripts  
As part of the integration between the CCSP and Elsbeth products, an audio file needs to be played back to a  
customer:  
l  
At the end of an abandoned call — PlaybackAudioCallAbandoned script  
l  
After a rebound call is queued — PlaybackAudioReboundQueued script  
l  
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. CCSP takes the URL of the file to play  
from the campaign parameters configured in CCSP 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 CCSP 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 CCSP Admin Campaign properties.  
CCSP Designer Help - Tenant - 7.4  
99

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 CCSP Admin Campaign properties.  
CCSP Designer Help - Tenant - 7.4  
100

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.  
CCSP Designer Help - Tenant - 7.4  
101

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.  
CCSP Designer Help - Tenant - 7.4  
102

The blocks function as follows.  
CCSP Designer Help - Tenant - 7.4  
103

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 LeaveVoicemail.xml  
script.  
CCSP Designer Help - Tenant - 7.4  
104

In Queue Timeout script  
This script is interpreted when the In Queue Timeout Event is fired. This script is empty.  
Pickup Timeout script  
This script is interpreted when the Pickup Timeout Event is fired. This script is empty.  
Voicemail processing scripts  
This subsection contains the following scripts and topic:  
l  
Leave Voicemail script  
l  
Retrieve Voicemail script  
l  
Voicemail processing prompts  
CCSP Designer Help - Tenant - 7.4  
105

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.  
CCSP Designer Help - Tenant - 7.4  
106

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  
C:/Program Files/Cosmocom/Server  
Components/Scripts/AllWaves/Voicemail.  
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$VoicemailWebService.asm  
x. This is the Voicemail web service URL.  
WSNamespace  
Assign variable =WSNamespace,  
value=http://CosmoCom.com/WebServices/VoicemailWS. This is  
the voicemail web service namespace.  
Condition  
ChkAgentMailboxDefined  
Conditional  
expression=$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:  
l  
StandardGreeting — greeting designed for all users in an  
organization.  
l  
PersonalGreeting — greeting recorded by a user for their  
own use.  
l  
PersonalSignature Greeting — greeting designed for all  
CCSP Designer Help - Tenant - 7.4  
107

Block  
Name  
Description  
users in an organization but with the insertion of a user's  
voice signature.  
Else branch: Go to PlayGreeting to play the standard  
greeting.  
PersonalGreeting branch: Go to PlayPersonalGreeting.  
PersonalSignaturebranch: Go to PlayPersonalSignature.  
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.  
CCSP Designer Help - Tenant - 7.4  
108

For the voice prompts used in this script, see Voicemail processing prompts.  
CCSP Designer Help - Tenant - 7.4  
109

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:  
l  
Create or change a Personal Identification Number (PIN).  
l  
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)  
CCSP Designer Help - Tenant - 7.4  
110

Flow of Message Retrieval (Page 2)  
Flow of Message Retrieval (Page 3)  
CCSP Designer Help - Tenant - 7.4  
111

For the voice prompts used in this script, see Voicemail processing prompts.  
CCSP Designer Help - Tenant - 7.4  
112

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  
And.wav  
… and …  
Arrived.wav  
Message arrived …  
DaysAgo.wav  
… days ago.  
EnterMailboxID.wav  
Please enter your mailbox ID  then press pound  
EnterMailboxPIN.wav  
Please enter your mailbox password then press pound  
From.wav  
… from…  
GetVoicemailFail.wav  
The request was unsuccessful. Please wait for few seconds.  
Goodbye.wav  
Goodbye.  
HoursAgo.wav  
… hours ago.  
InvalidMailboxIdOrPIN.wav  
The mailbox ID or password you entered is incorrect.  
InvalidSelection.wav  
You have made an invalid selection.  
LeaveVM.wav  
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.  
MainMenu.wav  
To listen to your messages press one.  
Message.wav  
… unheard message.  
MessageDeleted.wav  
Message deleted.  
MessageMenu1.wav  
To delete, press seven. To save, press nine. For more options, press  
zero  
MessageMenu2.wav  
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 .  
Messages.wav  
… unheard messages.  
MessageSaved.wav  
Message saved.  
MinutesAgo.wav  
… minutes ago.  
New.wav  
.. unheard and …  
CCSP Designer Help - Tenant - 7.4  
113

Filename  
Prompt  
NextMessage.wav  
Next Message.  
NoAgentMailboxDefined.wav  
The person you are calling does not have a mailbox defined.  
NoMessages.wav  
You have no messages.  
OldMessage.wav  
.. saved message.  
OldMessages.wav  
…saved messages.  
On.wav  
… on …  
OneDayAgo.wav  
… yesterday.  
OneHourAgo.wav  
… one hour ago.  
OneMinuteAgo.wav  
… one minute ago.  
TechnicalProblem.wav  
We are currently experiencing technical difficulties. Our apologies for  
the inconvenience. Please try again later.  
ThankYou.wav  
Thank you for leaving your message, Good bye.  
ThankYouForCalling.wav  
Thank you for calling.   
YouHave.wav  
You have …  
CCSP Designer Help - Tenant - 7.4  
114

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>  
CCSP Designer Help - Tenant - 7.4  
115

<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>  
CCSP Designer Help - Tenant - 7.4  
116

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:  
l  
Block availability per CallXML script  
See also Script blocks panel in the Workspace editor overview.  
CCSP Designer Help - Tenant - 7.4  
117

Block availability per CallXML script  
This table indicates which Designer blocks are available in particular CallXML scripts.  
Designer blocks are listed in the first column and the CallXML scripts in the subsequent column headings. The script names also correspond to CCSP 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  
CCSP Designer Help - Tenant - 7.4  
118

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 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  
Log  
Yes  
Yes  
Yes  
Yes  
Yes  
Yes  
Yes  
Yes  
Yes  
Yes  
Yes  
Yes  
Yes  
Yes  
Yes  
No  
No  
No  
Yes  
CCSP Designer Help - Tenant - 7.4  
119

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  
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  
(deprecated)  
Yes  
Yes  
No  
No  
No  
No  
Yes  
Yes  
No  
Yes  
No  
No  
No  
Yes  
No  
No  
No  
No  
Yes  
CCSP Designer Help - Tenant - 7.4  
120

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  
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 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  
Collect  
Yes  
Yes  
No  
No  
No  
No  
Yes  
Yes  
No  
Yes  
No  
No  
No  
Yes  
No  
No  
No  
No  
Yes  
Stop Video  
Yes  
Yes  
No  
No  
No  
No  
Yes  
Yes  
No  
Yes  
No  
No  
No  
Yes  
No  
No  
No  
No  
Yes  
CCSP Designer Help - Tenant - 7.4  
121

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  
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.  
CCSP Designer Help - Tenant - 7.4  
122

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  
The name of the skill to be added to the current call.  
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.  
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.  
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.  
You are not restricted to the values provided in the drop down list, so you can use  
an expression as the selected value.  
CCSP Designer Help - Tenant - 7.4  
123

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.  
CCSP Designer Help - Tenant - 7.4  
124

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 CCSP agent, based on the existing routing properties.  
For an example, see Agentless IVR script.  
CCSP Designer Help - Tenant - 7.4  
125

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 CCSP Admin, in the VCS parameters.  
Caution  
Your system administrator must create the ../VCSdebug/script\_answer\_supervision registry key and  
set it to 1.  
CCSP Designer Help - Tenant - 7.4  
126

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.  
CCSP Designer Help - Tenant - 7.4  
127

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  
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.  
CCSP Designer Help - Tenant - 7.4  
128

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 replaces 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:  
l  
TQoS — Target Quality of Service  
l  
Call Priority  
l  
Routing of calls  
CCSP Designer Help - Tenant - 7.4  
129

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  
l  
This message must not exceed 30K characters.  
l  
This message overwrites any previously configured Caller Message unless  
set to $CallData.CallerMessage$ New Message, for example.  
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.  
CCSP Designer Help - Tenant - 7.4  
130

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 CCSP.  
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:  
o  
Date  
o  
Day of week  
o  
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.  
CCSP Designer Help - Tenant - 7.4  
131

Configure date  
1.  
In the Date dialog, select a single date or a block of dates.  
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.  
CCSP Designer Help - Tenant - 7.4  
132

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  
l  
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  
l  
This file can be edited with a text editor and should be updated on a yearly basis, or as often as needed.  
CCSP Designer Help - Tenant - 7.4  
133

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.  
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.  
CCSP Designer Help - Tenant - 7.4  
134

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.  
CCSP Designer Help - Tenant - 7.4  
135

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 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.  
CCSP Designer Help - Tenant - 7.4  
136

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:  
l  
Refer to a built-in grammar.  
l  
Specify it by using XML.  
l  
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,  
request1@form-level.store.  
Comment  
Text  
User description of the block in the script.  
CCSP Designer Help - Tenant - 7.4  
137

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.  
CCSP Designer Help - Tenant - 7.4  
138

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.  
CCSP Designer Help - Tenant - 7.4  
139

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:  
l  
Low — low priority  
l  
Normal — normal priority  
l  
High — high priority  
l  
NoChange — leave as is  
Note  
l  
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.  
l  
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>  
CCSP Designer Help - Tenant - 7.4  
140

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.  
CCSP Designer Help - Tenant - 7.4  
141

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  
The location of the RTR server as a URL. For example,  
http://ais.yourcompany.com/rtrdll/  
This can be specified as an IP address.  
Caution  
This should be only the path to the RTRWeb.dll file, and should not  
include the RTRWeb.dll filename.  
RTR Timeout   
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.  
Group Name  
String  
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.  
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.  
CCSP Designer Help - Tenant - 7.4  
142

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.  
CCSP Designer Help - Tenant - 7.4  
143

Agent state  
Agent in call state  
Call type  
Initialized  
N/A  
N/A  
Available  
N/A (All)  
N/A (All)  
Ringing  
N/A (All)  
Available In Call  
In Call  
CallBack, Internet, Message, Multiple, Phone  
On Hold  
CallBack, Internet, Message, Multiple, Phone  
Preview  
Phone  
Ringing  
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  
CCSP Designer Help - Tenant - 7.4  
144

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.  
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  
CCSP Designer Help - Tenant - 7.4  
145

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.  
CCSP Designer Help - Tenant - 7.4  
146

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.  
CCSP Designer Help - Tenant - 7.4  
147

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  
This is either B for a business or C for a home consumer number.  
For B, a business, the following values appear:  
l  
.BusinessName — the name of this business  
l  
.BusinessSuffix — this field may show such information as "Inc.", "LTD", or "LLP".  
For C, a home consumer, the following values appear:  
l  
.LastName — the caller's last name  
l  
.FirstName — the caller's first name  
l  
.MiddleInitial— the caller's middle initial  
.TimeZone  
The caller's time zone  
CCSP Designer Help - Tenant - 7.4  
148

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,  
http://ais.yourcompany.com/rtrdll/  
This can be specified as an IP address.  
Caution  
This should be only the path to the RTRWeb.dll file, and should not  
include the RTRWeb.dll 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 CCSP Admin, use the toolbar to turn on Detail  
view and from the Action menu, select Customize and enable the  
First column in detail view.  
CCSP Designer Help - Tenant - 7.4  
149

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.  
CCSP Designer Help - Tenant - 7.4  
150

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:  
l  
The content type is application/ccu-integration.  
l  
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>  
CCSP Designer Help - Tenant - 7.4  
151

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 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 DBLookup.asp 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.  
CCSP Designer Help - Tenant - 7.4  
152

Hangup  
VCS only.  
This block terminates the call.  
Properties  
Values  
Description  
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).  
CCSP Designer Help - Tenant - 7.4  
153

IVR Recording  
VCS only.  
This block enables you to record parts of the IVR script.  
The block is allowed only in the Ivr 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  
l  
The IVR recording includes everything a customer says while IVR recording is on.  
l  
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.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
154

Examples  
In the following example, the script records the Play 1 and Play 3 blocks, but not the Play 2 block.  
CCSP Designer Help - Tenant - 7.4  
155

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.  
CCSP Designer Help - Tenant - 7.4  
156

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.  
CCSP Designer Help - Tenant - 7.4  
157

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:  
l  
beep\_short.wav  
l  
beep.wav (default)  
l  
beep\_long.wav  
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.  
CCSP Designer Help - Tenant - 7.4  
158

Properties  
Values  
Description  
All recorded files are in MP4 format.  
Note  
Only the MP4 file is saved, not the call information file.  
ACD Voicemail  
String  
Name of the tenant's voicemail folder. If left blank, the tenant's  
default voicemail folder is used (default).  
Interaction will ring at an agent as an ACD voicemail.  
If set to none and Recipient or CC properties match the rules for  
ACD voicemail, ACD voicemail is created with SMTP services only.  
If left blank or has a value other than none, ACD voicemail is  
created without SMTP services.  
Note  
See Examples below.  
Segregation  
Group Id  
Integer  
Group ID used to filter voicemail items in Supervisor Message  
View.  
Left blank by default.  
From  
String  
When sending ACD voicemail over SMTP or when sending  
voicemail as an email with an attachment, specify the email  
address of the sender.  
Recipients  
String  
Email addresses of recipients to which the voicemail will be sent  
over SMTP.  
Multiple email addresses can be delimited by a semi-colon (;).  
If a recipient email address starts with voicemail, an ACD  
voicemail interaction is created and rings at an agent as a  
voicemail.  
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.  
Left blank by default.  
Note  
See Examples below.  
CC  
String  
Email addresses of recipients that will receive a carbon copy of the  
voicemail being sent over SMTP.  
For additional rules, see the Recipients property.  
Note  
See Examples below.  
Subject  
String  
When sending voicemail as an email with an attachment, specify  
the email subject.  
Body  
String  
When sending voicemail as an email with an attachment, specify  
the email body text.  
CCSP Designer Help - Tenant - 7.4  
159

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:  
l  
folder@tenant.loc — the email address of a tenant’s CCSP folder defined in CCSP Admin  
l  
voicemailFolder@tenant.loc — the email address of a tenant’s folder for incoming voicemail only  
l  
someone@home.loc — the email address of a recipient external to CCSP  
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:  
l  
ACD Voicemail = none AND Recipient is empty AND CC is empty  
l  
ACD Voicemail = none AND Recipient/CC fields specify the email address of a tenant folder or the email  
address of a recipient external to CCSP  
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  
CCSP Designer Help - Tenant - 7.4  
160

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 CCSP service provider for guidelines in using Log blocks in Designer scripts.  
Properties  
Values  
Description  
Message  
String  
Use for debugging purposes only.  
Messages starting with the prefix:  
l  
"ERROR:", "WARNING:", or "INFO:" are printed in log files and also  
automatically placed in the Windows Event Log.  
l  
"DEBUG:" are printed only in log files.  
l  
"TRACE:" are printed only in log files, provided that trace is enabled in CCSP  
Admin.  
Messages without one of the above prefixes are written in log files, regardless of  
the level set in CCSP Admin (unless the setting is None).  
The log files are:  
l  
Designer’s dedicated log-per-call file (must be enabled in CCSP Admin)  
l  
Syslog log file (must be enabled in CCSP Admin)  
Note  
Prefixes must be uppercase.  
CCSP Designer Help - Tenant - 7.4  
161

Logging Control  
This block turns logging on or off for the duration of the current event.  
At the end of the CallTermination event, CCSP emails the entire log file as an attachment to the designated  
Recipient.  
Note  
l  
If you want to turn logging on or off for the whole call, you must put a Logging Control block in every  
event script.  
l  
If script logging is globally enabled and you disable logging via a Logging Control block for a certain  
event script, CCSP 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.  
l  
In the event script, logging follows the global script logging settings until CCSP 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 mailbox@domainname  
Ask your CCSP 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 mailbox@domainname  
This can be an email address outside of the CCSP  
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.  
CCSP Designer Help - Tenant - 7.4  
162

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,  
http://www.abc.com/soap/GetClientInformation.asmx  
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 Designer log as follows:  
No response was received from  
http://webserver/webservices/test.asmx. 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.  
CCSP Designer Help - Tenant - 7.4  
163

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.  
CCSP Designer Help - Tenant - 7.4  
164

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.  
CCSP Designer Help - Tenant - 7.4  
165

6.  
Repeat steps 1-5 to add the branches you need.  
7.  
Connect the Else branch  
to the appropriate block in the script.  
CCSP Designer Help - Tenant - 7.4  
166

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 EVCConfiguration for each call that you  
want processed. This parameter contains the name of the configuration to use.  
The actual value does not matter to CCSP, 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.  
CCSP UI CustomRing parameter  
Note  
l  
CCSP UI was previously called TouchPoint.  
l  
If an agent selects auto-answer then CCSP does not play the ring tone. This setting is only for CCSP UI,  
not IP phones or alternative devices.  
A unique optional parameter is available for CCSP UI: CustomRing.  
CCSP Designer Help - Tenant - 7.4  
167

Use this optional parameter to send a customized ring file location (URL) when a call is delivered to CCSP UI.  
This ring file can be used to announce any desired message to an agent before the call is picked up.  
CCSP UI looks for a CustomRing optional parameter. This parameter should contain the URL for the desired  
ring file.  
If this parameter is present, CCSP UI ignores ring preferences and uses the ring file specified in the  
CustomRing optional parameter.  
CCSP Designer Help - Tenant - 7.4  
168

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.  
3.  
Repeat steps 1-2 to add the percentages you need.  
CCSP Designer Help - Tenant - 7.4  
169

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.  
CCSP Designer Help - Tenant - 7.4  
170

Play Date  
VCS only.  
This block plays a specific date to the caller.  
Properties  
Values  
Description  
Date  
Digits only, no  
text.  
The date to play, as specified in the Date Format property. Must be in  
the same format as specified in the Date Format property.  
Typically, the value of a variable set earlier in the script is entered into  
this field, for example,  $TransactionDate$.  
Note  
Entering $Now$ 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.  
For example, if Date Format = yymmdd, this is a 6-digit number, such as  
010130  
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 playing the date. This  
should be set to No by default.  
Date Format  
yyyymmddhhnn  
yyyymmdd  
yymmddhhnn  
yymmdd  
mmddhhnn  
mmddhhnnss  
hhnnss  
hhnn  
Format in which the output date is spoken, where:  
l  
yyyy = four-digit year  
l  
yy = two-digit year  
l  
mm = two-digit month  
l  
dd = two-digit day  
l  
hh = two-digit hour  
l  
nn = two-digit minute  
l  
ss = two-digit seconds  
Note  
Default year: If yy and yyyy are omitted, the current year is played.  
Line  
Inbound  
The .wav file is played to the original (incoming) caller.  
CCSP Designer Help - Tenant - 7.4  
171

Properties  
Values  
Description  
Outbound  
The .wav file is played to the transferee (the party to whom the call is  
transferred).  
Async Mode  
Yes or No  
Allow asynchronous playback of a .wav file. 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.  
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.  
CCSP Designer Help - Tenant - 7.4  
172

Play Greeting  
This block plays a message to the caller and to the agent.  
Properties  
Values  
Description  
Greeting  
Text  
Path to .wav 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 .wav file to play or text to speech presented only to the agent  
before answering the call, for example, Incoming call from Sales.  
Note  
l  
To use Agent Greeting, you must also have a caller Greeting.  
l  
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 .wav file plays. The default is No.  
Note  
l  
If you use Async Mode, to ensure that CCSP 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, CCSP plays neither message because it executes the next block and  
connects the call.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
173

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 .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  
Allow asynchronous playback of a .wav file. 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.  
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.  
CCSP Designer Help - Tenant - 7.4  
174

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 .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  
Allow asynchronous playback of a .wav file. 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.  
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.  
CCSP Designer Help - Tenant - 7.4  
175

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).  
CCSP Designer Help - Tenant - 7.4  
176

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.  
CCSP Designer Help - Tenant - 7.4  
177

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:  
l  
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.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
178

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. CCSP takes the URL of the  
file to play from the campaign parameters configured in CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
179

Queue Name  
This block directs the call to a specific queue (callSetName).  
Properties  
Values  
Description  
Queue name  
String  
The name of the queue to which the current call is to be directed.  
Note  
Queue names should not be longer than 32 characters.  
This must be a valid queue name for the tenant, so spelling is critical but queue  
names are not case sensitive.  
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.  
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.  
CCSP Designer Help - Tenant - 7.4  
180

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  
l  
The name must start with an uppercase letter as in the SIP message and be  
exclusive, for example From, To, Via.  
l  
Do not include a colon (:).  
Header  
Value  
String  
The name of the variable to return the SIP header.  
CCSP Designer Help - Tenant - 7.4  
181

Record Audio  
VCS only  
This block captures a voice 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, C:\MyFiles\foo.wav, or just a filename, for  
example, foo.wav.  
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: message$line.lineID$.wav creates a wav file  
named message23.wav for line 23 on a VCS and is not overwritten until  
another caller on line 23 of that VCS records a .wav file. See How to use  
variables.  
If it is a fully-qualified name, the file is not automatically deleted.  
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. For more information, see XMLInterpreter Log  
File in the CCSP Registry Key Reference.  
All recorded files are in .wav format, 8KHz – 16 bit mono.  
If the filename specified does not contain the .wav extension, this extension  
is automatically appended to the filename.  
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 \*  
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. This should be  
set to No by default.  
Beep Type  
None  
Type of beep to play at start of recording time, or none.  
If left blank, the default is 0 (None)  
CCSP Designer Help - Tenant - 7.4  
182

Properties  
Values  
Description  
Short,  
Normal, or  
Long  
The filenames that correspond to the various beep lengths are:  
l  
beep\_short.wav  
l  
beep.wav  
l  
beep\_long.wav  
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 .wav 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 .wav 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).  
CCSP Designer Help - Tenant - 7.4  
183

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, C:\MyFiles\foo.avi, or just a filename, for  
example, foo.avi.  
If the specified filename does not contain the .avi 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  
1234567890\*#  
Clear Digits  
Yes or No  
Clear the digit buffer of any digits entered prior to recording. The default is  
No.  
CCSP Designer Help - Tenant - 7.4  
184

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  
l  
If the value (email address) starts with voicemail, it is mapped to ACD  
Voicemail Folder Name. Otherwise, it is mapped to External Mailbox  
Address.  
l  
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  
CCSP Designer Help - Tenant - 7.4  
185

Properties  
Values  
Description  
l  
If the value (email address) starts with voicemail, it is mapped to ACD  
Voicemail Folder Name. Otherwise, it is mapped to External Mailbox  
Address.  
l  
CC has a mapping precedence over BCC. The BCC property is ignored.  
BCC  
String  
This property is deprecated.  
In the Leave Voicemail block, this property is mapped to ACD Voicemail  
Folder Name or External Mailbox Address.  
Note  
If the value (email address) starts with voicemail, it is mapped to ACD  
Voicemail Folder Name. Otherwise, it is mapped to External Mailbox  
Address.  
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.  
CCSP Designer Help - Tenant - 7.4  
186

Recorder  
This block determines how calls are recorded.  
For more information on recording, see CallData object.  
Note  
l  
You cannot use this block in the CallAnswered script.  
l  
For email calls, agents can only control the screen recording. They cannot control the automatic email  
recording by the server.  
l  
CCSP 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.  
For detailed information about ffmpeg screen recording, see Screen  
recording default implementation in the CCSP UI Integrations Guide.  
Off  
When a call is recorded, the screens are not recorded.  
Agent Control  
(recording)  
String  
Allow  
The Record button on CCSP 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 CCSP 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 CCSP UI displays an indication if a call is being  
recorded.  
Hide  
The Record button on CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
187

Note  
If the Record button is added to the CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
188

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.  
CCSP Designer Help - Tenant - 7.4  
189

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:  
l  
Default — reject or queue the call as configured in the registry  
setting.  
l  
Queue — queue the call until an agent becomes available.  
l  
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  
l  
The default values are set in the AcceptOnlyIfThereAreActiveAgents and RejectIfBusy registry values in  
HKLM\SYSTEM\CurrentControlSet\Services\CCUCallManagementServer\Parameters.  
For details, refer to the CCSP Registry Key Reference.  
l  
The value Route is obsolete.  
CCSP Designer Help - Tenant - 7.4  
190

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:  
l  
CreateObject  
l  
Eval  
l  
Execute  
l  
Function  
l  
GetObject  
l  
GetRef  
l  
InputBox  
l  
Kill  
l  
LoadPicture  
l  
MsgBox  
l  
RGB  
l  
SetLocale  
l  
Stop  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
191

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 $line.ani$ 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 $Now$ and get the current date and time.  
This information should be obtained from the caller, from a Menu  
CCSP Designer Help - Tenant - 7.4  
192

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.  
CCSP Designer Help - Tenant - 7.4  
193

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  
l  
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.  
l  
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 CCSP Admin.  
l  
To be able to use the parameter defined through the Variable Name property, the Parameter Accepted  
option of the Entry Points dialog of CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
194

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  
$Now$ 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.  
CCSP Designer Help - Tenant - 7.4  
195

Properties  
Values  
Description  
Callback  
Number  
String  
The phone number to which the callback should be made.  
Generally would be $line.ani$ 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.  
CCSP Designer Help - Tenant - 7.4  
196

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  
Body of the email message, in valid HTML format, unless the Body Format value  
is plain text.  
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 (%%).  
For example, if you have a variable called average, you may use a string like  
this:  
The class average for the last text was %average% %%.  
This would result in the email body containing something like:  
The class average for this test was 85%.  
See How to use variables.  
Attachment  
(s)  
String  
Filename(s) of attachment(s) to include. Multiple attachments separated by a  
semicolon (;).  
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.  
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.  
CCSP Designer Help - Tenant - 7.4  
197

Properties  
Values  
Description  
Result  
String  
Enter a variable name to hold the return result of the send email. The value is  
l  
0 if successful  
l  
1 if an attachment failed to be loaded  
l  
2 if the send email failed  
l  
3 for an unknown error  
Minimum File  
Size  
Number  
If set, the email is not sent unless the attachment file size is equal or greater  
than this value. This value is in bytes.  
Note  
For CCSP wav files, 1 second equals 16000 bytes ex. Setting this field to 40000  
would cause emails with attached .wav files that are 2.5 seconds or less not to  
be sent.  
CCSP Designer Help - Tenant - 7.4  
198

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 application/ccu-integration 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.  
CCSP Designer Help - Tenant - 7.4  
199

Set Event  
This block subscribes to one of these events:  
l  
On Connected  
l  
On Disconnected  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
200

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.  
CCSP Designer Help - Tenant - 7.4  
201

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, Designer adds  
the appropriate header.  
Properties  
Values  
Description  
Header  
Name  
String  
The name of the added custom SIP header.  
Note  
l  
The name must contain only alphanumeric characters and hyphens (-). The  
maximum length is 30 characters (including the prefix).  
l  
It must not be an empty string or just the prefix.  
Header  
Value  
String  
The value of the added custom SIP header.  
Note  
l  
The name 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.  
l  
It must not be an empty string.  
CCSP Designer Help - Tenant - 7.4  
202

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 .wav file, plain text, or Speech Synthesis Markup  
Language (SSML) string.  
For a .wav file, do one of the following:  
l  
Enter a complete URL address to the file you wish to play.  
l  
If your system administrator created a shortcut to the .wav  
files, enter /[file\_name.wav].  
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:  
l  
Start of Input — the prompt is stopped as soon as the user  
starts to speak or presses a key on the phone keypad.  
l  
Recognition Complete — the prompt is stopped only after  
recognition is achieved according to the configured grammar.  
This is also known as Hot Word mode.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
203

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.  
CCSP Designer Help - Tenant - 7.4  
204

Configure the Speech Enabled Menu block  
1.  
Right-click the block and click Add case. The Speech Enabled Menu dialog appears.  
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  
l  
This is an alternative spoken response option for the caller.  
l  
The case accepts any one of your selected digits or words if the caller response is matched in  
this case.  
CCSP Designer Help - Tenant - 7.4  
205

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.  
CCSP Designer Help - Tenant - 7.4  
206

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:  
l  
Max Time = total time required for the speech recognition process to complete  
l  
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:  
l  
Max Time = 10, which exceeds the minimum of 9, calculated as 5 second input plus 2 x silence time of 2  
seconds  
l  
Max Silence = 2  
Properties  
Values  
Description  
Text or Audio File  
String  
Enter a .wav file, plain text, or Speech Synthesis Markup Language  
(SSML) string.  
For a .wav file, do one of the following:  
l  
Enter a complete URL address to the file you wish to play.  
l  
If your system administrator created a shortcut to the .wav files,  
enter /[file\_name.wav].  
Stop Prompt On  
Start of Input,  
Recognition  
Complete, or  
Don't Stop  
This property determines if caller barge-in is enabled or disabled:  
l  
Start of Input — the prompt is stopped as soon as the user starts  
to speak or presses a key on the phone keypad.  
l  
Recognition Complete — the prompt is stopped only after  
CCSP Designer Help - Tenant - 7.4  
207

Properties  
Values  
Description  
recognition is achieved according to the configured grammar. This  
is also known as Hot Word mode.  
l  
Don't Stop — no barge in during prompt.  
Grammar  
String  
The grammar determines what to recognize in a caller's speech when  
barge-in is enabled, and a barge-in occurs.  
Specify your grammar(s) in one of the following ways:  
l  
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  
built-in:grammar/boolean. The syntax is built-in:grammar/  
[your\_grammar].  
l  
Enter complete URL(s) to your grammar(s).  
l  
Type a list of comma-delimited words instead of a grammar.  
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.  
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.  
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  
minimum is 50 and the maximum is 100. The default is 50.  
Speech Rate  
Slow,  
Medium, or  
Fast  
The rate for reading a Text to Speech (TTS) message.  
Volume  
Soft, Medium,  
or Loud  
The volume level for reading a TTS message.  
Termination Digits  
0-9 # and \*  
Enter digit(s), # or \* . These value(s) can be used by the caller to  
indicate the end of their response.  
When a termination digit is pressed, playback stops, and the script  
executes the next block.  
Max Time  
Seconds  
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.  
This property applies only to Automatic Speech Recognition (ASR).  
Max Digits  
Number  
The maximum number of caller response DTMF digits. When the limit  
CCSP Designer Help - Tenant - 7.4  
208

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, CCSP  
routes the script to the else case.  
1.  
Right-click the block and click Add case. The Speech Play/Collect dialog appears.  
CCSP Designer Help - Tenant - 7.4  
209

2.  
From the SpeechPlayCollect list, select one of the following:  
o  
NoMatch — recognition did not succeed or caller pressed invalid DTMF.  
o  
NoInput — the caller did not respond.  
o  
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  
CCSP Designer Help - Tenant - 7.4  
210

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 #  
Any valid phone number to which to redirect the call, for example,  
+16315551212  
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,  
+$NumberDialed$.  
Note  
l  
The maximum outgoing call number or SIP URI is 370 characters.  
l  
This can also be an IP address, if you prefix the number with TA:, for  
example, TA:208.203.191.74.  
Source ID  
String  
Note  
This information is shown only if the Source display property is missing.  
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.  
This can be either an H323-ID or an E164 number:  
l  
H323-ID takes the format NAME:ARNIE or NAME:$anyvariable$  
l  
E164 takes the format TEL:1234 or  
TEL:$anyvalidnumberorvariable$  
All possible values for this property should be contained in a single string,  
such as:  
NAME:NJONES,NAME:FSCHMIDT,TEL:1234567  
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.  
You do not have to send both types of source address, but it is legal if  
CCSP Designer Help - Tenant - 7.4  
211

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 .wav file specified should be longer than the timeout value for the  
connection, or the caller hears silence after the .wav 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.  
CCSP Designer Help - Tenant - 7.4  
212

Configure the Supervised Transfer block  
1.  
Right-click the block and click Add case. The Transfer dialog appears.  
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.  
CCSP Designer Help - Tenant - 7.4  
213

Switch  
This block evaluates a case statement and branches according. For important information about variables in  
Designer, see How to use variables.  
Properties  
Values  
Description  
Expression  
The left side  
of any valid  
expression  
equals.  
(VBScript)  
Causes branching in the script.  
The name of the case branch you add to the block is the value to check the  
expression is equal to.  
For example, $Line.DNIS$ branches to 5551234 when the DNIS value for  
this call is 5551234.  
When the switch expression is $line.DNIS$, 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.  
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.  
CCSP Designer Help - Tenant - 7.4  
214

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:  
l  
Check Date  
l  
Check Time  
l  
Percent Allocated  
They all require Case statements to be evaluated to direct the call.  
CCSP Designer Help - Tenant - 7.4  
215

Table  
Note  
The Table block is obsolete because tables are now managed though 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.  
CCSP Designer Help - Tenant - 7.4  
216

Terminate  
This block sets the condition for termination.  
Properties  
Values  
Description  
Terminate  
Yes or  
No  
If Yes, the call terminates on completion of this script event and goes straight to  
the call terminate event.  
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.  
Note  
l  
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.  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
217

Transfer by Refer  
VCS only.  
This block transfers a SIP call without using IVR ports.  
Properties  
Values  
Description  
Phone Number  
0-9 \* and #  
Any valid phone number to which to redirect the call, for example,  
+16315551212  
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,  
+$NumberDialed$.  
After the Answer case, no blocks are executed.  
If the phone number is not reachable (404 SIP error code), the  
NoAnswer case is executed.  
Note  
l  
The maximum outgoing call number or SIP URI is 370 characters.  
l  
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  
Note  
This information is shown only if the Source display property is missing.  
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.  
E.164 takes the format TEL:1234 or  
TEL:$anyvalidnumberorvariable$  
All possible values for this property should be contained in a single  
string, such as: NAME:NJONES,NAME:FSCHMIDT,TEL:1234567  
The source address can be specified as an E.164 address denoted by  
the TEL: prefix.  
Source  Display  
String  
What to display as a source. This property is also known as CallerID or  
CallingPartyName.  
CCSP Designer Help - Tenant - 7.4  
218

Properties  
Values  
Description  
Connect Timeout  
Integer  
The number of seconds to wait for a connection. The default of 10  
seconds is used when the value is:  
l  
Left blank  
l  
0  
l  
Less than 0  
l  
Greater than 300  
Configure the Transfer by Refer block  
1.  
Right-click the block and click Add case. The Transfer dialog appears.  
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.  
CCSP Designer Help - Tenant - 7.4  
219

Transfer Call  
VCS only.  
This block redirects a call to another phone number.  
Properties  
Values  
Description  
Phone  
Number  
0-9 \* and  
#  
Any valid phone number to which to redirect the call, for example,  
+16315551212  
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, +$NumberDialed$.  
Note  
l  
The maximum outgoing call number or SIP URI is 370 characters.  
l  
This can also be an IP address. For H.323 calls, prefix the number with TA:, for  
example, TA:208.203.191.74. For SIP calls, use a prefix in the format  
sip:DNIS@IP\_ADDRESS, for example, sip:6111@192.168.100.100, where  
6111 is the DNIS.  
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.  
Transfer  
Type  
Outdial  
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.  
Source ID  
String  
Note  
This information is shown only if the Source display property is missing.  
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.  
This can be either an H.323-ID or an E.164 number:  
l  
H.323-ID takes the format NAME:ARNIE or NAME:$anyvariable$  
l  
E.164 takes the format TEL:1234 or TEL:$anyvalidnumberorvariable$  
CCSP Designer Help - Tenant - 7.4  
220

Properties  
Values  
Description  
All possible values for this property should be contained in a single string, such  
as: NAME:NJONES,NAME:FSCHMIDT,TEL:1234567  
The source address can be specified as either an H.323-ID value denoted by the  
NAME: prefix, or as an E.164 address denoted by the TEL: prefix  
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.  
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.  
CCSP Designer Help - Tenant - 7.4  
221

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:  
l  
The call is in the appropriate state (waiting for agent ring)  
l  
The script is in MusicOnHoldInQueue, InqueueTimeout, or PickupTimeout  
l  
The media is voicecall  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
222

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  
Note  
This information is only shown if the Source display property is missing.  
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.  
This can be either an H323-ID or an E164 number:  
l  
H323-ID takes the format NAME:ARNIE or NAME:$anyvariable$  
l  
E164 takes the format TEL:1234 or  
TEL:$anyvalidnumberorvariable$  
All possible values for this property should be contained in a single  
string, such as:  
NAME:NJONES,NAME:FSCHMIDT,TEL:123456 7  
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.  
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.  
Source Display  
String  
What to display as a source. This property is also known as CallerID or  
CallingPartyName.  
CCSP Designer Help - Tenant - 7.4  
223

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.  
CCSP Designer Help - Tenant - 7.4  
224

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.  
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.  
CCSP Designer Help - Tenant - 7.4  
225

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,  
http:// www.abc.com/soap/GetClientinformation.asmx  
Namespace  
String  
Web services namespace,http:// wsdl.use.abc.com  
Namespace Prefix  
String  
A namespace prefix for the xml request to be sent to the web  
services, for example, q0.  
Method  
String  
The web services method to be called, for example,  
GetClientInformation.  
Parameters  
String  
The key/value parameter pairs for the web services method, for  
example, PIN\_Number, 123456789, Surname, Doell, GivenName,  
John  
Parameter Wrapper  
String  
The nested wrapper parameters, for example,  
<Request><q0:RequestWrapper>%%1</  
q0:RequestWrapper></Request>.  
Note  
%%1 is required for indicating where the required parameters are  
being substituted.  
Envelope Prefix  
String  
The soap envelope prefix, for example, SOAP-ENV.  
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 theDesigner log as  
follows:  
No response was received  
from:http://webserver/webservices/test.asmx. Operation  
aborted.  
Result Variable  
String  
A variable name where responses are being prefixed. See XML  
response example.  
See Web services XML examples.  
CCSP Designer Help - Tenant - 7.4  
226

6: Objects  
This section describes the CCSP 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.  
CCSP 7.3.x continues to support the following objects for MCS, VCS, and ICS media and scripts:  
l  
CallData object — for messaging data.  
l  
CCLEvent object — for the call media type.  
l  
CosmoCall object — for callbacks.  
l  
Line object — for VoIP.  
l  
NewLine object — associated with the outbound call from a call transfer.  
l  
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:  
l  
Common media API — for identifying the call, its type, and common routing and history properties. Has  
multiple sub-objects.  
l  
CallbackCall object — for additional callback-specific information.  
l  
EmailCall object — for additional email-specific information.  
l  
OutgoingEmailCall object — for additional outgoing email-specific information.  
l  
VoicemailCall object — for additional voicemail-specific information.  
l  
Sub-objects:  
o  
AgeingInfo object — for additional aging information for email and voicemail.  
o  
DeferInfo object — for additional defer information for email and voicemail.  
o  
EmailCall.EmailData object — for additional information about the email data.  
o  
EmailCall.LastAgentInfo object — for additional information about the last agent to handle the email.  
o  
EmailCall.RecordingControl object — for additional information about the email recording.  
CCSP Designer Help - Tenant - 7.4  
227

o  
OutgoingEmailCall.OutgoingEmailData object — for additional information about the outgoing email  
data.  
o  
PullInfo object — for additional pull information for email and voicemail.  
o  
VoicemailCall.LastAgentInfo object — for additional information about the last agent to handle the  
voicemail.  
o  
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:  
l  
0 — Incoming  
l  
1 — Outgoing  
l  
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:  
l  
2 — Email  
l  
4 — Callback  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
228

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:  
l  
0 = not used  
l  
1 = least skilled  
l  
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:  
l  
0 = Default  
l  
1 = Route  
l  
2 = Queue  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
229

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.  
CCSP Designer Help - Tenant - 7.4  
230

Property name  
Type  
Description  
ErrorLevel  
Read  
Error level types:  
l  
0 = Info  
l  
1 = Warning  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
231

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:  
l  
Value=0 for initial callback  
l  
Value>0 for continuous callback  
MaxRescheduleCount  
Read  
Returns the callback's maximum allowed reschedule attempts.  
CCSP Designer Help - Tenant - 7.4  
232

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.  
CCSP Designer Help - Tenant - 7.4  
233

Property name  
Type  
Description  
AgedRefCount  
Read  
Number of ageing cycles.  
IsDeferAvailable  
Read/Write  
True or false.  
Write is performed through the Assign block.  
For email, set Variable Name to  
EmailCall.AgeingInfo.IsDeferAvailable and Variable Value to  
True.  
For voicemail, set Variable Name to  
VoiceMailCall.AgeingInfo.IsDeferAvailable and Variable Value  
to True.  
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  
Defer type:  
0 — Personal  
1 — Shared  
Write is performed through the Set Defer Type block.  
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  
The recipients. May be a list of values, separated by comma or semi-colon (;).  
To find out the destination of the email, use: $EmailData.To$  
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:  
CCSP Designer Help - Tenant - 7.4  
234

Property name  
Type  
Description  
l  
0 — Low  
l  
1 — Normal  
l  
2 — High  
l  
3 — No-change  
ReceivedTime  
Read  
The email arrival time.  
SentTime  
Read  
The email sent time.  
Body  
Read  
The part of the body sent from the MS to the MCS to expose to scripts,  
especially the Call Arrival script.  
Use the CCSP Admin to disable body fetching, set the body format, and, for  
text only, set the maximum size. For details, see Adding a folder in the CCSP  
Tenant Administrator Guide.  
KeepBody  
Read  
Whether and when to delete Body:  
l  
0 (default value) — clear on route. Clear MCS memory upon successful call  
routing to MCS.  
l  
1 — delete immediately. For example, if you only want to use Body in the  
Call Arrival script.  
l  
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.  
AgentGloibalId  
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.  
CCSP Designer Help - Tenant - 7.4  
235

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 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:  
l  
1 — pull from Deferred folder  
l  
2 — pull Not-Yet-Queued from ACD folder  
l  
3 — pull Queued from ACD folder  
l  
4 — pull from Rejected folder  
l  
5 — pull from Archived folder  
l  
6 — pull from Sent-Items folder  
l  
7 — pull from Outbox folder  
l  
8 — pull from Drafts folder  
CCSP Designer Help - Tenant - 7.4  
236

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.  
AgentGloibalId  
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.  
CCSP Designer Help - Tenant - 7.4  
237

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  
CCSP Designer Help - Tenant - 7.4  
238

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  
CCSP Designer Help - Tenant - 7.4  
239

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  
DirectRoutAgentId  
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  
CCSP Designer Help - Tenant - 7.4  
240

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  
CCSP Designer Help - Tenant - 7.4  
241

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  
CCSP Designer Help - Tenant - 7.4  
242

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 Designer blocks.  
Call priority takes precedence over call maturity within a  
particular queue.  
CallData.CallDirection  
Integer  
The values are:  
l  
0 — Incoming  
l  
1 — Outgoing  
l  
2 — Internal  
l  
3 — Agent to agent  
l  
4 — Callback request - initial ACD callback call (exists  
on MCS servers only when the callback reschedule  
count == 0)  
l  
5 — Callback reschedule - continuous ACD callback  
call (exists on MCS servers only when the callback  
reschedule count > 0)  
l  
6 — Callback response - outgoing VoIP callback call  
(exists on VCS servers only when dialed out as the  
CCSP Designer Help - Tenant - 7.4  
243

Property name  
Type  
Description  
result of a callback call)  
l  
7 — Predictive  
l  
8 — Predictive preview  
l  
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  
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.  
The value is in seconds.  
The measurement interval for the waiting time is  
estimated for at least 5 minutes and at least 10 calls, as  
follows:  
If there are more than 10 calls during the last 5 min.,  
the waiting time is calculated based on all the calls.  
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.  
CallData.HasSkill("SkillName")  
Boolean  
Tests if the supplied skill is attached to the call.  
CCSP Designer Help - Tenant - 7.4  
244

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:  
l  
0 — Chat call  
l  
1 — Phone call  
l  
2 — Email call  
l  
3 — VOIP call  
l  
4 — Callback call  
l  
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 CCSP 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  
CCSP Designer Help - Tenant - 7.4  
245

Property name  
Type  
Description  
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.  
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.  
CallData.line.SIPPrivacyList  
String  
Values from incoming call SIP Privacy header.  
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.  
The treatment of this information depends on your  
system administrator's settings for privacy in your  
network. See SIP privacy parameters in the CCSP  
Tenant Administrator Guide.  
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.SIPPANIAcessType  
String  
Value from incoming call PANI header. One of the  
components of the PANI header.  
CallData.line.SIPMediasList  
String  
List of media selected in the Session Description  
Protocol (SDP) part of the SIP INVITE message.  
Note  
SDP describes multimedia sessions for the purposes  
of session announcement, session invitation, and  
other forms of multimedia session initiation.  
CCSP Designer Help - Tenant - 7.4  
246

Playback-related properties  
The properties shown below are only available for Playback-enabled contact centers.  
Note  
l  
For information on how these values are set using the Playback block in Designer, see Playback.  
l  
CCSP 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 CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
247

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:  
l  
The VCS DWORD presentationIndicatorSupported registry key  
l  
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  
CCSP Designer Help - Tenant - 7.4  
248

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:  
l  
0 — Call was queued / call is valid  
l  
1 — Unknown error  
l  
2 — Insufficient memory  
l  
3 — No suitable agent was found  
l  
4 — Invalid queue  
l  
5 — Invalid Connection Server address  
l  
6 — No CallId  
l  
7 — Invalid external queue ID  
l  
8 — DID agent is not free  
l  
9 — CMS got an invalid EXTERNAL\_AGENT\_ID (ThirdPartyID)  
l  
10 — CMS failed to locate logged-in agent marked for  
external routing  
CCSP Designer Help - Tenant - 7.4  
249

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.  
CCSP Designer Help - Tenant - 7.4  
250

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).  
cosmocall.ttsInfo.ParamName  
String  
Returns the parameter name used during the TTS (only if set  
via selected EntryPoint).  
cosmocall.ttsInfo.ScriptName  
String  
Returns the script name used during the TTS (only if set via  
selected EntryPoint).  
cosmocall.ttsInfo.ApplicationID  
Integer  
Returns the application ID used during the TTS (only if set via  
selected EntryPoint).  
cosmocall.ttsInfo.CallType  
Integer  
Returns the TTS call type - one of the following values:  
l  
0 - main call (all media)  
l  
1 - outgoing TTS call (VoIP only)  
l  
2 - incoming TTS call (VoIP only)  
cosmocall.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 CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
251

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.  
CCSP Designer Help - Tenant - 7.4  
252

7: CallXML  
This section contains the following topics:  
l  
General elements  
l  
Call control elements  
l  
Test and debug elements  
l  
CallXML high level elements  
l  
Additional Designer elements  
This section describes how Designer supports the CallXML standard and explains how to use the additional  
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:  
l  
Media action elements such as <playAudio> and <recordAudio> to describe what to present to the user  
during a call.  
l  
Call action elements such as <answer>, <call>, and <hangup> to describe how to control and route the call  
itself.  
l  
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.  
l  
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.  
l  
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 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:  
l  
Correct — http://www.somesite.com/default.htm  
l  
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/csml\_2.0/home.htm.  
CCSP Designer Help - Tenant - 7.4  
253

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" />  
CCSP Designer Help - Tenant - 7.4  
254

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 CCSP 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: ""  
CCSP Designer Help - Tenant - 7.4  
255

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:  
l  
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 CCSP system.  
l  
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.  
l  
A CCSP file descriptor. This means that the wave file is stored in the CCSP database. The file descriptor name  
must match the name that is specified in CCSP Admin for the particular file.  
playDateformat  
<playDateformat="yymmddhhnn" default: as shown below  
value="9901111200" default: ""  
termDigits="1234567890\*#" default: ""  
clearDigits="TRUE|FALSE" default: "false" />  
The supported formats are:  
l  
yyyymmddhhnn  
l  
yyyymmdd  
l  
yymmddhhnn  
l  
yymmdd  
l  
mmddhhnnss  
l  
mmddhhnn  
CCSP Designer Help - Tenant - 7.4  
256

l  
hhmmss  
l  
hhmm  
Where:  
l  
yyyy is a 4 digit year  
l  
yy is a 2 digit year  
l  
mm is a 2 digit month  
l  
dd is a 2 digit day of the month  
l  
hh is a 2 digit hour  
l  
nn is a 2 digit minute  
l  
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.  
CCSP Designer Help - Tenant - 7.4  
257

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:  
l  
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.  
l  
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:  
l  
0 — for no beep  
l  
1 — for a short beep  
l  
2 — for a normal length beep  
l  
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 CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
258

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"  
CCSP Designer Help - Tenant - 7.4  
259

maxDigits="4"  
default: no limit  
maxTime="15s"  
default: maxSilence \* maxDigits  
maxSilence="5s" default: "3s" >  
Additional Designer elements  
Use the additional Designer elements as follows.  
addSkillvalue  
<addSkillvalue="defaultSkill" default: "" />  
Use this element to add CCSP 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 CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
260

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 CCSP 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 CCSP skills from the current call.  
sendEmail server  
<sendEmail server="MailServer" default: ""  
recipient=you@somewhere.com default: ""  
cc=aaa@cosmocom.com default: ""  
CCSP Designer Help - Tenant - 7.4  
261

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 CCSP 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.  
CCSP Designer Help - Tenant - 7.4  
262

# Contact Center: Service Provider APIs Guide Version 7.4

CCSP-APIG-74-R2-04/2025   
Contents   
1: Intro .............................................................................................................................................6   
What’s new ................................................................................................................................................. 6   
Legal disclaimer ......................................................................................................................................... 6   
Support ....................................................................................................................................................... 6   
2: Overview .....................................................................................................................................7   
3: CCSP Administrator API .............................................................................................................8   
High level architecture.............................................................................................................................. 9   
ConfigCom\_WP.exe......................................................................................................................... 10   
Password encryption .............................................................................................................................. 11   
AdminGUID .............................................................................................................................................. 12   
Parameters ............................................................................................................................................... 13   
Samples ..................................................................................................................................................... 13   
API methods sample ...................................................................................................................... 13   
AJAX sample ..................................................................................................................................... 14   
4: CCSP Agent AJAX API ............................................................................................................... 16   
5: CCSP Agent Proxy Service API ................................................................................................ 17   
Filtering/Sorting by optional parameters ............................................................................................ 18   
Syntax of JSON filter ....................................................................................................................... 18   
Specifying sort field ........................................................................................................................ 27   
6: CCSP Chat API .......................................................................................................................... 28   
Configuration ........................................................................................................................................... 29   
Flow diagrams .......................................................................................................................................... 30   
JoinChat flow .................................................................................................................................... 31   
SendMsg flow .................................................................................................................................. 31   
GetEvents flow ................................................................................................................................. 31   
LeaveChat flow ................................................................................................................................ 31   
JoinChat ..................................................................................................................................................... 32   
Request ............................................................................................................................................. 32   
Response .......................................................................................................................................... 36   
Errors ................................................................................................................................................ 36   
SendMSG .................................................................................................................................................. 38   
Request ............................................................................................................................................. 38   
Response .......................................................................................................................................... 38

CCSP APIs Guide - 7.4   
3   
Error .................................................................................................................................................. 38   
GetEvents .................................................................................................................................................. 39   
Request ............................................................................................................................................. 39   
Response .......................................................................................................................................... 40   
LeaveChat ................................................................................................................................................. 46   
Request ............................................................................................................................................. 46   
Response .......................................................................................................................................... 46   
FireEvent ................................................................................................................................................... 47   
Request ............................................................................................................................................. 47   
Response .......................................................................................................................................... 48   
Error .................................................................................................................................................. 48   
7: CCSP CIS to Connector API ..................................................................................................... 49   
High-level overview ................................................................................................................................. 50   
Display the HTML web page .................................................................................................................. 51   
Log in ......................................................................................................................................................... 51   
Request body format...................................................................................................................... 51   
Code snapshot................................................................................................................................. 52   
Example of HTTP POST Message being sent .............................................................................. 52   
Example of HTTP POST Message being received ....................................................................... 53   
Connect through WebSocket ................................................................................................................. 54   
Create a GET\_EVENTS request ............................................................................................................... 55   
Request body format...................................................................................................................... 55   
Example of HTTP POST Message being sent .............................................................................. 57   
Example of HTTP POST Message being received ....................................................................... 58   
Control the processing rate ................................................................................................................... 59   
Message body format .................................................................................................................... 59   
Code snapshot................................................................................................................................. 60   
Example of message being sent using WebSockets ................................................................. 60   
Example of error message sent from CIS ................................................................................... 60   
Receive messages ................................................................................................................................... 60   
Stop events ............................................................................................................................................... 61   
Request body format...................................................................................................................... 62   
Code snapshot................................................................................................................................. 62   
Example of HTTP POST Message being sent .............................................................................. 62   
Example of HTTP Response Message being received ............................................................... 63   
Log out ...................................................................................................................................................... 63   
Request body format...................................................................................................................... 63   
Code snapshot................................................................................................................................. 63   
Example of HTTP POST Message being sent .............................................................................. 64   
Example of HTTP POST Message being received ....................................................................... 64   
Add events to CISEvents.xml ................................................................................................................. 64   
Structure of CISEvents.xml ............................................................................................................ 64

CCSP APIs Guide - 7.4   
4   
Add events ....................................................................................................................................... 66   
8: CCSP Realtime API ................................................................................................................... 68   
ACalls ......................................................................................................................................................... 70   
Usage ................................................................................................................................................ 70   
Example ............................................................................................................................................ 70   
Elements and attributes ................................................................................................................ 71   
ACallsDetails ............................................................................................................................................. 74   
Usage ................................................................................................................................................ 74   
Example ............................................................................................................................................ 75   
Elements and attributes ................................................................................................................ 76   
QCalls ........................................................................................................................................................ 79   
Usage ................................................................................................................................................ 79   
Example ............................................................................................................................................ 80   
Elements and attributes ................................................................................................................ 80   
AConfigs .................................................................................................................................................... 83   
Usage ................................................................................................................................................ 83   
Example ............................................................................................................................................ 84   
Elements and attributes ................................................................................................................ 85   
CConfigs .................................................................................................................................................... 87   
Usage ................................................................................................................................................ 87   
Example ............................................................................................................................................ 87   
Elements and attributes ................................................................................................................ 88   
GConfigs ................................................................................................................................................... 89   
Usage ................................................................................................................................................ 89   
Example ............................................................................................................................................ 89   
Elements and attributes ................................................................................................................ 90   
QConfigs ................................................................................................................................................... 91   
Usage ................................................................................................................................................ 91   
Example ............................................................................................................................................ 91   
Elements and attributes ................................................................................................................ 92   
QHistory .................................................................................................................................................... 93   
Usage ................................................................................................................................................ 93   
Example ............................................................................................................................................ 93   
Elements and attributes ................................................................................................................ 96   
SServer ...................................................................................................................................................... 98   
Usage ................................................................................................................................................ 98   
Example ............................................................................................................................................ 98   
9: CCSP TWS integration ........................................................................................................... 103   
Architecture ............................................................................................................................................ 103   
Requirements ......................................................................................................................................... 104   
Web API ................................................................................................................................................... 104

CCSP APIs Guide - 7.4   
5   
Connect (agent, UPN)................................................................................................................... 104   
Disconnect (agent, UPN) .............................................................................................................. 105   
GetUsersPresences (agent, UPN) ............................................................................................... 105   
GetUsersPresencesSearch (agent, UPN, filter) ........................................................................ 107   
GetMyPresence (agent, UPN) ..................................................................................................... 108   
Error Codes .................................................................................................................................... 109   
SignalR Hub ............................................................................................................................................ 109   
Create connection ......................................................................................................................... 110   
Start ................................................................................................................................................. 110   
AddToGroup .................................................................................................................................. 110   
StatusUpdate ................................................................................................................................. 111   
RemoveAgent ................................................................................................................................ 112   
Client application flows ........................................................................................................................ 113   
Agent login ..................................................................................................................................... 113   
Authentication in TWS with AuthServer .................................................................................... 114   
Search users in Teams directory ................................................................................................ 115   
Agent changes presence ............................................................................................................. 116   
Another agent changes presence .............................................................................................. 116   
Agent logout .................................................................................................................................. 117   
Another agent logs out ................................................................................................................ 117   
10: CCSP UI Connector .............................................................................................................. 118   
A: CCSP Web Support integration ............................................................................................ 119   
Chat call initiation sequence ................................................................................................................ 120   
Web Support package components ................................................................................................... 120   
Configurator .................................................................................................................................. 120   
Quick start ...................................................................................................................................... 122   
Send external callbacks via SMTP ....................................................................................................... 123   
Extended SMTP headers .............................................................................................................. 123   
Sample script ................................................................................................................................. 126   
Chat extension ....................................................................................................................................... 128   
How to use the chat extension ................................................................................................... 128   
Chat extension components ....................................................................................................... 131   
Interfaces to the chat extension ................................................................................................. 135   
Customize the chat window ................................................................................................................. 139   
Increase the user input frame .................................................................................................... 139   
Show caller and agent names ..................................................................................................... 140   
Parameters ............................................................................................................................................. 141   
Options for launching calls .......................................................................................................... 141   
Call parameters ............................................................................................................................. 143   
ASP script parameters .................................................................................................................. 144

CCSP APIs Guide - 7.4   
6   
1: Intro   
This document is for developers who use the Contact Center: Service Provider (CCSP) APIs.   
What’s new   
•   
5: CCSP Agent Proxy Service API — added Filtering/Sorting by optional parameters (HF74-  
67429).   
•   
6: CCSP Chat API — the maximum total length for the optional parameters, including the   
optional parameter names, is 1024 bytes.   
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 content in this document, please open a case with   
Enghouse Support.

CCSP APIs Guide - 7.4   
7   
2: Overview   
CCSP provides a complete range of API interfaces using standards such as AJAX, JSON, SOAP Web   
Services, WSDL, XML requests, JS/VBS framework for integration, FTP repository system, and   
Angular JS framework.   
The interfaces are:   
•   
Agent API — for remote control, customizations, and specific applications, using AJAX, WSDL,   
SOAP, or Angular JS. See 4: CCSP Agent AJAX API and 5: CCSP Agent Proxy Service API.   
•   
CCSP UI Connector — provides a two-way integration hub on the client for integration into   
other desk applications or web applications. See 10: CCSP UI Connector.   
•   
TWS integration — enables a generic client application to integrate with Teams Web Service   
(TWS). See 9: CCSP TWS integration.   
•   
Administrator API — for custom GUI, automated config changes, config customizations, using   
SOAP or WSDL. See 3: CCSP Administrator API.   
•   
Chat API — for custom or mobile applications, using JSON or HTTP. See 6: CCSP Chat API and   
A: CCSP Web Support integration.   
•   
Central Integration Server (CIS) — provides a central integration point for all agent and call   
events. See 7: CCSP CIS to Connector API.   
•   
RTR web dll — for real time statistics pulling, using HTTP or XML. See 8: CCSP Realtime API.   
   
The scripting in the platform is done using XML scripts through the CCSP Designer tool, which   
offers open integration blocks to external integrations with web services or databases. For a   
description of the blocks and the scripting object model, see the CCSP Designer Help.   
The reporting database structure offers customers the ability to integrate directly with the   
historical data. For a description of the stored procedures and database tables, see the CCSP   
Historical Reports Technical Guide.

CCSP APIs Guide - 7.4   
8   
3: CCSP Administrator API   
Note   
Permission profiles are now enforced.   
   
The CCSP 7.4 Administrator API Library describes the SOAP Administrator methods for custom GUI,   
automated config changes, and config customizations.   
   
This chapter describes the high-level architecture, and configuration of the CCSP Administrator API   
web service. It also describes how to use the working sample of the API methods and the sample   
AJAX.

CCSP APIs Guide - 7.4   
9   
High level architecture   
The following diagram shows the high-level architecture.

CCSP APIs Guide - 7.4   
10   
ConfigCom\_WP.exe   
Upon successful logon to perform Administrator API operations, each landlord or tenant   
administrator is hosted in a separate out of process COM (ConfigCom\_WP.exe).   
The following diagram represents different ConfigCom logons hosted in separate out of process   
COMs.

CCSP APIs Guide - 7.4   
11   
Password encryption   
All APIs that require a password must be encrypted using RSA encryption. To use the same RSA   
encryption in the client application, the customer must contact Enghouse to obtain the key.   
For examples of how to use the RSA encryption, see the sample code listed in   
Samples\Includes\Crypto and Samples\WebAdminAPIClient directories under the Administrator API   
Web Services main directory.

CCSP APIs Guide - 7.4   
12   
AdminGUID   
The AdminGUID is the administrator globally unique identifier.   
To maintain the same logon session between API calls, a GUID is generated and passed back to the   
client upon a successful logon. Upon successful logon, this GUID can be obtained via the Login or   
GetLoginInfo API. All subsequent calls from the client must now pass the same AdminGUID as an   
internet header to identify his or her logon session.   
The following example demonstrates retrieving the tenant information.

CCSP APIs Guide - 7.4   
13   
Parameters   
The tenantId parameter for all APIs is mandatory. For landlord logon, the tenant id for the tenant   
to operate on must be specified. For tenant administrator logon, -1 can be specified as the tenant id   
or id of the tenant that this tenant administrator belongs to. When -1 is used as the tenant id, it is   
always under the assumption that this logon is a tenant administrator, not a landlord.   
Note   
When logging on a tenant administrator, the ID of the tenant that this administrator belongs to   
can be retrieved from the ConfigCom logon session.   
   
In all Create APIs, such as CreateQueue and CreateAgent, the name parameter is mandatory with   
the following exceptions:   
•   
CreateApplication — mandatory url and rule parameters   
•   
CreateAgent — mandatory RSA encrypted password   
   
For all Update APIs, such as UpdateQueue and UpdateAgent:   
•   
String parameters with empty string or null value are not updated.   
•   
All other parameters are updated based on the value provided in the API. For example, if for a   
numeric parameter the value 0 is passed, the parameter is updated to 0.   
Samples   
This section contains the following information:   
•   
API methods sample   
•   
AJAX sample   
API methods sample   
A working sample of the entire API methods can be found in the main folder of the Administrator   
API, under the Samples subfolder.   
This sample is written in html\javascript and is supported by these browsers:   
•   
Chrome — 23.0.1271.97   
•   
Firefox — 17.0.1   
   
To use the sample, update the server name for the sample application.   
To update the server name for the sample application:   
1.   
Navigate to C:\Program Files (86)\CosmoCom\Server Components\CCUWebAdminAPI\   
Samples\WebAdminAPIClient\WebAdminAPIService\.   
2.   
In a text editor, open WebAdminAPIService.js.   
3.   
Find the line this.serverName = '@ServerName@';

CCSP APIs Guide - 7.4   
14   
4.   
Replace @ServerName@ with your FQDN, for example, Server.Domain.com   
5.   
From a client machine browse to   
http://<ServerName>/CCUWebAdminAPI/samples/WebAdminAPIClient/main.htm, replacing   
<ServerName> with your FQDN.   
AJAX sample   
The Administrator API install package includes a sample AJAX (Asynchronous JavaScript and XML).   
The sample AJAX is used for verifying the available APIs in the Administrator API web service and as   
a guide for developing web applications for accessing any of the Administrator APIs.   
To verify the available APIs in the Administrator API web service:   
1.   
Use a supported browser to navigate to the AJAX sample. For example:   
http://localhost/CCUWebAdminAPI/Samples/WebAdminAPIClient/Main.htm   
If accessing the sample from a different machine, replace localhost with the server name.   
2.   
Log on as a landlord or tenant administrator.   
To access any other APIs, the client application must first successfully log on as a landlord or   
tenant administrator. The password is encrypted before being sent to the Administrator API   
web service. For details, see Password encryption.

CCSP APIs Guide - 7.4   
15   
3.   
Select the API to operate on from the menu.   
   
4.   
Use the standard CCSP Web Administrator or the API’s corresponding Get API to confirm the   
operation performed in the previous step. For example, if you used CreateApplication to create   
an item, use GetApplicationInfo to obtain the latest updated information for this item.

CCSP APIs Guide - 7.4   
16   
4: CCSP Agent AJAX API   
The CCSP 7.4 Agent AJAX API Library describes the AJAX Agent methods for remote control,   
customizations, and specific applications.   
You can find the CCSP 7.4 Agent AJAX API Library.zip file in your environment at:   
http://{servername.domain.loc}/TouchPoint/Ajax/CCSPAgentAjaxAPI/index.html

CCSP APIs Guide - 7.4   
17   
5: CCSP Agent Proxy Service API   
The CCSP 7.4 Agent Proxy Service API Library describes the SOAP Agent methods for remote control,   
customizations, and specific applications.

CCSP APIs Guide - 7.4   
18   
Note   
To validate the authenticity of the request, most of the AgentWS methods require a session   
based AgentGUID to be supplied as part of the request header.   
Due to the Messaging design, the following methods are unable to supply this parameter as part   
of the header:   
•   
FolderService class: GetAttachmentContent   
•   
FolderService class: GetAttachmentContent   
•   
FolderService class: GetAttachmentContentEmailPart   
•   
FolderService class: AddAttachment   
•   
FolderService class: GetMedia   
   
These methods use the global agent ID supplied as a query parameter. Because the global agent   
ID value is also session based, it can also be used to authenticate the request.   
Filtering/Sorting by optional parameters   
Starting from HF74-67429, the FolderService exposes a new API: GetFolderItemsEx2 with a new   
parameter named advancedFilter.   
This parameter allows you to specify a JSON string that describes a filter expression to allow   
filtering MSR items by optional parameters. See Syntax of JSON filter below.   
In addition, it is possible to sort MSR items by optional parameter value. The name of the optional   
parameter for sorting should be specified in the existing API parameter sortFieldName. See   
Specifying sort field.   
All other parameters to GetFolderItemsEx2 API have the same meaning and functionality as in   
GetFolderItemsEx API.   
Before continuing reading, we highly recommended that you first read the Managing call optional   
parameters section of the CCSP Operation Guide.   
Syntax of JSON filter   
About filters   
There are two types of filters:   
•   
Basic — filters that run a logical operator on an MSR field.   
•   
Compound — filters that run a logical operator on other filters.

CCSP APIs Guide - 7.4   
19   
Syntax conventions   
The following conventions are used when describing the JSON syntax.   
Convention   
Description   
<name>   
The <name> should be replaced by a real value according to the context of name.   
| (pipe)   
Only one of the values separated by the pipe should be selected.   
…   
Repeated value   
Basic filters   
ContainsSubstring   
Purpose: Matches item if it contains the specified optional parameter and its value contains the   
string specified in <value>.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "ContainsSubstring",   
 "ComparisonMode": "CaseInsensitive|CaseSensitive",   
 "Value": "<optional-parameter-value>"   
}   
Exists   
Purpose: Matches item if it contains the specified optional parameter.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "Exists"   
}

CCSP APIs Guide - 7.4   
20   
IsBetween   
Purpose: Matches item if it contains the specified optional parameter and its value is between the   
specified <LowerBound> and <UpperBound> values.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "IsBetween",   
 "LowerBound": "<optional-parameter-value>",   
 "UpperBound": "<optional-parameter-value>"   
}   
IsEqualTo   
Purpose: Matches item if it contains the specified optional parameter and its value is equal to the   
specified <value>.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "IsEqualTo",   
 "Value": "<optional-parameter-value>"   
}   
IsGreaterOrEqual   
Purpose: Matches item if it contains the specified optional parameter and its value is greater or   
equal to the specified <value>.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "IsGreaterOrEqual",   
 "Value": "<optional-parameter-value>"   
}

CCSP APIs Guide - 7.4   
21   
IsGreaterThan   
Purpose: Matches item if it contains the specified optional parameter and its value is greater than   
the specified <value>.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "IsGreaterThan",   
 "Value": "<optional-parameter-value>"   
}   
IsInList   
Purpose: Matches item if it contains the specified optional parameter and its value is one of the   
specified values in <PossibleValues> list.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "IsInList",   
 "PossibleValues": [<optional-parameter-value >, …]"   
}   
IsLessThan   
Purpose: Matches item if it contains the specified optional parameter and its value is less than the   
specified <value>.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "IsLessThan",   
 "Value": "<optional-parameter-value>"   
}

CCSP APIs Guide - 7.4   
22   
IsLessThanOrEqual   
Purpose: Matches item if it contains the specified optional parameter and its value is less than or   
equal to the specified <value>.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "IsLessThanOrEqual",   
 "Value": "<optional-parameter-value>"   
}   
IsLike   
Purpose: Matches item if it contains the specified optional parameter and its value matches the   
pattern specified in <value>. Pattern must follow the rules of SQL LIKE predicate.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "IsLike",   
 "Value": "<pattern>"   
}   
NotExists   
Purpose: Matches item if it does not contain the specified optional parameter.   
Syntax:   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "<optional-parameter-name>",   
 "Operator": "NotExists"   
}

CCSP APIs Guide - 7.4   
23   
Compound filters   
Not   
Purpose: Matches item if result of filter specified by <filter> is false.   
Syntax:   
{   
 "Operator": "Not”,   
 "InnerFilter": <filter>   
}   
SearchFilterGroup   
Purpose: Performs a logical operator on result of multiple filters. When LogicalOperator is OR then   
item is matched if result of any of listed filters is true. When LogicalOperator is AND then item is   
matched if result of all listed filters is true.   
Syntax:   
{   
 "Operator": "SearchFilterGroup”,   
 “LogicalOperator”: “OR|AND”,   
 "Filters": [ <filter> , <filter>, … ]   
}

CCSP APIs Guide - 7.4   
24   
Filters support matrix   
The following table lists how each filter supports the various optional parameter data types. The   
entire query will fail if you use an incompatible combination of filter/data type.   
Filter   
Supported for   
String optional   
parameter?   
Supported for   
Integer optional   
parameter?   
Supported for   
Datetime   
optional   
parameter?   
Optional   
parameter   
must be   
indexed?   
ContainsSubstring   
Yes   
No   
No   
Yes   
Exists   
Yes   
Yes   
Yes   
No   
IsBetween   
No   
Yes   
Yes   
Yes   
IsEqualTo   
Yes   
Yes   
Yes   
Yes   
IsGreaterOrEqual   
No   
Yes   
Yes   
Yes   
IsGreaterThan   
No   
Yes   
Yes   
Yes   
IsInList   
Yes   
Yes   
Yes   
Yes   
IsLessThan   
No   
Yes   
Yes   
Yes   
IsLessThanOrEqual   
No   
Yes   
Yes   
Yes   
IsLike   
Yes   
No   
No   
No   
Not   
Yes   
Yes   
Yes   
No   
NotExists   
Yes   
Yes   
Yes   
No   
SearchFilterGroup   
Yes   
Yes   
Yes   
No   
Examples   
Match all items where Optional parameter “CallCounter” = 3   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "CallCounter",   
 "Operator": "IsEqualTo",   
 "Value": "3"   
}

CCSP APIs Guide - 7.4   
25   
Match all items where Optional parameter “CallCounter” > 3   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "CallCounter",   
 "Operator": "IsGreaterThan",   
 "Value": "3"   
}   
Match all items where Optional parameter “CallCounter” is 3 or 4   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "CallCounter",   
 "Operator": "IsInList",   
 "PossibleValues": [ "3" , “4” ]   
}   
Match all items where Optional parameter “Category” = “Important”   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "Category",   
 "Operator": "IsEqualTo",   
 "Value": " Important "   
}   
Match all items where Optional parameter “Category” contains the word “Important” in   
case-sensitive.   
 "Property": "OptionalParameters",   
 "SubProperty": "Category",   
 "Operator": "ContainsSubstring",   
 "ComparisonMode": " CaseSensitive",   
 "Value": " Important "   
}

CCSP APIs Guide - 7.4   
26   
Match all items where Optional parameter “CallCounter” != 3   
{   
 "Operator": "Not”,   
 "InnerFilter": {   
 "Property": "OptionalParameters",   
 "SubProperty": "CallCounter",   
 "Operator": "IsEqualTo",   
 "Value": "3"   
 }   
}   
Match all items where Optional parameter “CallCounter” > 3 AND “Category” = “Important”   
{   
 "Operator": "SearchFilterGroup”,   
 “LogicalOperator”: “AND”,   
 "Filters": [   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "CallCounter",   
 "Operator": "IsGreaterThan",   
 "Value": "3"   
},   
{   
 "Property": "OptionalParameters",   
 "SubProperty": "Category",   
 "Operator": "IsEqualTo",   
 "Value": " Important "   
}   
 ]   
}

CCSP APIs Guide - 7.4   
27   
Specifying sort field   
To sort the MSR items list by an optional parameter, specify “OptionalParameters.<optional-  
parameter-name>” in the sort field.   
Note   
•   
You can sort by optional parameter even if that parameter is not part of the JSON filter.   
•   
The optional parameter used for sorting does not have to be configured to allow filtering.   
•   
If an item matches the filter but does not have a value in the optional parameter used for   
sorting, then the item will be ranked at the top of the matched items result set if the sort   
order is ascending, or at the bottom of the result set if the sort order is descending.   
Example   
To sort MSR items by optional parameter “CallCounter”, specify   
"OptionalParameters.CallCounter" in the sortFieldName parameter of GetFolderItemsEx2   
API.

CCSP APIs Guide - 7.4   
28   
6: CCSP Chat API   
This chapter describes the JSON functions in the ChatExtension.dll.   
This chapter contains the following information:   
•   
Configuration   
•   
Flow diagrams   
•   
JoinChat   
•   
SendMSG   
•   
GetEvents   
•   
LeaveChat   
•   
FireEvent   
   
The CCSP Chat API provides an easy to use API that can be used on mobile and non-mobile devices   
and different html pages.   
The JSON ChatExtension.dll provides just the data which customers can use to implement their own   
solutions.   
The JSON interface gets the requests inside the HTTP request and provides the response in JSON.   
The ChatExtension.dll recognizes the user by connection ID. If the user changes his IP, the   
ChatExtension.dll knows it is the same connection ID and therefore the same user.   
The original HTML ChatExtension.dll provides only configurable HTML pages. For details, see A: CCSP   
Web Support integration.   
The following table shows which JSON function replaces which HTML page in the original HTML   
ChatExtension.dll.   
JSON   
HTML   
JoinChat   
Connect   
SendMSG   
Write   
GetEvents   
Read and Status   
LeaveChat   
Close

CCSP APIs Guide - 7.4   
29   
Configuration   
To support cross domains on the API, you need to configure CCSP to support Cross-Origin Resource   
Sharing (CORS). CORS is a web browser technology specification that defines ways for a web server   
to allow its resources to be accessed by a web page from a different domain.   
Note   
The Chat Client host (where the Chat Client SDK runs) and the CCSP host (where chatextension dll   
runs) must be in the same zone..   
   
To configure CCSP to support CORS:   
1.   
On the CCSP server, where chatextension dll runs, open IIS Manager and add the following   
headers to HTTP Response Headers:   
o   
Access-Control-Allow-Origin: \*   
o   
Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept,   
AgentGUID, SOAPAction   
o   
Access-Control-Allow-Methods: POST,GET,OPTIONS   
   
2.   
If the chat client uses jQuery, on the server where the Chat Client SDK runs, set the following   
flag:   
jQuery.support.cors = true;

CCSP APIs Guide - 7.4   
30   
Flow diagrams   
The following diagram shows the flow between the client and the ChatExtension.dll.

CCSP APIs Guide - 7.4   
31   
JoinChat flow   
The following diagram show the flow between the client and the ChatExtension.dll for JoinChat.   
   
SendMsg flow   
The following diagram show the flow between the client and the ChatExtension.dll for SendMsg.   
   
GetEvents flow   
The following diagram show the flow between the client and the ChatExtension.dll for GetEvents.   
   
LeaveChat flow   
The following diagram show the flow between the client and the ChatExtension.dll for LeaveChat.

CCSP APIs Guide - 7.4   
32   
JoinChat   
JoinChat makes a new connection to the contact center. See JoinChat flow.   
Request   
The JoinChat POST method URL is:   
[path]/ChatExtension.dll?joinchat   
   
The JSON data in the body is:   
{   
 "API\_Version ":1.0 ,   
 "Call\_Center\_Address": "",   
 "Call\_Center\_QueueName": "",   
 "Call\_Center\_Port": ,   
 "Calling\_User\_Priority": ,   
 "Calling\_User\_URL": "",   
 "Calling\_User\_FirstName": "",   
 "Calling\_User\_LastName": "",   
 "TenantID": "",   
 "ApplicationID": ",   
 "AccountID": "",   
 "Calling\_User\_Skills":"",   
 "Routing\_Priority": ,   
 "TQOS": ,   
 "Calling\_User\_HardMessage":""   
 "OptionalParameterCount": [   
 {   
 "key":””,   
 "value":””   
 },   
 {   
 "key":””,   
 "value":””   
 },   
 {   
 "key": “”,   
 "value":””

CCSP APIs Guide - 7.4   
33   
 }   
 ]   
}   
For example:   
{   
 "API\_Version ":"1.0",   
 "Call\_Center\_Address": "10.116.200.50",   
 "Call\_Center\_QueueName": "Age",   
 "Call\_Center\_Port": 2324,   
 "Calling\_User\_FirstName": "Dana",   
 "Calling\_User\_LastName": "shimon",   
 "TenantID":"TT" ,   
 "ApplicationID": "TTAplication",   
 "Calling\_User\_Skills":"A+B+C"   
 "Calling\_User\_HardMessage":"I need help"   
 "OptionalParameterCount": [   
 {   
 "key": "title",   
 "value":"Mr"   
 },   
 {   
 "key": "empl\_Id",   
 "value":"802431330"   
 },   
 {   
 "key": "date",   
 "value":"16 Nov 2013"   
 }   
 ]   
 }

CCSP APIs Guide - 7.4   
34   
The following table describes the variables included in the JSON data.   
Name   
Description   
Required   
Default   
Type   
API\_Version   
API version   
Yes   
1.0   
String   
Call\_Center\_Address   
ACD address or name   
Yes   
Server name   
for the www   
String   
Call\_Center\_QueueName   
Name of queue to place   
caller in. This name must   
match a queue on the ACD.   
Yes   
DefaultQueue   
String   
Call\_Center\_Port   
Port for ACD Server   
communication   
Yes   
2324   
Integer   
Calling\_User\_Priority   
Assign priorities to calls.   
Calls with higher priority will   
always be routed to agents   
before the calls with lower   
priority in the same queue.   
No   
-1   
Integer   
Calling\_User\_FirstName   
Caller's first name   
No   
Caller   
String   
Calling\_User\_LastName   
Caller's last name   
No   
Caller   
String   
TenantID   
Tenant name to place the call   
in. This name must match a   
tenant on the ACD.   
Yes   
DefaultTenant   
String   
ApplicationID   
Application name to place   
the call in. This name can   
match an existing application   
name in the ACD. If a match   
is found, the selected   
application is run even if the   
application rule is not   
evaluated to TRUE. If a   
match is not found, the first   
application whose rule   
evaluates to TRUE is   
selected.   
Yes   
DefaultApplic  
ation   
String   
Calling\_User\_Skills   
List of required skills —   
separated by plus (+) signs   
No   
""   
String   
AccountID   
Associated with the   
customer account   
No   
"0"   
String

CCSP APIs Guide - 7.4   
35   
Name   
Description   
Required   
Default   
Type   
Routing\_Priority   
Dynamic routing. Route each   
call to the most qualified   
agent based on skills and the   
assigned priority (Call   
maturity).   
No   
0   
Integer   
TQOS   
Target Quality of Service —   
the number of seconds   
within which the call should   
be answered.   
No   
0   
Integer   
Calling\_User\_Hard   
Message   
First message agent will see   
when the caller connects   
No   
I need some   
help   
String   
Calling\_User\_URL   
URL to pass to agent   
No   
""   
String   
OptionalParameterCount   
Optional parameter values   
are available for the life of   
the call.   
Optional parameter   
structure in Chat API is:   
"OptionalParameterCount": [   
 {   
 "key":””,   
 "value":””   
 },   
 {   
 "key":””,   
 "value":””   
 },   
 {   
 "key": “”,   
 "value":””   
 }   
 ]   
Note   
The maximum total length   
for the optional parameters,   
including the optional   
parameter names, is 1024   
bytes.   
No   
""   
String

CCSP APIs Guide - 7.4   
36   
Response   
The response is 200 OK.   
It contains the new Connection\_ID, in the format: XXXXX\_YYYYY   
The client must send the same string in every Connection\_ID.   
The JSON data in the body is:   
{   
 "Connection\_ID":""   
}   
   
For example:   
{   
 "Connection\_ID":"6047936\_1379917213"   
}   
   
The Available-Connections key specifies the number of currently available simultaneous chat   
connections, that is, the maximum possible minus the number of connections currently in use. The   
maximum possible simultaneous chat connections is 899, because of IPWorks limitations. If the   
simultaneous chat calls limit is reached and a new call arrives, the JSON response to JoinChat is   
error code 18 (MAX\_CONNECTIONS\_REACHED).   
Errors   
If the server encounters an error, the response is 500 Internal Server Error.   
If a required field is empty, the port is invalid, or the application version is invalid, one of the   
following error codes can be added to the error 200 message, in the following format:   
{“Command”:”Error”, “Error\_Code”: X}   
   
Description   
Code   
INVALID\_PORT   
4   
INVALID\_QUEUE\_NAME   
8   
CONNECTION\_FAILED   
12   
INVALID\_TENANTID   
13   
INVALID\_APP\_VERSION   
14   
INVALID\_APPLICATIONID   
15   
MAX\_CONNECTIONS\_REACHED   
18

CCSP APIs Guide - 7.4   
37   
For example:   
{“Command”:”Error”, “Error\_Code”: 13}   
   
If a field is missing, the default value is used. For example, if only the following fields are available :   
{   
 "Call\_Center\_Address": "10.116.200.50",   
 "Call\_Center\_QueueName": "Age",   
 "Calling\_User\_FirstName": "Dana",   
 "Calling\_User\_LastName": "shimon",   
 "TenantID":"TT" ,   
 "ApplicationID": "TTAplication",   
 "Calling\_User\_Skills":"A+B+C"   
 "Calling\_User\_HardMessage":"I need help"   
}   
   
Then the missing fields use the default values:   
 "API\_Version ":"1.0"   
 "Call\_Center\_Port": 2324   
 "Calling\_User\_Priority":-1   
 "Calling\_User\_URL": "www.gmail.com"   
 "AccountID": "0",   
 "Routing\_Priority":0   
 "TQOS":0   
If Call\_Center\_Address is missing, the IP address of the ACD server on the WWW server is used.

CCSP APIs Guide - 7.4   
38   
SendMSG   
SendMSG sends a message to the agent. See SendMsg flow.   
Request   
The SendMSG POST method URL is:   
[path]/ChatExtension.dll?sendmsg   
   
The JSON data in the body is:   
{   
   
 "Connection\_ID ": "",   
 "Message\_Text": ""   
}   
   
The Connection\_ID format is XXXXX\_YYYYY and the client must send the same string in every   
Connection\_ID.   
The Message\_Text can be a maximum of 1024 characters.   
For example:   
{   
   
 "Connection\_ID ": "6047936\_1379917213",   
 "Message\_Text": "I have a question"   
}   
Response   
The response is 200 OK.   
Error   
If the server encounters an error, the response is 500 Internal Server Error.

CCSP APIs Guide - 7.4   
39   
GetEvents   
GetEvents returns pending requests for a new agent, and error or status messages. See GetEvents   
flow.   
Request   
The GetEvents GET method URL is:   
[path]/ChatExtension.dll? command=getevents& Connection\_ID= XXXXX\_YYYYY   
&Last\_Event\_ID=Z&GUID\_ID=b42f53a2-1297-4403-b669-a50eb611b7e7   
   
The Connection\_ID format is XXXXX\_YYYYY and the caller must send the same string in every   
connection ID.   
The Last\_Event\_ID is used by the client to acknowledge the last event message ID that was   
successfully received. If the last event is not the same as the one the server sent, the server will   
send the conversion again.   
Note   
In the URL, command must be in GET method.   
   
For example, if the last event the client received from the server was 4:   
ChatExtension.DLL? command=getevents& Connection\_ID=   
6047936\_1379917213&Last\_Event\_ID=4   
   
The GUID\_ID is an optional parameter and must be unique for each GetEvents send. It is used to   
track the GetEvents messages between the client and ChatExtension for debugging purposes.   
If not used, the GetEvents GUID\_ID is tracked only at ChatExtension level but not between the client   
and ChatExtension. The GetEvents GET method URL without GUI\_ID is:   
[path]/ChatExtension.dll? command=getevents& Connection\_ID= XXXXX\_YYYYY   
&Last\_Event\_ID=Z   
   
The user timeout period is configured on the ACD server, in the following registry key:   
HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\CosmoCom\Universe   
\CurrentVersion\ChatExtension\Parameters\<tenant name> UserTimeoutPeriod   
The default is 45.   
Each time the ACD server receives GetEvents, the server updates the last access. If last access +   
UserTimeoutPeriod < now, the CosmoCaller Extension deletes the user’s connection.

CCSP APIs Guide - 7.4   
40   
Response   
The response is 200 OK.   
The JSON data in the body is:   
{   
 "Connection\_ID":"",   
 Events:   
 [   
 {   
 "Command":"status",   
 "Status\_Code":,   
 "Additional\_Information":"", wating state (165), discconect   
   
   
   
   
   
   
   
   
(no agent/refused)(107)   
 "Estimated\_Time": , wating state(165)   
 "Event\_ID":   
 },   
 {   
 "Command":"read",   
 "Participant\_Name":"",   
 "Message\_Text":"",   
 “Pushed\_Url”:””, if URL pushed from the agent   
 "Event\_ID":   
 },   
 {   
 "Command":"Participantdisconnected",   
 "Participant\_Name":"",   
 "Event\_ID":   
 },   
 {   
 "Command":"Error",   
 "Error\_Code":,   
 "Event\_ID" :   
 }   
 ]   
}

CCSP APIs Guide - 7.4   
41   
GetEvents messages are added as follows:   
•   
read − any messages from the agent   
•   
Participantdisconnected − if the agent disconnects   
•   
Error − a wrong Event\_ID   
•   
500 Internal Server Error − an internal error   
•   
Event\_ID − sequential and bigger from message to message   
•   
Connection\_ID only – if the server has nothing to send   
   
The following example shows the Event\_ID flow.

CCSP APIs Guide - 7.4   
42   
Status codes   
The following table describes the status codes.   
Description   
Code   
Calling Call Center   
   
   
104   
Waiting for Agent   
   
105   
In Call   
106   
Not Connected   
107   
Connected to the call center   
133   
Call placed on hold   
157   
Waiting in queue   
165   
Call ID parameter   
166   
Chat Connection Server has reached its   
maximum capacity   
167   
   
Waiting in queue additional information status codes   
The following table describes the Waiting in queue additional information status codes.   
Description   
Code   
Less than   
1   
About   
2   
Longer than   
3

CCSP APIs Guide - 7.4   
43   
Not Connected additional information status codes   
The following table describes the Not Connected additional information status codes.   
Description   
Code   
Unknown reason   
6   
Insufficient memory   
7   
No satisfying agents   
8   
Invalid queue   
9   
No primary connection server address   
10   
No call ID   
11   
Invalid external queue   
12   
Unknown skill name is specified in call   
requirements   
13   
Stream with reference your already exist   
14   
Cannot find the call by your reference   
15   
Call with your reference cannot be accepted   
by the agent   
16   
There is no connection available now   
17   
Agent refused   
18   
Error code   
The following table describes the invalid Event\_ID error code.   
Description   
Code   
Invalid Event\_ID   
17

CCSP APIs Guide - 7.4   
44   
Examples   
In the following example, the agent sent a message, was in call, and then disconnected.   
{   
 "Connection\_ID":"6047936\_1379917213",   
 "Events":   
 [   
 {   
 "Command":"read",   
 "Participant\_Name":"Dana",   
 "Message\_Text":"How can I help you",   
 "Event\_ID":1   
 },   
 {   
 "Command":"status",   
 "Status\_Code":106,   
 "Event\_ID":2   
 },   
 {   
 "Command":"Participantdisconnected",   
 "Participant\_Name":"Dana",   
 "Event\_ID":3   
 }   
   
 ]   
}

CCSP APIs Guide - 7.4   
45   
In the following example, the agent disconnected:   
{   
 "Connection\_ID":"6047936\_1379917213",   
 "Events":   
 [   
 {   
 "Command":"Participantdisconnected",   
 "Participant\_Name":"Dana",   
 "Event\_ID":10   
 },   
 {   
 "Command":"status",   
 "Status\_Code":107,   
 "Event\_ID":11   
 }   
 ]   
}   
   
In the following example, the Event\_ID is invalid:   
{   
 "Connection\_ID":"6047936\_1379917213",   
 "Events":   
 [   
 {   
 "Command":"Error",   
 "Error\_Code":17,   
 "Event\_ID :20   
 },   
 ]   
}   
   
In the following example, the server has nothing to send, so only the Connection\_ID appears:   
{   
 "Connection\_ID":"6047936\_1379917213"

CCSP APIs Guide - 7.4   
46   
LeaveChat   
LeaveChat disconnects the caller from the chat. See LeaveChat flow.   
Request   
The LeaveChat POST method URL is:   
[path]/ChatExtension.dll?leavechat   
   
The JSON data in the body is:   
{   
 "Connection\_ID":""   
}   
   
The Connection\_ID format is XXXXX\_YYYYY and the client must send the same string in every   
Connection\_ID.   
For example:   
{   
   
 "Connection\_ID ": "6047936\_1379917213",   
}   
Response   
The response is 200 OK.

CCSP APIs Guide - 7.4   
47   
FireEvent   
The FireEvent sends a general chat message from the customer to the agent.   
Request   
The FireEvent POST method URL is:   
[path]/ChatExtension.dll? fireevent   
   
The JSON data in the body is:   
{   
"Connection\_ID": "",   
" Fire\_Event\_Name":   
" OptionalParametersCount ": (not mandatory)   
}   
   
The Connection\_ID format is XXXXX\_YYYYY and the client must send the same string in every   
Connection\_ID.   
The Fire\_Event\_Name parameter is a mandatory string value and cannot be empty.   
OptionalParametersCount is optional and, if it exists, includes the key and value pairs.   
For example:   
{   
"Connection\_ID": "6047936\_1379917213",   
"Fire\_Event\_Name":"evt\_5",   
"OptionalParameterCount":[{"key":"FirstKey","value":"FirstVal"},{"key":"Second  
Key","value":"SecondVl"}] }   
CCSP UI   
For CCSP UI, you can send a generic event that can eventually pass everything for chat.   
For example:   
If the Chat API sends the following FireEvent with next payload:   
{"Connection\_ID":"474793400\_1492938906","Fire\_Event\_Name":"SC\_ATTACHMENT","Opt  
ionalParameterCount":[   
{"key":"SC\_URL","value":"\_\_http://google.com"}   
]}

CCSP APIs Guide - 7.4   
48   
You can subscribe to NewChatEvent and extract parameters as below:   
window.app.PubSub.subscribe('NewChatEvent', null, function(evt){   
 var eventName = evt.EventName;   
 switch(eventName)   
{ case "SC\_ATTACHMENT": var scUrl = getValueByKey(evt.chatEvent.OptPar,   
"SC\_URL"); default: return; }   
});   
The following example shows the received object.   
   
Response   
The response is 200 OK (with {} in the body as requested by JSON).   
The following example shows the FireEvent flow.   
   
Error   
If Fire\_Event\_Name is empty or does not exist, the response is 501 Not Implemented.   
If the server encounters any other error, the response is 500 Internal Server Error.

CCSP APIs Guide - 7.4   
49   
7: CCSP CIS to Connector API   
Note   
You must work with Enghouse PS to build interfaces to the CIS. Currently, we do not support   
customer calls directly in Development related to CIS API issues.   
   
This chapter describes how to connect to the CCSP Central Integration Server (CIS).   
This chapter contains the following information:   
•   
High-level overview   
•   
Display the HTML web page   
•   
Log in   
•   
Connect through WebSocket   
•   
Create a GET\_EVENTS request   
•   
Control the processing rate   
•   
Receive messages   
•   
Stop events   
•   
Log out   
•   
Add events to CISEvents.xml

CCSP APIs Guide - 7.4   
50   
High-level overview   
The following diagram shows the communication between the CIS and the Connector in high-level   
steps.   
   
The following sections provide a simple connector code snapshot that demonstrates how to   
connect to the CIS.   
The client sample uses WebSocketSharp and Newtonsoft.JSON 3rd party library.   
The communication is based on JSON format.   
In the following example, the CIS configuration is:   
•   
CIS IP — 10.120.30.71   
•   
HTTP IP — 10.120.30.71 , HTTP PORT = 8088   
•   
WS IP — 10.120.30.71, WS PORT = 8089.

CCSP APIs Guide - 7.4   
51   
Display the HTML web page   
If you connect through a web browser, to display the html web page, browse to:   
http://<CIS HTTP IP>:<HTTP PORT>/cis/test   
Log in   
The first step to connect to the CIS, is the login process.   
To log in, the connector must issue an HTTP POST request to cis\login uri, with the username and   
password parameters represented in string.   
Request body format   
bodyText = {"username":"<user@domain>","password":"<user password>"}   
For example:   
bodyText = "username":"CCUAdmin@ex2010.loc","password":"pw"}   
   
The following table describes the parameters.   
Parameter name   
Type   
Mandatory   
username   
String   
Yes   
password   
String   
Yes   
   
If the request is successful, the CIS sends a reply with the following variables:   
•   
session\_id — represents the session ID. Keep this ID to use for further requests to the CIS.   
•   
tenant\_id — the tenant that is permitted to be monitored by this user. For the landlord, the   
value is -1.   
•   
assigned\_tenants — the tenants that are assigned to the CIS. If no tenants are assigned to   
the CIS, the CIS monitors all tenants, and the value is -1.   
•   
event\_providers — the list of event providers that are associated with the CIS.   
•   
supported\_messages — the list of messages that CIS can support.   
•   
web\_socket\_ip — the IP used by the CIS for web socket communication. If the value is 0.0.0.0,   
the CIS listens on all IPs assigned to it.   
•   
web\_socket\_port — the port used by the CIS for web socket connection, for example, 8089.   
•   
correlation\_id — is per GET\_EVENT request. Use to correlate the messages being received by   
the CIS to the request ID that was made by the connector.

CCSP APIs Guide - 7.4   
52   
Code snapshot   
 //HttpClient client;   
using (HttpClient client = new HttpClient())   
{   
 client.BaseAddress = new Uri(@"http://10.120.30.71:8088/cis/login");   
 client.DefaultRequestHeaders.Accept.Clear();   
 client.DefaultRequestHeaders.Accept.Add(new   
MediaTypeWithQualityHeaderValue("application/json"));   
   
   
 var data = new OrderedDictionary();   
 data["username"] = @"CCUAdmin@ex2010.loc";   
 data["password"] = @"pw";   
   
   
 var postDataJson = JsonConvert.SerializeObject(data);   
 var postdataString = new StringContent(postDataJson, new   
UTF8Encoding(), "application/json");   
   
 var response = await   
client.PostAsync("http://10.120.30.71:8088/cis/login", postdataString);   
   
 var stringContent = await response.Content.ReadAsStringAsync();   
   
 Console.WriteLine("{0}", stringContent);   
Example of HTTP POST Message being sent   
{"username":"CCUAdmin@ex2010.loc","password":"pw"}

CCSP APIs Guide - 7.4   
53   
Example of HTTP POST Message being received   
The following examples show the types of HTTP POST Message being received.   
Successful login   
{   
 "session\_id": "SSN6EEV5",   
 "result": "success",   
 "tenant\_id": -1,   
 "agent\_global\_id": 2234049340917612544,   
 "assigned\_tenants": -1,   
 "event\_providers": [   
 "VCS,11",   
 "VCS,30"   
 ],   
 "supported\_messages": {   
 "555": "CIS\_SNAPSHOT\_INFO",   
 "164": "CS\_CALL\_OFFERED",   
 "47": "CS\_CALLOUT",   
 "48": "CS\_ACK\_CALLOUT",   
 "49": "CS\_NACK\_CALLOUT",   
 "3": "CS\_CALLARRIVAL",   
 "5": "CS\_SCRIPTCOMPLETED",   
 "4": "CS\_ACKCALLARRIVAL",   
 "28": "AGENT\_RING",   
 "8": "CS\_PICKUPCALL",   
 "120": "CS\_AGENTCONNECTED",   
 "37": "CS\_NACKPICKUPCALL",   
 "9": "CS\_ACKCALLPICKUP",   
 "327": "CS\_LOG\_MediaData\_VCS",   
 "325": "CS\_CALL\_TERMINATED"   
 },   
 "https\_enabled": false,   
 "http\_ip": "10.120.30.71",   
 "http\_port": 8088,   
 "https\_ip": "10.120.30.71",   
 "https\_port": 8443,   
 "web\_socket\_ip": "0.0.0.0",

CCSP APIs Guide - 7.4   
54   
 "web\_socket\_port": 8089,   
 "web\_socket\_secure\_ip": "0.0.0.0",   
 "web\_socket\_secure\_port": 8431   
}   
Invalid user or password   
{   
 "result": "error",   
 "reason": "Authorization error - check your credentials or contact your   
administrator."   
}   
Invalid parameters   
For example, user instead of username.   
   
 {   
 "result": "error",   
 "reason": "Username or password are empty."   
 }   
Connect through WebSocket   
After the login pocess, you need to connect to the CIS through WebSocket.   
The client sample uses the WebSocketSharp library to implement the web socket.   
using (var ws = new   
WebSocketSharp.WebSocket(String.Format(@"ws://{0}:{1}/cis/{2}",   
data.web\_socket\_ip, data.web\_socket\_port, data.session\_id)))   
 {   
   
 ws.OnMessage += (sender, e) => Console.WriteLine(" <==== CIS Message : " +   
e.Data);   
 ws.Connect();   
}   
   
In the above example:   
•   
data.web\_socket\_ip — represents the websocket IP that was received in the logon response.   
•   
data.web\_socket\_port — represents the websocket port that was received in the port   
response.   
•   
data.session\_id — represents the session ID that was received in the logon response.

CCSP APIs Guide - 7.4   
55   
Create a GET\_EVENTS request   
After connecting to the CIS, you can create a GET\_EVENTS request.   
To send a get request, you need to send an HTTP POST request to cis\getevents.   
Request body format   
bodyText = {   
"session\_id":"<sessionId received in login response>",   
"event\_descriptors": [ <list of event descriptor>   
 {"groups":[<list of groupIds>],"tenant\_id":<requested tenant Id >,"event\_type":"< event type>",   
"event\_mask":[<list of event id>]},   
 ]   
}   
For example:   
bodyText = {   
"session\_id":"87HL4HAL",   
"event\_descriptors":[   
   
{"groups":[2],"tenant\_id":0,"event\_type":"both","event\_mask":[16,17,43,164]},   
 {"groups":[3,3],"tenant\_id":1,"event\_type":"both"}   
]   
}   
The following table describes the parameters.   
Parameter name   
Type   
Mandatory   
session\_id   
String   
Yes   
event\_descriptors   
Data structure of:   
tenant\_id, list of groups,   
event\_type, and event\_mask   
   
tenant\_id   
Int   
Yes. See Restrictions below.   
groups   
Array of int   
Yes. See Restrictions below.   
event\_type   
String   
Yes. See Restrictions below.   
event\_mask   
Array of int   
Yes. See Restrictions below.

CCSP APIs Guide - 7.4   
56   
The body format should be in json format.   
Restrictions   
The following parameters have restrictions:   
•   
groups — you cannot request to monitor tenant Id = -1, with list of groups. The group list   
must be empty.   
•   
tenant\_id — you cannot request to monitor tenant Id = -1, and an additional tenant. However,   
you can request to monitor several tenants, represented with a specific tenant ID.   
•   
event\_type — can contain the following: "agent", "call", "both". If the event\_type is not sent, it   
is interpreted by the CIS as "both".   
•   
event\_mask — represents the list of events (messages) that the connector is interested in. If   
it's not sent, it is interpreted by the CIS as all messages.   
   
If the request is successful, the CIS sends the response with a correlation\_id, more messages,   
including the state of all of the event providers, and the snapshot (if it's working in STAND-BY   
mode).   
The correlation\_id is per GET\_EVENT request, and should be used to correlate the messages being   
received by the CIS to the request ID that was made by the connector.   
Note   
It is important to count the number of events being received. For more information, see Control   
the processing rate.

CCSP APIs Guide - 7.4   
57   
var getEventsRequest= new OrderedDictionary();   
getEventsRequest["session\_id"] = data.session\_id;   
getEventsRequest["event\_descriptors"] = new List<EventDescriptor> {new   
EventDescriptor() { tenant\_id = data.tenant\_id, groups = new List<int>() ,   
event\_type = "both"}};   
   
var postGetEventsJson = JsonConvert.SerializeObject(getEventsRequest);   
var postGetEventsString = new StringContent(postGetEventsJson, new   
UTF8Encoding(), "application/json");   
   
response = client.PostAsync("http://10.120.30.71:8088/cis/getevents",   
postGetEventsString).Result;   
   
stringContent = response.Content.ReadAsStringAsync().Result;   
   
Console.WriteLine("{0}", stringContent);   
try   
{   
 data = JsonConvert.DeserializeObject<Data>(stringContent);   
}   
catch (Exception e)   
{   
   
}   
Example of HTTP POST Message being sent   
bodyText = {   
"session\_id":"87HL4HAL",   
"event\_descriptors":[   
   
{"groups":[2],"tenant\_id":0,"event\_type":"both","event\_mask":[16,17,43,164]},   
 {"groups":[3,3],"tenant\_id":1,"event\_type":"both"}   
]   
}

CCSP APIs Guide - 7.4   
58   
Example of HTTP POST Message being received   
{   
 "session\_id": "FRO4678R",   
 "result": "success",   
 "correlation\_id": 8   
}   
   
...   
   
 "message": "event\_provider\_status",   
 "session\_id": "W3EPLM9R",   
 "statuses": [   
 {   
 "event\_provider": "VCS,11",   
 "state": "connected"   
 },   
 {   
 "event\_provider": "VCS,30",   
 "state": "connected"   
 }   
 ]   
}   
   
…   
   
 "message": "snapshot",   
 "event\_provider": "VCS,11",   
 "infos": [   
 {   
 "message": "cis\_snapshot\_info",   
 "call\_id": 2236360475703377931,   
 "connection\_server\_id": 11,   
 "tenant\_id": 0,   
 "acount\_id": 0,   
 "queue\_id": 0,   
 "queue\_name": "",   
 "caller\_ani": "5010",

CCSP APIs Guide - 7.4   
59   
 "caller\_dnis": "5800",   
 "caller\_id": "\"AAAA\""   
 }   
 ],   
 "session\_id": "W3EPLM9R",   
 "correlation\_id": 7   
}   
   
…   
   
 "message": "snapshot",   
 "event\_provider": "VCS,30",   
 "infos": [],   
 "session\_id": "W3EPLM9R",   
 "correlation\_id": 7   
}   
Control the processing rate   
Slow processing of the events on the connector side can cause performance issues on the server   
side, because the messages queue grows.   
The CIS implements a simple rate control mechanism, which is based on feedback from the   
connector.   
Every 5 seconds, the connector should send a JSON-formatted message through WebSockets   
notifying how many events it received.   
Message body format   
bodyText = {"session\_id":<session\_id>,"event\_recieved":<number\_of\_events\_recieved>}   
For example:   
bodyText = {"session\_id":"MQLHHAQF","event\_recieved":238}   
   
If there is a difference of more than 10,000 events, the CIS closes the web socket connection with   
the following notification:   
{ "message" , "connector\_rate" } ,   
{ "error" , "connector\_not\_fast\_enough"}

CCSP APIs Guide - 7.4   
60   
Code snapshot   
var getEventsRequest = new OrderedDictionary();   
getEventsRequest["session\_id"] = data.session\_id;   
getEventsRequest["event\_recieved"] = NumberOfSamples;   
   
var postGetEventsJson = JsonConvert.SerializeObject(getEventsRequest);   
ws.Send(String.Format("{0}", postGetEventsJson));   
Example of message being sent using WebSockets   
bodyText = {"session\_id":"MQLHHAQF","event\_recieved":238}   
Example of error message sent from CIS   
If the messages are not being processed fast enough, the CIS terminates the session and sends the   
following notifiication:   
{ "message" , "connector\_rate" } ,   
{ "error" , "connector\_not\_fast\_enough"}   
Receive messages   
After connecting successfully to the CIS using the HTTP and WebSocket connections, and issuing a   
succesful GET\_EVENT REQUEST, the CIS starts sending message for the requested tenant, groups,   
event type, and event mask.   
   
 "session\_id": "W3EPLM9R",   
 "event\_provider": "VCS,11",   
 "correlation\_id": 7,   
 "message": {   
 "message": "cs\_call\_arrival",   
 "queue\_id": 1,   
 "queue\_name": "DefaultQueue",   
 "connection\_server\_id": 11,   
 "tenant\_id": 0,   
 "acount\_id": 0,   
 "routing\_priority": 0,   
 "call\_id": 2236360475703377931,   
 "ipc\_source\_address": "VCS,11",   
 "ipc\_dst\_address": "VCS,11",   
 "optional\_parameters": [

CCSP APIs Guide - 7.4   
61   
 {   
 "Item1": "parameter\_name",   
 "Item2": "APPselector\_RuleMatch"   
 },   
 {   
 "Item1": "parameter\_value",   
 "Item2": "TRUE with AppID: 1"   
 },   
 {   
 "Item1": "parameter\_name",   
 "Item2": "XMLInterpreter\_lineid"   
 },   
 {   
 "Item1": "parameter\_value",   
 "Item2": "3"   
 },   
 {   
 "Item1": "parameter\_name",   
 "Item2": "XMLInterpreter\_trunkid"   
 },   
 {   
 "Item1": "parameter\_value",   
 "Item2": "0"   
 }   
 ]   
 }   
}   
Stop events   
To stop getting events, the connector must issue an HTTP POST message to cis\stopevents.

CCSP APIs Guide - 7.4   
62   
Request body format   
bodyText = {"session\_id":<session\_Id>,"correlation\_id":<correlation id>}   
For example:   
bodyText = {"session\_id":"87HL4HAL","correlation\_id":3}   
   
The following table describes the parameters.   
Parameter name   
Type   
Mandatory   
session\_id   
String   
Yes   
correlation\_id   
int   
Yes   
Code snapshot   
var stopEventsRequest = new OrderedDictionary();   
stopEventsRequest["session\_id"] = getEventsData.session\_id;   
stopEventsRequest["correlation\_id"] = getEventsData.correlation\_id;   
   
var postStopEventsJson = JsonConvert.SerializeObject(stopEventsRequest);   
   
var postStopEventsString = new StringContent(postStopEventsJson, new   
UTF8Encoding(), "application/json");   
   
response = client.PostAsync("http://10.120.30.71:8088/cis/stopevents",   
postStopEventsString).Result;   
   
stringContent = response.Content.ReadAsStringAsync().Result;   
   
Console.WriteLine("{0}", stringContent);   
Example of HTTP POST Message being sent   
bodyText = {"session\_id":"90H4RTUL","correlation\_id":13}

CCSP APIs Guide - 7.4   
63   
Example of HTTP Response Message being received   
 "session\_id": "90H4RTUL",   
 "message": "$stop\_events$",   
 "reason": "user request",   
 "correlation\_id": 13   
}   
…   
{   
 "msg": "stopevents",   
 "reason": "user request",   
 "session\_id": "90H4RTUL",   
 "correlation\_id": 13   
}   
Log out   
To log out the user, the connector must issue an HTTP POST message to cis\logout.   
Request body format   
bodyText = {"session\_id":<session\_id>}   
For example:   
bodyText = {"session\_id":"MJL12VBK"}   
   
The following table describes the parameter.   
Parameter name   
Type   
Mandatory   
session\_id   
String   
Yes   
Code snapshot   
var logoutRequest = new OrderedDictionary();   
logoutRequest["session\_id"] = getEventsData.session\_id;   
   
var logoutJson = JsonConvert.SerializeObject(logoutRequest);   
var postLogoutString = new StringContent(logoutJson, new UTF8Encoding(),   
"application/json");

CCSP APIs Guide - 7.4   
64   
response = client.PostAsync("http://10.120.30.71:8088/cis/logout",   
postLogoutString).Result;   
   
stringContent = response.Content.ReadAsStringAsync().Result;   
   
Console.WriteLine("{0}", stringContent);   
Example of HTTP POST Message being sent   
bodyText = {"session\_id":"MJL12VBK"}   
Example of HTTP POST Message being received   
{   
 "result": "success",   
 "reason": "user logout",   
 "session": " MJL12VBK "   
}   
Add events to CISEvents.xml   
CISEvents.xml is a list of messages (events) that CIS supports and is located under Server   
Components\Configuration.   
In case of verification, the CIS serializes the CIS\_EVENT embedded message and forwards it to the   
web session.   
The communication between the CIS and the connector is in JSON format.   
Structure of CISEvents.xml   
CISEvents.xml contains the list of messages and FIDS that CIS will transfer to the web sessions. The   
default CISEvents.xml file appears as follows:   
   
<CISEvents>   
 <IPCMessage type="CS\_SCRIPTCOMPLETED" jsonname="script\_completed">   
 <Field fid="FID\_CALLID" jsonname="call\_id" />   
 <Field fid="FID\_TENANT\_ID" jsonname="tenant\_id" />   
 <Field fid="FID\_EVENTID" jsonname="event\_id" />   
 <Field fid="ERROR\_STATUS" jsonname="error\_status" />   
 <Field fid="IPC\_SOURCE\_ADDRESS" jsonname="ipc\_source\_address" />   
 <Field fid="IPC\_DESTINATION\_ADDRESS" jsonname="ipc\_dst\_address" />   
 </IPCMessage>

CCSP APIs Guide - 7.4   
65   
 <IPCMessage type="CS\_ACKCALLARRIVAL" jsonname="ack\_callarrival">   
 <Field fid="FID\_CALLID" jsonname="call\_id" />   
 <Field fid="FID\_TENANT\_ID" jsonname="tenant\_id" />   
 <Field fid="IPC\_SOURCE\_ADDRESS" jsonname="ipc\_source\_address" />   
 <Field fid="IPC\_DESTINATION\_ADDRESS" jsonname="ipc\_dst\_address" />   
 </IPCMessage>   
 <IPCMessage type="AGENT\_RING" jsonname="agent\_ring">   
 <Field fid="FID\_CALLID" jsonname="call\_id" />   
 <Field fid="FID\_TENANT\_ID" jsonname="tenant\_id" />   
 <Field fid="AGENT\_GLOBAL\_ID" jsonname="agent\_global\_id"/>   
 <Field fid="AGENT\_IPC\_CONTROL\_ADDRESS" jsonname="agent\_ais\_id"/>   
 <Field fid="AGENT\_PERSONID" jsonname="agent\_person\_id"/>   
 <Field fid="AGENT\_GROUP\_GLOBAL\_ID" jsonname="agent\_group\_id"/>   
 <Field fid="FID\_ISCIRCUITSWITCHED" jsonname="is\_curcuit\_switched"/>   
 <Field fid="FID\_AGENT\_FIRST\_NAME" jsonname="agent\_first\_name"/>   
 <Field fid="FID\_AGENT\_LAST\_NAME" jsonname="agent\_last\_name"/>   
 <Field fid="FID\_COUNTRYCODE" jsonname="agent\_country\_code"/>   
 <Field fid="FID\_AREACODE" jsonname="agent\_country\_code"/>   
 <Field fid="FID\_RECORDING" jsonname="recording\_enabled"/>   
 <Field fid="FID\_SELECTIVE\_RECORDING" jsonname="selective\_enabled"/>   
 <Field fid="FID\_SHOW\_RECORDING" jsonname="show\_recording"/>   
 <Field fid="FID\_SCREEN\_RECORDING" jsonname="screen\_recording"/>   
 <Field fid="FID\_TRANSFER\_RECORDING" jsonname="transfer\_recording"/>   
 <Field fid="IPC\_SOURCE\_ADDRESS" jsonname="ipc\_source\_address" />   
 <Field fid="IPC\_DESTINATION\_ADDRESS" jsonname="ipc\_dst\_address" />   
 </IPCMessage>   
</CISEvents>   
   
The file structure is as follows:   
•   
The xml file starts with “<CISEvents>“and ends with“ </CISEvents>“.   
•   
Every IPC message starts with “<IPCMessage>“and ends with “</IPCMessage>“.   
•   
jsonName is the name of the fid that will be presented by the connector.   
•   
The fid should be the same as the fid in the message: capital letters and underscores.

CCSP APIs Guide - 7.4   
66   
Add events   
Caution   
If CISEvents.xml is not configured correctly, CIS will not start.   
   
When adding events:   
•   
Every IPC message must start with “</IPCMessage>“and end with “</IPCMessage>“.   
•   
Every fid and message type must have jsonName.   
•   
Copy the message type and relevant fids from the message.   
•   
The message type must start with “<IPCMessage type=“and end with “>“.   
For example agent ring:   
<IPCMessage type="AGENT\_RING" jsonname="agent\_ring">   
•   
The Field fid must start with “ <Field fid= “ and end with “ />“.   
For example fid IPC\_DESTINATION\_ADDRESS:   
<Field fid="IPC\_DESTINATION\_ADDRESS" jsonname="ipc\_dst\_address" />   
   
The following example shows supporting messages that you can see in the login acceptance.   
   
STATUS =200   
----   
{   
 "session\_id": "LXU9GFL2",   
 "message": "http\_login\_request",   
 "result": "success",   
 "login\_time": "2015-08-17T09:17:32.2390972Z",   
 "tenant\_id": 0,   
 "agent\_global\_id": 2283010091194515456,   
 "assigned\_tenants": [   
 -1   
 ],   
 "event\_providers": [   
 "VCS,26",   
 "DPS,27"   
 ],   
 "supported\_messages": {   
 "4171": "CIS\_SNAPSHOT\_INFO",   
 "4173": "CIS\_GET\_EVENT\_INFO",   
 "164": "CS\_CALL\_OFFERED",   
 "47": "CS\_CALLOUT",

CCSP APIs Guide - 7.4   
67   
 "48": "CS\_ACK\_CALLOUT",   
 "49": "CS\_NACK\_CALLOUT",   
 "3": "CS\_CALLARRIVAL",   
 "5": "CS\_SCRIPTCOMPLETED",   
 "4": "CS\_ACKCALLARRIVAL",   
 "28": "AGENT\_RING",   
 "8": "CS\_PICKUPCALL",   
 "120": "CS\_AGENTCONNECTED",   
 "37": "CS\_NACKPICKUPCALL",   
 "9": "CS\_ACKCALLPICKUP",   
 "327": "CS\_LOG\_MediaData\_VCS",   
 "325": "CS\_CALL\_TERMINATED",   
 "721": "DPS\_AGENT\_MEDIA\_CONNECT",   
 "43": "CS\_HANGUP",   
 "16": "AGENT\_LOGON",   
 "17": "AGENT\_LOGOFF",   
 "20": "AGENT\_UNAVAILABLE",   
 "21": "AGENT\_AVAILABLE",   
 "106": "AGENT\_STATE\_CHANGE",   
 "377": "CS\_CONFERENCE\_CALLS",   
 "616": "CS\_CONFERENCE\_CALLID\_INFO"   
 },   
 "https\_enabled": false,   
 "http\_ip": "10.120.31.136",   
 "http\_port": 8088,   
 "https\_ip": "10.120.31.136",   
 "https\_port": 8443,   
 "web\_socket\_ip": "10.120.31.136",   
 "web\_socket\_port": 8089,   
 "web\_socket\_secure\_ip": "10.120.31.136",   
 "web\_socket\_secure\_port": 8431   
}

CCSP APIs Guide - 7.4   
68   
8: CCSP Realtime API   
This chapter describes the XML interfaces used to integrate with real-time reports.   
The Realtime API consists of filters used to extract the XML documents from the RTR ISAPI DLL   
(rtrweb.dll). These filters are:   
•   
ACalls − agent statistics and states data   
•   
ACallsDetails − agent statistics and calls data   
•   
QCalls − queue statistics data   
•   
AConfigs − agent configuration   
•   
CConfigs − campaign configuration   
•   
GConfigs − group configuration   
•   
QConfigs − queue configuration   
•   
QHistory − queue statistics historical data (previous 24 hours)   
•   
SServer − Statistics server snapshot data (used for debugging)   
   
Queue and agent filtering filters queue and agent items in accordance with a defined list on the   
RTR server side. The filter list is per RTR Dll (server) instance.   
The Permission Profile filters returned results in accordance with the list of allowed items.   
To extract XML documents for some or all of the filters, with the exception of SServer filter,   
separate the filters with the plus (+) sign. For example, to get data for QConfigs and AConfigs filters,   
use:   
http://localhost/rtrdll/rtrweb.dll?tenant=tnt0&Filter=qconfigs+aconfigs   
Note   
The filter and query attributes are not case sensitive.   
   
From CCSP 7.3, the query interface can include a list of agents, groups, teams, queues and media   
types that should be included in the response. Each attribute can specify a comma separated list of   
IDs.   
The additional query parameters are:   
•   
AgentId   
•   
GroupId   
•   
TeamId   
•   
QueueId   
•   
MediaType   
   
For the AConfigs, ACalls and ACallsDetails filters, you can use the three selection attributes that   
relate to agents, which are AgentId, GroupId and TeamId.

CCSP APIs Guide - 7.4   
69   
The following example will return a list of agents in groups 1 and 2, and also agents from team 3   
(even if they are not in group 1 or 2).   
http://localhost/rtrdll/rtrweb.dll?Filter=ACalls&GroupId=1,2&TeamId=3   
For the GConfigs filter, you can use the GroupId attribute.   
For the QConfigs filter, you can use the QueueId attribute.   
For the QCalls and QHistory filters, you can use the two selection attributes, QueueId and   
MediaType.   
The QCalls filter limits the output of the queues requested, and can also limit the output of the   
breakdown of media types.   
The media types are:   
•   
0 = Chat   
•   
2 = Email   
•   
3 = Phone   
•   
4 = Callback   
•   
5 = Voicemail   
   
To excude all breakdown of media type, use the value 99.   
The following example shows how to limit the response to queues 10 and 11, and also limit the   
breakdown of media types to just email (type 2) and phone (type 3):   
http://localhost/rtrdll/rtrweb.dll?Filter=QCalls&QueueId=10,11&MediaType=2,3   
   
The following example shows how to excude all breakdown of media type:   
http://localhost/rtrdll/rtrweb.dll?Filter=QHistory&MediaType=99

CCSP APIs Guide - 7.4   
70   
ACalls   
Note   
•   
The average talk time attributes (inatt, outatt, a2att, and tatt) do not include hold time.   
•   
The average duration attributes (ad, adchat, admail, advoip, advm, and adcb) include all the   
states of the call (preview, ringing, active, hold, and wrap).   
•   
ACallsDetails is a copy of ACalls with additional elements, and some changes. We recommend   
that you move to ACallsDetails because ACalls will be deprecated.   
   
The ACalls filter shows agent state and calls state handled by the agent filtered in accordance with   
the profile. If the supervisor is a tenant-wide supervisor or a landlord, the ACalls filter returns all the   
agents configured for the specified tenant.   
Usage   
 http://<ServerName>/realtimereports/rtrweb.dll? Tenant=<TenantName>   
&Filter=ACalls[DEV:&Username=<UserName>]   
   
Where:   
•   
<ServerName> is the IP address or domain name of the server hosting the RTR web site.   
•   
<TenantName> is the name of the tenant that you want to get statistics for. Only for landlord   
users.   
•   
<UserName> is the Windows user name of an agent supervisor or CCSP administrator. This   
parameter is used only if IIS is configured for anonymous access. Otherwise, RTR uses the user   
that authenticated with IIS.   
Example   
http://localhost/rtrdll/rtrweb.dll?Tenant=cosmocom&Filter=ACalls&Username=ekeselman   
   
<ACalls ts="Thu Jan 24 10:57:30 2019 UTC" cts="Thu Jan 24 10:57:27 2019 UTC"   
tzo="-3600" al="false">   
 <A id="11" tp="28" ldc="28" tl="88764" as="10" cc="2" all="2" aics="6,2;"   
 cid="0" at="88053" gid="6" dir="9" cct="6" cq="[Multiple]" inatt="35"   
 outatt="0" a2aatt="0" tatt="35" ad="49" adchat="0" admail="0"   
 advoip="49" advm="0" adcb="0"/>   
 <A id="3011" tp="158454" ctp="158434" ldc="158454" tl="179114" as="10"   
 cc="1" all="100" aics="5,1;" cid="0" at="179102" gid="6" dir="1"   
 cct="3" cq="DefaultQueue" inatt="0" outatt="0" a2aatt="0" tatt="0"   
 ad="0" adchat="0" admail="0" advoip="0" advm="0" adcb="0"/>   
</ACalls>

CCSP APIs Guide - 7.4   
71   
Elements and attributes   
The filter contains the following elements and attributes.   
Name   
Type   
Description   
ACalls   
Element   
This is the top level element and exists once in this document.   
al   
Attribute   
Agent limitation attribute that indicates whether the RTR view is   
filtered or not. False means that the logged in agent/supervisor is   
tenant-wide or landlord and does not have a profile. True means   
the user has a profile.   
cts   
Attribute   
Timestamp of the last configuration update received by the RTR   
web server, in UTC (Universal Time Coordinated)/GMT (Greenwich   
Mean Time).   
ts   
Attribute   
RTR web server time stamp in UTC (Universal Time   
Coordinated)/GMT (Greenwich Mean Time).   
tzo   
Attribute   
Timezone offset of the RTR web server. This is the number of   
seconds that the RTR web server differs from UTC/GMT.   
A   
Element   
There is one 'A' element for each agent that has had active   
calls this hour.   
a2aatt   
Attribute   
Agent-to-agent average talk time.   
ad   
Attribute   
Average duration of all media types.   
adchat   
Attribute   
Average duration of chat calls.   
admail   
Attribute   
Average duration of email calls.   
advoip   
Attribute   
Average duration of VoIP calls.   
advm   
Attribute   
Average duration of acd voicemail calls.   
adcb   
Attribute   
Average duration of callbacks.   
aics   
Attribute   
Agent in call state. List showing each state and number of calls in   
this state. Possible values are:   
5 – Wrapup   
6 − InCall   
7 − Ringing   
8 − Preview   
9 − OnHold   
all   
Attribute   
Agent load level. The sum of the weights of all the calls the agent   
is handling.

CCSP APIs Guide - 7.4   
72   
Name   
Type   
Description   
as   
Attribute   
Agent state. Possible values are:   
3 − Available   
4 − Unavailable   
10 – Available in call   
11 – Unavailable in call   
at   
Attribute   
Available time. How long the agent has been in the Available state.   
Includes Available and Available in call.   
cc   
Attribute   
Call count. The number of calls the agent is currently handling.   
cct   
Attribute   
Current call type. This value is actually the media stream type.   
Possible values are:   
0 − Chat   
2 − Email   
3 – VoIP   
4 − Callback   
5 – Voicemail   
6 – Multiple calls   
cid   
Attribute   
The dialer campaign ID of the agent (Campaign\_ID in the   
database). If not logged into a campaign, this value is 0.   
cq   
Attribute   
Current queue. If handling calls from different queues, [Multiple]   
appears.   
ctp   
Attribute   
Call time published.   
If call is active, ctp is the number of seconds since the call first   
became active.   
If call is ringing, on hold, or in wrapup, ctp is the number of   
seconds in that call state.   
Note   
This attribute exists only if the agent is handling a single call.

CCSP APIs Guide - 7.4   
73   
Name   
Type   
Description   
dir   
Attribute   
Call direction. Possible values are:   
0 − Out   
1 − In   
2 − A2A   
3 − Bridged   
4 − CallbackRequest   
5 − CallbackResponse   
6 − CallOutTypePDS   
7 − CallOutTypePPDS   
8 – CallOutTypeElsbeth   
9 – Multiple calls   
gid   
Attribute   
Group ID (Group\_ID in the database).   
id   
Attribute   
Agent ID (Person\_ID in the database).   
inatt   
Attribute   
Inbound average talk time.   
outatt   
Attribute   
Outbound average talk time.   
rc   
Attribute   
Release code used for agent released state. For example, if the   
agent selects "Coffee" as their released state, the release code   
"Coffee" is listed in parentheses next to Released as "Released   
(Coffee)."   
Note   
This attribute only appears when the agent is in the released state   
and has chosen a release code.   
tatt   
Attribute   
Total average talk time.   
tl   
Attribute   
Time logged in. The number of seconds that the agent has been   
logged in.   
tp   
Attribute   
Time published. The number of seconds since the agent state   
changed. This is the time in the agent state (as attribute).   
trc   
Attribute   
Time in release code.   
Note   
This attribute only appears when the agent is in the released state   
and has chosen a release code.

CCSP APIs Guide - 7.4   
74   
ACallsDetails   
Note   
•   
The average talk time attributes (inatt, outatt, a2att, and tatt) do not include hold time.   
•   
The average duration attributes (ad, adchat, admail, advoip, advm, and adcb) include all the   
states of the call (preview, ringing, active, hold, and wrap).   
•   
ACallsDetails is a copy of ACalls with additional elements, and some changes. We recommend   
that you move to ACallsDetails because ACalls will be deprecated.   
   
The ACallsDetails filter shows agent state and calls state handled by the agent filtered in   
accordance with the profile. If the supervisor is a tenant-wide supervisor or a landlord, the ACalls   
filter returns all the agents configured for the specified tenant.   
Usage   
 http://<ServerName>/realtimereports/rtrweb.dll? Tenant=<TenantName>   
&Filter=ACallsDetails[DEV:&Username=<UserName>]   
   
Where:   
•   
<ServerName> is the IP address or domain name of the server hosting the RTR web site.   
•   
<TenantName> is the name of the tenant that you want to get statistics for. Only for landlord   
users.   
•   
<UserName> is the Windows user name of an agent supervisor or CCSP administrator. This   
parameter is used only if IIS is configured for anonymous access. Otherwise, RTR uses the user   
that authenticated with IIS.

CCSP APIs Guide - 7.4   
75   
Example   
http://localhost/rtrdll/rtrweb.dll?Tenant=cosmocom&Filter=ACallsDetails&Username=ekeselman   
   
<ACallsDetails ts="Thu Jan 24 10:58:36 2019 UTC" cts="Thu Jan 24 10:57:27 2019   
UTC" tzo="-3600" al="false">   
 <A id="11" tp="94" ldc="94" tl="88830" as="10" cc="2" all="2" cid="0"   
 at="88119" gid="6" inatt="35" outatt="0" a2aatt="0" tatt="35" ad="49"   
 adchat="0" admail="0" advoip="49" advm="0" adcb="0">   
 <C ctp="95" ct="1" mt="0" cq="DefaultQueue" cs="6" tcs="95"   
 cid="2758136424493613063"/>   
 <C ctp="88" ct="1" mt="3" cq="DefaultQueue" cs="6" tcs="88"   
 cid="2758136441674661901"/>   
 </A>   
 <A id="3011" tp="158520" ldc="158520" tl="179180" as="10" cc="1" all="100"   
 cid="0" at="179168" gid="6" inatt="0" outatt="0" a2aatt="0" tatt="0"   
 ad="0" adchat="0" admail="0" advoip="0" advm="0" adcb="0">   
 <C ctp="158522" ct="1" mt="3" cq="DefaultQueue" cs="5" tcs="158501"   
 cid="2758136441673482253"/>   
 </A>   
</ACallsDetails>

CCSP APIs Guide - 7.4   
76   
Elements and attributes   
The filter contains the following elements and attributes.   
Name   
Type   
Description   
ACallsDetails   
Element   
This is the top level element and exists once in this   
document.   
al   
Attribute   
Agent limitation attribute that indicates whether the RTR view   
is filtered or not. False means that the logged in   
agent/supervisor is tenant-wide or landlord and does not   
have a profile. True means the user has a profile.   
cts   
Attribute   
Timestamp of the last configuration update received by the   
RTR web server, in UTC (Universal Time Coordinated)/GMT   
(Greenwich Mean Time).   
ts   
Attribute   
RTR web server time stamp in UTC (Universal Time   
Coordinated)/GMT (Greenwich Mean Time).   
tzo   
Attribute   
Timezone offset of the RTR web server. This is the number of   
seconds that the RTR web server differs from UTC/GMT.   
A   
Element   
There is one 'A' element for each agent that has had   
active calls this hour.   
a2aatt   
Attribute   
Agent-to-agent average talk time.   
ad   
Attribute   
Average duration of all media types.   
adchat   
Attribute   
Average duration of chat calls.   
admail   
Attribute   
Average duration of email calls.   
advoip   
Attribute   
Average duration of VoIP calls.   
advm   
Attribute   
Average duration of acd voicemail calls.   
adcb   
Attribute   
Average duration of callbacks.   
all   
Attribute   
Agent load level. The sum of the weights of all the calls the   
agent is handling.   
as   
Attribute   
Agent state. Possible values are:   
3 − Available   
4 − Unavailable   
10 – Available in call   
11 – Unavailable in call

CCSP APIs Guide - 7.4   
77   
Name   
Type   
Description   
at   
Attribute   
Available time. How long the agent has been in the Available   
state. Includes Available and Available in call.   
cc   
Attribute   
Call count. The number of calls the agent is currently   
handling.   
gid   
Attribute   
Group ID (Group\_ID in the database).   
inatt   
Attribute   
Inbound average talk time.   
outatt   
Attribute   
Outbound average talk time.   
rc   
Attribute   
Release code used for agent released state. For example, if   
the agent selects "Coffee" as their released state, the release   
code "Coffee" is listed in parentheses next to Released as   
"Released (Coffee)."   
Note   
This attribute only appears when the agent is in the released   
state and has chosen a release code.   
tatt   
Attribute   
Total average talk time.   
tl   
Attribute   
Time logged in. The number of seconds that the agent has   
been logged in.   
tp   
Attribute   
Time published. The number of seconds since the agent state   
changed. This is the time in the agent state (as attribute).   
trc   
Attribute   
Time in release code.   
Note   
This attribute only appears when the agent is in the released   
state and has chosen a release code.   
C   
Element   
There is one 'C' element for each call that the agent has.   
cq   
Attribute   
Call queue name.

CCSP APIs Guide - 7.4   
78   
Name   
Type   
Description   
ct   
Attribute   
Call type. Possible values are:   
0 − Out   
1 − In   
2 − A2A   
3 − Bridged   
4 − CallbackRequest   
5 − CallbackResponse   
6 − CallOutTypePDS   
7 − CallOutTypePPDS   
8 − CallOutTypeElsbeth   
ctp   
Attribute   
Call time published.   
If call is active, ctp is the number of seconds since the call first   
became active.   
If call is ringing, on hold, or in wrapup, ctp is the number of   
seconds in that call state.   
mt   
Attribute   
Media type. Possible values are:   
0 − Chat   
2 − Email   
3 – VoIP   
4 – Callback   
5 – Voicemail   
cs   
   
Call state. Possible values are:   
5 − Wrapup   
6 − Active   
7 − Ringing   
8 − Preview   
9 − OnHold   
cid   
Attribute   
Call ID (Call\_ID in the database).   
tcs   
Attribute   
Time in call duration. The number of seconds since the call   
was first answered. Includes wrapup and onhold time.

CCSP APIs Guide - 7.4   
79   
QCalls   
Note   
After StatisticServer failover, only the following queue counters are recovered:   
•   
Being Handled   
•   
Handled (only for calls currently in the system)   
•   
Offered (the recovered counter includes the Overflowed In counter values before the   
failover)   
•   
WaitingInQueue   
The counters are recovered only for the calls currently in the system that are not in their   
termination phase. All recovered counters are placed in the current hour slot. Counter recovery   
covers all the connection servers for all tenants.   
   
The QCalls filter displays queue statistics for the current hour in accordance with the profile. At the   
start of each hour, these counters are reset. If the supervisor is a tenant-wide supervisor or a   
landlord, the QCalls filter returns queue statistics for all the queues configured for the specified   
tenant.   
The only limitation is that the queue must have at least one call received in its lifetime otherwise no   
entry will be returned.   
Usage   
http://<ServerName>/realtimereports/rtrweb.dll? Tenant=<TenantName>   
&Filter=QCalls[DEV:&Username=<UserName>]   
Where:   
•   
<ServerName> is the IP address or domain name of the server hosting the RTR web site.   
•   
<TenantName> is the name of the tenant that you want to get statistics for. Only for landlord   
users.   
•   
<UserName> is the Windows user name of an agent supervisor or CCSP administrator. This   
parameter is used only if IIS is configured for anonymous access. Otherwise, RTR uses the user   
that authenticated with IIS.

CCSP APIs Guide - 7.4   
80   
Example   
 http://localhost/rtrdll/rtrweb.dll?Tenant=cosmocom&Filter=QCalls&Username=ekeselman   
<?xml version="1.0" encoding="UTF-8" ?>   
<QCalls ts="Wed Jul 10 12:33:30 2019 UTC" cts="Wed Jul 10 11:34:28 2019 UTC"   
 tzo="-10800" al="false">   
 <Q id="2" n="q1" wt="0" ch="2" cwt="920" co="2" cw="0" awt="0" act="5"   
 cbh="0" pfd="1" pfm="1" def="1" cwc="1" tct="5" twt="920">   
 <internet id="2" n="q1" />   
 <message id="2" n="q1" wt="0" ch="2" cwt="920" co="2" cw="0" awt="0"   
 act="5" cbh="0" pfd="1" pfm="1" def="1" cwc="1" tct="5" twt="920" />   
 <phone id="2" n="q1" />   
 <callback id="2" n="q1" />   
 <voicemail id="2" n="q1" />   
 </Q>   
</QCalls>   
Elements and attributes   
The filter contains the following elements and attributes.   
Name   
Type   
Description   
QCalls   
Element   
This is the top level element and exists one time in this   
document.   
al   
Attribute   
Agent limitation attribute that indicates whether the RTR view is   
filtered or not. False means that the logged in agent/supervisor is   
tenant-wide or landlord and does not have a profile. True means   
the user has a profile.   
cts   
Attribute   
Timestamp of the last configuration update received by the RTR   
web server, in UTC (Universal Time Coordinated)/GMT (Greenwich   
Mean Time).   
ts   
Attribute   
RTR web server time stamp in UTC (Universal Time   
Coordinated)/GMT (Greenwich Mean Time).   
tzo   
Attribute   
Timezone offset of the RTR web server. This is the number of   
seconds that the RTR web server differs from UTC/GMT.   
Q   
Element   
There is one 'Q' element for each queue that has had active   
calls this hour.

CCSP APIs Guide - 7.4   
81   
Name   
Type   
Description   
act   
Attribute   
Note   
Average call time is now calculated in the CCSP Dashboard client.   
This legacy attribute is inaccurate.   
awt   
Attribute   
Note   
Average wait time is now calculated in the CCSP Dashboard client.   
This legacy attribute is inaccurate.   
avlt   
Attribute   
Cumulative available time of the agents assigned to the campaign   
associated with this queue.   
ca   
Attribute   
Number of calls abandoned.   
catqos   
Attribute   
Calls abandoned within the target quality of service (TQOS).   
ch   
Attribute   
Number of calls handled. This does not include calls being   
handled - not completed.   
cbh   
Attribute   
Number of calls currently being handled (in call) by an agent.   
co   
Attribute   
Number of calls offered.   
cr   
Attribute   
Calls rejected.   
ct   
Attribute   
Number of calls handled and being handled within the TQOS   
threshold.   
cw   
Attribute   
Number of calls currently waiting in queue.   
cwc   
Attribute   
Cumulative waiting calls. The number of calls that waited in the   
queue to be handled.   
cwt   
Attribute   
Cumulative wait time, in seconds. This can be used to calculate the   
average wait time by dividing by calls handled ("ch").   
def   
Attribute   
Counts number of deferred emails within an hour.   
disp   
Attribute   
Number of calls delivered to the agents.   
dlro   
Attribute   
Number of calls terminated with dialer outcome code.   
id   
Attribute   
Queue ID (Queue\_ID in the database).   
n   
Attribute   
Queue name.   
ofi   
Attribute   
Number of calls overflowed into this queue.   
ofo   
Attribute   
Number of calls overflowed out of this queue.

CCSP APIs Guide - 7.4   
82   
Name   
Type   
Description   
rnt   
Attribute   
Campaign run time.   
pfd   
Attribute   
Counts number of items pulled from defer.   
pfm   
Attribute   
Counts number of queued items pulled from public.   
pfu   
Attribute   
Counts number of unqueued items pulled from public.   
stt   
Attribute   
Campaign start time.   
tct   
Attribute   
Total call time. The sum of talk time in the queue.   
tlkt   
Attribute   
Cumulative talk time of the agents assigned to the campaign   
associated with this queue.   
twt   
Attribute   
Total wait time, in seconds. The sum of wait time for all calls in the   
queue that waited to be handled.   
Note   
Average wait time is calculated as twt/cwc.   
wrpt   
Attribute   
Cumulative wrap-up time of the agents assigned to the campaign   
associated with this queue.   
wt   
Attribute   
Longest wait time, in seconds.   
D   
Element   
There is one 'D' element for each disposition code reported by   
an agent.   
cnt   
Attribute   
Number of calls with this disposition code.   
id   
Attribute   
Disposition code ID (DispositionCode\_ID in the database).   
tlkt   
Attribute   
Cumulative talk time for the calls with this disposition code.   
O   
Element   
There is one 'O' element for each outcome code reported by   
an agent.   
cnt   
Attribute   
Number of calls with this dialer outcome.   
id   
Attribute   
Dialer outcome code ID.

CCSP APIs Guide - 7.4   
83   
AConfigs   
The AConfigs filter displays a list of agents in accordance with the profile. If the supervisor is a   
tenant-wide supervisor or a landlord, the AConfigs filter returns all the agents configured for the   
specified tenant.   
Usage   
http://<ServerName>/realtimereports/rtrweb.dll? Tenant=<TenantName>   
&Filter=AConfigs[DEV:&Username=<UserName>]   
   
Where:   
•   
<ServerName> is the IP address or domain name of the server hosting the RTR web site   
•   
<TenantName> is the name of the tenant that you want to get stats for. Only for landlord   
users.   
•   
<UserName> is the Windows user name of an agent supervisor or CCSP administrator. This   
parameter is used only if IIS is configured for anonymous access. Otherwise, RTR uses the user   
that authenticated with IIS.

CCSP APIs Guide - 7.4   
84   
Example   
http://localhost/rtrdll/rtrweb.dll?Tenant=cosmocom&Filter=AConfigs&Username=ekeselman   
   
<AConfigs ts="Thu Jan 24 10:05:33 2019 UTC" tzo="-3600" al="false"   
smtpserver="localhost" loginid="13">   
 <A id="10" fn="admin" ln="admin" desc="" un="admin" sk="" tm=""   
ipn="Default Profile" bic="False" bldc="False" blc="False" bc="" isag="True"   
isad="True" issu="True">   
 <G id="-1"/>   
 </A>   
 <A id="11" fn="Agent" ln="One" desc="" un="agent1" sk="DefaultSkill"   
tm="TeamA" ipn="Default Profile" bic="False" bldc="False" blc="False" bc=""   
isag="True" isad="False" issu="False">   
 <G id="6" n="DefaultGroup"/>   
 </A>   
 <A id="12" fn="Agent" ln="Two" desc="" un="agent2" sk="DefaultSkill"   
tm="TeamB" ipn="Default Profile" bic="False" bldc="False" blc="False" bc=""   
isag="True" isad="False" issu="False">   
 <G id="2011" n="Group2"/>   
 </A>   
 <A id="13" fn="Supervisor" ln="One" desc="" un="super1" sk="DefaultSkill"   
tm="TeamA" ipn="Default Profile" bic="False" bldc="False" blc="False" bc=""   
isag="True" isad="False" issu="True">   
 <G id="6" n="DefaultGroup"/>   
 </A>   
</AConfigs>

CCSP APIs Guide - 7.4   
85   
Elements and attributes   
The filter contains the following elements and attributes.   
Name   
Type   
Description   
AConfigs   
Element   
This is the top level element and exists once in this   
document.   
al   
Attribute   
Agent limitation attribute indicates whether the RTR view is   
filtered or not. False means that the logged in agent/supervisor is   
tenant-wide or landlord and does not have a profile. True means   
the user has a profile.   
loginid   
Attribute   
User ID of the person accessing the RTR web service.   
smtpserver   
Attribute   
CosmoDashboard SMTP Server parameter defined in the Tenant   
Properties.   
ts   
Attribute   
RTR web server time stamp in UTC (Universal Time   
Coordinated)/GMT (Greenwich Mean Time).   
tzo   
Attribute   
Timezone offset of the RTR web server. This is the number of   
seconds that the RTR web server differs from UTC/GMT.   
A   
Element   
There is one 'A' element for each agent defined for this   
tenant.   
bc   
Attribute   
Agent's block codes. Multiple block codes are separated by   
commas.   
bic   
Attribute   
Block international calls flag. If it is set to '1', the agent is blocked   
from making international calls.   
blc   
Attribute   
Block local calls flag. If it is set to '1', the agent is blocked from   
making local calls.   
bldc   
Attribute   
Block long distance calls flag. If it is set to '1', the agent is blocked   
from making long distance calls.   
desc   
Attribute   
Agent's description.   
fn   
Attribute   
Agent's first name.   
id   
Attribute   
Agent ID (Person\_ID in the database).   
ipn   
Attribute   
Interaction profile name.   
isad   
Attribute   
CCSP administrator flag. If set to '1' then this agent is a CCSP   
administrator.

CCSP APIs Guide - 7.4   
86   
Name   
Type   
Description   
isag   
Attribute   
CCSP agent flag. If set to '1' then this agent is a CCSP agent.   
issu   
Attribute   
CCSP supervisor flag. If set to '1' then this agent is a CCSP   
supervisor.   
ln   
Attribute   
Agent's last name.   
sk   
Attribute   
Agent's skills. Multiple skills are separated by commas.   
tm   
Attribute   
Agent's teams. Multiple teams are separated by commas.   
un   
Attribute   
Agent's Windows username.   
G   
Element   
The group that this agent is assigned to.   
id   
Attribute   
Group ID (Group\_ID in the database).   
n   
Attribute   
Group name.

CCSP APIs Guide - 7.4   
87   
CConfigs   
The CConfigs filter displays a list of campaigns in accordance with the profile. If the supervisor is a   
tenant-wide supervisor or a landlord, the CConfigs filter returns all the campaigns configured for   
the specified tenant.   
Usage   
http://<ServerName>/realtimereports/rtrweb.dll?Tenant=<TenantName>   
&Filter=CConfigs[DEV:&Username=<UserName>]   
   
Where:   
•   
<ServerName> is the IP address or domain name of the server hosting the RTR web site.   
•   
<TenantName> is the name of the tenant that you want to get statistics for. Only for landlord   
users.   
•   
<UserName> is the Windows user name of an agent supervisor or CCSP administrator. This   
parameter is used only if IIS is configured for anonymous access. Otherwise, RTR uses the user   
that authenticated with IIS.   
Example   
http://localhost/rtrdll/rtrweb.dll?Tenant=cosmocom&Filter=CConfigs&Username=ekeselman   
   
<?xml version="1.0" encoding="UTF-8" ?>   
<CConfigs ts="Wed Jul 31 13:51:25 2013 UTC" tzo="14400" al="false">   
   
<C id="1" n="Campaign1" desc="">   
   
   
<Q id="2" n="Queue1" />   
   
   
<G id="1" n="DefaultGroup" />   
   
</C>   
   
<C id="2" n="Campaign2" desc="">   
   
   
<Q id="1" n="DefaultQueue" />   
   
   
<G id="2" n="Group1" />   
   
</C>   
   
<C id="3" n="CampaignDefault" desc="">   
   
   
<Q id="1" n="DefaultQueue" />   
   
   
<G id="2" n="Group1" />   
   
</C>   
</CConfigs>

CCSP APIs Guide - 7.4   
88   
Elements and attributes   
The filter contains the following elements and attributes.   
Name   
Type   
Description   
CConfigs   
Element   
This is the top level element and exists one time in this   
document.   
al   
Attribute   
Agent limitation attribute indicates whether the RTR view is   
filtered or not. False means that the logged in agent/supervisor is   
tenant-wide or landlord and does not have a profile. True means   
the user has a profile.   
ts   
Attribute   
RTR web server time stamp in UTC (Universal Time   
Coordinated)/GMT (Greenwich Mean Time).   
tzo   
Attribute   
Timezone offset of the RTR web server. This is the number of   
seconds that the RTR web server differs from UTC/GMT.   
C   
Element   
There is one 'C' element for each campaign defined for this   
tenant.   
id   
Attribute   
Campaign ID (Campaign\_ID in the database).   
n   
Attribute   
Campaign name.   
desc   
Attribute   
Campaign description.   
G   
Element   
There is one 'G' element for the group assigned to this   
campaign.   
id   
Attribute   
Group ID (Group\_ID in the database).   
n   
Attribute   
Group name.   
Q   
Element   
There is one 'Q' element for the queue assigned to this   
campaign.   
id   
Attribute   
Queue ID (Queue\_ID in the database).   
n   
Attribute   
Queue name.

CCSP APIs Guide - 7.4   
89   
GConfigs   
The GConfigs filter displays a list of agent groups in accordance with the profile. If the supervisor is   
a tenant-wide supervisor or landlord, the GConfigs filter returns all the groups configured for the   
specified tenant.   
Usage   
http://<ServerName>/realtimereports/rtrweb.dll? Tenant=<TenantName>   
&Filter=GConfigs[DEV:&Username=<UserName>]   
   
Where:   
•   
<ServerName> is the IP address or domain name of the server hosting the RTR web site.   
•   
<TenantName> is the name of the tenant that you want to get statistics for. Only for landlord   
users.   
•   
<UserName> is the Windows user name of an agent supervisor or CCSP administrator. This   
parameter is used only if IIS is configured for anonymous access. Otherwise, RTR employs the   
user that authenticated with IIS.   
Example   
 http://localhost/rtrdll/rtrweb.dll?Tenant=cosmocom&Filter=GConfigs&Username=ekeselman   
   
<?xml version="1.0" encoding="UTF-8" ?>   
<GConfigs ts="Fri Jun 28 19:23:57 2013 UTC" tzo="14400" al="false">   
   
<G id="1" n="DefaultGroup">   
   
   
<Q id="2" n="Queue1" pq="1" />   
   
   
<Q id="5" n="Queue4" pq="1" />   
   
   
<A id="1" fn="S" ln="1" />   
   
   
<A id="2" fn="S" ln="2" />   
   
</G>   
   
<G id="2" n="Group1">   
   
   
<Q id="1" n="DefaultQueue" pq="1" />   
   
   
<A id="4" fn="S" ln="4" />   
   
   
<A id="5" fn="Eugene" ln="Keselman" />   
   
</G>   
   
<G id="3" n="Group2">   
   
   
<Q id="3" n="Queue2" pq="1" />   
   
</G>   
</GConfigs>

CCSP APIs Guide - 7.4   
90   
Elements and attributes   
The filter contains the following elements and attributes.   
Name   
Type   
Description   
GConfigs   
Element   
This is the top level element and exists once in this   
document.   
al   
Attribute   
Agent limitation attribute indicates whether the RTR view is   
filtered or not. False means that the logged in agent/supervisor is   
tenant-wide or landlord and does not have a profile. True means   
the user has a profile.   
ts   
Attribute   
RTR web server time stamp in UTC (Universal Time   
Coordinated)/GMT (Greenwich Mean Time).   
tzo   
Attribute   
Timezone offset of the RTR web server. This is the number of   
seconds that the RTR web server differs from UTC/GMT.   
G   
Element   
There is one 'G' element for each CCSP group defined for this   
tenant.   
id   
Attribute   
Group ID (Group\_ID in the database).   
n   
Attribute   
Group name.   
Q   
Element   
There is one 'Q' element for each queue that belongs to this   
group.   
id   
Attribute   
Queue ID (Queue\_ID in the database).   
n   
Attribute   
Queue name.   
pq   
Attribute   
Primary queue flag. This is set to '1' if this queue is a primary   
queue in this group. It is set to '0' if it is a secondary queue in this   
group.   
A   
Element   
There is one 'A' element for each agent that belongs to this   
group.   
fn   
Attribute   
Agent's first name.   
id   
Attribute   
Agent ID (Agent\_ID in the database).

CCSP APIs Guide - 7.4   
91   
QConfigs   
The QConfigs filter displays a list of queues in accordance with the profile. If the supervisor is a   
tenant-wide supervisor or landlord, the QConfigs filter returns all the queues configured for the   
specified tenant.   
Usage   
http://<ServerName>/realtimereports/rtrweb.dll? Tenant=<TenantName>   
&Filter=QConfigs[DEV:&Username=<UserName>]   
   
Where:   
•   
<ServerName> is the IP address or domain name of the server hosting the RTR web site   
•   
<TenantName> is the name of the tenant that you want to get stats for. Only for landlord   
users.   
•   
<UserName> is the Windows user name of an agent supervisor or CCSP administrator. This   
parameter is only used if IIS is configured for anonymous access. Otherwise, RTR uses the user   
that authenticated with IIS.   
Example   
http://localhost/rtrdll/rtrweb.dll?Tenant=cosmocom&Filter=QConfigs&Username=ekeselman   
   
<?xml version="1.0" encoding="UTF-8" ?>   
<QConfigs ts="Fri Jun 28 19:34:40 2013 UTC" tzo="14400" al="false">   
   
<Q id="1" n="DefaultQueue" as="0" at="0" ap="120" ai="0">   
   
   
<G id="2" n="Group1" />   
   
</Q>   
   
<Q id="2" n="Queue1" as="0" at="0" ap="120" ai="0">   
   
   
<G id="1" n="DefaultGroup" />   
   
</Q>   
   
<Q id="3" n="Queue2" as="0" at="0" ap="120" ai="0">   
   
   
<G id="3" n="Group2" />   
   
</Q>   
   
<Q id="4" n="Queue3" as="0" at="0" ap="120" ai="0" />   
   
<Q id="5" n="Queue4" as="0" at="0" ap="120" ai="0">   
   
   
<G id="1" n="DefaultGroup" />   
   
</Q>   
   
<Q id="10" n="Queue9" as="0" at="0" ap="120" ai="0" />   
</QConfigs>

CCSP APIs Guide - 7.4   
92   
   
Elements and attributes   
The filter contains the following elements and attributes.   
Name   
Type   
Description   
QConfigs   
Element   
This is the top level element and exists once in this   
document.   
al   
Attribute   
Agent limitation attribute indicates whether the RTR view is   
filtered or not. False means that the logged in agent/supervisor is   
tenant-wide or landlord and does not have a profile. True means   
the user has a profile.   
ts   
Attribute   
RTR web server time stamp in UTC (Universal Time   
Coordinated)/GMT (Greenwich Mean Time).   
tzo   
Attribute   
Timezone offset of the RTR web server. This is the number of   
seconds that the RTR web server differs from UTC/GMT.   
Q   
Element   
There is one 'Q' element for each queue defined for this   
tenant.   
ai   
Attribute   
Assigned in queue timeout. The in queue time out assigned to   
this queue.   
ap   
Attribute   
Assigned pickup timeout. The pickup time out assigned to this   
queue.   
as   
Attribute   
Assigned service level. The service level assigned to this queue.   
at   
Attribute   
Assigned TQOS. The target quality of service assigned to this   
queue.   
aw   
Attribute   
Assigned wrap-up timeout in seconds.   
id   
Attribute   
Queue ID (Queue\_ID in the database).   
n   
Attribute   
Queue name.   
G   
Element   
The group that this queue is assigned to.   
id   
Attribute   
Group ID (Group\_ID in the database).   
n   
Attribute   
Group name.

CCSP APIs Guide - 7.4   
93   
QHistory   
The QHistory filter shows the queue statistics for the past 24 hours in accordance with the profile. If   
the supervisor is a tenant-wide supervisor or a landlord, the QHistory filter returns all the queues   
configured for the specified tenant.   
Each hourly attribute contains the number of calls for the particular counter from that hour until   
the current hour, not including the current hour. All hours are in UTC/GMT.   
This filter should be used in conjunction with the QCalls document to calculate statistics data from a   
particular hour. For example, the client is in GMT-5 timezone (Eastern USA) and you want to   
calculate the number of calls handled from 1 PM EST (13th hour EST, 18th hour UTC/GMT). You   
would take the value of the "t18" attribute from the "ch" element in QHistory and add it to the "ch"   
attribute of the QCalls document.   
Note   
QHIstory is reinitialized every time Statistics Server is restarted.   
Usage   
http://<ServerName>/realtimereports/rtrweb.dll?Tenant=<TenantName>   
&Filter=QHistory[DEV:&Username=<UserName>]   
   
Where:   
•   
<ServerName> is the IP address or domain name of the server hosting the RTR web site.   
•   
<TenantName> is the name of the tenant that you want to get statistics for. Only for landlord   
users.   
•   
<UserName> is the Windows user name of an agent supervisor or CCSP administrator. This   
parameter is used only if IIS is configured for anonymous access. Otherwise, RTR uses the user   
that authenticated with IIS.   
Example   
http://localhost/rtrdll/rtrweb.dll?Tenant=cosmocom&Filter=QHistory&Username=ekeselman   
...   
</message>   
<Q n="q1" id="2">   
 <ca t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0" t3="0"   
t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0" t17="0"   
t16="0" t15="0" t14="0" t13="0" />   
 <ch t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0" t3="0"   
t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0" t17="0"   
t16="0" t15="0" t14="0" t13="0" />

CCSP APIs Guide - 7.4   
94   
 <co t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0" t3="0"   
t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0" t17="0"   
t16="0" t15="0" t14="0" t13="0" />   
 <cwt t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <ct t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0" t3="0"   
t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0" t17="0"   
t16="0" t15="0" t14="0" t13="0" />   
 <ofi t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <ofo t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <catqos t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <cr t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0" t3="0"   
t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0" t17="0"   
t16="0" t15="0" t14="0" t13="0" />   
 <avlt t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <tlkt t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <wrpt t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <rnt t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <awt t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <act t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <pfd t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <pfu t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <pfm t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />

CCSP APIs Guide - 7.4   
95   
 <def t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <cwc t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <tct t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
 <twt t12="0" t11="0" t10="0" t9="0" t8="0" t7="0" t6="0" t5="0" t4="0"   
t3="0" t2="0" t1="0" t0="0" t23="0" t22="0" t21="0" t20="0" t19="0" t18="0"   
t17="0" t16="0" t15="0" t14="0" t13="0" />   
...

CCSP APIs Guide - 7.4   
96   
Elements and attributes   
The filter contains the following elements and attributes.   
Name   
Type   
Description   
QHistory   
Element   
This is the top level element and exists once in this document.   
ts   
Attribute   
RTR web server time stamp in UTC (Universal Time   
Coordinated)/GMT (Greenwich Mean Time).   
tzo   
Attribute   
Timezone offset of the RTR web server. This is the number of   
seconds that the RTR web server differs from UTC/GMT.   
Q   
Element   
There is one 'Q' element for each queue defined for this   
tenant.   
id   
Attribute   
Queue ID (Queue\_ID in the database).   
n   
Attribute   
Queue name.   
pfd   
Attribute   
Counts number of items pulled from defer.   
pfm   
Attribute   
Counts number of queued items pulled from public.   
pfu   
Attribute   
Counts number of unqueued items pulled from public.   
act   
Element   
Note   
Average call time is now calculated in the CCSP Dashboard client.   
This legacy element is inaccurate.   
avlt   
Element   
Cumulative available time of the agents assigned to the   
campaign associated with this queue. Each "tNN" attribute   
contains the count for that hour to the current hour, in UTC/GMT   
format   
awt   
Element   
Note   
Average wait time is now calculated in the CCSP Dashboard client.   
This legacy element is inaccurate.   
ca   
Element   
Calls abandoned. Each "tNN" attribute contains the count for   
that hour to the current hour, in UTC/GMT format.   
catqos   
Element   
Calls abandoned within the target quality of service (TQOS).   
Each "tNN" attribute contains the count for that hour to the   
current hour, in UTC/GMT format.   
ch   
Element   
Calls handled. Each "tNN" attribute contains the count for that   
hour to the current hour, in UTC/GMT format.

CCSP APIs Guide - 7.4   
97   
Name   
Type   
Description   
co   
Element   
Calls offered. Each "tNN" attribute contains the count for that   
hour to the current hour, in UTC/GMT format.   
cr   
Element   
Calls rejected. Each "tNN" attribute contains the count for that   
hour to the current hour, in UTC/GMT format   
cwc   
Element   
Cumulative waiting calls. The number of calls that waited in the   
queue to be handled.   
ct   
Element   
Calls handled within the target quality of service (TQOS)   
threshold. Each "tNN" attribute contains the count for that hour   
to the current hour, in UTC/GMT format.   
cwt   
Element   
Cumulative waiting time. Each "tNN" attribute contains the   
count for that hour to the current hour, in UTC/GMT format.   
rnt   
Element   
Campaign run time. Each "tNN" attribute contains the count for   
that hour to the current hour, in UTC/GMT format.   
ofi   
Element   
Calls overflowed into this queue. Each "tNN" attribute contains   
the count for that hour to the current hour, in UTC/GMT format.   
tct   
Element   
Total call time. The sum of talk time in the queue.   
ofo   
Element   
Calls overflowed out of this queue. Each "tNN" attribute   
contains the count for that hour to the current hour, in UTC/GMT   
format.   
tlkt   
Element   
Cumulative talk time of the agents assigned to the campaign   
associated with this queue. Each "tNN" attribute contains the   
count for that hour to the current hour, in UTC/GMT format   
twt   
Element   
Total wait time, in seconds. The sum of wait time for all calls in   
the queue that waited to be handled.   
Note   
AWT (Average waiting time) = (twt from QHistory + twt from QCalls) /   
(cwc from QHistory + cwc from QCalls)   
wrpt   
Element   
Cumulative wrap-up time of the agents assigned to the   
campaign associated with this queue. Each "tNN" attribute   
contains the count for that hour to the current hour, in UTC/GMT   
format

CCSP APIs Guide - 7.4   
98   
SServer   
The SServer filter shows raw data from the Statistics server. This data is internal data and should   
not be used in client application. This data can be used for logging and debugging.   
Usage   
http://<ServerName>/realtimereports/rtrweb.dll?Tenant=<TenantName>   
&Filter=SServer[DEV:&Username=<UserName>]   
   
Where:   
•   
<ServerName> is the IP address or domain name of the server hosting the RTR web site.   
•   
<TenantName> is the name of the tenant that you want to get stats for. Only for landlord   
users.   
•   
<UserName> is the Windows user name of an agent supervisor or CCSP administrator. This   
parameter is used only if IIS is configured for anonymous access. Otherwise, RTR uses the user   
that authenticated with IIS.   
Example   
 http://localhost/rtrdll/rtrweb.dll?Tenant=cosmocom&Filter=SServer&Username=ekeselman   
   
<?xml version="1.0" encoding="UTF-8" ?>   
<StatServer StartTime="1151505627">   
<Agents>   
   
<Agent PersonId="1" GlobalId="131072" FirstName="S" LastName="1"   
LoginName="s1" TenantId="0" AIS="9" Team="1" Group="1" State="2"   
MaximumNumberOfCalls="1" ReleaseCode="0" TimeInState="740">   
   
   
<MediaStreams />   
   
</Agent>   
</Agents>   
<Calls />   
<Servers>   
   
<Server ServerId="6" CCUApplicationType="2" Port="14006"   
IPAddress="1709100711" State="2" TimeInState="595183" />   
</Servers>   
<Trunks />   
<Campaigns />   
<StatisticCounters>   
   
<CounterTree CounterId="1001">   
   
   
<Counter>

CCSP APIs Guide - 7.4   
99   
   
   
   
<Counter GroupBy="7" />   
   
   
</Counter>   
   
</CounterTree>   
   
<CounterTree CounterId="1002">   
   
   
<Counter>   
   
   
   
<Counter GroupBy="0" />   
   
   
   
<Counter GroupBy="1" />   
   
   
</Counter>   
........................   
   
</StatisticCounters>   
<Enums>   
<CallStateEnum>   
   
<EnumMember Id="1" name="CallStateDoesntExist" />   
   
<EnumMember Id="2" name="CallStateInQueue" />   
   
<EnumMember Id="3" name="CallStateBeingServiced" />   
   
<EnumMember Id="4" name="CallStateAliveWithoutActiveStreams" />   
</CallStateEnum>   
<AgentStateEnum>   
   
<EnumMember Id="1" name="AgentStateLogout" />   
   
<EnumMember Id="2" name="AgentStateReleased" />   
   
<EnumMember Id="4" name="AgentStateUnavailable" />   
   
<EnumMember Id="5" name="AgentStateWrapUp" />   
   
<EnumMember Id="6" name="AgentStateInCall" />   
</AgentStateEnum>   
<TrunkStatesEnum>   
   
<EnumMember Id="1" name="TrunkStateDown" />   
   
<EnumMember Id="2" name="TrunkStateAvailable" />   
   
<EnumMember Id="3" name="TrunkStateInCall" />   
   
<EnumMember Id="4" name="TrunkStateOutCall" />   
   
<EnumMember Id="5" name="TrunkStateAgentCircuit" />   
   
<EnumMember Id="6" name="TrunkStateOutOfService" />   
</TrunkStatesEnum>   
<TrunkTypesEnum>   
   
<EnumMember Id="1" name="TrunkIN" />   
   
<EnumMember Id="2" name="TrunkOUT" />   
   
<EnumMember Id="3" name="TrunkCIRCUIT" />

CCSP APIs Guide - 7.4   
100   
   
<EnumMember Id="4" name="TrunkINOUT" />   
   
<EnumMember Id="5" name="TrunkOUTOFSERVICE" />   
</TrunkTypesEnum>   
<CallTypesEnum>   
   
<EnumMember Id="0" name="CallOut" />   
   
<EnumMember Id="1" name="CallIn" />   
   
<EnumMember Id="2" name="CallAgentToAgent" />   
   
<EnumMember Id="3" name="CallBridgedCall" />   
   
<EnumMember Id="4" name="CallbackRequest" />   
   
<EnumMember Id="5" name="CallbackResponse" />   
</CallTypesEnum>   
<ServerStatesEnum>   
   
<EnumMember Id="1" name="ServerStateDown" />   
   
<EnumMember Id="2" name="ServerStateUp" />   
</ServerStatesEnum>   
<CampaignStatesEnum>   
   
<EnumMember Id="1" name="CampaignStateDoesntExist" />   
   
<EnumMember Id="2" name="CampaignStateRunning" />   
   
<EnumMember Id="3" name="CampaignStateStopped" />   
   
<EnumMember Id="4" name="CampaignStateSuspended" />   
</CampaignStatesEnum>   
<StatisticLevels>   
   
<EnumMember Id="0" name="TENANT\_STAT" />   
   
<EnumMember Id="1" name="CS\_STAT" />   
   
<EnumMember Id="2" name="AIS\_STAT" />   
   
<EnumMember Id="3" name="QUEUE\_STAT" />   
   
<EnumMember Id="4" name="TEAM\_STAT" />   
   
<EnumMember Id="6" name="SKILLSET\_STAT" />   
   
<EnumMember Id="7" name="MEDIATYPE\_STAT" />   
   
<EnumMember Id="8" name="TRUNKGROUP\_STAT" />   
   
<EnumMember Id="9" name="TRUNK\_STAT" />   
   
<EnumMember Id="10" name="IPLINK\_STAT" />   
   
<EnumMember Id="11" name="AGENT\_STAT" />   
   
<EnumMember Id="12" name="CALL\_STAT" />   
   
<EnumMember Id="13" name="APP\_STAT" />   
   
<EnumMember Id="14" name="DISPOSITION\_STAT" />   
   
<EnumMember Id="15" name="DIALEROUTCOME\_STAT" />

CCSP APIs Guide - 7.4   
101   
   
<EnumMember Id="16" name="CAMPAIGN\_STAT" />   
</StatisticLevels>   
<StatisticCounters>   
   
<EnumMember Id="1001" name="AvailableForOutDialTrunks" />   
   
<EnumMember Id="1002" name="TotalTrunkUtilization" />   
   
<EnumMember Id="1003" name="TotalTimeAllTrunksBusy" />   
   
<EnumMember Id="1004" name="NumberOfTrunks" />   
   
<EnumMember Id="1005" name="NumberOfTrunksBusy" />   
   
<EnumMember Id="1006" name="NumberOfTrunksOutOfService" />   
   
<EnumMember Id="1007" name="AllTrunksList" />   
   
<EnumMember Id="2001" name="AllIPLinksList" />   
   
<EnumMember Id="2002" name="TotalTimeAllIPLinksBusy" />   
   
<EnumMember Id="2003" name="TotalIPLinkUtilization" />   
   
<EnumMember Id="3001" name="AllLoggedInAgentsList" />   
   
<EnumMember Id="3002" name="ListOfAgentsInCall" />   
   
<EnumMember Id="4001" name="CallsWaiting" />   
   
<EnumMember Id="4002" name="CallsBeingHandled" />   
   
<EnumMember Id="4003" name="CallsOverflowedIn" />   
   
<EnumMember Id="4004" name="CallsOverflowedOut" />   
   
<EnumMember Id="4005" name="CallsAliveWithoutStreamsList" />   
   
<EnumMember Id="4006" name="CallsBeingHandledList" />   
   
<EnumMember Id="4007" name="CallsWaitingList" />   
   
<EnumMember Id="5001" name="AllServersList" />   
   
<EnumMember Id="6001" name="PDSNumberOfCallsByDisposition" />   
   
<EnumMember Id="6002" name="PDSCumulativeCallsLengthByDisposition" />   
   
<EnumMember Id="6003" name="PDSAvailableTime" />   
   
<EnumMember Id="6004" name="PDSTalkTime" />   
   
<EnumMember Id="6005" name="PDSWrapTime" />   
   
<EnumMember Id="6006" name="CampaignRunTime" />   
   
<EnumMember Id="6007" name="PDSNumberOfCallsByOutcome" />   
   
<EnumMember Id="6008" name="AllCampaignsList" />   
   
<EnumMember Id="33287" name="CallsHandled" />   
   
<EnumMember Id="33282" name="CallsAbandoned" />   
   
<EnumMember Id="33288" name="CallsOffered" />   
   
<EnumMember Id="33306" name="ServiceLevel" />   
   
<EnumMember Id="4100" name="CallsAbandonedWithinTQOS" />   
   
<EnumMember Id="4101" name="CallsRejected" />

CCSP APIs Guide - 7.4   
102   
   
<EnumMember Id="33304" name="LongestWaitingCall" />   
   
<EnumMember Id="33299" name="SpeedToAnswerTime" />   
</StatisticCounters>   
</Enums>   
</StatServer>

CCSP APIs Guide - 7.4   
103   
9: CCSP TWS integration   
This chapter explains how to use a generic client application to integrate with Teams Web Service   
(TWS). It details the TWS Web API, the SignalR Hub, and includes some code snippets to facilitate   
the integration. Sequence diagrams with the main flows are also included.   
This chapter contains the following information:   
•   
Architecture   
•   
Requirements   
•   
Web API   
•   
SignalR Hub   
•   
Client application flows   
Architecture   
The following diagram shows the TWS integration architecture.

CCSP APIs Guide - 7.4   
104   
The Teams Web Service (TWS) is a C# .NET Core Application. It can be deployed as a self-hosted   
web application running as a windows service.   
TWS integrates with Microsoft Teams by calling methods on the MS Graph API to get the presence   
of Teams users.   
TWS subscribes to presence change notifications using the Graph API. Whenever a Teams user   
subscribed in TWS changes its presence, a notification is received by TWS.   
A Web API is exposed by TWS so that a client application can use it to integrate with teams.   
TWS provides an ASP.NET Core SignalR Hub to propagate events to a client application.   
Both the TWS Web API and the SignalR Hub are secured using JWT Bearer Authentication. Tokens   
can be obtained from CCSP’s Auth Server or from another token issuer.   
Requirements   
For the integration between CCSP and MS Teams to work correctly, the following requirements   
must be met:   
1.   
The first part of the User Principal Name in teams must match the Login ID in CCSP. Example:   
if a Teams user with userPrincipalName “PattiF@ccspteams2.onmicrosoft.com” is to work as   
an agent in CCSP it must have the Login ID “Pattif” in CCSP. The Teams user must have at least   
one phone number.   
2.   
As the TWS Web API methods are authenticated, the client application must get a bearer token   
before calling the methods and send it in the HTTP Authorization header. For the client   
application to connect to the SignalR Hub it must also send the same token.   
   
The client application may get the bearer token from CCSP’s AuthService or from another token   
issuer. When using a token issuer different from AuthService, the TWS has be configured with token   
parameters Issuer, Audience and OAuthSecretKey to validate the tokens. For more details on   
authentication in TWS with Auth Server, please see the Authentication in TWS with AuthServer flow.   
Web API   
TWS Web API methods are accessible at [TWSBaseAddress]/TeamsPresence/.   
These are the available methods.   
Connect (agent, UPN)   
This method adds the agent to the list of active agents in the TWS. It should be called when the   
agent logs in CCSP.   
Request parameters   
•   
agent – the CCSP agent Login ID. Example “agent01”.   
•   
UPN – the User Principal Name suffix that identifies the tenant. Example “t1.loc”.

CCSP APIs Guide - 7.4   
105   
Example request URL   
GET https://localhost:6001/TeamsPresence/Connect?agent=agent01&UPN=t1.loc   
Return value   
On success (HTTP 200 OK) the method returns an empty body.   
Note   
This method doesn’t need to be called explicitly by the Client Application as the Get\* methods   
already execute the same logic in TWS internally.   
   
Disconnect (agent, UPN)   
This method removes the agent from the list of active agents in TWS. It should be called when the   
agent logs out from CCSP.   
Request parameters   
•   
agent – the CCSP agent Login ID. Example “agent01”.   
•   
UPN – the User Principal Name suffix that identifies the tenant. Example “t1.loc”.   
Example request URL   
GET https://localhost:6001/TeamsPresence/Disconnect?agent=agent01&UPN=t1.loc   
Return value   
On success (HTTP 200 OK) the method returns an empty body.   
GetUsersPresences (agent, UPN)   
This method gets the presences of the currently active agents in the TWS. It should be called only   
when connecting or reconnecting to the server.   
Request parameters   
•   
agent – the CCSP agent Login ID. Example “agent01”.   
•   
UPN – the User Principal Name suffix that identifies the tenant. Example “t1.loc”.   
Example request URL   
GET https://localhost:6001/TeamsPresence/GetUsersPresences?agent=agent01&UPN=   
t1.loc

CCSP APIs Guide - 7.4   
106   
Return value   
The method returns the presence information for each user. It’s an array of JSON objects.   
As an example, here’s a response body with presence information for 2 users.   
[   
 {   
 "userPrincipalName": "PattiF@ccspteams2.onmicrosoft.com",   
 "displayName": "Patti Fernandez",   
 "givenName": "Patti",   
 "surname": "Fernandez",   
 "number": "+1 502 555 0144",   
 "mail": "PattiF@ccspteams2.onmicrosoft.com",   
 "department": "Executive Management",   
 "activity": "Available",   
 "availability": "Available",   
 "id": "fe4135db-175b-4149-978e-53c5a947c107"   
 },   
 {   
 "userPrincipalName": "NestorW@ccspteams2.onmicrosoft.com",   
 "displayName": "Nestor Wilke",   
 "givenName": "Nestor",   
 "surname": "Wilke",   
 "number": "+1 206 555 0105",   
 "mail": "NestorW@ccspteams2.onmicrosoft.com",   
 "department": "Operations",   
 "activity": "Busy",   
 "availability": "Busy",   
 "id": "7c9e9ed7-7761-4688-afc9-0e11086d1301"   
 }   
]   
Note   
The field “activity” has the Teams presence information for the users. In the example above,   
PattiF@ccspteams2.onmicrosoft.com is “Available” and NestorW@ccspteams2.onmicrosoft.com   
is “Busy”.

CCSP APIs Guide - 7.4   
107   
GetUsersPresencesSearch (agent, UPN, filter)   
This method is used to search for contacts in the Teams directory. Please note that the contacts   
don’t have to be CCSP users. The search will return the teams users that match the filter criteria.   
The number of results is limited by configuration.   
The requests to this method are cached on the server, meaning that every search filter used by an   
agent is stored in a memory cache and then reused for its lifetime (configurable).   
Parameters   
•   
agent – the CCSP agent Login ID. Example “agent01”.   
•   
UPN – the User Principal Name suffix that identifies the tenant. Example “t1.loc”.   
•   
filter – a string to filter the users in the Teams directory.   
Example request URL   
GET https://localhost:6001/TeamsPresence/GetUsersPresencesSearch?agent   
=agent01&UPN=t1.loc&Filter=m   
Return value   
The method returns the presence information for each user. It’s an array of JSON objects.   
As an example, here’s a response body with presence information for 3 users which is the result of   
searching with filter “m”. This will return all users in the Teams directory that have a name or   
surname starting with “m”.   
 [   
 {   
 "userPrincipalName": "MeganB@ccspteams2.onmicrosoft.com",   
 "displayName": "Megan Bowen",   
 "givenName": "Megan",   
 "surname": "Bowen",   
 "number": "+1 412 555 0109",   
 "mail": "MeganB@ccspteams2.onmicrosoft.com",   
 "department": "Marketing",   
 "activity": "Away",   
 "availability": "Away",   
 "id": "4cd43515-7986-437a-a4e3-fbbd66b74472"   
 },   
 {   
 "userPrincipalName": "MiriamG@ccspteams2.onmicrosoft.com",   
 "displayName": "Miriam Graham",   
 "givenName": "Miriam",

CCSP APIs Guide - 7.4   
108   
 "surname": "Graham",   
 "number": "+1 858 555 0109",   
 "mail": "MiriamG@ccspteams2.onmicrosoft.com",   
 "department": "Sales & Marketing",   
 "activity": "Offline",   
 "availability": "Offline",   
 "id": "abe9cc47-ad70-4527-b92a-41e2197456ee"   
 },   
 {   
 "userPrincipalName": "HenriettaM@ccspteams2.onmicrosoft.com",   
 "displayName": "Henrietta Mueller",   
 "givenName": "Henrietta",   
 "surname": "Mueller",   
 "number": "+1 954 555 0118",   
 "mail": "HenriettaM@ccspteams2.onmicrosoft.com",   
 "department": "R&D",   
 "activity": "BeRightBack",   
 "availability": "BeRightBack",   
 "id": "fe7ae738-f5fe-4134-b1b3-487a2dcfe3ae"   
 }   
]   
GetMyPresence (agent, UPN)   
This method gets the presence of the current user.   
Request parameters   
•   
agent – the CCSP agent Login ID. Example “agent01”.   
•   
UPN – the User Principal Name suffix that identifies the tenant. Example “t1.loc”.   
Example request URL   
GET https://localhost:6001/TeamsPresence/GetMyPresence?agent   
=agent01&UPN=t1.loc

CCSP APIs Guide - 7.4   
109   
Return value   
The method returns the presence information for the user. It’s a JSON object.   
As an example, here’s a response body with presence information for the user.   
 {   
 "userPrincipalName": "NestorW@ccspteams2.onmicrosoft.com",   
 "displayName": "Nestor Wilke",   
 "givenName": "Nestor",   
 "surname": "Wilke",   
 "number": "+1 206 555 0105",   
 "mail": "NestorW@ccspteams2.onmicrosoft.com",   
 "department": "Operations",   
 "activity": "Available",   
 "availability": "Available",   
 "id": "7c9e9ed7-7761-4688-afc9-0e11086d1301"   
 }   
   
In the example above NestorW@ccspteams2.onmicrosoft.com is “Available”.   
Error Codes   
The error codes returned by the API methods are:   
•   
HTTP 400 Bad Request – when the agent or the UPN Request parameters are empty.   
•   
HTTP 401 Unauthorized – when the methods are called without sending the Bearer token in   
the Authorization header of the HTTP request or when sending a token that has expired.   
•   
HTTP 403 Forbidden – when the claims sent in the Bearer token are not valid.   
•   
HTTP 404 Not Found – when the agent or the UPN Request parameters are not found in the   
server.   
SignalR Hub   
The SignalR Hub is accessible at address [TWSBaseAddress]/hub/teams   
Notifications of Teams presence changes are sent from TWS to a client application using ASP.Net   
Core SignalR.   
The client application subscribes to notifications from TWS by connecting to the SignalR Hub.   
Note   
For information about developing a SignalR client in JavaScript please check   
https://docs.microsoft.com/en-us/aspnet/core/signalr/javascript-client?view=aspnetcore-6.0.

CCSP APIs Guide - 7.4   
110   
Create connection   
First the client application must create a connection to the TWS SignalR Hub. This depends on the   
programming language in which the client is written but it’s usually done by calling   
HubConnectionBuilder passing the Hub address. As the SignalR Hub is secured using JWT   
authentication, the access\_token must be sent in the requests.   
Here’s an example using JavaScript.   
connection = new signalR.HubConnectionBuilder()   
.withUrl(https://localhost:6001/hub/teams, {accessTokenFactory: () =>   
access\_token})   
.build();   
   
Note   
When creating the SignalR connection it’s possible to configure what happens when the   
connection with the server is lost. For more details check: https://docs.microsoft.com/en-  
us/aspnet/core/signalr/javascript-client?view=aspnetcore-6.0&tabs=visual-studio#reconnect-  
clients.   
Start   
After having the connection object, the client application must call a method called start or similar.   
For example, in JavaScript:   
connection.start()   
   
At this point the connection with the TWS Hub is established.   
AddToGroup   
After starting the connection, the client application must call method AddToGroup to pass the   
tenant UPN information to TWS. This step is needed for Teams Web Service to notify only the   
agents from the same tenant when a presence update occurs. For example, in JavaScript:   
connection.start().then(function () {   
console.log("SignalR client connected");   
connection.invoke("AddToGroup", "t1.loc");   
}

CCSP APIs Guide - 7.4   
111   
The following notifications are sent by SignalR to the client application.   
StatusUpdate   
Notifies clients that a presence update occurred in one Teams user.   
The StatusUpdate message sends the presence update information of a single Teams user. It’s a   
JSON object. Here’s an example of the StatusUpdate message:   
 {   
 "userPrincipalName": "PattiF@ccspteams2.onmicrosoft.com",   
 "displayName": "Patti Fernandez",   
 "givenName": "Patti",   
 "surname": "Fernandez",   
 "number": "+1 502 555 0144",   
 "mail": null,   
 "department": null,   
 "activity": "Available",   
 "availability": "Available",   
 "id": "fe4135db-175-4149-978e-53c5a947c107"   
 }   
   
Note   
The field “activity” has the Teams presence information for the user. In the example above the   
teams user PattiF@ccspteams2.onmicrosoft.com became “Available”.   
   
   
Here’s the JavaScript sample code to handle the notification.   
connection.on("StatusUpdate", function (message) {   
console.log("Received StatusUpdate:" + message.userPrincipalName + " is " + me  
ssage.activity);   
//add additional logic here   
}

CCSP APIs Guide - 7.4   
112   
RemoveAgent   
Notifies the client application that an agent is no longer active in the Teams Web Service. The   
RemoveAgent message is a JSON object. Here’s an example message:   
 {   
 "userPrincipalName": "NestorW@ccspteams2.onmicrosoft.com",   
 "displayName": "Nestor Wilke",   
 "givenName": "Nestor",   
 "surname": "Wilke",   
 "number": "+1 206 555 0105",   
 "mail": null,   
 "department": null,   
 "activity": "Offline",   
 "availability": "Offline",   
 "id": "7c9e9ed7-7761-4688-afc9-0e11086d1301"   
 }   
   
In this example, the CCSP agent NestorW is no longer active in the Teams Web Service.   
Here’s the JavaScript code to handle the notification:   
connection.on("RemoveAgent", function (message) {   
console.log("Received RemoveAgent:" + message.userPrincipalName);   
//add additional logic here   
});

CCSP APIs Guide - 7.4   
113   
Client application flows   
In this section the main client application flows are shown as sequence diagrams.   
Agent login   
When the agent clicks the “Start” button, the following flow is executed. The initial step of the flow   
is the authentication of the Client Application with AuthServer as explained in the next flow.   
   
Note   
connection.start() is automatically called by SignalR after the connection with the Hub is   
established.

CCSP APIs Guide - 7.4   
114   
Authentication in TWS with AuthServer

CCSP APIs Guide - 7.4   
115   
   
In the above diagram we can see how to call TWS with authenticated requests. These are the steps   
in more detail:   
1.   
The Client Application calls the /init endpoint on AuthServer to get an RSA key pair.   
2.   
The Client Application encrypts the agent password with the RSA key pair.   
3.   
The Client Application sends a request to the /login endpoint of AuthServer with the   
username, encrypted password, teams\_client\_id (a specific client\_id for teams in AuthServer)   
and grant\_type = ‘password’. The response to this request will have an access\_token and a   
refresh\_token.   
4.   
Requests to TWS must include “Bearer “ concatenated with the teams\_access\_token in the   
HTTP Authorization header to be valid.   
5.   
When a request with an expired access\_token is done, TWS replies with “401 Unauthorized”   
and the Client Application must get a new access\_token sending a request to the /login   
endpoint of AuthServer with the teams\_client\_id, refresh\_token and grant\_type =   
‘refresh\_token’. The response to this request will have a new access\_token that must be used   
to authenticate the requests to TWS.   
6.   
When a request to AuthServer to refresh the access\_token is done with an expired   
refresh\_token, AuthServer replies “401 Unauthorized” and the Client Application must   
authenticate again with AuthServer starting in point 1 above.   
   
Note   
AuthServer default lifetime for an access\_token is 15 minutes and for a refresh\_token is 10 hours.   
   
Search users in Teams directory   
The agent searches users in the Teams directory.

CCSP APIs Guide - 7.4   
116   
Agent changes presence   
The agent changes presence in the Teams client.   
   
   
Another agent changes presence   
Another agent changes presence in the Teams client.

CCSP APIs Guide - 7.4   
117   
Agent logout   
The agent logs out.   
   
Another agent logs out   
Another agent logs out.

CCSP APIs Guide - 7.4   
118   
10: CCSP UI Connector   
The CCSP UI Connector (previously called the CCSP TouchPoint Connector) provides a two-way   
integration hub on the client for integration into other desk applications or web applications.   
For a description of the architecture and .NET API Interface, see the CCSP UI Connector Guide.   
You can find the CCSP UI Connector Guide.pdf in the same location as this guide.

CCSP APIs Guide - 7.4   
119   
A: CCSP Web Support integration   
Note   
CCSP now has a JSON-based Chat API that can be used on mobile and non-mobile   
devices and different html pages.   
We recommend that you use the JSON-based CCSP Chat API instead of the CCSP Web Support   
package, because the CCSP Chat API:   
•   
Is more flexible and can be customized as required.   
•   
Uses less bandwidth than the html-based CCSP Web Support package that contains extra   
data such as design.   
•   
Will have new features developed for it, such as typing. No new features will be developed for   
the CCSP Web Support package.   
   
For more information, see 6: CCSP Chat API.   
   
This appendix contains the following information:   
•   
Chat call initiation sequence   
•   
Web Support package components   
•   
Send external callbacks via SMTP   
•   
Chat extension   
•   
Chat extension components   
•   
Interfaces to the chat extension   
•   
Customize the chat window   
•   
Parameters   
   
The CCSP Web Support package provides an easily configurable method to enable chat calls and   
callback requests to your contact center.   
Contact-enabling your website involves adding a configured chat button to your website and   
embedding tags to set the parameters that are needed to initiate the call.   
You use the Configurator to set the defaults that should be used for all chat calls. These defaults   
are saved in an Active Server Pages (ASP) script. You can use the ASP components to quickly   
contact-enable your website using these defaults and by gathering actual caller information.

CCSP APIs Guide - 7.4   
120   
Chat call initiation sequence   
To initiate a chat call, the following sequence must occur:   
1.   
User browses to the contact center's webpage.   
2.   
If cookies are used, a cookie is set with parameters appropriate for the call.   
3.   
User clicks the configured button.   
4.   
The script grabs the parameters from the cookie, URL query, or form.   
5.   
The script creates a query string and redirects to ChatExtension.dll. If any parameters are   
missing, the script uses the defaults set in the Configurator.   
6.   
The HTML chat window is launched on the caller's computer and the chat is initiated.   
Web Support package components   
The Web Support package includes:   
•   
Configurator — used to enter the defaults for all chat calls, including file and directory   
locations, optional parameters, and other information needed by the Web Support package to   
launch chats. See Configurator.   
•   
ASP scripts — the quickest way to contact-enable your website. These scripts are generated   
using Configurator. See Quick start.   
•   
Chat extension — an ISAPI extension DLL that runs on your Microsoft IIS Web Server. The   
extension provides a set of APIs that you can use on a webpage to communicate with the CCSP   
ACD. Parameters can be passed to the APIs as either HTML form fields or a URL query. See   
Chat extension.   
Configurator   
Use the Configurator to create a configuration that contains the defaults to be used for all chat   
calls, including file and directory locations, optional parameters, and other information needed to   
launch chats from a configured webpage. The defaults are used if the associated parameters are   
not passed to the ASP script when the call is initiated.   
The configuration information is stored in ASP scripts. You can create multiple configurations,   
possibly one for each of your webpages. Each configuration that you create must reside in a unique   
directory that has script execute and read privileges.   
To create a new configuration or make changes to an existing configuration:   
1.   
Click Start, point to All programs, click CCSP, and then click CCSP ASP File Configurator. The   
CCSP ASP File Configurator dialog box appears.   
2.   
On the File menu, click New Configuration or to edit an existing one, click Open   
Configuration.   
3.   
Type the new or existing configuration name, and click OK.   
4.   
In Directory (Physical Path), enter the local path to your scripts directory.   
5.   
On the Call Center Information tab, enter defaults for your contact center.

CCSP APIs Guide - 7.4   
121   
6.   
Click the No Agents Action tab and enter the following information:   
o   
No Agents Action — a webpage that handles rejected chat calls.   
o   
No Agents Message — a continuous string of text. Any hard returns are removed.   
7.   
Click the Caller Information tab and enter the following information:   
o   
Caller Phone Type — chat is always the default for HTML and Java callers.   
o   
Caller URL — the URL to forward to the agent at the start of the chat.   
o   
HTML Caller Initial URL — the URL that should appear in the right frame of the HTML   
chat window.   
o   
Caller Message — the message that the agent should see.   
o   
Caller Name — the name that the agent should see.   
o   
Call Center Queue and Tenant — routing information to use if no customer parameters   
are set.   
o   
Skills Required — use a comma, space, semicolon, or a plus sign (+) to separate multiple   
skills.   
o   
Caller Language — language of the HTML chat window, for example, 1033 for English.   
For other languages, substitute the correct Windows locale ID code.   
8.   
Click the Optional Params tab and enter the optional parameters. You can’t use blank spaces   
in the parameter names. Some optional parameters that you might need to use for integration   
with another application include: application\_name, customer\_ID, shopping\_cart\_ID,   
trouble\_ticket, order\_number, and tracking\_number.   
9.   
On the File menu, click Save Configuration.   
When you save a configuration, a number of sample .asp files are created in the directory you   
selected for the configuration.   
File   
Description   
call.asp   
ASP script that contains the default values you want to use for   
all email and chat calls configured to use this script.   
makecall.asp   
Places the call to the ChatExtension.dll using the parameters   
from call.asp.   
call\_request\_demo.htm   
Sample HTML form for testing purposes.   
cookiecaller.txt   
Inserts button image and link into page on caller's browser.   
Reads parameters from cookie and creates a URL with   
parameters (used for URL query). Included by cosmotest.asp.   
cosmocallasp.asp   
Sample HTML form that can be used to collect caller data.   
cosmotest.asp   
Sample script used for testing the three parameter-gathering   
methods: cookie, URL query, and form. Works with   
call\_request\_demo.htm.   
talk\_to\_us4.gif   
Button logo. Requires your \scripts directory to have read   
access privileges.

CCSP APIs Guide - 7.4   
122   
Quick start   
The quickest way to contact-enable your website is to use the generated ASP scripts. There are   
several ways to use the ASP scripts to initiate a call, however, the simplest way is to include the   
following on your webpage:   
<a href=“<path>/call.asp”><img src=“<path>/talk\_to\_us4.gif”></a>   
where <path> indicates the fully qualified URL to the ASP script you created.   
When the user clicks this button, call.asp initiates a call using the default values that were set with   
the Configurator.   
This method is suitable for testing or evaluation purposes and very basic websites. In a real   
operation it is likely that you want to use CCSP's intelligent routing and application integration   
features which utilize information about the customer. This information can be gathered from a   
form or from the pages the caller visited on your website, or it can be derived from a server-side   
application such as a shopping cart or help desk application.

CCSP APIs Guide - 7.4   
123   
Send external callbacks via SMTP   
This section contains the following information:   
•   
Extended SMTP headers   
•   
Sample script   
   
This section describes an open API for sending callback requests into the CCSP platform using the   
SMTP protocol.   
Note   
To be able to use SMTP callbacks with the MSR DB and an external mailbox, the email data store   
must be assigned with an MSR DB in CCSP Admin.   
   
When using the Tenant Exchange Database as the Messaging store:   
•   
Each tenant requires a mail-enabled folder for collecting these callback requests. The folder   
email address is Callback@<tenantSuffix>. The <tenantSuffix> is located in Tenant properties, in   
the CCSP Administrator.   
   
When using the Tenant Messaging Database as the Messaging store:   
•   
Each tenant requires at least one folder associated with an external mailbox.   
•   
The email address of that folder can be used to send callback requests using the SMTP   
protocol.   
   
Application developers include the relevant tenant and routing context data from the defined CCSP   
configuration.   
Extended SMTP headers   
Extended SMTP headers allow for the transmission of extra information along with callback   
requests. This can include such information as the caller's name and a message, and scheduling   
information for callbacks and routing.   
All headers are strings.   
Header field   
Description   
X-CosmoCall-AccountName   
Name of the account to use for generated call, for example,   
"Microsoft Account"   
X-CosmoCall-AgentID   
Designated agent’s configuration ID.   
Note   
If a personal callback is scheduled without the   
SegregationGroupID, the AgentID is removed, an error is   
logged, and the callback is treated as a non-personal callback   
and routed to the next suitable agent.

CCSP APIs Guide - 7.4   
124   
Header field   
Description   
X-CosmoCall-AttachmentUrl   
URL to send with the call. This URL is used for backward   
compatibility for integration purposes on the client application   
side.   
X-CosmoCall-CallerANI   
Caller's ANI to send with the call.   
X-CosmoCall-CallerDNIS   
Caller's DNIS to send with the call.   
X-CosmoCall-CallerIvrDigits   
Caller's IVR digits to send with the call.   
X-CosmoCall-CallerName   
Caller's name to send with the call.   
X-CosmoCall-  
DirectInwardDialing   
Whether direct route to agent specified via X-CosmoCall-  
AgentID field. If yes, set to “True”   
X-CosmoCall-ImmediateDialing   
Whether callback outdialing address should be dialed   
immediately without agent intervention when the agent picks   
up the callback call. If yes, set to “True”   
X-CosmoCall-Message   
Text message to send with the call and to present to the agent,   
for example, "This call arrived from auto-callback   
script"   
X-CosmoCall-Option#   
Optional parameters to send with the call, for example,   
"Name>Value"   
Optional parmeters must be sequential and their value must   
be in the form “{key}>{value}”   
For example:   
X-CosmoCallOption1=key1>value1   
X-CosmoCallOption2=key2>value2   
Note   
Out-of-sequence optional parameters are ignored.   
X-CosmoCall-OriginatingCallId   
Originating (linked) call's call ID.   
X-CosmoCall-  
OriginatingCallType   
Originating (linked) call's media stream type.   
X-CosmoCall-  
OutDialingAddress   
Callback outdialing address, usually a number to dial out   
(required).   
X-CosmoCall-QueueName   
Name of the queue to which call is to be routed   
(recommended).   
If this header is not in the database, MCS routes to the default   
queue. If the default queue is not defined for the tenant, the   
callback is rejected and moved to the Rejected Callback folder.

CCSP APIs Guide - 7.4   
125   
Header field   
Description   
X-CosmoCall-ScheduleTime   
Callback schedule time, in.NET parseable format (required).   
X-CosmoCall-  
SegregationGroupID   
Enables segregation of callback items by group in Message   
View.   
Note   
From CCSP 7.3.x onward, personal callbacks scheduled without   
the SegregationGroupID are treated as non-personal   
callbacks.   
X-CosmoCall-SkillsNames   
List of skills assigned to call, separated by semicolon.   
For example, "s1;s2”

CCSP APIs Guide - 7.4   
126   
Sample script   
The following script uses some of the headers described above.   
' Reference KB: https://msdn.microsoft.com/en-us/library/   
 ms526453(v=exchg.10).aspx   
   
Const cdoSendUsingPickup = 1   
Const cdoSendUsingPort = 2 'Must use this to use Delivery Notification   
Const cdoBasic = 1 ' clear text   
Const cdoNTLM = 2 'NTLM   
   
Dim objConf   
Set objConf = CreateObject("CDO.Configuration")   
With objConf   
 .Fields("http://schemas.microsoft.com/cdo/configuration/sendusing") =   
cdoSendUsingPickup   
 .Fields.Update   
End With   
   
Dim objMessage   
Set objMessage = CreateObject("CDO.Message")   
With objMessage   
.Configuration = objConf   
.From = "svccosmocall@pj12.loc"   
.To = "callback@defTenant.net"   
.Fields("urn:schemas:mailheader:X-CosmoCall-Message") = "Callback sent via   
automatic script"   
.Fields("urn:schemas:mailheader:X-CosmoCall-CallerName") = "Customer Name"   
.Fields("urn:schemas:mailheader:X-CosmoCall-AttachmentUrl") =   
"http://global.sales.domain.com/callback"   
.Fields("urn:schemas:mailheader:X-CosmoCall-SkillsNames") = "s1;s2"   
.Fields("urn:schemas:mailheader:X-CosmoCall-QueueName") = "q1"   
.Fields("urn:schemas:mailheader:X-CosmoCall-OutDialingAddress") = "+1-631-940-  
4200"   
.Fields("urn:schemas:mailheader:X-CosmoCall-Option1") = "CUSTOMERID>1234"   
.Fields("urn:schemas:mailheader:X-CosmoCall-Option2") = "ACCOUNTID>5678"   
.Fields("urn:schemas:mailheader:X-CosmoCall-ScheduleTime") = "10:00+02:00"   
.Fields.Update   
.Send

CCSP APIs Guide - 7.4   
127   
End With   
   
Set objMessage = nothing

CCSP APIs Guide - 7.4   
128   
Chat extension   
This section contains the following information:   
•   
How to use the chat extension   
•   
Chat extension components   
•   
Interfaces to the chat extension   
   
One of the central components of the CCSP Web Support package is the chat extension that   
enables chat calls to be made from websites.   
The chat extension is an ISAPI extension DLL that runs on your Microsoft IIS Web Server. The   
extension provides a set of APIs that you can use on a webpage to communicate with the CCSP   
ACD. Parameters can be passed to the APIs as either HTML form fields or a URL query.   
The extension maintains a chat conversation for each connected caller. Using the APIs exposed as   
URLs, the webpages can connect and disconnect callers, send messages to the agent, and read the   
chat conversation messages and status. A connection ID is used to identify each chat call.   
How to use the chat extension   
The ChatExtension.dll is installed in the IIS scripts directory, normally c:\inetpub\scripts. If there   
are separate physical directories on your virtual web server, the ChatExtension.dll is installed in   
each one.   
The HTML and JavaScript files must be in a real or virtual \CosmoCall directory. If it is a real   
directory, it must be off the root of the web server. The default path for these components is   
c:\inetpub\wwroot\cosmocall\1033\default. The 1033 portion of the path is for English. For other   
languages, substitute the correct Windows locale ID code.   
The chat extension DLL requires that the \scripts directory has execute privileges.   
You may modify the HTML and JavaScript files to meet your implementation requirements.   
Minimum requirements   
To use the chat extension:   
•   
JavaScript needs to be enabled in the caller's browser, for example, Chrome.   
•   
The directory containing ChatExtension.dll must have ISAPI caching enabled.   
Configuration   
The chat extension uses several entries in the system registry for configuration. Default values are   
established when the extension is first used, so it is not necessary to predefine configuration data.

CCSP APIs Guide - 7.4   
129   
The following keys are stored on the ACD server in the registry:   
HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\CosmoCom\Universe\CurrentVersion\ChatExtension\Parameters   
\[tenant name]   
Registry key   
Type   
Default value   
Description   
ChatAgentNameColor   
String   
#0000FF   
(Blue) HTML color code for the agent name in the   
chat window.   
ChatAgentTextColor   
String   
#0000FF   
(Blue) HTML color code for the agent text in the   
chat window.   
ChatCallerName   
String   
Me   
Name that is displayed in the chat window to   
identify the caller. To use the   
Calling\_User\_FirstName parameter, leave this   
setting blank.   
ChatSystemMsgColor   
String   
#FF0000   
(Red) HTML color code for the system message in   
the chat window.   
ChatSystemTimeColor   
String   
#FF0000   
(Red) HTML color code for the system message   
time stamp in the chat window.   
ChatSystemTimeFormat   
String   
%H:%M   
Format of system time stamp in chat window. You   
can add :%S for seconds. If you do not want the   
time displayed, leave this setting blank   
ChatUserNameColor   
String   
#000000   
(Black) HTML color code for user name in the chat   
window.   
ChatUserTextColor   
String   
#000000   
(Black) HTML color code for the user text in the chat   
window.   
ChatWindowBackground  
Color   
String   
#FFFFFF"   
background="/Cosmocall/spiral.jpg   
(White) HTML color code for the chat window   
background.

CCSP APIs Guide - 7.4   
130   
Registry key   
Type   
Default value   
Description   
ChatWindowRefreshPeri  
od   
DWORD   
5   
The number of seconds between automatic   
refreshes of the chat window.   
ChatWindowRowFormat   
String   
<TR><TD>&nbsp;&nbsp;&nbsp;&nbsp;   
&nbsp;&nbsp;</TD><TD valign="top">   
<FONT Size="1" COLOR="%s"   
FACE="Verdana, Arial">%s</FONT></TD>   
<TD valign="top" width="180"><FONT   
size="1"   
FACE="Verdana, Arial"   
COLOR="%s">%s</font>   
</TD></TR>   
Format (font, size) of text in chat window.   
ChatWindowSplit   
DWORD   
80   
Percent of chat table for the text column versus the   
name column.   
DisplayFrame   
String   
\_blank   
When the agent pushes a URL to the caller, the   
caller's browser opens that URL in the named   
window. The default name, "\_blank", causes the   
browser to always open a new window. Otherwise,   
it always goes to the same named window in   
Netscape browsers.   
StatusBackgroundColor   
String   
#FFFFFF"   
background="/Cosmocall/statusBack.gif   
(Gray) HTML color code for the status window   
background.   
StatusMsgColor   
String   
#000000   
(Black) HTML color code for the status line message   
text.   
StatusRefreshPeriod   
DWORD   
5   
The number of seconds between automatic   
refreshes of the status line window.

CCSP APIs Guide - 7.4   
131   
Registry key   
Type   
Default value   
Description   
StatusRowFormat   
String   
<TR><TD valign=\"top\"><FONT   
Size=\"1\"   
FACE=\"Verdana, Arial\"&nbsp;   
COLOR=\"%s\">%s</FONT></TD>   
<TD align=\"right\" width=\"25%%\">   
<FONT Size=\"1\" FACE=\"Verdana,   
Arial\"   
COLOR=\"%s\">%s</font></TD></TR>   
Format (font, size) of text in status line.   
TimeFormat   
String   
hh:mm tt   
   
URLGo   
DWORD   
1   
Displays pushed URLs in URL frame of chat window.   
URLShow   
DWORD   
1   
Shows URL string in chat conversation frame on   
caller's browser.   
UserTimeoutPeriod   
DWORD   
25   
The number of seconds without contact before the   
chat extension deletes a user's connection.   
Chat extension components   
The chat extension consists of these components.   
Component   
File   
Description   
Server   
ChatExtension.dll   
The chat extension DLL that runs on your Microsoft IIS Web Server.   
HTML template file   
used during normal   
operation   
ChatExtensionClose.txt   
This template defines a form for disconnecting from the contact center. The   
Close ISAPI command is used in this form.   
HTML file used to   
report error conditions   
ChatExtensionDefault.htm   
This page is returned when an ISAPI URL name that is not recognized by the   
chat extension DLL is accessed. This does not occur during normal use.

CCSP APIs Guide - 7.4   
132   
Component   
File   
Description   
HTML template file   
used during normal   
operation   
ChatExtensionFrame.txt   
After a successful connection is made using the chat extension with the   
Connect command, the chat extension sends a dynamically generated page   
to the caller's browser. This page is based on the ChatExtensionFrame.txt   
template file.   
The template file is processed by the chat extension and is delivered with a   
connection ID for the caller. The dynamically generated page is frames-  
based and defines a user input frame, a status line frame, and a chat   
conversation frame. Other frames may be added if needed.   
The user input frame initially references the ChatExtensionNoInput.htm   
page. When an agent takes the call, the chat extension changes the   
reference to a dynamically generated page based on the   
ChatExtensionInput.txt template file.   
The status line frame directly references the Status command. The chat   
extension returns a single line with the current status and time. The   
returned data includes an HTML Refresh command, so that the status line   
requests updates as needed.   
The chat conversation frame directly references the Read command. The   
chat extension returns the conversation data.   
The close frame references a dynamically generated page based on the   
ChatExtensionClose.txt template file.   
HTML Template file   
used during normal   
operation   
ChatExtensionInput.txt   
This template defines a form for sending user entered text to the agent and   
another form for closing the connection to the chat extension. The Write   
ISAPI command is used in this frame.   
HTML file used to   
report error conditions   
ChatExtensionInternalError.ht  
m   
This page is returned if an internal error is detected in the chat extension.   
Currently, this can only occur during Connect if there is not enough memory   
available to create a connection ID for the user.

CCSP APIs Guide - 7.4   
133   
Component   
File   
Description   
HTML file used to   
report error conditions   
ChatExtensionInvalidConnecti  
onID.htm   
This page is returned when the Write, Escalate, or Close APIs are used, and   
the connection ID is not recognized by the chat extension. This should not   
occur during normal use because the Close API causes the connection ID to   
expire.   
HTML file used during   
normal operation   
ChatExtensionNoInput.htm   
This page defines the user entered text input window before the caller is   
actually connected. There is no Send button.   
HTML file used to   
report error   
conditions   
ChatExtensionNotConnected.  
htm   
This page is returned when the Read, Write, Escalate, or Close APIs are used   
when the user is not connected to a chat session using the chat extension.   
Web Server   
ChatExtensionUnsupportedBr  
owser.htm   
This page provides notification about an unsupported browser.   
HTML file used during   
normal operation   
ChatExtensionWrapUp.htm   
This page displays a message or URL after the caller closes the connection.   
JS file   
ChatScrollScript.js   
This script is sent during Chat message refreshes. It is used to put new   
messages at the bottom of the chat conversation frame.   
JS file   
DisconnectedScript.js   
This script is sent once, during status message refreshes, when a disconnect   
is detected. It contains two substitutable parameters. The first, %1, sets the   
location of the input window to the NoInput HTML file (alias of   
ChatExtensionNoInput.htm). The second, %2, sets the location of the   
disconnect frame to the Wrapup HTML file (alias of   
ChatExtensionWrapup.htm).

CCSP APIs Guide - 7.4   
134   
Component   
File   
Description   
JS file   
OpenWindowScript.js   
This script is sent when the user connects. It is used to determine whether   
the chat opens in a new window or in the existing window in which the caller   
placed the call.   
If you want to open a new window, use %1, which provides the reference to   
the method of ChatExtension.dll, which returns the content of the new   
window.   
If you want to use the existing window, use an empty file that just has %2 in   
it. The content of the HTML page is then displayed in the existing window.   
Web Server   
Phrases.txt   
This is the translation file for ChatExtension.dl.   
JS file   
PushUrlScript.js   
This script is sent when the agent pushes a URL to the user. It contains two   
substitutable parameters. The first, %1, is used by the script as the name of   
the window to open, and the second, %2, is used as the URL to direct that   
window to.   
JS file   
RefreshScript.js   
This script is sent during Status message refreshes. It contains no   
substitutable parameters and it causes the main chat window to reload.   
JS file   
RingScript.js   
This script is sent once, during Status message refreshes, when an agent   
has taken the call and the caller needs to be notified. It contains three   
substitutable parameters. The first, %1, and second, %2, are both   
substituted with Ringout.wav so the refresh can audibly signal the browser.   
The third, %3, sets the location of the input frame to the Input HTML file,   
generated from the ChatExtensionInput.txt.

CCSP APIs Guide - 7.4   
135   
Interfaces to the chat extension   
This section contains the following information:   
•   
Access the interfaces   
•   
Connect   
•   
Read   
•   
Status   
•   
Write   
•   
Close   
•   
Registry entries   
•   
Diagnostic function   
Access the interfaces   
The interfaces to the chat extension are accessed in the same way as standard CGI web server   
scripts. For example, to connect, the browser must reference the URL:   
http://server/scripts/chatextension.dll?Connect?   
This is normally accomplished using the generated ASP script.   
If you are not using the generated ASP scripts and are writing your own forms, you have to do this   
through an HTML form. The required parameters for the Connect function are stored in the form   
fields, like the methods described in Options for launching calls, with one exception: user-defined   
parameters require special formatting. You need to supply the parameter count and values   
described below. If you want to pass optional, user-defined parameters, use the following.   
Parameter   
Description   
OptionalParameterCount   
Defines how many OptionalParameters are included, for example:   
<input type="hidden" name="OptionalParameterCount"   
value="2">   
OptionalParametern   
(base n = 0)   
Used for custom parameters, for example:   
<input type="hidden" name="OptionalParameterCount"   
value="2">   
<input type="text" name="OptionalParameter0"   
value="customer\_id,123">   
<input type="text" name="OptionalParameter1"   
value="Ticket,188">

CCSP APIs Guide - 7.4   
136   
For example:   
<html>   
<body>   
<form action="/scripts/chatextension.dll?command=connect" method="GET"   
name="ChatForm">   
<input type="hidden" name="Call\_Center\_Address" value="ics.yoursite.com">   
<input type="hidden" name="Call\_Center\_Name" value="NY Sales Center">   
<input type="hidden" name="Calling\_User\_Skills" value="English">   
<input type="text" name="Calling\_User\_FirstName" value=""> First Name<BR>   
<input type="text" name="Calling\_User\_LastName" value=""> Last Name<BR>   
<input type="text" name="Calling\_User\_HardMessage" value="Sales question">   
Message<br>   
<input type="hidden" name="OptionalParameterCount" value="2">   
<input type="text" name="OptionalParameter0" value="customer\_id,123">   
<input type="text" name="OptionalParameter1" value="Ticket,188">   
<p><input type="submit" value="Start Chat"></p>   
</form>   
</body>   
</html>   
   
This form collects the caller's name and a message to be sent to the agent. The contact center   
address, name, and skill set are established in hidden fields since these are not caller defined.   
If you want to use the POST method, you must change the method and action and add a command   
field as shown below:   
<form method="POST" action="/scripts/ChatExtension.dll">   
<input type="hidden" name="command" value="connect">   
Connect   
The Connect URL uses the form fields listed in the Call parameters section:   
•   
For POST method this is   
path/chatextension.dll?connect   
•   
For GET method this is   
path/chatextension.dll?command=connect   
   
When a successful connection is established, the chat extension returns the contents of the file   
dynamically generated from the ChatExtensionFrame.txt template file to the caller and sends an   
HTML page with an embedded connection ID used to maintain a link between the caller and the   
contact center.

CCSP APIs Guide - 7.4   
137   
Read   
The Read URL is used by the file dynamically generated from the ChatExtensionFrame.txt template   
file to fill in the chat conversation frame. The Read URL uses the connection ID There are no other   
inputs. The chat extension returns an HTML formatted table of the conversation text and a Refresh   
command.   
Status   
The Status URL is dynamically generated from the ChatExtensionFrame.txt template file to fill in the   
status line frame. The Status URL uses the connection ID. There are no other inputs. The chat   
extension returns an HTML formatted table of the status and a Refresh command. This table   
contains a single row.   
Write   
The Write URL uses the connection ID and transmits the content of the form field named text to the   
agent. If there is more than one field named text, both are transmitted. The chat extension returns   
the contents of a file dynamically generated by the ChatExtensionInput.txt template file, which   
effectively clears the caller input.   
Close   
The Close URL is used by a file dynamically generated by the ChatExtensionClose.txt template file   
for the Disconnect button. The Close URL uses the connection ID. There are no other inputs. The   
chat extension returns the contents of the ChatExtensionWrapup.htm file.

CCSP APIs Guide - 7.4   
138   
Registry entries   
For reference only.   
The first time ChatExtension.dll is loaded, to keep functions with files, it adds the following registry   
entries to HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node \CosmoCom   
\ChatExtension\Files:   
   
"OpenWindowScript"="/CosmoCall/OpenWindowScript.js"   
"MainFrame"="/CosmoCall/ChatExtensionFrame.txt"   
"RefreshScript"="/CosmoCall/RefreshScript.js"   
"DisconnectedScript"="/CosmoCall/DisconnectedScript.js"   
"Wrapup"="/CosmoCall/ChatExtensionWrapup.htm"   
"NoInput"="/CosmoCall/ChatExtensionNoInput.htm"   
"RingScript"="/CosmoCall/RingScript.js"   
"Input"="/CosmoCall/ChatExtensionInput.txt"   
"Ringout"="/CosmoCall/Ringout.wav"   
"NotConnected"="/CosmoCall/ChatExtensionNotConnected.htm"   
"InvalidConnection"="/CosmoCall/ChatExtensionInvalidconnectionID.htm"   
"Default"="/CosmoCall/ChatExtensionDefault.htm"   
"InternalError"="/CosmoCall/ChatExtensionInternalError.htm"   
"PushUrlScript"="/CosmoCall/PushUrlScript.js"   
"ChatScrollScript"="/CosmoCall/ChatScrollScript.js"   
Diagnostic function   
CCSP has a diagnostic function built into the chat extension that:   
•   
Allows you to view key registry entries   
•   
Checks that the required files are readable   
•   
Lists basic information about connected users   
   
To display this information in your browser, type http://<path>/chatextension.dll?diagnose? where   
<path> indicates the path to the ChatExtension.dll.

CCSP APIs Guide - 7.4   
139   
Customize the chat window   
This section contains the following information:   
•   
Increasing the user input frame   
•   
Showing caller and agent names   
   
The following diagram shows the components of the chat extension frame.   
   
Increase the user input frame   
By default the user input frame contains a single text entry line. This allows the caller to type text   
and press Enter to send the text. The maximum length of this field is 1024 characters.   
Optionally, you can make the input area a multi-line entry. This allows large chunks of text to be   
copied and pasted into the entry field. The caller then has to click Send to send the text. The Enter   
key does not send text because it is used to move to the next line in the text entry area.   
To make the input area a multi-line entry:   
1.   
Change the following line in the ChatExtensionInput.txt and ChatExtensionNoInput.htm files:   
From:   
document.write("<input type="text/" size=/"size=/"",InputLen,"/"   
maxlength=/"1024/" name=/"text/" style=/"background-image:   
url('/CosmoCall/typehere.bmp')/">");

CCSP APIs Guide - 7.4   
140   
To:   
document.write("<textarea rows=/"2/" name=/"text/" cols=/"40/"   
style=/"background-image: url('/CosmoCall/typehere.bmp')/"></textarea>");   
The textarea rows parameter can be larger than 2.   
2.   
Increase the value of the InputSize parameter in the ChatExtensionFrame.txt file:   
InputSize = 70;   
if (bAppName.indexOf("Netscape") >= 0) InputSize = 85;   
document.write("<frameset rows=/"\*,40,",InputSize,",25/" border=/"0/"   
frameborder=/"no/">");   
The frameset rows parameter controls the left-side of the chat window:   
frameset rows=/"\*,40,",InputSize,",25/"   
In the above example:   
o   
\* = the chat conversation frame   
o   
40 = the Hang Up and Help button frame.   
o   
InputSize = the user input frame (where you type the text to send).   
o   
25 = the status line frame.   
Show caller and agent names   
To configure the chat window to show actual caller and agent names:   
1.   
Open the resource file ..\Inetpub\wwwroot\CosmoCall\<LocaleID>\<TenantName>\Phrases.txt   
The last lines of this file contain the following:   
159 "Caller"   
 Me   
160 "Rep"   
Rep   
2.   
Replace Me and Rep with arbitrary strings that you want to be presented to the caller when a   
chat takes place between the caller and the agent. You can use %1 and %2 to indicate caller or   
agent first and last name.   
The following example displays the full name of the caller but uses Rep instead of the agent   
name:   
159 "Caller"   
%1 %2   
160 "Rep"   
Rep

CCSP APIs Guide - 7.4   
141   
Parameters   
This section contains the following information:   
•   
Options for launching calls   
•   
Call parameters   
•   
ASP script parameters   
   
To launch email and chat calls, a number of parameters must be sent to the ACD server. These   
parameters are case sensitive and need to match what you are using in your URL query, form, or   
cookie as well as what you entered in the Configurator.   
Options for launching calls   
There are three methods you can use to send parameters to the ACD server to launch a call:   
•   
URL query   
•   
Form   
•   
Cookie   
   
You can also automate most of the process. See Automated method.   
URL query   
The query string must look like the following:   
path/call.asp?ret=query&Callertype=HTML&parameter=value&parameter=value...   
The parameters and values are explained in Call parameters. Be sure to match the case because   
the command is case sensitive. If parameters are not found, the defaults from the ASP script are   
used.   
Form   
The form method uses a form to gather caller information. A form must use an HTTP POST.   
The form should contain the following:   
<form method="POST" action="call.asp?ret=form">   
<input type="text" name="parameter" value="value">...   
   
The parameters and values are explained in Call parameters. If parameters are not found, the   
defaults from the ASP script are used.

CCSP APIs Guide - 7.4   
142   
Here is an example of an HTML page that may be submitted:   
<form action="/scripts/call.asp?ret=form" method="POST" name="ChatForm">   
<input type="hidden" name="Call\_Center\_Address" value="ics.yoursite.com">   
<input type="hidden" name="Call\_Center\_Name" value="NY Sales Center">   
<input type="hidden" name="Calling\_User\_Skills" value="English">   
<input type="text" name="Calling\_User\_FirstName" value="> First Name<BR>   
<input type="text" name="Calling\_User\_HardMessage" value="Sales question">   
Message<br>   
<p><input type="submit" value="Start Chat"></p>   
</form>   
Cookie   
The cookie method is best to use when you have customer information already in a database from   
a shopping cart or help desk application or if you can gather information from pages where the   
caller has visited on your website. Cookies can be set on any page of your website. The cookie   
name must be the same as the parameter name. For a complete list of parameter names, see Call   
parameters.   
The cookies can be read by the ASP script:   
call.asp?ret=cookie   
Automated method   
To help you automate much of the process of gathering parameters and passing them to the ACD,   
CCSP provides the following method which can be used on any ASP page:   
1.   
Put the following Include statement on the ASP page where you want to set parameters:   
<!--#Include File="path/setparams.txt"-->   
2.   
Call the Setparams function wherever you want to begin setting a cookie.   
Setparams("parameter","value")   
3.   
Put the following Include statement on your ASP webpage where you want the CCSP logo to   
appear:   
<!--#Include File="path/cookiecaller.txt"-->   
   
This inserts the CCSP logo and link on your webpage, reads the parameters from the cookie, and   
creates a URL query. If parameters are not found, the defaults from the ASP scripts are used.

CCSP APIs Guide - 7.4   
143   
Call parameters   
Call parameters are case sensitive and need to match what you are using in your URL query, form,   
or cookie as well as what you entered in the Configurator.   
Parameter   
Description   
Required   
Call\_Center\_Name   
Logical name of contact center.   
Yes   
Call\_Center\_Address   
ICS IP address or FQDN.   
Yes   
Call\_Center\_QueueName   
Name of queue to place caller in. This name   
must match a queue on the ACD.   
Yes   
Call\_Center\_Port   
Port for ACD server communication.   
Yes   
Calling\_User\_PhoneType   
Phone type (string). Valid values are   
NETMEETING or CHAT.   
Yes   
Calling\_User\_URL   
URL to pass to agent.   
No   
Calling\_User\_Skills   
List of required skills, separated by a plus   
sign (+).   
No   
Calling\_User\_FirstName   
Caller's first name.   
No   
Calling\_User\_LastName   
Caller's last name.   
No   
Calling\_User\_IDMode   
Caller identification.   
No   
Calling\_User\_HardMessage   
Message to send to the agent.   
No   
Calling\_User\_Initial\_URL   
URL to be displayed in the right frame of the   
chat.   
No   
TQoS   
Number of seconds within which the call   
should be answered. Overrides the TQoS   
configured for the queue.   
No   
No\_Agents\_Msg   
Message to send caller when no agents are   
logged into the ACD. Overrides the default   
message for such an event.   
No   
No\_Agents\_URL   
URL to push to caller when no agents are   
logged into the ACD. Overrides the default   
URL for such an event. Must be used with the   
No\_Agents\_Msg field.   
No

CCSP APIs Guide - 7.4   
144   
Parameter   
Description   
Required   
DisplayFrame   
Determines which frame is used to display a   
URL pushed from an agent.   
Caution   
Do not Use \_self, \_parent, or \_top as they may   
wipe away the caller to put the URL in the   
current frame.   
No.   
OriginalEmail   
The email address of the customer who sent   
an email.   
No. For web-  
generated email   
only.   
SupportStaffEmail   
The return address for agents replying to   
emails that come through chats. This field   
overrides the value of the Reply to Address   
field configured in the Message   
Communication System (MCS).   
No. For web-  
generated email   
only.   
MailUID   
Unique ID for email message. Stored in log   
database   
No. For email only.   
Optional parameters   
User-defined parameters can also be passed   
to the ACD Server. The easiest way to define   
these parameters is by using the   
Configurator. If you are not using the   
generated ASP scripts, see Chat extension.   
For information about optional parameters   
used in email, see Send external callbacks via   
SMTP.   
No   
ASP script parameters   
You need to use two special parameters with the ASP scripts:   
1.   
ret= is used to indicate the method being used to send parameters to the ACD server. The   
valid inputs are query, form, and cookie.   
2.   
Callertype=HTML

# Contact Center: Service Provider CCSP UI CSS Customization Guide Version 7.3.5

CCSP-UICCG-735-R2-12/2023   
Contents   
1: Intro .............................................................................................................................................3   
What’s new ................................................................................................................................................. 3   
Legal disclaimer ......................................................................................................................................... 3   
Support ....................................................................................................................................................... 3   
2: CCSP UI CSS structure ................................................................................................................4   
3: CSS customization ......................................................................................................................5   
AdminPoint ................................................................................................................................................. 5   
4: Customization options ...............................................................................................................7   
Complete facelift to the .css files ............................................................................................................ 7   
Customization using custom.less ........................................................................................................... 7   
Specific change to a single CSS property ............................................................................................... 8   
5: Example .......................................................................................................................................9   
Appendix A: Compiling a .less file to a .css file ......................................................................... 10   
Installing the Less compiler ................................................................................................................... 10   
Creating the .css file ................................................................................................................................ 11

CCSP UI CSS Customization Guide - 7.3   
3   
1: Intro   
This document is for the UI designers responsible for customizing the CCSP UI (previously called   
TouchPoint) look and feel by using custom .css files.   
The UI designer must possess working knowledge and skills to adapt colors and styles in CSS   
format, and to compile .less files to .css files.   
What’s new   
Version 7.3.5   
•   
TouchPoint is now called CCSP UI.   
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 content in this document, please open a case with   
Enghouse Support.

CCSP UI CSS Customization Guide - 7.3   
4   
2: CCSP UI CSS structure   
Because CCSP UI is made up of gadgets, the CSS structure is designed so that each gadget has its   
own .css file.   
The styles folder under Server Components\TouchPoint\Content\styles, contains .less and .css files for   
all the system gadgets and functions, such as callBack.less, chat.less, email.less, assist.less, and   
login.less.   
To enable a common look and feel and make the files simpler to use, some .less files contain   
variables. You can find these variables in custom.less.   
For example, the OK button properties are:   
@okButton-color: #7dbe47;   
@okButton-disabled-color: #cad7e6;   
@okButton-hover-color: #a8df7c;   
@okButton-hover-disabled-color: #E4E4E4;   
@okButton-active-color: #649938;   
   
Changing these properties in the custom.less file changes all the OK buttons in CCSP UI.

CCSP UI CSS Customization Guide - 7.3   
5   
3: CSS customization   
You customize the CCSP UI look and feel by loading a customized .css file.   
To do this, you copy the styles folder and edit it. You then use AdminPoint to load a single.css file   
that includes all the required changes.   
Note   
•   
The very first time an agent logs on to CCSP UI, it uses the default login.css. CCSP UI saves the   
tenant ID provided by the agent in local storage on the client PC. When the agent logs on   
again, CCSP UI uses the branded login.less.   
•   
Changes to the LookAndFeel path in AdminPoint affect CCSP UI only when the agent logs on   
again.   
•   
If the LookAndFeel path in AdminPoint is invalid, the default CCSP UI .css file is used.   
•   
There is a delay of a few seconds while the loggedout.js and ClientInstaler.js pages load, during   
which the default blue screen appears.   
AdminPoint   
In AdminPoint, on the Gadgets subtab, click Settings. In the Settings dialog box, in Look and Feel,   
point to the customization .css file.   
The .css file has the same classes and overrides the default CCSP UI design.   
As with any AdminPoint definition, the customization .css file can be set on   
Landlord/Tennant/Group/Roll level and be overridden in any lower level.

CCSP UI CSS Customization Guide - 7.3   
6   
Note   
•   
The path is relative to CosmoCom\Server Components\TouchPoint. To avoid path errors, we   
recommend that you locate your new styles folder under Touchpoint\Content as in the image   
above.   
•   
To import images to your custom stylesheet, copy the folder CosmoCom\Server   
Components\TouchPoint\Content\images to the directory above your work directory. This action   
is required because some of the .less files contain imports from ../images.

CCSP UI CSS Customization Guide - 7.3   
7   
4: Customization options   
   
Note   
•   
A lookAndFeel.less file is provided in the styles folder. This file contains imports for all CCSP UI   
.less files. Compiling this file generates a .css file containing all the elements.   
•   
The logo appears in the login.css and helpAboutTouchPoint.css files. All references are included   
in the logo.less file.   
   
The customization options are:   
•   
Complete facelift to the .css files — use this option for a large scale customization that involves   
changes in many .css files.   
•   
Customization using custom.less — use this option to change any of the common properties   
defined in the custom.less file that will affect all references to this property.   
•   
Specific change to a single CSS property — use this option for a very specific change to any of   
the properties in one of the .less files used by CCSP UI.   
Complete facelift to the .css files   
To perform a complete facelift to the .css files:   
1.   
Copy all files in the styles folder to a working directory.   
2.   
Change any of the .less files.   
3.   
Compile lookAndFeel.less to a .css file to generate the customization .css file. (See Appendix A:   
Compiling a .less file to a .css file.)   
4.   
Set the LookAndFeel property in AdminPoint to point to the compiled lookAndFeel.css file.   
Customization using custom.less   
To perform a customization using custom.less:   
1.   
Copy all files in the styles folder to a working directory.   
2.   
Change custom.less. For example, change a background color or the OK or Cancel buttons.   
3.   
Compile lookAndFeel.less to a .css file because the variables may be included in various .less files   
that now also require a recompilation.   
4.   
Set the LookAndFeel property in AdminPoint to point to the compiled lookAndFeel.css file.

CCSP UI CSS Customization Guide - 7.3   
8   
Specific change to a single CSS property   
Note   
Every .less file change requires .css compilation of the changed file and all affected files. For   
example, if you change a Less variable, you must compile all files in which the variable is used.   
   
To perform a specific change to a single CSS property:   
1.   
Copy the .less file you want to modify from the styles folder to a working directory.   
2.   
Modify the .less file as required.   
3.   
Compile the modified .less to .css file to generate the customization .css file. (See Appendix A:   
Compiling a .less file to a .css file.)   
4.   
Set the LookAndFeel property in AdminPoint to point to the compiled .css file.

CCSP UI CSS Customization Guide - 7.3   
9   
5: Example   
To customize the OK button using custom.less:   
1.   
Copy the styles folder to <cssFolder> under   
 ..\CosmoCom\Server Components\TouchPoint\Content.   
2.   
In custom.less, change @okButton-color to another color, for example:   
@okButton-color: blue;   
3.   
Compile lookAndFeel.less to a .css file.   
4.   
Upload this file to AdminPoint.   
   
Before:   
   
 After:

CCSP UI CSS Customization Guide - 7.3   
10   
Appendix A: Compiling a .less file to a   
.css file   
You can use the free Less compiler to compile a .less file to a .css file. After installing node.js, you can   
install node.js modules via Node Package Manage (NPM).   
Installing the Less compiler   
To install the Less compiler:   
1.   
Install node.js from https://nodejs.org/en/.   
2.   
Install Less globally by opening a cmd window as administrator and running the following   
command:   
$ npm install –g less

CCSP UI CSS Customization Guide - 7.3   
11   
Creating the .css file   
To create the .css file:   
1.   
Open npm cmd as administrator on your CSS custom folder.   
2.   
Create the css file by running the following command:   
lessc lookAndFeel.less lookAndFeel.css --relative-urls -–js   
   
A lookAndFeel.css file is created in the same folder.

# Contact Centre: Service Provider CCSP UI Integrations Guide Version 7.3 and 7.4

CCSP-UIIG-73/74-R3-02/2024   
Contents   
1: Intro .............................................................................................................................................7   
What’s new ................................................................................................................................................. 7   
Legal disclaimer ......................................................................................................................................... 7   
Support ....................................................................................................................................................... 7   
2: Overview .....................................................................................................................................8   
3: Events ..........................................................................................................................................9   
Overview ..................................................................................................................................................... 9   
Publishing and subscribing to events .................................................................................................. 10   
window.app.PubSub ....................................................................................................................... 11   
eventServiceWrapper ..................................................................................................................... 13   
subscriberBase ................................................................................................................................ 13   
Event lists .................................................................................................................................................. 14   
Server events ................................................................................................................................... 14   
UI event list ...................................................................................................................................... 14   
UI command list .............................................................................................................................. 18   
Custom events ......................................................................................................................................... 22   
Custom CCSP UI Connector events .............................................................................................. 22   
4: CCSP UI AngularJS services .................................................................................................... 23   
agentServiceWrapper ............................................................................................................................. 23   
Properties ......................................................................................................................................... 23   
Functions .......................................................................................................................................... 23   
customCSSService ................................................................................................................................... 24   
Properties ......................................................................................................................................... 24   
Functions .......................................................................................................................................... 24   
DALService ................................................................................................................................................ 25   
Properties ......................................................................................................................................... 25   
Functions .......................................................................................................................................... 25   
dialogService ............................................................................................................................................ 26   
Properties ......................................................................................................................................... 26   
Functions .......................................................................................................................................... 26   
dialService ................................................................................................................................................. 27   
Properties ......................................................................................................................................... 27   
Functions .......................................................................................................................................... 27   
localSettingsService ................................................................................................................................ 30   
Properties ......................................................................................................................................... 30   
Functions .......................................................................................................................................... 30

CCSP UI Integrations Guide - 7.3 and 7.4   
3   
loggerService ........................................................................................................................................... 32   
Properties ......................................................................................................................................... 32   
Functions .......................................................................................................................................... 32   
oidcService ................................................................................................................................................ 33   
Properties ......................................................................................................................................... 33   
Functions .......................................................................................................................................... 33   
translationService .................................................................................................................................... 38   
Properties ......................................................................................................................................... 38   
Functions .......................................................................................................................................... 38   
viewModelStoreService .......................................................................................................................... 40   
Properties ......................................................................................................................................... 40   
Functions .......................................................................................................................................... 40   
voipService................................................................................................................................................ 41   
Properties ......................................................................................................................................... 41   
Functions .......................................................................................................................................... 41   
5: Gadgets .................................................................................................................................... 44   
Overview ................................................................................................................................................... 44   
Example gadgets ............................................................................................................................ 44   
What is a gadget? ........................................................................................................................... 44   
Gadget capabilities ......................................................................................................................... 44   
Gadget types ................................................................................................................................... 45   
Gadget creation ....................................................................................................................................... 46   
Standard or functional gadget?.................................................................................................... 46   
Demonstration gadgets ................................................................................................................. 46   
CCSP AdminPoint ............................................................................................................................ 47   
Adding the Demo Chat Prompt functional gadget .................................................................... 47   
Adding the Demo CRM standard gadget .................................................................................... 48   
Adding the gadgets to a role ........................................................................................................ 48   
Testing the gadgets ........................................................................................................................ 49   
Gadget refresh ................................................................................................................................ 49   
Gadget properties ................................................................................................................................... 50   
Default properties ........................................................................................................................... 50   
Adding custom properties ............................................................................................................. 53   
Standard gadget development ............................................................................................................. 54   
Standard gadget structure ............................................................................................................ 54   
Standard gadget as an AngularJS module .................................................................................. 55   
Functional gadget development ........................................................................................................... 59   
Functional gadget code structure ................................................................................................ 59   
Using CCSP UI services and objects ...................................................................................................... 61   
Subscribing to events ..................................................................................................................... 61   
Publishing events ............................................................................................................................ 62   
Accessing the Agent object ........................................................................................................... 63

CCSP UI Integrations Guide - 7.3 and 7.4   
4   
Making calls ..................................................................................................................................... 65   
General call handling ..................................................................................................................... 66   
Transferring calls ............................................................................................................................ 70   
Attended transfer and conferencing ........................................................................................... 80   
Multiple call handling ..................................................................................................................... 83   
Chat call handling ........................................................................................................................... 87   
Email call handling .......................................................................................................................... 92   
Wrapping up calls ......................................................................................................................... 107   
Raising alerts and prompts ......................................................................................................... 108   
System gadget customization ............................................................................................................. 110   
AngularJS Directives ..................................................................................................................... 110   
Overriding CCSP UI directives ..................................................................................................... 110   
Overriding CCSP UI HTML only ................................................................................................... 112   
Overriding CCSP UI CSS ............................................................................................................... 113   
Location of CCSP UI directives .................................................................................................... 115   
Managing Release updates to CCSP UI directives ................................................................... 115   
Reading ConfigItem data ..................................................................................................................... 116   
Example ConfigItem ..................................................................................................................... 116   
Helper Functions ........................................................................................................................... 118   
Extracting data from a ConfigItem ............................................................................................ 121   
Standard gadget properties ................................................................................................................ 125   
Accessing properties of a standard gadget .............................................................................. 125   
Adding an example custom property ........................................................................................ 126   
Accessing the custom property in the code ............................................................................. 126   
Overriding a custom property .................................................................................................... 126   
Accessing the custom property object in normal JavaScript .................................................. 126   
windowObject.Gadget properties .............................................................................................. 126   
Gadget instance properties ......................................................................................................... 131   
Standard gadget properties example ....................................................................................... 132   
Changing the default values of the gadget properties ........................................................... 133   
Changing the gadget's default size and position as a floating window .............................. 134   
Changing the gadget's default size and position as a popout window ............................... 135   
Changing the gadget's appearance .......................................................................................... 135   
Changing the title of a gadget instance .................................................................................... 141   
Functional gadget properties .............................................................................................................. 142   
Using custom properties in a functional gadget ..................................................................... 142   
Disabling a functional gadget ..................................................................................................... 144   
Accessing other gadgets and windows .............................................................................................. 145   
Using the GadgetComm object .................................................................................................. 145   
GadgetComm functions ............................................................................................................... 145   
Inter-gadget access example ...................................................................................................... 149   
Using the Controller URL property............................................................................................. 158   
Persisting data between gadgets and logins .................................................................................... 158   
CCSP UI and web storage ............................................................................................................ 159

CCSP UI Integrations Guide - 7.3 and 7.4   
5   
Using viewModelStoreService ..................................................................................................... 159   
Using DALService and localSettingsService .............................................................................. 162   
Saving and loading gadget properties ...................................................................................... 167   
6: Migrating 6.1.X Web Agent integrations ............................................................................. 169   
Legacy Integrations.js in CCSP UI ....................................................................................................... 169   
Integrations gadget structure .................................................................................................... 169   
Adding a 6.1.X integration to CCSP UI ................................................................................................ 171   
Adding the Integrations functional gadget .............................................................................. 171   
Adding the Integrations standard gadget ................................................................................ 172   
6.1.X Integration.js script migration .......................................................................................... 172   
Examples ........................................................................................................................................ 175   
Troubleshooting ............................................................................................................................ 175   
7: Co-browsing integration ...................................................................................................... 176   
Overview ................................................................................................................................................. 176   
Concepts ................................................................................................................................................. 176   
Co-browsing session lifecycle .............................................................................................................. 176   
CCSP co-browsing support .................................................................................................................. 178   
Co-bowsing gadget components ............................................................................................... 179   
CoBrowsingAPI interface ..................................................................................................................... 180   
Session functions .......................................................................................................................... 180   
Gadget controls functions ........................................................................................................... 180   
Initialization functions .................................................................................................................. 181   
Events ............................................................................................................................................. 182   
Writing co-browsing integrations ....................................................................................................... 183   
Configuring the gadget ............................................................................................................... 183   
CoBrowsingImplementation object ........................................................................................... 183   
Implementing the session functions ......................................................................................... 184   
CreateSession implementation .................................................................................................. 184   
Exiting a session ............................................................................................................................ 186   
SwitchController implementation .............................................................................................. 187   
GoToURL implementation ........................................................................................................... 187   
Implementing the gadget control functions ............................................................................ 187   
Raising events ............................................................................................................................... 188   
Co-Browsing gadget properties .......................................................................................................... 189   
Accessing the gadget properties ................................................................................................ 190   
Examples ................................................................................................................................................. 190   
Template ........................................................................................................................................ 190   
Test .................................................................................................................................................. 190   
Surfly ............................................................................................................................................... 191   
Temasys Skylink ............................................................................................................................ 191   
Points to note ......................................................................................................................................... 192

CCSP UI Integrations Guide - 7.3 and 7.4   
6   
8: Screen Recording gadget implementation override ......................................................... 193   
Overview ................................................................................................................................................. 193   
Screen Recording gadget architecture .............................................................................................. 193   
Screen Recording gadget interface .................................................................................................... 195   
Functions ........................................................................................................................................ 195   
Constants ....................................................................................................................................... 196   
Example .......................................................................................................................................... 197   
Screen recording sequence ................................................................................................................. 197   
Screen recording default implementation ........................................................................................ 198   
Overiding the screen recording implementation ............................................................................. 200   
Example .......................................................................................................................................... 201   
Points to note ......................................................................................................................................... 201   
9: Single Sign On Client development ..................................................................................... 202   
Overview ................................................................................................................................................. 202   
Relevance ................................................................................................................................................ 202   
Scope ....................................................................................................................................................... 202   
General SSO concepts ........................................................................................................................... 202   
Single Sign On (SSO) .................................................................................................................... 202   
Auth Server .................................................................................................................................... 202   
Resource Client ............................................................................................................................. 202   
Resource Server ............................................................................................................................ 203   
Access token .................................................................................................................................. 203   
Refresh token ................................................................................................................................ 203   
OpenID Connect (OIDC)............................................................................................................... 203   
Authorization code grant ............................................................................................................. 203   
Refresh token grant ...................................................................................................................... 203   
CCSP UI SSO Concepts .......................................................................................................................... 204   
Basic architecture ......................................................................................................................... 204   
Detailed architecture .................................................................................................................... 205   
CCSP UI Gadget with SSO ..................................................................................................................... 207   
Example gadget with OIDC ......................................................................................................... 208   
Main gadget script (demoSSOGadget.js) .................................................................................. 208   
Gadget OIDC Page (demoOIdc.html) ........................................................................................ 209   
Gadget OIDC Class (demoOidc.js) .............................................................................................. 210

CCSP UI Integrations Guide - 7.3 and 7.4   
7   
1: Intro   
This document is for engineers and developers who need to integrate third-party applications with   
CCSP UI (previously called TouchPoint). A working knowledge of HTML and JavaScript is required.   
Experience with AngularJS is also beneficial.   
What’s new   
Version 7.4   
•   
TouchPoint is now called CCSP UI.   
•   
dialService — changed communicatorContacts description to Teams or your custom   
application.   
•   
oidcService — added this service.   
•   
9: Single Sign On Client development — added this chapter.   
Version 7.3   
•   
Removed references to Internet Explorer.   
•   
Replaced window.onload with angular.element(document).ready.   
•   
8: Screen Recording gadget implementation override — added new chapter.   
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 content in this document, please open a case with   
Enghouse Support.

CCSP UI Integrations Guide - 7.3 and 7.4   
8   
2: Overview   
CCSP UI provides customization features that allow it to be greatly extended and modified by third   
parties. The available integration methods are:   
•   
Integration.js (legacy)   
•   
Ajax API   
•   
Gadgets   
•   
CCSP UI Connector   
   
The following table summarizes the integration methods.   
   
Integration.js   
Ajax API   
Gadgets   
Connector   
Language   
JavaScript   
JavaScript   
JavaScript   
C#/(JavaScript)   
AngularJS   
Yes   
Yes   
Yes   
No   
UI Events   
Yes   
No   
Yes   
No   
AngularJS   
Services access   
Yes   
No   
Yes   
No   
Access to Agent   
object   
Yes   
Yes   
Yes   
No   
Access to Call   
object   
Yes   
Yes   
Yes   
No   
Documentation   
CCSP UI   
Integrations   
Guide   
CCSP APIs Guide   
/CCSP Agent Ajax   
API Reference   
CCSP UI   
Integrations   
Guide   
CCSP UI   
Connector Guide   
   
Table 1: Integration methods summary   
This document provides a reference point for Professional Services, Customer Services, third-party   
engineers and other interested parties who need to add business logic, custom functionality, and   
displays to CCSP UI to integrate it with other applications.

CCSP UI Integrations Guide - 7.3 and 7.4   
9   
3: Events   
This chapter contains the following information:   
•   
Overview   
•   
Publishing and subscribing to events   
•   
Event lists   
•   
Custom events   
Overview   
CCSP UI is an Agent GUI written as an AngularJS web app application. It uses the CCSP AJAX API to   
communicate with the Agent Web Server to log in to the Contact Center as an agent or supervisor,   
make and receive calls, view reports, and perform other tasks as required. The interaction between   
CCSP UI, the AJAX API, and Agent WS is driven by requests and events as shown in a simplified form   
in the following diagram.   
   
Figure 1: Interaction between CCSP UI, AJAX, and AgentWS   
CCSP UI uses the AJAX API to make requests to the AgentWS Web Services. Responses to these   
requests are received from AgentWS either synchronously or more typically asynchronously in the   
form of events. The events are distributed to the CCSP UI implementations of the AJAX API   
EventsHandlers and broadcast to the rest of the CCSP UI application using eventServiceWrapper.   
Some events may be handled directly by the Interface classes themselves. For example, the Agent   
class defines default implementations of the LoginSuccessHandler and LoginErrorHandler to

CCSP UI Integrations Guide - 7.3 and 7.4   
10   
process the LoginSuccess and LoginError events respectively. However, the event is still passed to   
the CCSP UI implementation of the LoginSuccessHandler and LoginErrorHandler.   
Publishing and subscribing to events   
There are two objects that can be used to publish and subscribe to events and one higher-level   
object that can be used to manage event subscriptions:   
•   
window.app.PubSub: this is the lowest level object and provides the core functions for   
publishing and subscribing to events. The event-handling functionality is pure JavaScript for it   
to be included in non-AngularJS pages. The CCSP UI app creates one instance of this object.   
•   
eventServiceWrapper: this is an AngularJS service that wraps and abstracts the   
window.app.PubSub object. It is designed to be injected as a service into controllers and   
directives and provides no functionality over and above the window.app.PubSub functions.   
•   
SubscriberBase: this is a JavaScript object that provides simplified management of   
subscriptions. Each subscription to an event should be unsubscribed when it is no longer   
required, otherwise a callback function can be erroneously invoked multiple times. The   
SubscriberBase object keeps a record of all subscriptions and provides a convenience function   
to clear them all in one call.

CCSP UI Integrations Guide - 7.3 and 7.4   
11   
window.app.PubSub   
This object contains the following functions.   
clearSubscriptions   
Stop   
This function must never be called by customer integrations.   
   
Purpose   
Removes all subscriptions. Not for use by integrations   
Signature   
clearSubscriptions()   
Parameter   
Type   
Description   
-   
-   
-   
Returns   
-   
-   
getSubscriptionsForEvent   
Purpose   
Returns an array of subscription Ids for the given event and if specified the   
given context   
Signature   
getSubscriptionsForEvent(event, context)   
Parameter   
Type   
Description   
event   
string   
Event name   
context   
string   
Context, for example, call Id or null if not required   
Returns   
Array   
List of subscription Ids for the given event or null if none found   
NewEvent   
Purpose   
Publishes an event with the given context and data and automatically   
outputs a message using the loggerService   
Signature   
NewEvent(event, severity, logMessage, parameters, context)   
Parameter   
Type   
Description   
event   
string   
Event name   
severity   
string   
Warning level of the output. Possible values are defined in Global.js   
in the CCSP APIs Guide   
logMessage   
string   
Text to be output   
parameters   
Any   
Null or any valid JavaScript type or object   
context   
string   
Context, for example, call Id or null if not required   
Returns   
-   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
12   
publish   
Purpose   
Publishes an event with the given context and data   
Signature   
publish(event, context, data)   
Parameter   
Type   
Description   
event   
string   
Event name   
context   
string   
Context, for example, call Id or null if not required   
data   
Any   
Null or any valid JavaScript type or object   
Returns   
-   
-   
subscribe   
Purpose   
Subscribes to a given event name with optional context   
Signature   
subscribe(event, context, callback, priority)   
Parameter   
Type   
Description   
event   
string   
Event name   
context   
string   
Context, for example, call Id or null if not required   
callback   
function   
Callback to invoke when the event is raised. Cannot be null   
priority   
integer   
Priority of this subscriber when event raised with 0 being   
highest   
Returns   
string   
GUID string   
unsubscribe   
Purpose   
Unsubscribes from a given event name with optional context   
Signature   
unsubscribe(subscriptionId)   
Parameter   
Type   
Description   
event   
string   
Event name   
context   
string   
Context, for example, call Id or null if not required   
Returns   
-   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
13   
eventServiceWrapper   
See window.app.PubSub for functions.   
subscriberBase   
This object contains the following functions.   
addSubscription   
Purpose   
Subscribes to a given event name with optional context and stores the   
subscription Id   
Signature   
addSubscription(event, context, callback, priority)   
Parameter   
Type   
Description   
event   
string   
Event name   
context   
string   
Context, for example, call Id or null if not required   
callback   
function   
Callback to invoke when the event is raised. Cannot be null   
priority   
integer   
Priority of this subscriber when event raised with 0 being   
highest   
Returns   
-   
-   
clearSubscriptions   
Purpose   
Removes all subscriptions held by this object. If a context is supplied, removes   
only those subscriptions for that context   
Signature   
clearSubscriptions(context)   
Parameter   
Type   
Description   
context   
string   
Context, for example, call Id or null if not required   
Returns   
-   
-   
removeSubscription   
Purpose   
Unsubscribes from the given event with the optional context   
Signature   
removeSubscription(event, context)   
Parameter   
Type   
Description   
event   
string   
Event name   
context   
string   
Context, for example, call Id or null if not required   
Returns   
-   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
14   
Event lists   
CCSP UI works with three types of events: Server events, UI events, and Client Services events.   
Server events are events that have been received from the AgentWS through the Ajax API. See the   
CCSP APIs Guide for a description of each event and its parameters.   
Note   
Each event is published within CCSP UI using eventServiceWrapper and the parameters are the   
properties of the event object, except where stated.   
   
UI events and commands are raised and consumed by the CCSP UI functionality in response to   
user action or triggered by a server event. They can be used to open and close window. These are   
mainly for internal use.   
Connector events are raised and consumed by Client Services applications such as Client Tray App,   
Communicator App, and Screen Recording. These are received and sent by CCSP UI through the   
CCSP UI Connector and control specific functionality within CCSP UI.   
Server events   
All events received from the server are described in the CCSP Agent Ajax API Reference.   
Stop   
These events must not be published by integrations or custom gadgets.   
UI event list   
The following events are available for integrations to subscribe to.   
Stop   
These events must not be published by integrations or custom gadgets.   
callMenuOpenOrClose   
Trigger   
The call menu has been opened or closed   
Context   
null   
Data   
Object:   
{   
callMenuVisible (Boolean)   
callMenuWidth (integer)   
gadgetoryWidth (integer)   
}

CCSP UI Integrations Guide - 7.3 and 7.4   
15   
ChatEnded   
Trigger   
A chat call is destructed   
Context   
null   
Data   
Call object   
ChatStarted   
Trigger   
A chat call goes to the "InCall" state or when monitoring of a chat call begins   
Context   
null   
Data   
Call object   
Boolean: indicates that the new chat window grabs the focus   
focusCall   
Trigger   
Call has been selected as the active call card.   
Context   
Call Type   
Data   
Call Id   
popoutWindow\_Close   
Trigger   
A pop-out window has been closed   
Context   
null   
Data   
Window Id   
popoutWindow\_CloseOnRemoteLogout   
Trigger   
A pop-out window is closed due to a remote logout   
Context   
null   
Data   
Window Id   
popoutWindow\_Open   
Trigger   
A pop-out window is opened   
Context   
null   
Data   
Window Id

CCSP UI Integrations Guide - 7.3 and 7.4   
16   
popoutWindow\_Popin   
Trigger   
A pop-out window is docked into the main window   
Context   
null   
Data   
Window Id   
windowClosing   
Trigger   
A window is being closed   
Context   
null   
Data   
Window Id   
wrapupWindowClosing   
Trigger   
The WrapUp window is closed   
Context   
Call Id   
Data   
null   
wrapupWindowEnded   
Trigger   
The WrapUp window is closed and its data is being submitted   
Context   
Call Id   
Data   
Object:   
{   
selectedWrapupCode (integer),   
note (string),   
PDCallbackTime (Date object),   
scheduleCallback (Boolean)   
}   
WrapupWindowIsOpened   
Trigger   
The WrapUp window is being opened   
Context   
Call Id   
Data   
null

CCSP UI Integrations Guide - 7.3 and 7.4   
17   
wrapupWindowSaving   
Trigger   
The WrapUp information is being submitted to the server   
Context   
Call Id   
Data   
Object:   
{   
selectedWrapupCode (integer),   
note (string),   
PDCallbackTime (Date object),   
scheduleCallback (flag)   
}

CCSP UI Integrations Guide - 7.3 and 7.4   
18   
UI command list   
The following events are available for publishing by integrations   
closeChatApp   
Description   
Closes the Chat window   
Context   
null   
Data   
null   
GetTranslations   
Purpose   
Requests the translation data for the language currently loaded for CCSP UI   
Context   
null   
Data   
null   
GoCallBack   
Purpose   
Displays the Schedule Callback window   
Context   
Call Id   
Data   
null   
GoCallTransfer   
Purpose   
Displays the Call Transfer window   
Context   
Call Id   
Data   
null   
GoRecord   
Purpose   
Toggles the call recording   
Context   
Call Id   
Data   
null   
GoWebCollab   
Purpose   
Starts the Web Collaboration   
Context   
Call Id   
Data   
null

CCSP UI Integrations Guide - 7.3 and 7.4   
19   
GoWrapUp   
Purpose   
Displays the WrapUp window   
Context   
Call Id   
Data   
null   
LogoutAgent   
Purpose   
Starts the Logout process   
Context   
null   
Data   
null   
openAboutTouchPoint   
Purpose   
Opens the About CCSP UI window   
Context   
null   
Data   
null   
openAssistApp   
Purpose   
Displays the Assist window   
Context   
null   
Data   
null   
openChangePassword   
Purpose   
Displays the Change Password window   
Context   
null   
Data   
null   
openChatApp   
Purpose   
Displays the Chat window   
Context   
null   
Data   
null

CCSP UI Integrations Guide - 7.3 and 7.4   
20   
openDefer   
Description   
Opens the Defer window   
Context   
null   
Data   
data   
{   
 call: Call object   
}   
OpenNewCallDialog   
Description   
Displays the Outgoing Call window   
Context   
null   
Data   
null   
openPreferences   
Description   
Displays the Preferences window   
Context   
null   
Data   
null   
openReportProblem   
Description   
Displays the Report Problem window   
Context   
null   
Data   
null   
openSupervisorApp   
Description   
Displays the Supervisor window. If the Supervisor window is already open,   
specifying an agent Id causes the window to display the page that contains   
the given agent   
Context   
null   
Data   
null   
OR   
Object   
{   
Id : Agent Config Id   
}

CCSP UI Integrations Guide - 7.3 and 7.4   
21   
openTouchPointHelp   
Description   
Displays the CCSP UI Help window   
Context   
null   
Data   
null   
ShowEmailCall   
Description   
Displays an email call in a new window   
Context   
null   
Data   
data   
{   
call: Call object   
windowId: GUID string   
}   
showReplayPlayerWindow   
Description   
Displays the Replayer/Player window   
Parameters:   
null

CCSP UI Integrations Guide - 7.3 and 7.4   
22   
Custom events   
The events that are subscribed to and published by third-party integrations are not limited to those   
in the above lists or the ones raised by the AgentWS. Any integration or gadget can raise its own set   
of events, simply by using the publish function. Conversely, an integration or gadget can subscribe   
to any of these custom events. This can enable, for example, a third-party application running on   
the user's desktop to publish an event to CCSP UI that only a custom gadget within CCSP UI is   
subscribed to. Within CCSP UI, third-party gadgets can communicate with each other using custom   
events.   
Custom CCSP UI Connector events   
To add a cusom event that is published to or received over the CCSP UI Connector, edit the file   
C:\Program Files (x86)\CosmoCom\Server Components\TouchPoint\Connector\   
api\connectorapi.events.js.   
This file contains two arrays:   
•   
ConnectorToTouchPointEvents for events received from the CCSP UI Connector.   
•   
TouchPointToConnectorEvents for events published to the CCSP UI Connector.   
   
Add the names of the custom events to the appropriate array.   
Note   
You don’t need to specify the additional null element, unless the event can be specific to an   
instance of CCSP UI in a terminal services context.

CCSP UI Integrations Guide - 7.3 and 7.4   
23   
4: CCSP UI AngularJS services   
This chapter describes key AngularJS services that are available in CCSP UI to perform day-to-day   
tasks such as dialing numbers, transferring calls, and publishing and subscribing to events. There   
are many more services used by CCSP UI than those documented here, however the majority are   
for internal use and should not be used by integrations without specific guidance from CCSP   
Customer Service.   
agentServiceWrapper   
This service contains the Agent object representing the logged-in user. Injecting this service into a   
controller or directive provides access to all the properties and functions of the Agent class that are   
described in the CCSP Agent AJAX API Reference.   
Properties   
agent   
Type: Object   
Description: Represents the logged in agent and implements the AJAX API Agent interface class.   
For details of properties and methods, see the web page in the CCSP Agent AJAX API Reference.   
Functions   
None

CCSP UI Integrations Guide - 7.3 and 7.4   
24   
customCSSService   
This service enables additional and modified CSS files to be loaded and applied to the CCSP UI. It is   
used internally by the CCSP UI application to load the file defined for the Settings gadget Look and   
Feel property that contains customized CSS styles. However, you can use it to load additional or   
modified CSS files at any point.   
Properties   
None   
Functions   
reloadCustomCSS   
Purpose   
Load styles contained in a given file and apply to the CCSP UI application   
Signature   
reloadCustomCSS(customCssPath)   
Parameter   
Type   
Description   
customCssPath   
String   
URL of the CSS file to load. Can be relative to the server   
(/TouchPoint/newCSS/newStyles.css) or absolute

CCSP UI Integrations Guide - 7.3 and 7.4   
25   
DALService   
DAL is for Data Access Layer and the service is used to write data to and retrieve data from the   
browser's web storage (see CCSP UI and web storage). Data is stored as a key-value pair, so the key   
used must be unique. The data stored can be in any form including objects and arrays.   
Properties   
None   
Functions   
DeleteValue   
Purpose   
Deletes the data referenced by the given key from local storage   
Signature   
DeleteValue(key)   
Parameter   
Type   
Description   
key   
String   
Unique identifier of the data value to be deleted   
GetValue   
Purpose   
Retrieves the data referenced by the given key from local storage   
Signature   
GetValue(key)   
Parameter   
Type   
Description   
key   
String   
Unique identifier of the data to be retrieved   
SetValue   
Purpose   
Saves the data with the given key in local storage   
Note   
Calling this function with a null data value is the same as calling DeleteValue.   
   
Signature   
SetValue(key, value)   
Parameter   
Type   
Description   
key   
String   
Unique identifier of the data value to be stored   
value   
Any   
The data to be stored

CCSP UI Integrations Guide - 7.3 and 7.4   
26   
dialogService   
This service is used to output messages, alerts and prompts to the user.   
Properties   
None   
Functions   
closeModal   
Purpose   
Closes the dialog box   
Signature   
closeModal()   
Parameter   
Type   
Description   
-   
-   
-   
showMessageBox   
Purpose   
Displays an alert box to the user with the title and message specified.   
Can be modified on display to present different button configurations   
and accepts a callback function for these buttons   
Signature   
showMessageBox(title, message, messageBoxButtons, resultAction,   
opts)   
Parameter   
Type   
Description   
title   
string   
Text to be displayed in the message box title bar   
message   
string   
Text to be output in the main area of the message box   
messageBoxButtons   
enum   
Button configuration to display. Enum is   
window.enghouse.enums.MessageBoxButtons which has   
the values:   
•   
OK   
•   
OKCancel   
•   
YesNo   
•   
YesNoCancel   
resultAction   
function   
Function to call when a button is pressed. The signature   
must match function(result) where result is a string with the   
possible values:   
•   
"ok"   
•   
"cancel"   
•   
"yes"   
•   
"no"

CCSP UI Integrations Guide - 7.3 and 7.4   
27   
opts   
obj  
ect   
Further options   
dialService   
This service is used to dial out or transfer a voice call to a given number, agent, or system entry   
point.   
Properties   
None   
Functions   
Dial   
Purpose   
Makes a new voice call to the given target entity   
Signature   
Dial(type, name, target, needParam, parameterValue, configId, attendable,   
call)   
Parameter   
Type   
Description   
type   
string   
Call destination type. Possible values:   
•   
"callback" — callback   
•   
"communicatorContacts" — Teams or your custom application   
•   
"dialpad" — number   
•   
"incoming" — equivalent to "dialpad"   
•   
"speedDial" — speed dial entry   
•   
"system" — system entry point   
•   
"users" — agent   
•   
"voicemail" — dial agent's voicemail   
name   
string   
Display name of the destination. For dial type:   
•   
"callback" — use caller name, if available   
•   
"communicatorContacts" — use contact display name   
•   
"dialpad" — use callee name   
•   
"incoming" — use ""   
•   
"speedDial" — use speed dial name   
•   
"system" — use entry point   
•   
"users" — use agent name   
•   
"voicemail" — use "Voicemail"   
target   
string   
The value to be dialed: For dial type:   
•   
"callback" — use callback number

CCSP UI Integrations Guide - 7.3 and 7.4   
28   
•   
"communicatorContacts" — use SIP URI   
•   
"dialpad" — use number   
•   
"incoming" — use number   
•   
"speedDial" — use speed dial number   
•   
"system" — use system entry point config Id   
•   
"users" — use Agent Config Id   
•   
"voicemail" — use agent's private voicemail number   
needParam   
string   
Indicates that a parameter can be specified. This applies to the   
"system" dial type only   
parameterValue   
string   
Extra data that can be associated with the call. For dial type:   
•   
"communicatorContacts" — the contact Skye for Business   
presence   
•   
"system" — the parameter accepted by the destination script   
•   
"users" — the destination agent's state   
attendable   
string   
Call is attendable. Only applies for "system" dial type   
call   
object   
If non-null, the dialing action is performed from this object instead   
of the agent object   
Transfer   
Purpose   
Performs a blind transfer of a chat or email call to the given target. Do not   
use this for VoIP calls.   
Note   
This only supports blind transfers to another agent or to System. The only   
parameters that are used are type, target, parameterValue, and call. All   
other values should be null.   
   
Signature   
Dial(type, name, target, needParam, parameterValue, configId, attendable,   
call)   
Parameter   
Type   
Description   
type   
string   
Call destination type. Possible values:   
•   
"callback" — Callback   
•   
"communicatorContacts" — Teams or your custom application   
•   
"dialpad" — number   
•   
"incoming" — equivalent to "dialpad"   
•   
"speedDial" — speed dial entry   
•   
"system" — system entry point   
•   
"users" — agent   
•   
"voicemail" — dial agent's voicemail

CCSP UI Integrations Guide - 7.3 and 7.4   
29   
name   
string   
Display name of the destination. For dial type:   
•   
"callback" — use caller name, if available   
•   
"communicatorContacts" — use contact display name   
•   
"dialpad" — use ""   
•   
"incoming" — use ""   
•   
"speedDial" — use speed dial name   
•   
"system" — use entry point   
•   
"users" — use agent name   
•   
"voicemail" — use "Voicemail"   
target   
string   
The value to be dialed: For dial type:   
•   
"callback" — use callback number   
•   
"communicatorContacts" — use SIP URI   
•   
"dialpad" — use number in tel: format   
•   
"incoming" — use number in tel: format   
•   
"speedDial" — use speed dial number   
•   
"system" — use system entry point config Id   
•   
"users" — use Agent Config Id   
•   
"voicemail" — use agent's private voicemail number   
needParam   
string   
Indicates that a parameter can be specified. This applies to the   
"system" dial type only   
parameterValue   
string   
Extra data that can be associated with the call. For dial type:   
•   
"communicatorContacts" — the contact Skye for Business   
presence   
•   
"system" — the parameter accepted by the destination script   
•   
"users" — the destination agent's state   
attendable   
string   
Call is attendable. Only applies for "system" dial type   
call   
object   
The dialing action is performed from this object

CCSP UI Integrations Guide - 7.3 and 7.4   
30   
localSettingsService   
This service provides a means of storing data within CCSP UI so that is retrievable by a user-  
supplied key. Data can be stored only for the current login session or saved to web storage using   
DALService for subsequent login sessions. Data is stored as a key-value pair, so the key used must   
be unique. The data stored can be in any form including objects and arrays.   
Properties   
None   
Functions   
DeleteValue   
Purpose   
Deletes the data referenced by the given key from current login session and   
from the local store, if specified   
Signature   
DeleteValue(key, removeFromDAL)   
Parameter   
Type   
Description   
key   
String   
Unique identifier of the data value to be deleted   
removeFromDAL   
Boolean   
Flag indicating that the data is to be deleted from local storage if   
present   
GetValue   
Purpose   
Retrieves the data referenced by the given key. Retrieves it from local   
storage, if specified by the given flag   
Signature   
GetValue(key, getFromDAL)   
Parameter   
Type   
Description   
key   
String   
Unique identifier of the data to be retrieved   
getFromDAL   
Boolean   
Flag indicating that the data is to be retrieved from local storage   
if present

CCSP UI Integrations Guide - 7.3 and 7.4   
31   
SetValue   
Purpose   
Saves the data with the given key. This is saved to local storage if specified by   
the given flag   
Note   
Calling this function with a null data value is the same as calling DeleteValue.   
   
Signature   
SetValue(key, value, saveToDAL)   
Parameter   
Type   
Description   
key   
String   
Unique identifier of the data value to be stored   
value   
Any   
The data to be stored   
saveToDAL   
Boolean   
Flag indicating that the data is also to be saved to local storage

CCSP UI Integrations Guide - 7.3 and 7.4   
32   
loggerService   
This service is used to output debugging information to the browser console and/or file log.   
Properties   
None   
Functions   
SetUserName   
Purpose   
Sets the user name to the value given to indicate the source of the logging   
Signature   
SetUserName(username)   
Parameter   
Type   
Description   
username   
string   
Username to be set   
Returns   
-   
WriteLine   
Purpose   
Writes a line of text to the selected outputs with the given log severity and   
identifying caller   
Signature   
WriteLine(message, severity, callerName)   
Parameter   
Type   
Description   
message   
string   
Text to be output   
severity   
string   
Warning level of the output. Possible values are defined in Global.js   
in the CCSP APIs Guide   
callerName   
string   
Used to indicate where the text originated from, typically the name   
of the calling function or class   
Returns   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
33   
oidcService   
This service provides an API to the Javascript object used to initiate and mange authentication and   
authorization of named clients with the CCSP Auth Server according to the OpenID Connect   
Authorization flow. The OpenID Connect Authorization flow and the use of this service is explained   
in 9: Single Sign On Client development.   
Properties   
None   
Functions   
addClient   
Purpose   
Adds an oidcClient object to its internal collection for performing   
authentication and authorization requests on behalf of a named   
client   
Signature   
addClient({client, clientId, redirectUri, redirectUriLogout,   
refreshTokenRequired, responseModePostRequired,   
refreshSlidingTime, refreshTokenCallback})   
Parameter   
Type   
Description   
client   
string   
Unique name to identify the client within the CCSP UI   
clientId   
string   
The ID by which the client is known in the OIDC Clients   
table and is presented as the identifier for the client   
when making requests to the CCSP Auth Server   
redirectUri   
string   
Location that is used by the CCSP Auth Server to help   
identify the client and where the CCSP Auth Server   
directs the browser with an authorization code if the   
user is authenticated   
redirectUriLogout   
string   
Location that is used by the CCSP Auth Server to help   
validate the client’s request and where the CCSP Auth   
Server directs the browser with an authorization code   
if the user is authenticated   
refreshSlidingTime   
int   
Time in milliseconds before the current access token   
expires that a new access token will be requested   
when refreshTokenRequired is true   
refreshTokenCallback   
string   
Function to be called when the result of an access   
token request using the refresh token is received   
refreshTokenRequired   
Boolean   
Flag indicating that a refresh token is also required   
when requesting an access token and that the access   
token should be automatically renewed using the   
OIDC Refresh flow

CCSP UI Integrations Guide - 7.3 and 7.4   
34   
responseModePostRequired   
Boolean   
Not currently used   
Returns   
Array of Translation objects defined as {Key, Value}   
getAccessToken   
Purpose   
Gets the current access token for the named client   
Signature   
getAccessToken(client)   
Parameter   
Type   
Description   
client   
string   
Unique name to identify the client within the CCSP UI   
Returns   
Access token, or null if no token defined   
getUserData   
Purpose   
Gets data from the CCSP Auth Server for the currently authenticated user   
Signature   
async getUserData(client)   
Parameter   
Type   
Description   
client   
string   
Unique name to identify the client within the CCSP UI   
Returns   
Object defined as {dataReceived, data} where:   
•   
dataReceived is Boolean indicating the success of the operation   
•   
data is the user data returned by the CCSP Auth Server. If dataReceived   
is false, this object is null;   
When defined, data contains:   
o   
FirstName – string - user’s first name   
o   
FullName – string - user’s combined first and last name   
o   
LastName – string - user’s last name   
o   
Name – string - user’s login name   
o   
PersonClaimType - int - type of user: 0 (Landlord), 1 (Reseller), 2   
(Tenant)   
o   
PersonId – int – user’s unique ID in Configuration Database   
o   
Profile\_ID – string – Profile ID or null if not defined   
o   
Role – int – user’s Role ID or null if not defined   
o   
SystemParameters {   
   
   
InitialDC – Domain controller being used   
   
   
UseRootOU – Boolean – indicates that CCSPRoot is being   
   
   
used   
   
}   
o   
TenantId – string – Tenant’s unique ID in Configiration Database   
or null if not a tenant user   
TenantInfo – not currently defined

CCSP UI Integrations Guide - 7.3 and 7.4   
35   
o   
upnSuffix – string – tenant or domain suffix (note lowercase ‘u’)   
o   
UserName – string – full login id with suffix   
hasClient   
Purpose   
Checks if the oidcService contains the named client   
Signature   
hasClient(client)   
Parameter   
Type   
Description   
client   
string   
Unique name to identify the client within the CCSP UI   
Returns   
Boolean that is true if client exists   
isRedirectCallbackCode   
Purpose   
Checks if the current or given window location contains an authorization   
code in the query   
Signature   
async isRedirectCallbackCode(client, location)   
Parameter   
Type   
Description   
client   
string   
Unique name to identify the client within the CCSP UI   
Location   
object   
Window location object defining the URL to be checked. If this is   
null, the current window location object will be used   
Returns   
Boolean that is true if the authorizatrion code has been found   
login   
Purpose   
Initiates the OIDC authorization code flow to authenticate the user and   
client   
Signature   
async login(client, basePath, language, locationObj)   
Parameter   
Type   
Description   
client   
string   
Unique name to identify the client within the CCSP UI   
basePath   
string   
Redirect URL for the authorizatioin code response. This will replace   
the redirect URL specified when the client was added to the   
oidcService. If the original redirect URI is to be used, set this   
parameter to null   
language   
string   
Language code to be used by the CCSP Auth Server login pages.   
This should always be null when the oidcService is used by gadgets   
as user will aready have been authenticated   
locationObj   
object   
Window location object defining the URL to be checked. If this is   
null, the current window location object will be used

CCSP UI Integrations Guide - 7.3 and 7.4   
36   
Returns   
Empty object if the login request is successful, or object containing:   
•   
error – string – Error message   
•   
status – Request status code   
Note that when function is successful, the window location will change to   
the requested redirect URI.   
logout   
Purpose   
Logs out the user from the CCSP Auth Server.   
Stop   
This method should never be called from a gadget within CCSP UI.   
   
Signature   
async logout(client)   
Parameter   
Type   
Description   
client   
string   
Unique name to identify the client within the CCSP UI   
Returns   
“ok” if logout was successful, else the error code in the event of failure   
processCodeCallback   
Purpose   
Makes a request to the CCSP Auth Server for an access token and optionally   
a refresh token   
Signature   
getAccessToken(client)   
Parameter   
Type   
Description   
client   
string   
Unique name to identify the client within the CCSP UI   
basePath   
string   
Redirect URL for the authorizatioin code response. This will replace   
the redirect URL specified when the client was added to the   
oidcService. If the original redirect URI is to be used, set this   
parameter to null   
language   
string   
Language code to be used by the CCSP Auth Server login pages.   
This should always be null when the oidcService is used by gadgets   
as user will aready have been authenticated   
locationObj   
object   
Window location object defining the URL to be checked. If this is   
null, the current window location object will be used

CCSP UI Integrations Guide - 7.3 and 7.4   
37   
Returns   
Object defined as {tokenReceived, error} where:   
•   
tokenReceived is Boolean indicating the success of the operation   
•   
error is the error message returned by the CCSP Auth Server if the   
operation failed. It the operation succeeded, this object is not defined   
Note   
This function does not return the access token itself. Client gadgets must   
use the getAccessToken method and should not store the access token   
locally. They should only get the token when they need to send it to the   
Resouce Server in a request.   
   
removeClient   
Purpose   
For the named client, cancels any ongoing token refresh, deletes any tokens,   
and removes the client from the oidcService   
Signature   
removeClient(client)   
Parameter   
Type   
Description   
client   
string   
Unique name to identify the client within the CCSP UI   
Returns   
-   
stopRefreshingToken   
Purpose   
For the named client, cancels any ongoing token refresh and deletes any   
tokens   
Signature   
stopRefreshingToken(client)   
Parameter   
Type   
Description   
client   
string   
Unique name to identify the client within the CCSP UI   
Returns   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
38   
translationService   
This service is used to load default and custom xml translation files and provide functions to   
translate text into the required language according to the information provided in those files.   
Properties   
None   
Functions   
getAllTranlsations   
Purpose   
Gets all the loaded translations   
Signature   
getAllTranslations()   
Parameter   
Type   
Description   
-   
-   
-   
Returns   
Array of Translation objects defined as {Key, Value}   
getLanguagesList   
Purpose   
Gets the names of the language files available in the system   
Signature   
getLanguagesList()   
Parameter   
Type   
Description   
-   
-   
-   
Returns   
Array of Language file objects defined as {filename, NativeName}   
where:   
•   
filename is the name of the translation file without the extension   
•   
NativeName is the full language name, full region/country name and the   
optional script   
getTranlsation   
Purpose   
Gets the translated text for the given key in the set   
Signature   
getTranslation(key)   
Parameter   
Type   
Description   
key   
string   
Key phrase to translate. See relevant translations xml file for   
possible values   
Returns   
string translation or null if no translation found

CCSP UI Integrations Guide - 7.3 and 7.4   
39   
getTranlsationAsync   
Purpose   
Gets the translated text for the given key in the set language   
asynchronously. Returns a promise object   
Signature   
getTranslationAsync(key)   
Parameter   
Type   
Description   
key   
string   
Key phrase to translate. See relevant translations xml file for   
possible values   
Returns   
Promise object where the then function returns the translated string or null   
if no translation found   
 initTranslationObject   
Purpose   
Initializes the translation service to use the given language   
Signature   
initTranlsationObject(language)   
Parameter   
Type   
Description   
language   
string   
Language key. Currently supported values are:   
•   
"de" — German   
•   
"en-us" — English-United States   
•   
"fr" — French   
•   
"ja" — Japanese   
Returns   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
40   
viewModelStoreService   
This service is used to store and retrieve single instances of objects that are global to the CCSP UI   
application. As the name suggests, these are typically ViewModels that may each represent an   
email, or a window that can be referenced by different parts of CCSP UI, but of which only one   
instance must exist. Integrations can use this to store data that needs to be available to different   
gadgets.   
Properties   
None   
Functions   
checkIn   
Purpose   
Adds a ViewModel to the store that is referenced by the given name   
Signature   
checkIn(name, vm)   
Parameter   
Type   
Description   
name   
String   
Name of the ViewModel by which it is referenced   
vm   
Object   
ViewModel to store   
Returns   
-   
checkOut   
Purpose   
Retrieves a ViewModel from the store that is referenced by the given name   
Signature   
checkOut(name)   
Parameter   
Type   
Description   
name   
String   
Name of the ViewModel to retrieve   
Returns   
ViewModel that is referenced by the given name or null if it is not found   
Remove   
Purpose   
Removes the ViewModel from the store that is referenced by the given   
name   
Signature   
remove(name)   
Parameter   
Type   
Description   
name   
String   
Name of the ViewModel to delete   
Returns   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
41   
voipService   
This service provides the high level interface to the media devices on the agent's desktop or the   
agent's alternative voice device if selected.   
Properties   
isMicMuted   
Type: Boolean   
Description: Flag that indicates if the currently selected audio input device is muted.   
isSpeakerMuted   
Type: Boolean   
Description: Flag that indicates if the currently selected audio output device is muted.   
micSliderVal   
Type: integer   
Description: Audio level of the currently selected input device on a scale of 0 to 100.   
speakerSliderVal   
Type: integer   
Description: Audio level of the currently selected output device on a scale of 0 to 100.   
Functions   
Connect   
Purpose   
Establishes a media connection for a given Call object using the selected   
audio devices on the agent's desktop   
Signature   
Connect(call)   
Parameter   
Type   
Description   
call   
Object   
Call object to establish the connection for   
Returns   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
42   
ConnectAlternateDeliveryCall   
Purpose   
Establishes a media connection for a given Call object using the agent's   
alternative voice device   
Signature   
ConnectAlternateDeliveryCall(call)   
Parameter   
Type   
Description   
call   
   
Object   
Call object to establish the connection for   
Returns   
-   
MicrophoneMute   
Purpose   
Mutes or unmutes the selected audio input device according to the input   
Note   
Changes in the microphone mute status can be caused by different parts of   
CCSP UI so the current value of isMicMuted should always be checked   
before using this function.   
   
Signature   
MicrophoneMute(newStatus)   
Parameter   
Type   
Description   
newStatus   
Boolean   
Flag indicating whether or not microphone is to be muted   
Returns   
-   
PreConnect   
Purpose   
Establishes a media connection for the given Call object using the selected   
audio devices on the agent's desktop when the call is incoming   
Signature   
PreConnect(call)   
Parameter   
Type   
Description   
call   
   
Object   
Call object to establish the connection for   
Returns   
-   
PreConnectAlternativeDelivery   
Purpose   
Establishes a media connection for the given Call object using the agent's   
alternative voice device when the call is incoming   
Signature   
PreConnectAlternativeDelivery(call)   
Parameter   
Type   
Description   
call   
   
Object   
Call object to establish the connection for   
Returns   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
43   
SpeakerMute   
Purpose   
Mutes or unmutes the selected audio output device according to the input   
Note   
Changes in the speaker mute status can be caused by different parts of   
CCSP UI so the current value of isSpeakerMuted should always be checked   
before using this function.   
   
Signature   
SpeakerMute(newStatus)   
Parameter   
Type   
Description   
newStatus   
Boolean   
Flag indicating whether or not speaker is to be muted   
Returns   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
44   
5: Gadgets   
Overview   
This section describes the CCSP UI gadget capability, what gadgets are, how they are created, and   
how they can be written to greatly extend the capabilities of CCSP UI and integrate it with a wider   
set of customer applications and websites.   
Note   
This guide uses AngularJS in the example gadgets and we recommend that you familiarize   
yourself with the basic concepts of this framework.   
Example gadgets   
The full versions of all example gadgets given in this guide are available in the installed CCSP UI   
folder under Server Components\TouchPoint\App\gadgets\Examples in their respective folder. When   
creating the example gadgets as part of the walkthrough, create a folder under Server   
Components\TouchPoint\App\gadgets for each example gadget for your own work.   
Note   
The code walkthroughs in this guide purposely provide very little in the way of CSS styling to   
avoid cluttering the HTML. This does mean that the gadgets produced during the walkthroughs   
do not look very polished and it is up to you to introduce styles as you see fit.   
What is a gadget?   
A gadget is a facility in CCSP UI that can be added, removed, or customized to help build bespoke   
GUIs designed to meet the particular needs of a tenant group or role. CCSP UI comes with its own   
set of customizable default gadgets and it enables the addition of third-party gadgets, to extend   
the functionality and integrate CCSP UI with other applications.   
Gadget capabilities   
Some of the capabilities and benefits of using gadgets are:   
•   
Ability to listen to all UI and AgentWS events raised in CCSP UI and perform bespoke actions as   
a result.   
•   
Ability to raise their own events that can be subscribed to by other gadgets or other   
applications that are listening through CCSP UI Connector.   
•   
Ability to manipulate calls held in the Agent.   
•   
Ability to manipulate the agent status and settings.

CCSP UI Integrations Guide - 7.3 and 7.4   
45   
Gadget types   
CCSP UI gadgets fall into three categories: system, standard, and functional.   
System gadgets   
These are the default gadgets that are developed and packaged as part of the CCSP UI application   
itself. Certain aspects of these gadgets can be customized for each tenant, group, or role.   
Standard gadgets   
Standard gadgets are custom windows that can host HTML or ASP pages that are as simple or   
complex as necessary, or provide access to different websites within the CCSP UI window. The   
content of a standard gadget is limited only by the normal restrictions for iframes (that is, certain   
public websites such as Google or BBC do not allow their content to be displayed in an iframe) and   
can be used to display information from a CRM application or to activate an application on the   
client desktop.   
Standard gadgets can be opened from the gadgetory bar in CCSP UI and can be opened as a   
tabbed, floating, or separate window and resized and repositioned. They have a range of properties   
that can be used to control their appearance and position and can be assigned a stylesheet.   
Functional gadgets   
Functional gadgets are scripts that are added to the CCSP UI application on agent login. They have   
no UI and usually just sit in the background listening for specific CCSP UI events and execute the   
required business logic. An example use for a functional gadget is to open a website in a new   
browser window when the agent receives a specific kind of call. The functional gadget can pass   
information from that call to the new web page.   
Standard vs. functional gadgets   
The main difference between standard and functional gadgets from a user perspective is that a   
standard gadget has a window and can be activated from the gadgetory, while a functional gadget   
for all intents and purposes is invisible. From a development point of view there are the following   
important differences:   
•   
Standard gadgets are registered for events for the duration of their lifetime (from being   
opened to being closed), whereas functional gadgets are registered for the entirety of the   
agent's login session.   
•   
Standard gadgets are refreshed each time they are closed and then re-opened. This means   
that when developing a gadget, it is not necessary for a test agent to log out and back into   
CCSP UI every time a change is made. Functional gadgets are loaded on agent login, so a   
change made to the script code requires a relogin.   
•   
A standard gadget can be written as an AngularJS module and utilize the CCSP UI services   
using service injection. Functional gadgets can only access these services using the AngularJS   
injector.

CCSP UI Integrations Guide - 7.3 and 7.4   
46   
Note   
A standard gadget does not have to be written as an AngularJS module as it also can use the   
injector.   
Combining standard and functional gadgets   
A functional gadget can be used to display a standard gadget and pass arguments to it. An   
example of this to ensure that a standard gadget is always displayed on a new call. Standard   
gadgets can also use functions and objects that have been defined in a functional gadget (see   
Inter-gadget access example).   
Gadget creation   
Standard or functional gadget?   
Before starting to create a gadget, you need to know the purpose of the gadget because this   
determines the type of gadget to be created. Ask the following questions:   
•   
Does the gadget need to display any information to the agent?   
•   
Does the gadget need to register for events on agent login?   
•   
What does the gadget do? For example, does it open a web page, run an application, pop up a   
window, or raise an event?   
   
If the gadget needs to present a page to the agent that can be used to manipulate the agent's   
login session or a call, then you need a standard gadget. If the gadget just needs to send data to a   
CRM database every time a certain event occurs, a functional gadget is sufficient.   
Demonstration gadgets   
In this document, we develop the following gadgets to illustrate the various concepts and   
capabilities of custom gadgets in CCSP UI:   
•   
Chat Prompt — standard gadget   
•   
Chat Prompt Func — functional gadget   
•   
Custom Email — standard gadget   
•   
Demo CRM — standard gadget   
•   
Demo Chat Prompt — functional and standard gadgets   
•   
Simple Gadget — standard gadget   
   
The Demo CRM gadget is loosely based on the MiniCRM and demonstrates access to and   
manipulation of the Agent and Call objects, advanced call handling, and how to access and use   
various AngularJS services provided within the CCSP UI application.

CCSP UI Integrations Guide - 7.3 and 7.4   
47   
The Demo Chat Prompt example starts off as a simple functional gadget to answer a chat call and   
extract information from the caller. It is later extended into a standard gadget that pops up on a   
new call and demonstrates the addition and customization of gadget properties.   
CCSP AdminPoint   
You administer CCSP UI integrations and gadgets using the CCSP AdminPoint application. Before   
any custom gadget can be used in CCSP UI, it must be added first to the system repository of   
gadgets and secondly to the available gadgets for a tenant, group, or role. For full details about   
using CCSP AdminPoint, see CCSP AdminPont Help.   
The basic steps to add the two demonstration gadgets are as follows.   
Adding the Demo Chat Prompt functional gadget   
To add the Demo Chat Prompt functional gadget:   
1.   
Log in to CCSP AdminPoint using the appropriate credentials.   
2.   
Select the Gadgetory tab, select the Gadgets subtab, click the Add + button and select   
Functional Gadget.   
3.   
Type Demo Chat Prompt for the name of the new gadget.   
4.   
In Script Url, type the following URL:   
app/gadgets/DemoChatPrompt/DemoChatPromptFunc.js   
5.   
Click Save.   
   
Create the folder app/gadgets/DemoChatPrompt underneath the physical TouchPoint folder on the   
server and inside it, create the first version of the DemoChatPromptFunc.js file with the following   
code.   
function StartUpDemoChatPrompt() {   
 alert("The Demo Chat Prompt gadget has been loaded");   
}   
   
StartUpDemoChatPrompt();

CCSP UI Integrations Guide - 7.3 and 7.4   
48   
Adding the Demo CRM standard gadget   
To add the Demo CRM standard gadget:   
1.   
Log in to CCSP AdminPoint using the appropriate credentials.   
2.   
Select the Gadgetory tab, select the Gadgets subtab, click the Add Gadget + button and select   
Standard Gadget.   
3.   
Type Demo CRM for the name of the new gadget.   
4.   
In Target URL (HTML), type the following URL: app/gadgets/DemoCRM/DemoCRM.html.   
5.   
Leave all the other properties at their default values for now.   
6.   
Click Save.   
   
Create the folder app/gadgets/DemoCRM underneath the physical TouchPoint folder on the server   
and inside it, create the first version of the DemoCRM.html file with the following code.   
<!DOCTYPE html>   
<html xmlns="http://www.w3.org/1999/xhtml">   
<head>   
 <title>Demo CRM</title>   
</head>   
<body>   
 <h1>Hello, welcome to Demo CRM</h1>   
</body>   
</html>   
Adding the gadgets to a role   
When the gadgets are in the system, you can add them to any of the tenants or their contained   
groups or roles. To minimize the visibility of the two demo gadgets just added, we recommend that   
you create a specific Testing role for the tenant using CCSP Web Administrator and the two gadgets   
added to this role in CCSP AdminPoint. However, you can add the gadgets at the group or tenant   
level if required.   
For details about adding a group or role to a tenant and customizing it to add the two demo   
gadgets, see CCSP AdminPoint Help.

CCSP UI Integrations Guide - 7.3 and 7.4   
49   
Testing the gadgets   
After you add the gadgets at the appropriate level, they are available for use within that level in the   
next login session. Log on to CCSP UI using an agent belonging to the role that contains the   
gadget. A normal browser alert should be displayed with the message The Demo Chat Prompt   
gadget has been loaded. Acknowledge this alert and check the Gadgetory tool bar. This should have   
an additional icon as below.   
   
Clicking on this icon shows the new Demo CRM gadget as a tabbed window in CCSP UI.   
Gadget refresh   
During gadget development, adding any gadget and modifying it in the system and at the different   
customization levels within CCSP AdminPoint, requires a new login session for the changes to take   
effect in CCSP UI, that is, a currently logged on user must log out and back on again.   
Changes to a standard gadget's code and files, however, do not require a new login session   
provided the file specified as the Target URL (HTML) remains the same. To reload a standard   
gadget with the changes, close the gadget in CCSP UI and re-open it. This does not apply to   
functional gadgets because CCSP loads them at login, so a new login session is required for any   
code changes to take effect.   
This can be demonstrated by logging onto the user with access to the DemoCRM gadget, adding   
the following line to the DemoCRM.html in the body section and saving the updated file.   
<h2>This is the gadget used for the standard gadget development   
walkthrough</h2>   
   
While still logged on, open the Demo CRM and it displays with the new text.   
Conversely, changing the alert message displayed in the Demo Chat Prompt functional gadget,   
while the user is logged on has no effect.

CCSP UI Integrations Guide - 7.3 and 7.4   
50   
Gadget properties   
Standard gadgets are created with a number of default properties and these can be supplemented   
with custom properties that are available for access from within the gadget itself.   
Default properties   
The table below lists the default properties for a standard gadget.   
Name   
Usage   
Notes   
Disabled   
When selected, the gadget is no   
longer part of the customization   
level and does not appear in CCSP   
UI.   
   
Can Maximize   
NOT USED   
DO NOT USE   
Can Hide   
When selected, the Minimize option   
is added to the gadget's window   
menu so that it can be removed   
from display while still running.   
   
Can Close   
When selected, a close (  
) button   
is added to the gadget's window   
menu if the gadget is displayed as a   
window in CCSP UI.   
A close button is always added if   
the gadget is being displayed as a   
tab or pop out window.   
Can Popout   
When selected, a Pop out button is   
added to the gadget's tool bar or   
window menu when that displays   
the gadget in its own browser   
window when activated.   
This property must be enabled   
when the Default State property is   
set to Popout.   
Can Tab   
When selected, a maximize (  
)   
button is added to the gadget's tool   
bar when it is displayed as a   
floating or pop out window. This   
docks the gadget as a tab page in   
CCSP UI when activated.   
This property must be enabled   
when the Default State property is   
set to Tab.   
Can Float   
When selected, a Float button is   
added to the gadget's window   
menu when it is displayed as a tab   
page. This changes the gadget to a   
floating window within CCSP UI   
when activated.   
This property must be enabled   
when the Default State property is   
set to Float.

CCSP UI Integrations Guide - 7.3 and 7.4   
51   
Name   
Usage   
Notes   
Can Fold   
When selected, the gadget's icon in   
the gadgetory is removed if the   
CCSP UI window height is   
decreased beyond a point where all   
gadgets can be displayed. The   
gadget is placed in a subgadgetory   
window, accessible by hovering   
over the plus icon in the gadgetory   
as below.   
   
   
When this property is not enabled,   
the gadget remains in the main   
gadgetory. If the CCSP UI height   
shrinks below a certain value, the   
gadget is not shown.   
Is Supervisor   
Gadget   
When selected, the gadget is only   
available in the Gadgetory if the   
logged on user is a supervisor.   
   
Is Bottom   
When selected, the gadget appears   
at the bottom of the gadgetory   
below the Settings gadget.   
   
Can Expand   
When selected, a toggle (  
) button   
is added to the gadget's window   
when it is displayed as a floating   
window. When activated, the   
"popoutWindow\_Expand" event is   
raised and the button changes to   
. When activated again, the   
"popoutWindow\_Contract" event is   
raised.   
The gadget implementation must   
subscribe to this event and perform   
any actions required.   
No Icon in   
Gadgetory   
When selected, no icon is displayed   
in the gadgetory for the gadget,   
therefore it can only be displayed   
programmatically, for example, in   
response to an event.   
   
Float Default   
Position   
Specifies the initial X and Y co-  
ordinates of the gadget when   
displayed as a floating window.   
X increases the value from left to   
right and Y increases it from top to   
bottom.

CCSP UI Integrations Guide - 7.3 and 7.4   
52   
Name   
Usage   
Notes   
Float Default Size   
Specifies the initial width and   
height of the gadget when   
displayed as a floating window.   
   
Popout Default   
Position   
Specifies the initial X and Y co-  
ordinates of the gadget when   
displayed in a separate browser   
window.   
X increases the value from left to   
right and Y increases it from top to   
bottom.   
Popout Default   
Size   
Specifies the initial height and   
width of the gadget when displayed   
in a separate browser window.   
   
Default State   
Specifies the default layout of the   
gadget when it is opened. Can be   
Float, Popout or Tab.   
The corresponding property flag   
must be enabled for the gadget to   
appear in the selected state, for   
example, Can Tab must be enabled   
if Tab is selected.   
Allowed Instances   
Specifies the number of separate   
instances of the gadget that can be   
opened.   
   
Target URL (HTML)   
   
Specifies the HTML file that   
represents the gadget.   
This file must be in a location   
relative to the TouchPoint virtual   
folder.   
Controller URL   
   
Specifies a JavaScript file to load for   
the gadget on login.   
This file must be in a location   
relative to the TouchPoint virtual   
folder.   
Icon   
Enables a custom image file to be   
used to represent the gadget in the   
gadgetory.   
The image file is loaded into the   
Configuration Database, so the file   
can be located anywhere.   
SelectedIcon   
Enables a custom image file to be   
used to represent the highlighted   
gadget in the gadgetory.   
The image file is loaded into the   
Configuration Database, so the file   
can be located anywhere.   
Stylesheet   
Specifies a CSS file to be used for   
the gadget.   
This file must be in a location   
relative to the TouchPoint virtual   
folder.   
   
Table 2: Standard gadget default properties

CCSP UI Integrations Guide - 7.3 and 7.4   
53   
In CCSP AdminPoint, go to the Gadgets subtab and edit the Demo CRM gadget. Experiment with   
changing the different property values and relogin to see the effects on the gadget.   
The table below lists the default properties for a functional gadget.   
Name   
Usage   
Notes   
Disabled   
When enabled, the gadget is not   
loaded in CCSP UI   
   
Script URL   
Specifies the JavaScript file that   
represents the gadget.   
This file must be in a location   
relative to the TouchPoint virtual   
folder.   
   
Table 3: Functional gadget default properties   
Adding custom properties   
Custom properties can be added to both standard and functional gadgets. The following types of   
properties can be added:   
•   
Boolean   
•   
Integer   
•   
String   
•   
Coordinates   
•   
Dimension   
•   
ValuesList   
   
Custom properties can only be added to a gadget at the system level, not at the different   
customization levels. In each case, the name of the property must be specified and an optional   
default value. For the ValuesList, which is simply an array of integer and/or string values, a Default   
value must be specified from those added.   
The name of a custom property can contain spaces. In the code, the space is removed from the   
name.   
The addition and usage of Custom Properties is explored in further detail with examples in Adding   
an example custom property.

CCSP UI Integrations Guide - 7.3 and 7.4   
54   
Standard gadget development   
Customized standard gadgets are HTML-based pages that contain UI and business logic to extend   
the CCSP UI functionality. Because CCSP UI is an AngularJS –based application, standard gadgets   
can be written as AngularJS modules and use the services provided by CCSP UI through the normal   
injection into the module. This section describes how to write such a gadget.   
This guide provides the minimum information on AngularJS to produce a usable gadget that   
demonstrates how services and objects from the main CCSP UI application are used to extend its   
functionality. AngularJS is a powerful development framework and it can be used to create gadgets   
with a rich UI and complex business logic. Such usage is beyond the scope of the document and for   
further information, see the official AngularJS website: https://angularjs.org/.   
Standard gadget structure   
HTML file   
This is the file that must be used as the Target URL (HTML) parameter for the gadget in CCSP   
AdminPoint. This file may be all that is needed if the gadget is very simple, for example, if it just   
displays a non-interactive web page.   
CSS file   
This file can be specified in the Stylesheet property of the gadget in CCSP AdminPoint or it can be   
one of the links within the HTML file.   
Code files   
This file is the source code that is included by the HTML file. This file can be anything but access to   
the essential objects such as Agent and Call and the various services is provided using JavaScript. If   
the gadget is written as an AngularJS module, this file should contain the page's controller.   
Depending on the complexity of the gadget, multiple code files may be required, which can all be   
included by the HTML file.

CCSP UI Integrations Guide - 7.3 and 7.4   
55   
Standard gadget as an AngularJS module   
The Demo CRM gadget created for this guide currently contains only the following lines in the   
HTML file.   
<!DOCTYPE html>   
<html xmlns="http://www.w3.org/1999/xhtml">   
<head>   
 <title>Demo CRM</title>   
</head>   
<body>   
 <h1>Hello, welcome to Demo CRM</h1>   
</body>   
</html>   
   
To convert this to an AngularJS module, you need to provide as a minimum the following:   
•   
Includes for the AngularJS code   
•   
An AngularJS module for the page ("demoCRMApp")   
•   
An AngularJS controller for the body ("demoCRMController")   
•   
The module definition and the controller code   
Converting the Demo CRM HTML to AngularJS   
Add the following script includes to the DemoCRM.html file.   
<script type="text/javascript" src="/TouchPoint/Scripts/jquery-  
2.0.1.js"></script>   
<script type="text/javascript" src="/TouchPoint/Scripts/angular.js"></script>   
<script type="text/javascript"   
src="/TouchPoint/App/common/gadgetInit.js"></script>   
<script type="text/javascript" src="/TouchPoint/Ajax/helpers.js"></script>   
<script type="text/javascript" src="DemoCRM.js"></script>   
   
The list of includes covers the AngularJS source and other useful files:   
•   
jquery.js facilitates the selection and manipulation of the HTML elements that are added later.   
•   
angular.js provides the necessary objects and functions for creating the module and controller.   
•   
gadgetInit.js defines functions for accessing global functions such as GetRootWindow().   
•   
helpers.js includes various utility functions for extracting data from the Agent and Call objects.   
•   
DemoCRM.js is the code file for the gadget and contains the controller that is added later.

CCSP UI Integrations Guide - 7.3 and 7.4   
56   
The demoCRMApp module is added to the html declaration and the demoCRMController as an   
attribute of the body as below.   
<html ng-app="demoCRMApp" xmlns="http://www.w3.org/1999/xhtml">   
…   
<body ng-controller="demoCRMController">   
The html should now look like this.   
<!DOCTYPE html>   
<html ng-app="demoCRMApp" xmlns="http://www.w3.org/1999/xhtml">   
<head>   
 <title>Demo CRM</title>   
 <script type="text/javascript" src="/TouchPoint/Scripts/jquery-  
2.0.1.js"></script>   
 <script type="text/javascript"   
src="/TouchPoint/Scripts/angular.js"></script>   
 <script type="text/javascript"   
src="/TouchPoint/App/common/gadgetInit.js"></script>   
 <script type="text/javascript" src="/TouchPoint/Ajax/helpers.js"></script>   
 <script type="text/javascript" src="DemoCRM.js"></script>   
</head>   
<body ng-controller="demoCRMController">   
 <h1>Hello, welcome to Demo CRM</h1>   
   
<h2>This is the gadget used for the standard gadget development   
walkthrough</h2>   
</body>   
</html>

CCSP UI Integrations Guide - 7.3 and 7.4   
57   
Defining the Demo CRM module and controller   
The demoCRMApp and demoCRMController are defined in a separate DemoCRM.js file. It could   
actually all be defined in the HTML file, but for clarity and best practice, we use a separate file.   
1.   
Create the DemoCRM.js file in the same DemoCRM folder as the HTML file and open for editing.   
2.   
Get a reference to the root CCSP UI window, which has visibility of the modules and AngularJS   
services.   
//get reference to root window - the one containing our main modules and   
services   
var \_root = GetRootWindow();   
3.   
Create the demoCRMApp application module.   
//create module   
var mainModule = angular.module("demoCRMApp", []);   
4.   
Access the required services. For now, only loggerService is used to write some debug output   
to the Client log file on the agent's machine.   
var directServices = ["loggerService"];   
//use this method to inject all the services above to your module   
\_root.appendServiceProxiesToModule(mainModule, directServices);   
5.   
Create the controller for the DemoCRM page as part of the demoCRMApp application module,   
with the loggerService as an injected service.   
mainMoule.controller("demoCRMController", ["$scope", "loggerService",   
"eventServiceWrapper",   
 function ($scope, loggerService, eventServiceWrapper) {   
 loggerService.WriteLine("Demo CRM has been opened",   
\_root.enghouse.enums.Severity.Debug, "demoCRM");   
 }]);   
6.   
Optionally define the ready event handler for the document, which should call the   
GadgetComm.windowReady() function. This is actually only required when the gadget is   
expected to be opened by another gadget such as a functional gadget (see Accessing other   
gadgets and windows).   
angular.element(document).ready(function ()   
{\_root.app.GadgetComm.windowReady(window);}   
);

CCSP UI Integrations Guide - 7.3 and 7.4   
58   
The complete code should be as follows.   
//get reference to root window - the one containing our main modules and   
services   
var \_root = GetRootWindow();   
   
//create module   
var mainModule = angular.module("demoCRMApp", []);   
   
//declare the services you wish to inject to your module   
var directServices = ["loggerService"];   
   
//use this method to inject all the services above to your module   
\_root.appendServiceProxiesToModule(mainModule, directServices);   
   
//declare your controller with its dependencies - angular or custom ones.   
mainModule.controller("demoCRMController", ["$scope", "loggerService",   
 function ($scope, loggerService) {   
 loggerService.WriteLine("Demo CRM has been opened",   
\_root.enghouse.enums.Severity.Debug, "demoCRM");   
 }]);   
   
angular.element(document).ready(function ()   
{\_root.app.GadgetComm.windowReady(window);}   
);   
   
On closing the DemoCRM gadget and re-opening it, an entry is written into the current ClientLog   
file indicating that the DemoCRM gadget has been opened.   
   
Note   
Every major browser provides the ability to debug javascript code using F12 or a menu option to   
invoke the Developer Tools. This includes the ability to examine and set breakpoints in source   
code and we highly recommend its use in developing gadgets.   
Because gadget files are dynamically loaded, you may have a problem seeing this code in the   
debugger applications of some browsers, such as Chrome. To ensure that the code is accessible   
in the debugger by the given name, add the following line at the end of a gadget’s javascript file:   
//# sourceURL=<filename>.js   
   
Replace <filename.js> with the name of your file (full path is not necessary).

CCSP UI Integrations Guide - 7.3 and 7.4   
59   
Functional gadget development   
Functional gadgets comprise a single script file that is the value of the Script URL for the gadget in   
CCSP AdminPoint. The script is loaded on user login by appending it to the main document of the   
CCSP UI web page, so it has access to all the functions defined in the scripts included by the main   
application. Although it has no associated HTML, it can still make use of AngularJS services and   
scopes by using an AngularJS injector. This section show how to achieve this.   
Note   
You can have multiple code files for a functional gadget by adding the files dynamically. The   
IntegrationsFunc.js file that is part of the framework to migrate legacy integrations to CCSP UI   
uses this technique. For an example of this, see the code in the IntegrationsFunc.js that is part of   
the Integrations.js gadget.   
Functional gadget code structure   
The Demo Chat Prompt functional gadget code currently looks like this.   
function StartUpDemoChatPrompt() {   
 alert("The Demo Chat Prompt gadget has been loaded");   
}   
StartUpDemoChatPrompt();   
   
Note   
Wrapping the functional gadget code in its own function or object is considered good practice   
and strongly recommended to prevent declared variables and functions from the gadget   
unnecessarily being made part of the global scope.   
   
To convert this code to a more useful function using AngularJS:   
1.   
As with the standard gadget, get a reference to the root CCSP UI window.   
//get reference to root window - the one containing our main modules and   
services   
var \_root = GetRootWindow();   
   
2.   
Get the AngularJS injector that provides access to the main application's services and objects.   
var localInjector = \_root.angular.element("[ng-app]").injector();   
   
3.   
To demonstrate that you can now access any of the CCSP UI services, as for the standard   
gadget, use the loggerService to output a line of text to the client log in the file system.   
var loggerService = localInjector.get('loggerService');   
   
 loggerService.WriteLine("Demo Chat Prompt Functional Gadget has been   
loaded", \_root.enghouse.enums.Severity.Debug, "demoChatPromptFunc");

CCSP UI Integrations Guide - 7.3 and 7.4   
60   
The complete code should look like this.   
function StartUpDemoChatPrompt() {   
   
 //get reference to root window - the one containing our main modules and   
services   
 var \_root = GetRootWindow();   
   
 //This line obtains the TouchPoint application's angularJS injector which   
provides access to all services and directives   
 var localInjector = \_root.angular.element("[ng-app]").injector();   
   
 //Can now get any service required   
 var loggerService = localInjector.get('loggerService');   
   
 loggerService.WriteLine("Demo Chat Prompt Functional Gadget has been   
loaded", \_root.enghouse.enums.Severity.Debug, "demoChatPromptFunc");   
}   
StartUpDemoChatPrompt();   
   
To see this output, relogin to the user and then check the client log file.

CCSP UI Integrations Guide - 7.3 and 7.4   
61   
Using CCSP UI services and objects   
Subscribing to events   
For the DemoCRM, we use eventServiceWrapper to subscribe to the AgentStateChange event.   
The signature for the eventServiceWrapper subscribe() function is as follows.   
eventServiceWrapper.subscribe(event, context, callback, priority)   
In this case, the event is "AgentStateChange", the context is always null and no priority needs to be   
specified. In the callback, the current state of the agent is displayed on the HTML page.   
In the body element of the DemoCRM page, add the following.   
<p>   
 Current Agent State is: <label id="agentState" />   
</p>   
   
The "agentState" label displays the state from the AgentStateChange event.   
In the DemoCRM.js add the subscription to the AgentStateChange event as follows.   
var agentStateChangeId = eventServiceWrapper.subscribe("AgentStateChange",   
null, function (newState) {   
 document.getElementById("agentState").innerHTML = newState;   
});   
   
Note   
The subscribe() function returns an id for that subscription, which should be stored so that it can   
be unsubscribed from later.   
   
The unsubscribe action should be done when the window is unloaded. It is good practice in   
AngularJS to use an injection of the window object rather than the window directly, so add this to   
the controller definition, as follows.   
mainModule.controller("demoCRMController", ["$scope", "$window",   
"loggerService", "eventServiceWrapper",   
 function ($scope, $window, loggerService, eventServiceWrapper) {   
   
Then add a handler for the unload() function.   
$window.onunload = function () {   
 loggerService.WriteLine("Demo CRM has been closed",   
\_root.enghouse.enums.Severity.Debug, "demoCRM");   
 eventServiceWrapper.unsubscribe(agentStateChangeId);   
};   
   
To test the DemoCRM, reload the DemoCRM and change the availability of the agent a few times.

CCSP UI Integrations Guide - 7.3 and 7.4   
62   
Publishing events   
Stop   
In general, user interface commands such as opening and closing windows can safely be   
published by custom gadgets. Any event that is received from the Agent Web Service must not be   
published by custom gadgets as this could seriously compromise the stability of CCSP UI.   
   
Events can be published using the publish() method of eventServiceWrapper, which has the   
following signature.   
eventServiceWrapper.publish(event, context, data)   
   
The event to open the Preferences dialog is openPreferences and it has no context or data. To   
demonstrate this event, add a button to the DemoCRM to display the Preferences dialog.   
Add the button to the HTML as below.   
<p>   
 <button id="openPreferencesButton" ng-  
click="openPreferences()">Preferences</button>   
</p>   
   
The ng-click attribute is the AngularJS equivalent of "onclick". The openPreferences() method is   
defined in the DemoCRM.js as below.   
$scope.openPreferences = function(){   
   
eventServiceWrapper.publish("openPreferences", null, null);   
};   
   
Reload the updated gadget and click the Preferences button that is now on the page. The   
Preferences dialog appears.   
You can display various other windows using this method. For example, you can display the   
Outgoing Call window by publishing the event OpenNewCallDialog. Add the following code to the   
HTML.   
<form id="callForm" submit="callNumber()">   
<label for="numberToCall">Number to call</label>   
 <input type="number" placeholder="e.g. 12345" id="numberToCall" /><input   
type="submit" ng-click="callNumber()" value="Call"/>   
</form>   
   
This code creates a simple form on the page with an input for a number and button to click to call.   
This step is actually in preparation for a later stage in the walkthrough, but for now all this form   
does is open the Outgoing Call dialog.

CCSP UI Integrations Guide - 7.3 and 7.4   
63   
In the DemoCRM.js, add the callNumber() function.   
$scope.callNumber = function () {   
 eventServiceWrapper.publish("OpenNewCallDialog", null, null);   
};   
   
Reload the updated gadget and click the Call button. The Outgoing Call dialog now appears with   
the dialpad selected.   
For more examples of publishing simple UI commands, see the MiniCRM code in Server   
Components\TouchPoint\App\gadgets\MiniCRM\MiniCRM.js.   
Accessing the Agent object   
The DemoCRM shows blank for the agent state when it is first loaded and a value only appears   
when the state is changed in the availability gadget. To show the state on loading the page, you   
need to access the Agent object.   
In the DemoCRM.js file, add the agentServiceWrapper to the list of services available to the   
DemoCRM gadget, as follows.   
//declare the services you wish to inject to your module   
var directServices = ["loggerService", "eventServiceWrapper",   
"agentServiceWrapper"];   
   
Also add it to the demoCRMController definition, as follows.   
mainModule.controller("demoCRMController", ["$scope", "$window",   
"loggerService", "eventServiceWrapper", "agentServiceWrapper",   
 function ($scope, $window, loggerService, eventServiceWrapper,   
agentServiceWrapper) {   
   
The agentServiceWrapper contains only one object, which is agent, but this encapsulates all the   
properties and actions associated with the logged in agent. To show the initial state of the agent on   
display of the DemoCRM, add the following.   
document.getElementById("agentState").innerHTML =   
agentServiceWrapper.agent.State;   
   
Caution   
The agentServiceWrapper provides the same Agent object that is used throughout CCSP UI. It is   
not a clone and it is not read-only, so any operations performed on this object is global. It is   
therefore imperative that customizations do not directly change properties on the Agent object   
(for example, never use agent.State = "Available").   
To avoid introducing errors such as the one above, use the constant-first method in   
comparisons, for example:   
if ("Available" == agent.State)…

CCSP UI Integrations Guide - 7.3 and 7.4   
64   
Use the Agent class methods to access properties and manipulate the Agent. The methods and   
properties are described in the Agent AJAX API Reference.   
Manipulating the Agent   
The Agent class provides methods for manipulating the Agent object from changing the agent state   
to logging out the agent. To demonstrate changing the agent state, add the following include to   
the HTML.   
<script type="text/javascript"   
src="/TouchPoint/Ajax/Interface/Agent.js"></script>   
   
This include is required to access the AgentState enumerate values.   
In the body, add a button to toggle the agent state.   
<p>   
 <input type="button" ng-click="toggleAgentState()" value="Toggle Agent   
Availability" />   
</p>   
   
Implement the toggleAgentState() function in DemoCRM.js as below.   
$scope.toggleAgentState = function () {   
 if ("Available" == agentServiceWrapper.agent.State) {   
 agentServiceWrapper.agent.GoReleased(AgentState.GenericReleased)   
 } else {   
 agentServiceWrapper.agent.GoAvailable();   
 }   
};   
   
Now when you reload the DemoCRM, clicking on the toggle button switches the agent state   
between Available and Released.   
As a further exercise, you can use the property agent.ReleaseCodesList to obtain the Release codes   
allocated to the Agent. Use this data to populate a drop down list of release codes that can be   
selected.

CCSP UI Integrations Guide - 7.3 and 7.4   
65   
Making calls   
Making calls using the Agent   
There are two general methods of making outgoing calls. One is to use the dial service, the other is   
to use the AJAX API method on the Agent class. Using the latter, you can replace the publish   
OpenNewCallDialog event with the use of the agent.CallOutPhone() method to directly initiate the   
call.   
Comment out the existing code in the callNumber() method and add the following.   
agentServiceWrapper.agent.CallOutPhone(document.getElementById("numberToCall")  
.value);   
   
Reload the DemoCRM and enter a valid phone number in the Number to Call text input and press   
Return or click Call. An outgoing call is now made to this number.   
Making calls using the dialService   
To use the dialService, it needs to be added to the list of injected services for the demoCRMApp   
module and the demoCRMController. The dialing action is then completed by using the Dial()   
function of the service, which has the following signature.   
Dial(type, name, target, needParam, parameterValue, configId, attendable,   
call)   
   
See the function description in the dialService Section for a full explanation of all the parameters,   
but for the purposes of replacing the use of the current CallOutPhone() call, only the type, name   
and target parameters need to be specified.   
Create another form on the web page by copying and pasting the current one, as follows.   
<form id="dialForm" submit="dialNumber()">   
 <label for="numberToDial">Number to dial</label>   
 <input type="number" placeholder="e.g. 12345" id="numberToDial" />   
 <label for="calleeName">Callee name</label>   
 <input type="text" placeholder="e.g. A Customer" id="calleeName" /><input   
type="submit" ng-click="dialNumber()" value="Dial" />   
</form>   
   
There are two inputs here, one for the number itself and the other for the callee name. In the   
DemoCRM.js, define the $scope.dialNumber() function and include the following single statement.   
dialService.Dial("dialpad", document.getElementById("calleeName").value,   
document.getElementById("numberToDial").value, "", "", "", "", null);   
   
When you reload the DemoCRM gadget, the Number to dial form appears. Input a valid number   
and a name for the Callee, and then click Dial. The outgoing call is now made to this number, as   
with the CallOutPhone() method on the Agent.

CCSP UI Integrations Guide - 7.3 and 7.4   
66   
dialService.Dial vs. Agent.CallOutPhone   
The Dial() function is the higher-level CCSP UI method of making calls, whereas the CallOutPhone()   
function is part of the low-level AJAX interface. The Dial() function actually uses CallOutPhone and   
its sibling functions (CallToSystem(), CallToAgent()), thereby providing a single, versatile dialing   
interface.   
The other important aspect is due to the dialService being part of the CCSP UI GUI. Calls made   
using the dialService are logged in the Recent calls list, whereas calls using the lower level functions   
are not. To see this difference, make calls to two different numbers using the "Call" and "Dial"   
forms in the DemoCRM gadget and then opening the Outgoing Call window. The call made using   
the Dial form is present in the Recent Calls list but the call made using the "Call" form is not.   
   
Note   
The value specified for the callee name (the "name" parameter in the Dial() function call) is not   
passed through the AJAX API to the server. It is purely for use within the CCSP UI as an extra   
identifier on the Recent Calls tab.   
General call handling   
The DemoCRM can currently make calls, but the call card needs to be used to hold and transfer,   
and hang up the call that has been made. To perform these functions from the DemoCRM, you   
need to obtain the Call object.   
Getting the Call object   
The most direct way of obtaining the Call object is to subscribe to the NewCall event as below.   
var newCallId = eventServiceWrapper.subscribe("NewCall", null, function   
(callEvent) {   
});   
   
Note   
The parameter passed to the NewCall event callback function is an object containing the Call   
object, not the call object itself.   
   
When the NewCall event is received, the gadget must first decide if this is a call that it is interested   
in. An important point to note is that NewCall events are raised for all calls, including voice, email,   
outgoing and incoming calls, agent to agent calls, and monitoring calls. There are therefore a   
number of criteria that a gadget must consider before doing any further processing on the call.   
When the gadget decides that this is a call of interest, the gadget must do the following:   
•   
Store a reference to the Call object   
•   
Subscribe to the Call object's state changes   
Storing a reference to the call is crucial, because subsequent events that are raised against the call   
do not include the Call object in the callback function data, only its call Id.

CCSP UI Integrations Guide - 7.3 and 7.4   
67   
Caution   
As with the Agent object, there is only one instance of each Call object in the CCSP UI application,   
so it must be handled with care by gadgets and integrations. Any manipulation of the Call object   
should be done through its exposed methods, and its properties should only be used for   
examination and never assigned to.   
   
To be notified of the call's state changes, the CallStateChange event is subscribed to using the   
context of the call's CallId, as in the following example.   
eventServiceWrapper.subscribe("CallStateChange", callEvent.call.CallId,   
$scope.OnCallStateChange);   
   
For the DemoCRM, the gadget is initially only interested in new voice calls that it has originated. So   
the criteria to check here is for the call type being "VOIP", the direction being outgoing, and the   
called number matching the input in the forms as below.   
var dialledNumber = "";   
   
var newCallId = eventServiceWrapper.subscribe("NewCall", null, function   
(callEvent) {   
 if ("VOIP" == callEvent.call.CallInfo.Type.toUpperCase() && "Outgoing" ==   
callEvent.call.CallInfo.Direction && dialledNumber ==   
callEvent.call.CallInfo.Caller.Name) {   
 eventServiceWrapper.subscribe("CallStateChange",   
callEvent.call.CallId, onCallStateChange);   
 }   
});   
   
A new variable has been introduced. The dialledNumber is set to the value of either the   
numberToCall or numberToDial input elements in the $scope.callNumber or $scope.dialNumber   
functions respectively, as in the following example.   
dialledNumber = document.getElementById("numberToCall").value;   
   
Despite its name, the CallInfo.Caller object for outgoing calls actually contains information related   
to the destination (callee). Unless an IVR has been set up to do so, the majority of the information is   
unpopulated and the Name property actually contains the dialed number (URI if a SIP address is   
used). If the callee is another agent, the value of Name is the destination agent's display name.   
The onCallStateChange() function initially just updates a new label in the HTML page that shows the   
current outgoing call state.   
<p>   
 Current Outgoing Call State: <label id="callState"></label>   
</p>

CCSP UI Integrations Guide - 7.3 and 7.4   
68   
The function is defined as follows.   
onCallStateChange = function (callEvent) {   
 document.getElementById("callState").innerHTML = callEvent.state;   
};   
   
Reload the DemoCRM and call a number using either of the Call or Dial forms. The new label shows   
the InCall state after the call is connected. In the call card, hold and unhold the call and observe the   
changes in the state value in the DemoCRM gadget.   
When you hang up the call, the state goes to WrapUp (if WrapUp codes have been defined for the   
Agent) and on completion of WrapUp the state goes to destructed. If you make a voice call from the   
Outgoing call window directly, or email and chat calls are received, these are ignored by the   
DemoCRM gadget.   
Holding the Call   
To perform operations on the Call, a reference to it must be maintained within the gadget. For the   
DemoCRM, it is a property of the demoCRMController scope, declared as below.   
$scope.outgoingCall = null;   
   
This is now usable in the HTML (without the $scope prefix) and is set in the callback for the NewCall   
event, as below.   
$scope.outgoingCall = callEvent.call;   
$scope.$apply();   
   
The $scope.$apply() line is required to update the binding on the HTML page when the   
outgoingCall variable has been updated, because the NewCall event callback is outside of the   
AngularJS framework.   
Note   
Add a $scope.$apply() line at the end of the onCallStateChange() function for the same reason.   
   
With a reference to the Call object, the gadget can now display its properties and use its methods   
directly. Add a button in the HTML code to the Call state line to use to hold and pick up the   
outgoing call. To demonstrate the use of the outgoingCall object's properties, write the button as   
follows.   
<button ng-click="toggleHold()">{{outgoingCall.State.toUpperCase()=="ONHOLD" ?   
"Pick Up Call" : "Hold Call"}}</button>   
   
The code between two curly braces uses the Call object's state to display the correct available   
operation. Reload the gadget to see this in action. On making an outgoing call, the hold button   
shows the caption "Hold Call". When putting the call on hold from the call card, the button's   
caption changes to "Pick Up Call".   
To actually perform the hold/pick up operation, implement the toggleHold() function in the   
DemoCRM.js as below.

CCSP UI Integrations Guide - 7.3 and 7.4   
69   
$scope.toggleHold = function () {   
 if ("InCall" == $scope.outgoingCall.State) {   
 $scope.outgoingCall.PutOnHold();   
 } else {   
 $scope.outgoingCall.PickUp();   
 }   
};   
   
The function uses the Call object's PutOnHold() and PickUp() methods to perform the actions.   
Reload the DemoCRM gadget and use the Hold Call button to hold and retrieve the call.   
Hanging up the Call   
To hang up the call, you need to use the Call object's HangUp() method. Add a new button to the   
HTML code as below.   
<button ng-click="hangUp()" ng-disabled="outgoingCall == null ||   
outgoingCall.State == 'Destructed' || outgoingCall.State == 'WrapUp'">Hang   
Up</button>   
   
In this case, the availability of the button is determined by the state of the outgoing call. The   
hangUp() function should be implemented in the DemoCRM.js as below.   
$scope.hangUp = function () {   
 $scope.outgoingCall.HangUp();   
};   
   
On reloading the DemoCRM gadget, the Hang Up button is displayed but disabled. Make a call   
from the gadget, and the Hang Up button becomes enabled. Click the Hang Up button in the   
gadget to end the call.

CCSP UI Integrations Guide - 7.3 and 7.4   
70   
Transferring calls   
Transferring a call to another destination involves a number of steps that are not complicated but   
have to be followed for the correct result to be achieved. These steps are normally performed using   
the GUI, for example, displaying the Outgoing Call window during a live call, entering the number,   
waiting for the destination to pick up the call and then transferring the call. It is possible to perform   
all these steps with a single button click from the gadget page, in what is known as a one-touch   
blind transfer.   
The three most common scenarios for a call transfer are:   
•   
Transfer to another number   
•   
Transfer to a System Entry Point   
•   
Transfer to another agent   
   
The steps are basically the same because only the initial function calls vary. All three are   
demonstrated with the DemoCRM.js.   
Basic steps in transferring a call   
For a call to be transferred by a gadget, the following must occur:   
•   
The destination number must be dialed and a call established   
•   
The new call (slave) must be associated with the original (master) call   
•   
If the slave call is accepted at the destination, the media for the slave call must be connected   
•   
After the slave call reaches the Active InCall state, it can be bridged with the master call   
   
After the slave call is bridged, the Agent's and therefore the gadget's involvement with the call   
ends.   
Initiating the transfer process can be performed using:   
•   
The Call object   
•   
The dialService   
Transferring to a number using the Call object   
To start a transfer to a number using the Call object, you must use its CallOutPhone() method.   
There are specific methods for transferring to another agent or to system but not for transferring   
to a dialed number. Add another form to the HTML code as below.   
<form title="Transfer via Call" id="transferUsingCallForm"   
submit="transferUsingCall()">   
 <label for="transferDestCall">Number to transfer to</label>   
 <input type="number" placeholder="e.g. 12345" id="transferDestCall"   
/><input type="submit" ng-click="transferUsingCall()" value="Transfer using   
Call Object" />   
</form>

CCSP UI Integrations Guide - 7.3 and 7.4   
71   
As indicated this is used to initiate the Transfer using the Call object. In the DemoCRM.js, the   
transferUsingCall() function is implemented as follows.   
$scope.transferUsingCall = function () {   
 if ($scope.outgoingCall) {   
 transferDestination =   
document.getElementById("transferDestCall").value;   
 $scope.outgoingCall.PutOnHold();   
 $scope.outgoingCall.CallOutPhone(transferDestination);   
 }   
};   
   
This initiates a new call that must now be captured by the gadget when its corresponding event is   
raised in CCSP UI.   
Note   
Before making the call, the outgoing call must be put on hold by the gadget. The NewCall event   
callback function therefore needs to be modified to check for the call being a slave call and   
having a master call with the same Id as the original outgoing call.   
   
else if (true == callEvent.call.CallInfo.IsBridgeSlave &&   
$scope.outgoingCall.CallId == callEvent.call.BridgeMasterCall.CallId) {   
$scope.slaveCall = callEvent.call;   
   
If the call satisfies this criteria, the gadget must now store this as the slave call and subscribe to its   
CallStateChange event to invoke the original call's Bridge() function when it goes to the InCall state.   
eventServiceWrapper.subscribe("CallStateChange", $scope.slaveCall.CallId,   
function (callEvent) {   
 //Only care about the inCall and Destructed states   
 if (callEvent.state == "InCall") {   
 $scope.outgoingCall.Bridge($scope.slaveCall.CallId);   
 } else if (callEvent.state == "Destructed") {   
 $scope.slaveCall = null;   
 }   
});   
$scope.$apply();

CCSP UI Integrations Guide - 7.3 and 7.4   
72   
The slave call only reaches the InCall state if the media is connected. This is usually done by the   
Outgoing Call window but the DemoCRM gadget is bypassing this completely to achieve the one-  
touch transfer. To connect the media, the voipService must be introduced into the demoCRM   
module and controller alongside the other injected services.   
In the same branch as the subscription to the slave call's state change event, add the following line.   
$scope.ConnectCall($scope.slaveCall);   
   
The ConnectCall() function uses the voipService to connect the media to the call as below.   
$scope.ConnectCall = function (call) {   
 if (!agentServiceWrapper.agent.Preferences.IsAlternateDelivery()) {   
 //Connect the SIP device if this is selected in the Agent Preferences   
 if (!("0" == agentServiceWrapper.agent.InitiatePDSCallValue &&   
call.CallInfo.IsPredictive)) {   
 if (("Outgoing" == call.CallInfo.Direction) ||   
(call.CallInfo.IsPredictive)) {   
 voipService.Connect(call);   
 } else {   
 voipService.PreConnect(call);   
 }   
 }   
 } else {   
 //Connect the alternative device if this is selected in the Agent   
Preferences   
 if (("Outgoing" == call.CallInfo.Direction) ||   
(call.CallInfo.IsPredictive)) {   
 voipService.ConnectAlternateDeliveryCall(call);   
 } else {   
 voipService.PreConnectAlternateDeliveryCall(call);   
 }   
 }   
};   
   
The function uses the appropriate voipService functions to connect the call using either the SIP   
devices or the agent's alternative voice device.   
Reload the DemoCRM gadget and initiate a call to a number using either the Call or Dial forms.   
After the call is established, enter a valid number in the Transfer via Call form and press Enter. The   
destination device should ring immediately and because it's a blind transfer, the original call is   
hung up at the Agent.

CCSP UI Integrations Guide - 7.3 and 7.4   
73   
Transferring to a number using the dialService   
The process for transferring a call to another number using the dialService is almost exactly the   
same, the only difference is that it is the dialService's Dial() function that is used to start the   
process. In DemoCRM.html, copy and paste the existing Transfer via Call form and replace "Call"   
with "Dial" in the new form. In DemoCRM.js, do the same for the transferUsingCall() function and   
also replace the CallOutPhone call with the following.   
dialService.Dial("dialpad", "", transferDestination, "", "",   
transferDestination, "", $scope.outgoingCall);   
   
Everything else remains the same. Reload the gadget and repeat the test from the previous section,   
but using the Transfer via Dial form. The call is transferred in the same manner as before.   
Transferring to another agent   
The process to transfer the call to another agent is again exactly the same except for the function   
to start the process. When using the Call object, you can use either of the following functions:   
•   
CallToAgent()   
•   
Transfer() – blind transfer for email and chat calls only, do not use for VoIP.   
   
When using the dialService, you can use either of the following functions:   
•   
Dial()   
•   
Transfer() – blind transfer for email and chat calls only, do not use for VoIP.   
   
The usage is similar to the CallOutPhone() and Dial() calls already created in the DemoCRM gadget.   
The common element with all these functions is the requirement for the destination Agent's global   
Id. This is not available until the transfer list for the transferring agent has been created and   
obtaining this value is the focus of this section.   
Obtaining the list of logged on agents   
The list of agents currently logged on can be obtained by using either of two methods on the Agent   
object:   
•   
AgentsTransferee()   
•   
RefreshAgentList()   
   
The AgentsTransferee() method requests the list of all logged in agents in the tenant. As this could   
be an extremely large number depending on the tenant size, we do not recommend this unless the   
tenant size is known to be small. The preferred method is RefreshAgentList() which allows the   
specification of various parameters to filter the results that are returned.   
For the DemoCRM gadget, the only filter used is the agent name search pattern and the results are   
displayed as a list of agent names. The gadget then allows a name to be selected from the list and   
the associated global Id of the agent is used as the transfer destination value.   
Add the following div element to the DemoCRM.html file.

CCSP UI Integrations Guide - 7.3 and 7.4   
74   
<div>   
 <p>   
 <label for="agentName">Pattern to search for</label><br/>   
 <input type="text" id="agentName" placeholder="e.g. Agent One"   
width="300" />   
 <button ng-click="findAgents()">Find Agents</button>   
 </p>   
 <p>   
 <div id="agentList" style="width: 300px; height: 200px; overflow-y: auto;   
background-color: aliceblue">   
 <ul>   
 <li ng-repeat="agent in foundAgents" style="list-style: none" ng-  
click="selectAgent(agent)">   
 {{agent.name}}   
 </li>   
 </ul>   
 </div>   
 </p>   
 <p>   
 <label for="selectedAgentName">Selected Agent</label>   
 <input type="text" id="selectedAgentName" ng-disabled="true" value="{{   
selectedAgent.name}}" />   
 <button ng-click="transferToAgent()">Transfer to Agent</button>   
 </p>   
</div>   
   
The first input element takes the agent name string to search for and the first button triggers the   
function that creates the RefreshAgentList request. The div contains the list of Agents that is being   
created here using the AngularJS ngRepeat directive. This is being used for convenience as it   
eliminates the need to loop through the returned agent list and create each individual <li> element   
in the JavaScript code. The second input and button pair display the name of the selected agent and   
trigger the transfer.   
In the DemoCRM.js code, add the following scope variables at the top of the demoCRMController()   
function.   
$scope.foundAgents = [];   
$scope.selectedAgent = null;   
var refreshAgentsId = null;   
   
The foundAgents array points to the list of agents received and selectedAgent is the agent selected   
from that list in the gadget. The refreshAgentsId value is the subscription Id for the   
AgentsTransfereeRefreshed event.

CCSP UI Integrations Guide - 7.3 and 7.4   
75   
Define the findAgents() function as below.   
$scope.findAgents = function () {   
 var pattern = document.getElementById('agentName').value;   
 agentServiceWrapper.agent.RefreshAgentList(null, 100, PageDirection.None,   
pattern, 0, 0, "");   
   
 refreshAgentsId =   
eventServiceWrapper.subscribe("AgentsTransfereeRefreshed", null, function   
(data) {   
 eventServiceWrapper.unsubscribe(refreshAgentsId);   
 $scope.foundAgents = [];   
 $scope.foundAgents = data.agentsList;   
 $scope.$apply();   
 });   
};   
   
For details of the RefreshAgentList parameters, see the Agent AJAX API Reference. For the DemoCRM   
gadget, the list is limited to 100 entries and each call is a refresh (paging direction is None). The   
only criteria specified is the search pattern, so all teams, groups and agent states are returned.   
As soon as the request is made, the gadget subscribes to the event that contains the search results.   
As this gadget is only making one-off requests, it doesn't want to be notified of any subsequent   
AgentTransfereeRefreshed events, hence the unsubscribe call. The list of agents for display is then   
updated.   
Reload the DemoCRM gadget and without making an initial call, enter a string in the new text box   
that matches at least one other logged in agent, and then click the Find Agents button. The list of   
matching agents appears in the box below and clicking on a name in the list causes it to appear in   
the non-editable text box underneath.   
Refreshing the list of logged on agents   
It should be noted that the AgentsTransfereeRefreshed event is also used by the Outgoing Call   
window to populate the list of agents in the Users tab. When the window is opened, it makes its   
own RefreshAgentList call based on the currently selected criteria. However, it uses the result of   
any AgentsTransfreeRefreshed event that is raised, regardless of what has prompted it, so the list   
shown in the window can be affected by other uses of RefreshAgentList.   
To see this, open the Outgoing Call window, select the Users tab and then use the Find Agents   
button in the DemoCRM gadget. The Users tab is updated to show the results of the DemoCRM   
request, but should shortly revert back to the expected output. This is because the Users tabs   
periodically sends the RefreshAgentList request. Because the DemoCRM gadget is unsubscribing   
from the AgentsTransfereeRefreshed event, its output is unaffected by the User tab's requests.

CCSP UI Integrations Guide - 7.3 and 7.4   
76   
Completing the transfer to agent   
To actually transfer the call to the destination agent, implement the transferToAgentUsingCall()   
function as below.   
$scope.transferToAgent = function () {   
 if ($scope.outgoingCall && $scope.selectedAgent) {   
 transferDestination = $scope.selectedAgent.agentId;   
 $scope.outgoingCall.PutOnHold();   
 $scope.outgoingCall.CallToAgent(transferDestination);   
 }   
};   
   
It follows the same format as the other transfer functions with the main difference being the use of   
the CallToAgent() function. Reload the gadget and this time make an outgoing call from it. Enter a   
search pattern for an agent and select an appropriate one from the results list. Click the Transfer   
to Agent button and the call rings at the destination agent and is removed from CCSP UI.   
To use the dialService Dial() function, substitute the line below for the Call object CallToAgent()   
statement in transferToAgentUsingCall().   
dialService.Dial("users", $scope.selectedAgent.name, transferDestination, "",   
"", transferDestination, "", $scope.outgoingCall);   
Transferring to system   
Transferring a call to system requires the system entry point Id for the required destination to be   
obtained, so the processing is similar to that for retrieving the agent transferee list, however the   
information is held within the Agent.

CCSP UI Integrations Guide - 7.3 and 7.4   
77   
Obtaining the entry points   
Add the following div element to the DemoCRM.html file.   
<div>   
 <p>   
 <button ng-click="getEntryPoints()">Get System Entry   
Points</button><br/>   
 <div id="systemList" style="width: 300px; height: 200px; overflow-y:   
auto; background-color: aliceblue">   
 <ul>   
 <li ng-repeat="point in foundPoints" style="list-style: none"   
ng-click="selectEntryPoint(point)">   
 {{point.Name}}   
 </li>   
 </ul>   
 </div>   
 </p>   
 <p>   
 <label for="selectedPointId">Selected Entry Point</label>   
 <input type="text" id="selectedPointId" ng-disabled="true" value="{{   
selectedPoint.Name}}" />   
 <button ng-click="transferToSystem()">Transfer to System</button>   
 </p>   
</div>   
   
This produces a similar set of elements as for the Agent Transferee list display and selection. In the   
DemoCRM.js, declare the appropriate scope variables as below and define the implementation of   
the getEntryPoints() function, which is very simple.   
$scope.foundPoints = [];   
$scope.selectedPoint = null;   
…   
$scope.getEntryPoints = function () {   
 $scope.foundPoints = agentServiceWrapper.agent.EntryPointOrganized;   
};

CCSP UI Integrations Guide - 7.3 and 7.4   
78   
Transfer to system using the Call object   
The transferToSystem() function follows the familiar format, with the Call object's CallToSystem()   
function being used.   
$scope.transferToSystem = function () {   
 if ($scope.outgoingCall && $scope.selectedPoint) {   
 transferDestination = $scope.selectedPoint.Id;   
 $scope.outgoingCall.PutOnHold();   
 $scope.outgoingCall.CallToSystem(transferDestination);   
};   
Reload the gadget and make an outgoing call from it. Click the Get System Entry Points button   
and the available entry points for the Agent are displayed.   
Note   
Entry points are only loaded on login, so any changes to the available ones for the Agent require   
the agent to log out of CCSP UI and back in again.   
   
On selection of an entry point, it appears in the text box. Clicking Transfer to System causes the call   
to be transferred to the next available agent associated with the entry point.   
Transfer to system using the dialService   
To use the higher level dialService Dial() function, replace the CallToSystem() line with the following.   
dialService.Dial("system", $scope.selectedPoint.Name, transferDestination,   
$scope.selectedPoint.NeedParameter, $scope.selectedPoint.ParameterValue,   
transferDestination, $scope.selectedPoint.Attendable, $scope.outgoingCall);   
Calling another agent   
To make a normal outgoing call to another agent, you can use either of the following:   
•   
Agent.CallToAgent()   
•   
dialService.Dial()   
   
This is demonstrated by using a single function for the two call methods.   
Add the following buttons to the <p> element containing the "Selected Agent" elements in the   
demoCRM.html.   
<button ng-disabled="outgoingCall != null" ng-click="callAgent('agent')">Call   
Agent</button>   
<button ng-disabled="outgoingCall != null" ng-click="callAgent('dial')">Dial   
Agent</button>

CCSP UI Integrations Guide - 7.3 and 7.4   
79   
The two buttons use a common function with a parameter to specify the calling method to be used.   
As an aside, the ng-disabled directive is an optional addition that is only used here to prevent calls   
being made when an outgoing call is already in progress from the gadget. The callAgent() function   
is then written to choose the appropriate method according to the parameter.   
$scope.callAgent = function (serviceToUse) {   
 if ($scope.selectedAgent) {   
 if (serviceToUse == "agent") {   
   
agentServiceWrapper.agent.CallToAgent($scope.selectedAgent.agentId);   
 } else if (serviceToUse == 'dial') {   
 dialService.Dial("users", $scope.selectedAgent.name,   
$scope.selectedAgent.agentId, "", "", $scope.selectedAgent.agentId, "", null);   
 }   
 }   
};   
   
On reloading the gadget, the difference between using the Call Agent button and the Dial Agent   
button can be seen in the Recent tab of the Outgoing Call window. An agent called using the Dial   
Agent button appears in the recent calls, an agent called with the Call Agent button does not.   
Calling a system entry point   
The code written for calling an agent can be reworked to do a call to system, using the   
dialService.Dial() function or the Agent.CallToSystem() function. The Dial() function uses the same   
values as when transferring to an entry point, apart from the call object parameter which is null.   
The CallToSystem() function takes the entry point Id.   
As an additional exercise, you can produce the appropriate code to allow the DemoCRM gadget to   
make calls to system entry points using the two methods.

CCSP UI Integrations Guide - 7.3 and 7.4   
80   
Attended transfer and conferencing   
The transfers implemented in the DemoCRM gadget thus far have all been blind transfers of voice   
calls. The key functionality is the automatic bridging of the slave call when it reaches the InCall   
state, regardless of whether or not the destination party has accepted it. To add attended transfer   
and conference capability, the gadget just needs to bypass the automatic bridging and accept the   
required action from the user.   
Attended transfer and conferencing with a number   
In the DemoCRM.html code, add the following.   
<p>   
 <label for="autoTransferChecked">Auto Transfer</label>   
 <input id="autoTransferChecked" type="checkbox" ng-  
model="autoTransferEnabled" />   
 <span ng-show="slaveCall != null && !autoTransferEnabled">   
 <button ng-disabled='slaveCall.State != 'InCall'" ng-  
click="bridgeComplete('transfer')">Transfer</button>   
 <button ng-disabled='slaveCall.State != 'InCall'" ng-  
click="bridgeComplete('conference')">Conference</button>   
 <button ng-click="bridgeComplete('cancel')">Cancel</button>   
 </span>   
</p>   
   
This adds a new paragraph of three buttons and a check box. The check box is used to control   
whether or not slave calls are automatically bridged. The buttons provide the Transfer, Conference   
and Cancel actions on the slave call.   
Note   
The check box uses the ngModel directive to bind its value to the new scope variable   
autoTransferEnabled. This variable is used in the span element containing the buttons to control   
whether or not the span and therefore the buttons are shown.   
   
   
In the DemoCRM.js code, the autoTransferEnabled variable should be initialized at the top of the   
demoCRMController and the bridgeComplete() function is implemented to call the required   
function on the Call object.

CCSP UI Integrations Guide - 7.3 and 7.4   
81   
$scope.bridgeComplete = function (op) {   
 if ($scope.outgoingCall != null && $scope.slaveCall != null) {   
 switch (op) {   
 case 'transfer':   
 $scope.outgoingCall.Bridge($scope.slaveCall.CallId);   
 break;   
 case 'conference':   
 $scope.outgoingCall.Conference($scope.slaveCall.CallId);   
 break;   
 case 'cancel':   
 $scope.slaveCall.HangUp();   
 break;   
 default:   
 break;   
 }   
 }   
};   
The autoTransferEnabled flag needs to be used in the CallStateChange handler for the slave call to   
control the automatic bridging that is currently always done.   
if ("InCall" == callEvent.state) {   
 if ($scope.autoTransferEnabled) {   
 $scope.outgoingCall.Bridge($scope.slaveCall.CallId);   
 }   
}   
   
One more change is required before testing. When the slave call is successfully bridged, no   
CallStateChange event is raised to say that it has been destructed. Therefore to tidy up the gadget,   
subscribe to the call's ConferenceStatus event alongside the subscription to CallStateChange.   
eventServiceWrapper.subscribe("ConferenceStatus", $scope.slaveCall.CallId,   
function (eventData) {   
 if (eventData.status == "Success") {   
 $scope.slaveCall = null;   
 }   
 $scope.$apply();   
});

CCSP UI Integrations Guide - 7.3 and 7.4   
82   
With everything in place, reload the DemoCRM to test the functionality. Make an outgoing call to a   
number and wait until it is picked up. Disable Auto Transfer and initiate a transfer to another   
number. The call goes on hold as usual but now the Transfer, Conference and Cancel buttons   
appear while the slave call is made to the transferee number. When the call is picked up by the   
transferee, click the Conference button and a three way conference is now enabled. The Transfer,   
Conference and Cancel buttons should now be hidden again.   
You can repeat the test with the Transfer and Cancel buttons. Transferring and conferencing with   
another agent or an entry point in the gadget works in the same way.   
Note   
It could be said that the Transfer and Conference buttons should not be enabled until the   
transfer destination has picked up the slave call. If the Transfer button is used before the slave   
call is picked up, it is the equivalent of doing a blind transfer. If the Conference button is used,   
this just sets up the conference, which the transfer destination may or may not decide to join.

CCSP UI Integrations Guide - 7.3 and 7.4   
83   
Multiple call handling   
The DemoCRM gadget is so far only dealing with a single outgoing call that has been initiated from   
the gadget itself. This section demonstrates how to deal with multiple calls, in this instance,   
multiple incoming voice calls.   
Note   
This section reinforces the point that gadgets must keep track of the call objects they are   
interested in, because the events that are received for these calls never pass the call object itself,   
only the call Id.   
   
In the DemoCRM.js, an array property is added to the demoCRMController scope to contain the calls   
and this is used to populate a table element on the gadget's page. Each row displays:   
•   
Call Id   
•   
Call Type   
•   
Current State   
•   
Pick Up, Hold, Refuse and Hang Up buttons   
Managing multiple calls   
Add the following declaration to the demoCRMController.   
$scope.incomingCalls = [];   
   
In the callback function for the NewCall event, add the following else if branch within the   
processing for the VOIP calls:   
else if ("Incoming" == callEvent.call.CallInfo.Direction) {   
 $scope.incomingCalls.push({ callId: callEvent.call.CallId, call:   
callEvent.call });   
 eventServiceWrapper.subscribe("CallStateChange", callEvent.call.CallId,   
onIncomingCallStateChange);   
 $scope.$apply();   
}   
   
This registers the gadget's interest in incoming VOIP calls and adds them to the incomingCalls   
array. The callback function onIncomingCallState is implemented as below.

CCSP UI Integrations Guide - 7.3 and 7.4   
84   
onIncomingCallStateChange = function (stateEvent) {   
 if ("Destructed" == stateEvent.state) {   
 var x = 0;   
 while ($scope.incomingCalls[x].callId != stateEvent.callId && x <   
$scope.incomingCalls.length) {   
 x++;   
 }   
 if (x < $scope.incomingCalls.length) {   
 $scope.incomingCalls.splice(x, 1);   
 }   
 }   
 $scope.$apply();   
};   
   
The only explicit processing needed for this event is to remove the call from the array after it has   
been destructed. Changes in the call state are automatically updated in the page due to the   
$apply() call.   
In the DemoCRM.html, add the div element below after the <p> element for the Current Outgoing   
Call State section.   
<div title="Received Voice Calls">   
 <table style="border-style: solid">   
 <thead>   
 <tr>   
 <td>Call Id</td>   
 <td>Call Type</td>   
 <td>State</td>   
 <td>Actions</td>   
 </tr>   
 </thead>   
 <tr ng-repeat="callWrapper in incomingCalls">   
 <td>{{ callWrapper.callId }}</td>   
 <td>{{ callWrapper.call.CallInfo.Type }}</td>   
 <td>{{ callWrapper.call.State }}</td>   
 <td>   
 <span>   
 <button ng-click="pickUpHoldCall(callWrapper.call)">{{   
("Ringing" == callWrapper.call.State || "OnHold" == callWrapper.call.State) ?   
"Pick Up" : "Hold" }}</button>

CCSP UI Integrations Guide - 7.3 and 7.4   
85   
 <button ng-click="hangUpCall(callWrapper.call)">{{   
"Ringing" == callWrapper.call.State ? "Refuse" : "Hang Up" }}</button>   
 </span>   
 </td>   
 </tr>   
 </table>   
</div>   
   
This sets up the table as described before with a minimal border style just to distinguish it on the   
page. The ngRepeat directive populates the table with the data from the call objects themselves   
and the call state determines the appearance of the two buttons.   
The functionality of the buttons must therefore depend on the context in which they are being   
invoked. In the DemoCRM.js, add the definitions for pickUpHoldCall() and hangUpCall as follows.   
$scope.pickUpHoldCall = function (call) {   
 if ('Ringing' == call.State || 'OnHold' == call.State) {   
 call.PickUp();   
 } else {   
 call.PutOnHold();   
 }   
};   
   
$scope.hangUpCall = function (call) {   
 if ('Ringing' == call.State) {   
 call.Refuse();   
 } else {   
 call.HangUp();   
 }   
};   
   
The two functions call the appropriate methods on the Call object according to its state. It is very   
important to invoke the correct method according to the state as for example, attempting to hang   
up a ringing call may cause incorrect behavior.

CCSP UI Integrations Guide - 7.3 and 7.4   
86   
Accounting for Call Id changes   
CCSP only routes a single voice ACD call to an agent at a time. For an agent to receive multiple voice   
calls, they must be transferred from another agent. This means that the Agent actually receives two   
calls: the first is the secondary (slave) call used to set up the transfer, the second is the slave call   
merged with the original one. As it stands, the DemoCRM gadget does not pick up the transition   
between these calls as the Call Id changes, so it must register for that event and take the   
appropriate action.   
eventServiceWrapper.subscribe("CallIdChange", null, function (idEvent) {   
 var x = 0;   
 while ($scope.incomingCalls[x].callId != idEvent.oldCallId && x <   
$scope.incomingCalls.length) {   
 x++;   
 }   
 if (x < $scope.incomingCalls.length) {   
 $scope.incomingCalls[x].callId = idEvent.newCallId;   
 eventServiceWrapper.subscribe("CallStateChange", idEvent.newCallId,   
onIncomingCallStateChange);   
 $scope.$apply();   
 }   
});   
   
The gadget subscribes to the CallIdChange event and updates the inComingCalls element with the   
new Call Id in the callback function. The callback then subscribes to call state changes for that new   
Call Id.   
To test the new functionality, reload the DemoCRM gadget and initiate an ACD voice call. A new row   
appears with the Call information and the action buttons. You can use these buttons to pick up,   
refuse, hold and hang up the call as required. Transferring further calls into the Agent from other   
agents produces new rows.   
As a further exercise, add a transfer button to each row. You can reuse or copy the functions that   
have already been written for performing blind or attended transfers for the single outgoing call to   
do the same for a received call.

CCSP UI Integrations Guide - 7.3 and 7.4   
87   
Chat call handling   
The Demo CRM gadget is only set up to handle incoming and outgoing voice calls. To demonstrate   
how the Call object can be used to interact with chat calls, the Demo Chat Prompt functional gadget   
is used.   
The Demo Chat Prompt Func gadget initially just automatically answers a chat call and requests   
some information from the caller in the form of a yes or no response to a question. It then opens a   
web page in a new browser window with the response received from the caller.   
Note   
Ensure that in the Preferences for the test agent being used, the Answer calls automatically and   
Use greeting check boxes are both unchecked.   
   
At the moment, the DemoChatPromptFunc.js file should contain the following.   
function StartUpDemoChatPrompt() {   
   
 //get reference to root window - the one containing our main modules and   
services   
 var \_root = GetRootWindow();   
   
 //This line obtains the TouchPoint application's angularJS injector which   
provides access to all services and directives   
 var localInjector = \_root.angular.element("[ng-app]").injector();   
   
 //Can now get any service required   
 var loggerService = localInjector.get('loggerService');   
   
 loggerService.WriteLine("Demo Chat Prompt Functional Gadget has been   
loaded", \_root.enghouse.enums.Severity.Debug, "demoChatPromptFunc");   
}   
StartUpDemoChatPrompt();   
   
As for the Demo CRM standard gadget, the eventServiceWrapper needs to be obtained in to   
subscribe to the NewCall event.   
var eventServiceWrapper = localInjector.get('eventServiceWrapper');   
   
Also need to use the agentWrapperService to obtain information from the Agent object.   
var agentServiceWrapper = localInjector.get('agentServiceWrapper');   
   
The gadget stores chat calls in the following array.   
var chatCalls = [];

CCSP UI Integrations Guide - 7.3 and 7.4   
88   
The question and response acknowledgment strings that the gadget sends are as follows.   
var question = "Please answer the following question with yes or no. Do you   
have an existing policy with us?";   
var responseIsYes = "Thank you, please wait a moment while I retrieve your   
information";   
var responseIsNo = "Thank you, would you like to open a new policy with us?";   
var responseIsNeither = "Sorry, I did not understand that response. ";   
   
The eventServiceWrapper is then used to subscribe to the NewCall event with only chat calls being   
processed by this gadget.   
eventServiceWrapper.subscribe("NewCall", null, function (callEvent) {   
 if ("CHAT" == callEvent.call.CallInfo.Type.toUpperCase()) {   
 eventServiceWrapper.subscribe("CallStateChange",   
callEvent.call.CallId, onCallStateChange);   
 eventServiceWrapper.subscribe("NewChatItem", callEvent.call.CallId,   
onNewChatItem);   
 chatCalls.push(   
 {   
 callId: callEvent.call.CallId,   
 call: callEvent.call,   
 questionAsked: false,   
 });   
 loggerService.WriteLine("Demo Chat Prompt Functional gadget has   
subscribed to " + callEvent.call.CallId, \_root.enghouse.enums.Severity.Debug,   
"demoChatPromptFunc");   
 }   
});   
   
The gadget is subscribing to both the Call object's CallStateChange event and its NewChatItem   
event. It subscribes to the first event to be able to manage its call list and respond automatically to   
calls and to the second event to examine the messages from the caller and make a response. Each   
received call is added to the chatCalls array with its Id and a flag to indicate if the caller has been   
asked the question. The loggerService line just provides some useful debug info.

CCSP UI Integrations Guide - 7.3 and 7.4   
89   
Auto answering a chat call and sending a message   
The gadget needs to auto answer the call and send the question, so the onCallStateChange()   
function is implemented as below.   
onCallStateChange = function (stateEvent) {   
 var x = getItemIndex(chatCalls, 'callId', stateEvent.callId);   
 if (x != -1) {   
 switch (stateEvent.state) {   
 case "Ringing":   
 chatCalls[x].call.PickUp();   
 break;   
 case "InCall":   
 if (!chatCalls[x].questionAsked) {   
 chatCalls[x].call.SendChatMessage("Hi, my name is " +   
agentServiceWrapper.agent.Name + ". " + question);   
 chatCalls[x].questionAsked = true;   
 }   
 break;   
 case "Destructed":   
 chatCalls.splice(x, 1);   
 break;   
 default:   
 break;   
 }   
 }   
};   
   
The getItemIndex() function is a convenience function defined in helpers.js for extracting the index   
of a call object from an array by its Id. The call is picked up if the state is currently ringing and after   
it moves to the InCall state, the SendChatMessage() method of the Call object is used to send the   
question to the caller.   
Note   
Only voice calls have the FirstTimeInCall property set to true when a call is moved to the InCall   
state for the first time. So to avoid the question being sent again if the Call is placed then taken   
off hold, the questionAsked flag is checked. This also means that the question is asked if the call   
has been transferred from another agent.

CCSP UI Integrations Guide - 7.3 and 7.4   
90   
Receiving chat messages   
To handle received chat messages for a call the onNewChatItem() function is implemented as   
follows.   
onNewChatItem = function (chatEvent) {   
 if (chatEvent.chatItem.Who == "Caller") {   
 var x = getItemIndex(chatCalls, 'callId', chatEvent.callId);   
 if (x != -1) {   
 if (chatCalls[x].questionAsked) {   
 var response =   
chatEvent.chatItem.Text.match(/\byes\b|\bno\b/i);   
 if (response != null && response.length > 0){   
 var responseText = response[0].toUpperCase();   
 var infoURL = \_root.constants.rootPath +   
   
"app/gadgets/ChatPrompt/chatanswer.html?CallerName=" +   
chatCalls[x].call.CallInfo.Caller.Name + "&answer=" + responseText;   
 window.open(infoURL);   
 if (responseText == "YES") {   
 chatCalls[x].call.SendChatMessage(responseIsYes);   
 } else {   
 chatCalls[x].call.SendChatMessage(responseIsNo);   
 }   
 chatCalls[x].questionAsked = false;   
 } else {   
 chatCalls[x].call.SendChatMessage(responseIsNeither +   
question);   
 }   
 }   
 }   
 }   
};   
   
The implementation is first checking that the message is from the caller then retrieves the call   
object. It uses a regular expression to parse the caller's response and then opens a web page with   
information from the caller and the response itself. It then sends an acknowledgement for the   
response to the caller.   
The web page that is being opened can be anything but for the purposes of this guide, it just   
displays the parameters passed to it.

CCSP UI Integrations Guide - 7.3 and 7.4   
91   
<!DOCTYPE html>   
   
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">   
<head>   
 <meta charset="utf-8" />   
 <title></title>   
</head>   
<body>   
 Caller Name is: <label id="callerName"></label><br />   
 Caller Answer to question is <label id="callerAnswer"></label>   
 <script type="text/javascript">   
 var parameters = location.search.substring(1).split("&");   
 var temp = parameters[0].split("=");   
 var callerName = unescape(temp[1]);   
 temp = parameters[1].split("=");   
 var callerAnswer = unescape(temp[1]);   
 document.getElementById("callerName").innerHTML = callerName;   
 document.getElementById("callerAnswer").innerHTML = callerAnswer;   
 </script>   
</body>   
</html>   
   
With the code and the sample web page in place, test the gadget by re-logging onto the test Agent   
and sending in a chat call. The call is automatically answered and the question sent out to the   
caller. A "yes" or "no" response results in the ChatAnswer page appearing in a new window and an   
acknowledgement being sent. Any other response results in the question being asked again.

CCSP UI Integrations Guide - 7.3 and 7.4   
92   
Email call handling   
Email call classes   
There are three different AJAX API classes that can be passed to gadgets when handling email calls:   
•   
EmailCall: Represents an incoming email call   
•   
EmailCallPart: Represents a reply or forward draft associated with an incoming call   
•   
OutgoingEmailCall: Represents an outgoing email initiated by user   
   
The EmailCall and OutgoingEmailCall classes inherit from the Call class and the EmailCallPart and   
OutgoingEmailCall classes also have similar methods but none of the classes can be used   
interchangeably.   
Custom email standard gadget   
The functionality available to gadgets for creating, editing and handling emails is currently limited   
to the following:   
•   
Basic call handling functionality   
•   
Initiate outgoing emails   
•   
Create reply/forwarded emails   
   
Note   
There is no available CCSP UI service that allows a gadget to interact with an active email, so for   
example, it is not possible to update the From, To, CC Subject fields of an email draft. The gadget   
only has visibility of the email through the EmailCall object, which also does not present an API to   
safely manipulate the email data.   
   
To demonstrate this functionality, create the Custom Email standard gadget. In CCSP AdminPoint,   
create the gadget with the following default property values:   
•   
Allowed Instances: 1   
•   
Default State: Float   
•   
Float Default Size W: 800, H: 400   
•   
Float Default Position X: 0, Y: 450   
•   
Target URL (HTML): app/gadgets/CustomEmail/CustomEmail.html   
   
Add the gadget to the customization that is applicable for the test Agent.   
Create the TouchPoint\App\gadgets\CustomEmail folder and create the files CustomEmail.html and   
CustomEmail.js.   
The Custom Email gadget presents the following:   
•   
List of received email calls   
•   
List of outgoing email calls   
•   
Buttons to initiate new email calls and handle incoming email calls

CCSP UI Integrations Guide - 7.3 and 7.4   
93   
Custom email gadget implementation   
Edit the CustomEmail.html file and add the following code.   
html ng-app="customEmailApp" xmlns="http://www.w3.org/1999/xhtml">   
<head>   
 <title>Custom Email</title>   
 <script type="text/javascript" src="/TouchPoint/Scripts/jquery-  
2.0.1.js"></script>   
 <script type="text/javascript"   
src="/TouchPoint/Scripts/angular.js"></script>   
 <script type="text/javascript"   
src="/TouchPoint/App/common/gadgetInit.js"></script>   
 <script type="text/javascript" src="/TouchPoint/Ajax/helpers.js"></script>   
 <script type="text/javascript" src="customEmail.js"></script>   
</head>   
<body ng-controller="customEmailController">   
</body>   
</html>   
   
This sets customEmailApp as the page's ngApp attribute and customEmailController as the body's   
ngController.   
In the body, insert the code below to create a New Email button to initiate outgoing email calls and   
simple div elements to show two lists, one for outgoing email calls and one for received email calls   
with action buttons.

CCSP UI Integrations Guide - 7.3 and 7.4   
94   
<div>   
 {{ lastActionMessage }}   
</div>   
<br />   
<br />   
<div>Outgoing Emails</div>   
<button ng-click="newEmail()">New Email</button>   
<div ng-repeat="wrapper in outgoingCalls">   
<button ng-disabled='!wrapper.isInCall' ng-  
click="sendOutgoingEmail(wrapper)">Send New Email</button>   
<br />   
</div>   
<br />   
<br />   
<div>Incoming Emails</div>   
<div style="border-color: black; border-width: thin" ng-repeat="wrapper in   
incomingCalls">   
 <label>Email received from: {{wrapper.call.CallInfo.EmailInfo.From.Address   
}}</label><br />   
 <button ng-disabled='wrapper.isInCall' ng-click="pickUp(wrapper)">Answer   
Email</button>   
 <button ng-disabled='wrapper.isInCall' ng-  
click="pickUpInExternalApp(wrapper)">Answer Email Externally</button>   
 <button ng-disabled='wrapper.isInCall' ng-click="refuse(wrapper)">Refuse   
Email</button>   
 <div ng-show="wrapper.isInCall">   
 <button ng-click="createClaimReply(wrapper, 'rejected')">Reject   
Claim</button>   
 <button ng-click="createClaimReply(wrapper, 'accepted')">Accept   
Claim</button>   
 <button ng-click="createClaimForward(wrapper, 'rejected')">Forward   
Claim with Reject</button>   
 <button ng-click="createClaimForward(wrapper, 'accepted')">Forward   
Claim with Accept</button>   
 <button ng-click="defer(wrapper, 'PersonalDefer')">Defer to   
Personal</button>   
 <button ng-click="defer(wrapper, 'SharedDefer')">Defer to   
Shared</button>   
 <button ng-click="discardDraft(wrapper)">Discard Draft</button>   
 <button ng-click="send(wrapper)">Send</button>   
 <button ng-click="hangUp(wrapper)">Hang Up</button>

CCSP UI Integrations Guide - 7.3 and 7.4   
95   
 <button ng-disabled="wrapper.call.State != 'WrapUp'" ng-  
click="wrapUp(wrapper)">Wrap Up</button>   
 </div>   
 <br />   
</div>   
<br />   
   
Edit the CustomEmail.js file and add the following for the initial version.   
var \_root = GetRootWindow();   
   
//create module   
var mainModule = angular.module("customEmailApp", []);   
//declare the services you wish to inject to your module   
   
var directServices = ["loggerService", "eventServiceWrapper",   
"agentServiceWrapper"];   
//use this method to inject all the services above to your module   
\_root.appendServiceProxiesToModule(mainModule, directServices);   
   
//declare your controller with its dependencies - angular or custom ones.   
mainModule.controller("customEmailController", ["$scope", "$window",   
"loggerService", "eventServiceWrapper", "agentServiceWrapper",   
 function ($scope, $window, loggerService, eventServiceWrapper,   
agentServiceWrapper) {   
 loggerService.WriteLine("Custom Email has been opened",   
\_root.enghouse.enums.Severity.Debug, "customEmail");   
   
 $scope.incomingCalls = [];   
 $scope.outgoingCalls = [];   
 $scope.lastActionMessage = "";   
   
 var newCallId = eventServiceWrapper.subscribe("NewCall", null,   
function (callEvent) {   
 });   
 if ("EMAIL" == callEvent.call.CallInfo.Type.toUpperCase()) {   
 if ("Incoming" == callEvent.call.CallInfo.Direction) {   
 $scope.incomingCalls.push(   
 {   
 callId: callEvent.call.CallId,   
 call: callEvent.call,

CCSP UI Integrations Guide - 7.3 and 7.4   
96   
 isInCall: false,   
 isRepliedTo: false,   
 replyType: "",   
 isForwarded: false,   
 isDeferred: false   
 });   
 $scope.$apply();   
 eventServiceWrapper.subscribe("CallStateChange",   
callEvent.call.CallId,   
 function (event) { onCallStateChange(event,   
$scope.incomingCalls) });   
 } else {   
 $scope.outgoingCalls.push(   
 {   
 callId: callEvent.call.CallId,   
 call: callEvent.call,   
 isInCall: false   
 });   
 eventServiceWrapper.subscribe("CallStateChange",   
callEvent.call.CallId,   
 function (event) { onCallStateChange(event,   
$scope.outgoingCalls) });   
 }   
 }   
   
 $window.onunload = function () {   
 loggerService.WriteLine("Demo Custom Email has been closed",   
\_root.enghouse.enums.Severity.Debug, "customEmail");   
 eventServiceWrapper.unsubscribe(newCallId);   
 };   
 }]);   
   
This sets up the customEmailApp and customEmailController for the gadget and subscribes to the   
NewCall event, listening specifically for email calls. Every incoming email call received is added to   
the scope's incomingCalls array for display in the gadget with a number of properties that are used   
later. Outgoing emails calls are added to the equivalent outgoingCalls array.   
Implement the onCallStateChange() function as follows.   
onCallStateChange = function (stateEvent, callArray) {   
 var x = getItemIndex(callArray, "callId", stateEvent.callId);   
 if (x != -1) {

CCSP UI Integrations Guide - 7.3 and 7.4   
97   
 if ("InCall" == stateEvent.state) {   
 callArray[x].isInCall = true;   
 } else if ("Destructed" == stateEvent.state) {   
 callArray.splice(x, 1);   
 }   
 }   
 $scope.$apply();   
   
Test the initial version by logging onto the test Agent, opening the gadget and sending in an email.   
When the email call rings in CCSP UI, an entry appears in the gadget showing the email sender's   
address and the two answer buttons. When the email is picked up from the call card, the rest of the   
call handling buttons for the email appear in the gadget.   
Making new email calls   
Outgoing Email calls are initiated using the Agent function InitiateNewOutgoingEmail(). Implement   
the gadget's newEmail() function as below.   
$scope.newEmail = function () {   
 agentServiceWrapper.agent.InitiateNewOutgoingEmail();   
};   
   
Now when the gadget is reloaded, activating the New Email button opens a new email call.   
Note   
The NewCall event that is received by CCSP UI in response to this action contains an   
OutgoingEmailCall object.   
Answering and refusing email calls   
   
Received email calls are picked up and refused using the EmailCall object's PickUp and Refuse   
functions, so the event handles for the corresponding buttons in the gadget can be implemented   
as below.   
$scope.pickUp = function (wrapper) {   
 wrapper.call.PickUp();   
};   
   
$scope.refuse = function (wrapper) {   
 wrapper.call.Refuse();   
};   
   
When the gadget is reloaded an email call is received, the Answer and Refuse buttons can be used   
instead of the call card.

CCSP UI Integrations Guide - 7.3 and 7.4   
98   
Hanging up email calls   
Caution   
The HangUp() function can be used to end an email call, however this should be used with   
caution. Because the function is being called directly on the AJAX API layer object, it bypasses   
CCSP UI, so calling HangUp() immediately terminates the call. No option is presented to the user   
to save any draft for outgoing emails and email call parts, or defer the email for an incoming   
email.   
If a gadget calls the HangUp() function itself, then it should also manage the UI aspect. This is   
especially important when hanging up on incoming calls with active call parts because the   
command is rejected by the server and may leave the email call in an indeterminate state.   
   
Because the Custom Email gadget is not currently managing outgoing email calls, it only allows the   
user to hang up on incoming emails. The gadget's hangUp() function is implemented as follows.   
$scope.hangUp = function (wrapper) {   
 if (wrapper.call.CallParts.length == 0) {   
 wrapper.call.HangUp();   
 $scope.lastActionMessage = "Hung up email from " +   
wrapper.call.CallInfo.EmailInfo.From.DisplayName;   
 } else {   
 $scope.lastActionMessage = "Active Call Parts present, cannot hang up   
email from " + wrapper.call.CallInfo.EmailInfo.From.DisplayName;   
 }   
};   
   
When the gadget is reloaded and an incoming email is answered, activating the email's Hang Up   
button in the gadget immediately terminates the call.   
Note   
Replies and forwarding emails are represented as EmailCallPart objects and are accessible from   
the original email call as elements of its CallParts array property. The gadget's hangUp() function   
uses this to check if a reply or forwarding email has been initiated for the call. If so, the HangUp   
command is not sent.   
   
The HangUp() method is also used to end an outgoing email call. In the Custom Email gadget, its   
hangUpOutgoingEmail() function can be implemented as follows.   
$scope.hangUpOutgoingEmail = function (wrapper) {   
 wrapper.call.HangUp();   
};   
   
When the gadget is reloaded and a new email call is initiated, activating the email's Hang Up   
button in the gadget immediately discards the draft email and ends the call.

CCSP UI Integrations Guide - 7.3 and 7.4   
99   
Creating reply and forwarding emails   
EmailCall objects (incoming email) have Reply and Forward methods. These are used to initiate   
replies and forwarded emails and they allow for the setting of the Subject line.   
For example, the following creates a reply to the email represented by a call with the Subject "New   
Subject Line".:   
call.Reply("New Subject Line");   
   
The Custom Email gadget provides the buttons Accept Claim and Reject Claim for each email, which   
each create a reply draft with the appropriate response appended to the original email's subject   
line. The ngClick event handler, createClaimReply is implemented as follows.   
$scope.createClaimReply = function (wrapper, replyType) {   
 if (!wrapper.repliedTo) {   
 wrapper.call.Reply("RE: " + wrapper.call.CallInfo.EmailInfo.Subject +   
": Claim is " + replyType.toUpperCase(), true);   
 wrapper.repliedTo = true;   
 wrapper.replyType = replyType;   
 }   
 $scope.lastActionMessage = "Created " + replyType + " claim reply email   
for " + wrapper.call.CallInfo.EmailInfo.From.DisplayName;   
};   
   
The same is done for the Forward Claim with Reject and Forward Claim with Accept buttons' event   
handler, creteClaimForward.   
$scope.createClaimForward = function (wrapper, forwardType) {   
 if (!wrapper.isForwarded) {   
 wrapper.call.Forward("FW: " + wrapper.call.CallInfo.EmailInfo.Subject   
+ ": Recommend that claim is " + forwardType.toUpperCase(), true);   
 wrapper.isForwarded = true;   
 }   
 $scope.lastActionMessage = "Created " + forwardType + " claim forwarding   
email for " + wrapper.call.CallInfo.EmailInfo.From.DisplayName;   
};   
   
On reloading the gadget and sending in an email, it is now possible to create replies and   
forwarding emails for the call by using the Accept, Reject and Forward… buttons.

CCSP UI Integrations Guide - 7.3 and 7.4   
100   
Sending emails   
Caution   
Attempting to send any new, reply or forwarding email when no recipient has been provided,   
causes the server to reject the request. Because CCSP UI is being bypassed, this may leave the   
email in an indeterminate state if the rejection is not handled correctly, it may not be possible to   
hang up the email call or re-send the email draft, and may prevent the Agent from logging out.   
We therefore strongly recommend that any gadget that needs to send an email, performs checks   
for valid email addresses in the To, CC and BCC fields, before attempting to send the email.   
   
Emails are sent using either respective Send() method of either the OutgoingEmailCall or   
EmailCallPart object, as follows.   
outgoingCall.Send(emailData);   
callPart.Send();   
   
Just as with all the Call objects, there is only one instance of an EmailCallPart object for a   
reply/forwarding email in the system. Therefore activating the Send() function of the call part sends   
whatever information is present in the email draft at the time. Conversely, any change to the email   
in the UI updates the EmailCallPart object.   
Sending reply and forwarding emails   
To send a reply or forwarding email, use the Send() method of the corresponding EmailCallPart   
object. The easiest way to access this object is from the owning call's CallParts array property.   
Because it is essential that there is at least one recipient in the To field, the Custom Email gadget   
must examine the EmailCallParts emailData property to check that one exists   
This is shown in the implementation for the Custom Email's Send button click event handler.   
$scope.send = function (wrapper) {   
 if (wrapper.call.CallParts.length > 0) {   
 var callPart = wrapper.call.CallParts[0];   
 if (callPart.emailData.To.EmailAddress !== undefined &&   
callPart.emailData.To.EmailAddress.length != 0) {   
 callPart.Send();   
 $scope.lastActionMessage = "Attemping to send call part for email   
from " + wrapper.call.CallInfo.EmailInfo.From.DisplayName;   
 } else {   
 $scope.lastActionMessage = "Cannot send call part for email from "   
+ wrapper.call.CallInfo.EmailInfo.From.DisplayName;   
 }   
 }   
};

CCSP UI Integrations Guide - 7.3 and 7.4   
101   
The EmailAddress property can also be checked in the BCC and CC properties of emailData. It is an   
array of EmailAddress objects which are documented in the AJAX API Reference.   
Test this by reloading the gadget and answering an email call in the gadget. Initiate a reply to the   
email using one of the Accept or Reject buttons, enter some text in the reply draft and then use the   
gadget's Send button to send the reply. You can also test this with forwarded emails because the   
check on the To field prevents the email from being sent with no recipient.   
Sending new emails   
The Send() function of the OutgoingEmailCall class requires the data that defines the email to be   
specified as a parameter. Unless the contents of the email are coming from another application   
outside of CCSP UI, this email data should simply be the CallInfo.EmailInfo property of the call   
object. The Custom Email gadget implements the sendOutgoingEmail() function as below.   
$scope.sendOutgoingEmail = function (wrapper) {   
 if (wrapper.call.CallInfo.EmailInfo.To.EmailAddress !== undefined &&   
wrapper.call.CallInfo.EmailInfo.To.EmailAddress.length != 0) {   
 $scope.lastActionMessage = "Sending email to " +   
wrapper.call.CallInfo.EmailInfo.To.EmailAddress[0].Address;   
 wrapper.call.Send(wrapper.call.CallInfo.EmailInfo);   
 } else {   
 $scope.lastActionMessage = "Could not send email";   
 }   
};   
   
It makes the check for a valid Email Address before calling send with the call's EmailInfo. Because   
there is only one instance of the object, this contains whatever has been added from the UI up to   
that point.

CCSP UI Integrations Guide - 7.3 and 7.4   
102   
Saving drafts   
The OutgoingEmailCall and EmailCallPart objects both have SaveDraft() functions which are   
intended to be used to save emails in progress before they are sent. However, there is currently no   
way to retrieve a draft from the server when saved, therefore we strongly recommend that for now,   
these functions are not used by gadgets.   
Discarding reply/forwarding drafts   
To abort a reply or forwarding email that is in progress, use the corresponding EmailCallPart   
object's DiscardDraft() method.   
In CustomEmail.js, implement the ngClick event handler for the incoming call Discard button as   
follows.   
$scope.discardDraft = function (wrapper) {   
 if (wrapper.call.CallParts.length > 0) {   
 wrapper.call.CallParts[0].DiscardDraft();   
 $scope.lastActionMessage = "Discarding draft for for email from " +   
wrapper.call.CallInfo.EmailInfo.From.DisplayName;   
 } else {   
 $scope.lastActionMessage = "Cannot discard draft for email from " +   
wrapper.call.CallInfo.EmailInfo.From.DisplayName;   
 }   
};   
   
Reload the gadget, accept an email call, create a reply to the email, and click the corresponding   
Discard button for the email. The reply is deleted.   
Note   
Drafts for outgoing email calls are discarded by hanging up the call so there is no Discard()   
method for the OutgoingEmailCall class.

CCSP UI Integrations Guide - 7.3 and 7.4   
103   
Deferring emails   
Received emails can be deferred to be processed by either the Agent receiving or another Agent at   
a later date. The operation is done by using the EmailCall.Defer() method, passing either   
'PersonalDefer' or 'SharedDefer' for the destination folder.   
For the Custom Email gadget, implement the Defer… buttons' ngClick event handler as follows.   
$scope.defer = function (wrapper, deferType) {   
 if (!wrapper.isDeferred) {   
 wrapper.call.Defer(deferType);   
 wrapper.isDeferred = true;   
 $scope.lastActionMessage = "Deferred to " + deferType + " email from "   
+ wrapper.call.CallInfo.EmailInfo.From.DisplayName;   
 }   
};   
   
Reload the gadget, send in an email call, and click either of the Defer… buttons. When deferred, the   
email is visible in the Message View gadget.   
Getting attachments from incoming emails   
Attachments are held in the EmailInfo property of the EmailCall object. It is an object that contains   
either a single AttachmentInfo object or an array of AttachmentInfo objects if more than one. The   
AttachmentInfo object is described in the AJAX API Reference.   
To see this in the Custom Email gadget, add the following code to CustomEmail.html in the div   
element that appears when the email is in call.   
<div>   
 Attachments:   
 <ul>   
 <li ng-repeat="attachment in wrapper.attachments">   
 {{ attachment.Filename }}: {{ attachment.FileSize }}   
 </li>   
 </ul>   
</div>   
   
Then in the CustomEmailjs add the following line to the "InCall" branch of the onCallStateChange()   
function.   
callArray[x].attachments = (callArray[x].call.CallInfo.EmailInfo.Attachments   
!== undefined) ?   
fixArray(callArray[x].call.CallInfo.EmailInfo.Attachments.AttachmentInfo) :   
[];

CCSP UI Integrations Guide - 7.3 and 7.4   
104   
The fixArray function is from helpers.js and is used to return an array object if the input parameter   
is actually just a singular object.   
Test the updated gadget with an incoming email containing an attachment. When the email is   
picked up, the attachment info appears with it.   
Downloading attachments   
Attachments can be downloaded using the GetAttachment() method of the EmailCall object and   
requires the Id property of the AttachmentInfo object, for example:   
call.GetAttachment(isEmbedded, attachment.Id);   
   
Note   
The attachment's Id property cannot be used directly. It must be encoded using the browser's   
encodeURIComponent() function to enable it to be passed through the AJAX API.   
   
Add a Download button to the attachment <li> element in the CustomEmail.html file as below.   
<button ng-click="getAttachment(wrapper, attachment)">Download</button>   
   
Implement the getAttachment() function in CustomEmail.js as follows.   
$scope.getAttachment = function (wrapper, attachment) {   
 wrapper.call.GetAttachment(false, encodeURIComponent(attachment.Id));   
};   
   
Reload the gadget and test with another incoming email containing an attachment. Click the   
Download button. The attachment is retrieved it from the server.   
Adding attachments   
Adding an attachment to an email relies on the CCSP UI due to the selection of the attachment   
using a file dialog. It is therefore not advised to attempt this from a gadget, unless that uses a   
similar mechanism. If this is really required, please consult with Professional Services.   
Hiding the email window   
There may be occasions when the email calls are actually handled by external applications, in which   
case the email call is answered in CCSP UI but the email is not actually displayed. It is possible to   
hide the Email window from a gadget. It requires a subscription to the ShowEmailCall event and the   
use of the GadgetComm.closeComponentWindow() function which is described in Section   
GadgetComm functions.   
This is demonstrated in the Custom Email gadget by implementing the ng-click event handler for   
the Answer Email Externally button and adding a simple emailEdit.html file to display the caller and   
subject line.   
In the CustomEmail.js file, add the following properties to the object added to the incomingCalls   
array.

CCSP UI Integrations Guide - 7.3 and 7.4   
105   
$scope.incomingCalls.push(   
 …   
 editInExternalApp: false,   
 windowId: "",   
 externalWindow: null   
   
Implement the function pickUpInExternalApp(), which is the handler for the Answer Email   
Externally button as below.   
$scope.pickUpInExternalApp = function (wrapper) {   
 wrapper.editInExternalApp = true;   
 wrapper.call.PickUp();   
};   
   
This is simply setting the flag to be examined in the event handler for ShowEmailCall. The gadget   
should subscribe to this event after subscribing to NewCall and the callback functionality should be   
as follows.   
var showEmailId = eventServiceWrapper.subscribe("ShowEmailCall", null,   
function (showEvent) {   
 //Close the window   
 var x = getItemIndex($scope.incomingCalls, "callId",   
showEvent.call.CallId);   
 if (x != -1 && $scope.incomingCalls[x].editInExternalApp) {   
 //Only hide the call window the first time round, it is   
 //displayed if user activates it from the call card again   
 if ($scope.incomingCalls[x].windowId == "") {   
 $scope.incomingCalls[x].windowId = showEvent.windowId;   
 \_root.app.GadgetComm.closeComponentWindow(showEvent.windowId);   
 var editURL = \_root.constants.rootPath +   
 "app/gadgets/CustomEmail/emailEdit.html?from=" +   
$scope.incomingCalls[x].call.CallInfo.EmailInfo.From.Address + "&subject=" +   
 $scope.incomingCalls[x].call.CallInfo.EmailInfo.Subject;   
 $scope.incomingCalls[x].externalWindow = $window.open(editURL);   
 }   
 }   
});

CCSP UI Integrations Guide - 7.3 and 7.4   
106   
The callback function checks if the call should be externally opened then stores the event's   
windowId. It then uses the closeComponentWindow() function to close the Email window.   
Note   
The returned id from the subscribe call is stored and should be used to unsubscribe from the   
event in the $window.onunload() function.   
   
The gadget then opens a new window passing in information from the call to simulate an external   
application handling it, although this can be anything that is required by the customization. For   
completeness, example code for the emailedit.html file is presented below.   
<!DOCTYPE html>   
   
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">   
<head>   
 <meta charset="utf-8" />   
 <title></title>   
</head>   
<body>   
 From: <label id="from"></label><br />   
 Subject: <label id="subject"></label>   
 <script type="text/javascript">   
 var parameters = location.search.substring(1).split("&");   
   
 var temp = parameters[0].split("=");   
 var from = unescape(temp[1]);   
 temp = parameters[1].split("=");   
 var subject = unescape(temp[1]);   
 document.getElementById("from").innerHTML = from;   
 document.getElementById("subject").innerHTML = subject;   
 </script>   
</body>   
</html>   
   
Test this functionality by reloading the gadget and sending an email into the Agent. Click the   
Answer Email Externally button to pick up the email. In CCSP UI, the email is still briefly displayed   
before the new browser window is opened, showing the email's From and Subject parameters.   
Back in CCSP UI, click anywhere on the call card to display the email again.

CCSP UI Integrations Guide - 7.3 and 7.4   
107   
Wrapping up calls   
Wrap up of calls is completed by calling the FinishWrapUp() method of the Call object, which has   
the following signature.   
FinishWrapUp(wrapUpCodeId, comment)   
   
The wrapUpCodeId is the Identifier for the Wrap Up code that is part of the owning Agent's   
WrapupCodesList array property. This function is demonstrated using the Custom Email gadget   
written in the last section, although the functionality applies to all call objects, irrespective of their   
type.   
In the CustomEmail.html, the Wrap Up button for received email call is defined as below.   
 <button ng-disabled="wrapper.call.State != 'WrapUp'" ng-  
click="wrapUp(wrapper)">Wrap Up</button>   
   
The use of the call's State property is to ensure that the Wrap Up button is only enabled in the   
correct call context. In the CustomEmail.js file, implement the wrapUp() function as below.   
$scope.wrapUp = function (wrapper) {   
 var comment = "Claim from " + wrapper.call.CallInfo.EmailInfo.From.Address   
+ " has been ";   
 comment+= wrapper.replyType != "" ? wrapper.replyType : "deferred";   
 if (agentServiceWrapper.agent.WrapupCodesList.length > 0) {   
   
wrapper.call.FinishWrapUp(agentServiceWrapper.agent.WrapupCodesList[0].code,   
comment);   
 } else {   
 wrapper.call.FinishWrapUp(0, comment);   
 }   
};   
   
The function uses the first entry in the Agent object's WrapupCodesList array to provide the Id and   
creates a comment based on whether or not the email has been replied to.   
To test this, ensure that the test Agent has Wrap Up codes available, reload the Custom email   
gadget and accept an incoming email call. Hang up the call and when the Wrap Up window   
appears, click the Wrap Up button for the call in the Custom Email window. The Wrap Up window is   
closed and the call is finished.

CCSP UI Integrations Guide - 7.3 and 7.4   
108   
Raising alerts and prompts   
Any alerts and prompts to the Agent from a gadget should be displayed using the dialogService   
instead of the default JavaScript alert function. This avoids the CCSP UI application being blocked   
while waiting for the user to respond. Prompts and alerts are displayed using the function   
showMessageBox() (see dialogService), which has the following signature.   
showMessageBox(title, message, messageBoxButtons, resultAction, opts)   
   
Note   
The opts parameter is not mandatory.   
   
In the Demo CRM gadget, add the following div element to the DemoCRM.html file.   
<div>   
 <form title="Display Alert" id="alertUser" submit="displayAlert()">   
 <label for="displayAlert">Enter Alert title</label>   
 <input type="text" placeholder="e.g. System Message" id="alertTitle"   
/>   
 <br />   
 <label for="alertButtons">Select buttons</label>   
 <br />   
 <input type="radio" name="alertButtons" value="OK" checked />OK<br />   
 <input type="radio" name="alertButtons" value="OKCancel" />OK,   
Cancel<br />   
 <input type="radio" name="alertButtons" value="YesNo" />Yes, No<br />   
 <input type="radio" name="alertButtons" value="YesNoCancel" />Yes, No,   
Cancel<br />   
 <label for="alertText">Enter Alert text</label>   
 <input type="text" placeholder="e.g. Message for the user"   
id="alertMessage" />   
 <br />   
 <input type="submit" ng-click="displayAlert()" value="Display Alert"   
/>   
 </form>   
 <label id="alertResult"></label>   
</div>   
   
This creates a form that takes a title and message for the alert prompt and provides radio buttons   
to select the button format to be displayed. The actual button clicked from the alert is then shown.

CCSP UI Integrations Guide - 7.3 and 7.4   
109   
In the DemoCRM.js file, implement the displayAlert() function as below.   
$scope.displayAlert= function () {   
 var buttons = {};   
 var selected = "OK";   
 var checked = document.querySelector('input[name=alertButtons]:checked');   
 if (checked != null){   
 selected = checked.value;   
 }   
 switch (selected)   
 {   
 case 'OK':   
 buttons = \_root.enghouse.enums.MessageBoxButtons.OK;   
 break;   
 case 'OKCancel':   
 buttons = \_root.enghouse.enums.MessageBoxButtons.OKCancel;   
 break;   
 case 'YesNo':   
 buttons = \_root.enghouse.enums.MessageBoxButtons.YesNo;   
 break;   
 case 'YesNoCancel':   
 buttons = \_root.enghouse.enums.MessageBoxButtons.YesNoCancel;   
 break;   
 default:   
 buttons = \_root.enghouse.enums.MessageBoxButtons.OK;   
 break;   
 }   
 dialogService.showMessageBox   
 (document.getElementById('alertTitle').value,   
 document.getElementById('alertMessage').value,   
 buttons,   
 function (result) {   
 document.getElementById('alertResult').innerHTML = "The " + result   
+ " button was clicked";   
 });   
};   
   
The function gets the selected radio button and assigns the corresponding enum value from   
MessageBoxButtons. It then calls the showMessageBox() function with the inputs from the form.

CCSP UI Integrations Guide - 7.3 and 7.4   
110   
To test this, reload the gadget, enter text in the title and message fields, and click the Yes, No   
button. Click on Display Alert and the prompt appears with the selected configuration and text.   
Click on a button in the prompt to close the window. The result appears below the form.   
To close the prompt without any user input, use the closeModal() function of the dialogService. This   
is demonstrated in the Demo CRM gadget by adding another button to the form as follows.   
<button ng-click="displayAndCloseAlert()">Display and Close Alert</button>   
   
Implement the displayAndCloseAlert() function as below.   
$scope.displayAndCloseAlert = function () {   
 $scope.displayAlert();   
 setTimeout(function () { dialogService.closeModal() }, 5000);   
};   
Reload the gadget and click the Display and Close Alert button. The alert appears and then closes   
after 5 seconds.   
System gadget customization   
AngularJS Directives   
CCSP UI is an AngularJS based application and the windows that are displayed are composed of   
directives. Each system gadget is defined by a directive and windows that aren’t gadgets, such as   
the Wrap Up window and call card are also directives. Each directive consists of:   
•   
Presentation: an HTML file that provides the template or view   
•   
Business Logic: a javscript file that provides the controller   
   
The controller contains the functionality that does all the background work, using services, events,   
view models, and AJAX API as required.   
Overriding CCSP UI directives   
Starting from 1.5.X, AngularJS provides the capability to override directive functionality in run time.   
You can use this capability to override the factory code of a directive and replace it with customized   
code. For example, you can replace the factory implementation of a system gadget with different   
functionality.   
To modify system gadget functionality:   
1.   
Identify the HTML and javascript files for the UI component you wish to override   
2.   
Create customized HTML and javascript versions of these files for the new functionality.   
3.   
Load the customized code in a similar way to the following example. This code must be added   
as a functional gadget to execute it when CCSP UI starts.

CCSP UI Integrations Guide - 7.3 and 7.4   
111   
The example below shows how the HTML and javascript for the chatMessages directive would be   
replaced by a customized version.   
   
//This component replaces html with a different one   
(function () {   
 'use strict';   
   
angular.module("app.main").customCompileProvider.component("chatMessages",   
 {   
 templateUrl: window.constants.rootPath +   
"App/gadgets/myChat/myChat.html",   
 bindings:   
 {   
 chatObject: '=',   
 timeFormats: '<'   
 },   
 controller: function ($scope) {   
 var self = this;   
 // Custom code goes in here   
 }   
 });   
})();   
   
//This function is a delegate, which is responsible for selecting which   
//directive to use   
angular.module("app.main").customProvide.decorator('chatMessagesDirective',   
 [   
 '$delegate',   
 function ($delegate) {   
 var chosenComponent = $delegate[0];   
 for (var i = 0; i < $delegate.length; i++) {   
 if ($delegate[i].templateUrl.includes("myChat"))   
 { chosenComponent = $delegate[i]; break; }   
 }   
 return ([chosenComponent]);   
 }   
 ]);

CCSP UI Integrations Guide - 7.3 and 7.4   
112   
Overriding CCSP UI HTML only   
If you just want to disable a specific button from a system gadget or window, remove it altogether,   
or make some other change that only affects the view and not the controller functionality, you can   
just variant the HTML file without touching the controller script file. You can just replace the   
templateUrl property of the directive with a version of the HTML file that has your change in it.   
For example, if a customer doesn’t want the Callback button to be shown in the Wrap Up window,   
you can create a version of the wrapup.html file without that button element and place it in another   
location within the TouchPoint folder structure (for example,   
App/gadgets/templateOverrides/wrapup.html). You can then create a functional gadget in CCSP   
AdminPoint and add it to the required customization.   
The only code required in its script file is shown below.   
angular.module("app.main").customProvide.decorator('wrapupDirective',   
 [   
 '$delegate',   
 function ($delegate) {   
 var chosenComponent = $delegate[0];   
 chosenComponent.templateUrl = window.constants.rootPath +   
"App/gadgets/templateOverrides/wrapup.html";   
 return $delegate;   
 }   
 ]);   
   
On agent login, this script replaces the “view” that was originally associated with the Wrap Up   
window with the customized one that does not contain the Callback button.

CCSP UI Integrations Guide - 7.3 and 7.4   
113   
Overriding CCSP UI CSS   
You can also disable or remove parts of the UI or modify any of the visual aspects by changing the   
CSS style associated with the specific DOM elements. You can identify the DOM element you wish to   
modify and its associated CSS selector by inspecting the CCSP UI using the browser’s Developer   
Tools. Activate Developer Tools in the browser by any of the following methods:   
•   
Pressing F12 (All major browsers)   
•   
Selecting the option from the Settings menu (All major browsers)   
•   
CTRL-Shift-I (Chrome and Firefox)   
   
Within the Developer Tools, use the arrow button in the leftmost tab for Chrome, Firefox, and Edge   
to inspect individual DOM elements in the CCSP UI application. Clicking on an element will display   
the HTML and all the CSS styles, including those that are inherited.   
After the CSS selector for an element has been identified, you can use this to modify the style using   
code in a functional gadget. For example, to change the background color of CCSP UI, you can use   
this basic javascript code:   
var matches = document.querySelectorAll('div.mainFrameHolder');   
if (matches.length > 0) {   
 matches[0].style.backgroundColor = 'palevioletred';   
}   
   
You obtain the CSS selector for the background using Developer Tools and then it’s very simple to   
update the style of the node element. When combined with the use of events and data from the   
AJAX API Agent and Call objects, you can use this to create powerful UI customizations.   
CSS in gadgets and windows   
Static elements such as the CCSP UI background or gadgetory bar are created upon login and   
therefore the CSS for these can be immediately modified by code in a functional gadget. Elements   
that are part of a system gadget or a window such as the Wrap Up window do not exist until that   
gadget or window is displayed.   
To modify an element that is part of a gadget or window, add a subscription to an event that is   
raised when that window has been displayed and in the callback function for the event, perform the   
required changes on the DOM element which should now be defined.   
This can be demonstrated by extending the Callback removal functionality to hide the Callback   
button from the call card. Using Developer Tools, the CSS selector for the Callback button in the call   
card can been identified as 'div.smallIcon.callback'. However, no call card is displayed until a call is   
actually in the system, so we need to subscribe to an appropriate event that will indicate when it   
has been created.

CCSP UI Integrations Guide - 7.3 and 7.4   
114   
In our functional gadget, we subscribe to the callMenuOpenOrClose event and implement the CSS   
modification in the callback function:   
var localInjector = \_root.angular.element("[ng-app]").injector();   
var eventServiceWrapper = localInjector.get("eventServiceWrapper");   
   
eventServiceWrapper.subscribe("callMenuOpenOrClose", null, onShowCallCard)   
function onShowCallCard(data) {   
 if (data.callMenuVisible === true) {   
 var matches = document.querySelectorAll('div.smallIcon.callback');   
   
 if (matches.length > 0) {   
 for (var x = 0; x < matches.length; x++) {   
 matches[x].style.display = "none";   
 }   
 }   
 }   
})   
Note   
•   
In this instance we loop through the matches to ensure that we update all the occurrences of   
the Callback button if multiple call cards are displayed. If further call cards are created after   
the initial opening of the Call Menu, the Callback button will be displayed in each one, so   
subscribe to the additional focusCall event to ensure this is covered.   
eventServiceWrapper.subscribe("focusCall", "VOIP", onShowCallCard)   
   
•   
The focusCall event requires the Call Type to be specified as the context.

CCSP UI Integrations Guide - 7.3 and 7.4   
115   
Location of CCSP UI directives   
The javascript files for all CCSP UI directives are located in the folder structure under   
app\common\directives. Directives for gadgets and windows have an html file defined in the   
templateUrl property of the directive function and are named for the window or gadget they   
represent. Lower level directives that are basically just library functions do not have this. The html   
files for the directives are located in app\common\views.   
Note   
Most of the directives for system gadgets and windows are fairly large, with the following   
exceptions:   
•   
The Chat window is broken up into different directives for the header, top, messages, and   
bottom parts.   
•   
The Outgoing Call gadget is divided into separate directives for each of its tabs.   
Managing Release updates to CCSP UI directives   
With ongoing updates and improvements to the CCSP UI application, it is possible that the original   
directive on which a customized version is based is updated. Check any updates and merge them   
into the customized version where applicable to avoid gaps in functionality occurring. We   
recommend that you develop customized directives within a well-defined change control process   
so that you can merge updates to the core code in an efficient manner.

CCSP UI Integrations Guide - 7.3 and 7.4   
116   
Reading ConfigItem data   
Almost every piece of data that is received by CCSP UI from the Agent Web Services via the CCSP   
Agent AJAX API, is part of a ConfigItem object. This object is a node structure that contains the   
following properties:   
•   
Id (string) and either or both of:   
o   
Value (string)   
o   
Item (a single ConfigItem or array of ConfigItems)   
   
The flexibility of this structure enables the receipt of complex data through the API such as the   
Agent and Call objects.   
Example ConfigItem   
You may encounter ConfigItem objects if you examine the optional parameter property of the Call   
object or if you request Config data via the CCSP Agent AJAX API methods, for example Release   
Codes and Wrap Up Codes. Requesting the Wrap Up Code config triggers the   
GetWrapUpCodesSuccess event. In the success handler for this event, you receive a ConfigItem that   
contains the wrap up codes. If only one wrap up code has been defined, it is a single item,   
otherwise it is an array of ConfigItems as seen in the example below. The example shows an array   
containing 2 wrap up codes and is in JSON format.   
   
[   
 {   
 "Item": [   
 {   
 "Id": "AgentPDDispositionCode\_ID",   
 "Value": "1"   
 },   
 {   
 "Id": "Description",   
 "Value": "My Wrap Up Code 1"   
 },   
 {   
 "Id": "Name",   
 "Value": "WrapUp 1"   
 },   
 {   
 "Item": {   
 "Id": "2",   
 "Value": ""

CCSP UI Integrations Guide - 7.3 and 7.4   
117   
 },   
 "Id": "Queues"   
 }   
 ],   
 "Id": "3"   
 },   
 {   
 "Item": [   
 {   
 "Id": "AgentPDDispositionCode\_ID",   
 "Value": "8"   
 },   
 {   
 "Id": "Description",   
 "Value": "My Wrap Up Code 2"   
 },   
 {   
 "Id": "Name",   
 "Value": "Wrap Up 2"   
 },   
 {   
 "Item": {   
 "Id": "2",   
 "Value": ""   
 },   
 "Id": "Queues"   
 }   
 ],   
 "Id": "4"   
 }   
]   
   
The highlighted fields are the configuration database IDs of the respective wrap up codes. Each   
wrap up code ConfigItem contains an Item array of further ConfigItems. Within this array, the   
ConfigItem element identified by the ID "Queues" contains a further sub-array that has only one   
ConfigItem. If a wrap up code is associated with multiple queues, there are multiple elements in the   
"Queues" array.

CCSP UI Integrations Guide - 7.3 and 7.4   
118   
Note   
Due to the complex and dynamic nature of the configuration data, the "raw" ConfigItem is not   
documented. If you need to see which Ids and properties are defined for a specific ConfigItem   
object, the best method is to:   
1. Write code in a gadget to examine it or assign it to a variable.   
2. Use Developer Tools in the browser to set a source code breakpoint at the point where the   
ConfigItem is available.   
3. Run CCSP UI until it reaches the breakpoint.   
4. Examine the ConfigItem by using the command JSON.stringify(objectName) from the   
Console.   
Helper Functions   
The CCSP Agent AJAX API converts most of the ConfigItems into individual Javascript objects with   
named properties by using utility functions from the file TouchPoint\Ajax\helpers.js. For example, the   
CallInfo object in the Call class is based on the ConfigItem object of the same name that is received   
in the NewCall event. The properties of the CallInfo object are populated by getting the values of   
the individual items within the ConfigItem.   
To set the the call Type property for instance, the following statement is used in the Call object.   
this.CallInfo.Type = getValueByKey(this.CallInfo.Item, "Type");   
The function getValueByKey is from helpers.js and returns the ConfigItem whose Value property   
matches the given string.   
The following functions are useful when you need to read data directly from ConfigItems.   
ArrangeInSortedList   
Purpose   
Sorts an array of ConfigItems by a particular field that is identified by its   
position in each ConfigItem   
Signature   
ArrangeInSortedList(dataArray, keyPositon)   
Parameter   
Type   
Description   
dataArray   
Array   
Array of ConfigItems to be sorted   
keyPosition   
Integer   
Index of the field in the ConfigItem object on which to sort   
Returns   
Array   
A sorted array of objects with the properties:   
{   
name: (ConfigItem Id)   
code: (ConfigItem Value)   
}   
Example   
agent.Groups = ArrangeInSortedList(groupsItem,   
getPositionByKey(groupsItem[0].Item, "Name"));

CCSP UI Integrations Guide - 7.3 and 7.4   
119   
fixArray   
Purpose   
Converts a single ConfigItem into an array so that it can be processed by the   
same Array handling code   
Signature   
fixArray(array)   
Parameter   
Type   
Description   
array   
ConfigItem   
ConfigItem to be converted which may or may not already be   
an array   
Returns   
Array   
Either the original array or a new array containing the single   
item   
Example   
var groupsItem = fixArray(getItemByKey(response.ConfigData.Item, "Groups"));   
getItemByKey   
Purpose   
Retrieves a ConfigItem from an array whose Id property matches the given   
key   
Signature   
getItemByKey(array, key)   
Parameter   
Type   
Description   
array   
Array   
Array of ConfigItems   
Key   
String   
Key to match to ConfigItem Id   
Returns   
ConfigItem   
A match for the given key or null if none found   
Example   
var groupsItem = getItemByKey(response.ConfigData.Item, "Groups");   
getPositionByKey   
Purpose   
Retrieves the array index of the ConfigItem whose Id property matches the   
given key   
Signature   
getPositionByKey(array, key)   
Parameter   
Type   
Description   
array   
Array   
Array of ConfigItems   
Key   
String   
Key to match to ConfigItem Value   
Returns   
Integer   
Index of the matched ConfigItem or null if none found   
Example   
var nameIndex = getPositionByKey(agent.releaseItem[0].Item, "Name");

CCSP UI Integrations Guide - 7.3 and 7.4   
120   
getValueByKey   
Purpose   
Retrieves the Value property of the ConfigItem whose Id property matches   
the given key   
Signature   
getValueByKey(array, key)   
Parameter   
Type   
Description   
Array   
Array   
Array of ConfigItems   
Key   
String   
Key to match to ConfigItem Value   
Returns   
Any   
The Value property of the matched ConfigItem or null if none   
found   
Example   
this.CallInfo.Caller.Name = getValueByKey(this.CallInfo.Caller, "Name");

CCSP UI Integrations Guide - 7.3 and 7.4   
121   
Extracting data from a ConfigItem   
By combining the knowledge of a specific ConfigItem and the helper functions, you can extract data   
from the ConfigItem that cannot otherwise be accessed in the Agent or Call objects by referencing   
a named property. The following example accesses the PD Disposition and Description associated   
with the wrap up code selected by the agent after a call has ended.   
Getting ConfigItem data   
The wrap up information for an agent is only partially stored in the Agent object when the agent   
logs in. The Agent object contains the array WrapupCodesList, which contains the Name, Id and   
Queues data from each Wrap Up Config Item. To get the other information such as Description and   
PD Disposition Code, you need to get the complete wrap up data set from the server itself.   
1.   
Create a new functional gadget called ReadConfigItemExample with the javascript file   
readConfigItemExample.js.   
2.   
Following the established best practice, the functionality is contained in a function   
ReadConfigItemExample(), which performs the familiar setup procedure of getting the root   
window and the required services as below.   
   
function ReadConfigItemExample() {   
 // get reference to root window   
 var \_root = GetRootWindow();   
   
 // Get the TouchPoint application's angular injector   
 var localInjector = \_root.angular.element("[ng-app]").injector();   
   
 // Get the required services   
 var eventServiceWrapper = localInjector.get('eventServiceWrapper');   
 var agentServiceWrapper = localInjector.get('agentServiceWrapper');   
 var dialogService = localInjector.get('dialogService');   
   
 // Array to store the received Wrap Up data   
 var fixedWrapupItems = [];

CCSP UI Integrations Guide - 7.3 and 7.4   
122   
3.   
Using the eventServiceWrapper, subscribe to a specific AJAX API event   
GetWrapUpCodesSuccess that contains the full wrap up data. In the event handler, use the   
fixArray function to ensure that you are always dealing with an array of data.   
eventServiceWrapper.subscribe("GetWrapUpCodesSuccess", null, function   
(eventData) {   
   
 console.log("readConfigItemExample: GetWrapUpCodesSuccess")   
   
// Ensure that the wrap up item is treated as an array even if is   
// actually a single object   
 fixedWrapupItems = fixArray(eventData.response.ConfigData.Item);   
 });   
   
Note   
Observe how the actual ConfigItem is stored within the structure of the eventData   
object, that is, eventData.response.ConfigData.Item.   
This structure is followed for all Get…Success events raised by the AJAX API.   
   
4.   
Subscribe to a CCSP UI event that informs you when the Wrap Up window has been closed   
after a call. The event used is wrapupWindowEnded. When this event is received, use the   
getItemByKey() function to get the wrap up Item, then the getValueByKey() function to get the   
individual properties of the item.   
eventServiceWrapper.subscribe("wrapupWindowEnded", null, function   
(wrapupInfo) {   
 // Get the wrap up item that is identified by the selected wrapup   
code   
 var requiredItem = getItemByKey(fixedWrapupItems,   
wrapupInfo.selectedWrapupCode);   
 if (requiredItem !== null) {   
   
 // Get the disposition code from the item   
 var dispositionCode =   
getValueByKey(requiredItem, "AgentPDDispositionCode\_ID");   
 var description = getValueByKey(requiredItem, "Description");   
 dialogService.showMessageBox   
 ("ReadConfigItem Output", "Disposition code is " +   
dispositionCode +   
 ", Description is " + description,   
 \_root.enghouse.enums.MessageBoxButtons.OK);   
 }   
 });

CCSP UI Integrations Guide - 7.3 and 7.4   
123   
The dialogService is used instead of a javascript alert to display the retrieved wrap up   
information so as not to hold up the single thread of execution.   
5.   
To complete the example, call the AJAXI API GetWrapUpCodes() method on the Agent object   
   
 agentServiceWrapper.agent.GetWrapUpCodes();   
   
The complete code is as follows.   
function ReadConfigItemExample() {   
 // Get reference to root window   
 var \_root = GetRootWindow();   
   
 // Get the TouchPoint application's angular injector   
 var localInjector = \_root.angular.element("[ng-app]").injector();   
   
 // Get the required services   
 var eventServiceWrapper = localInjector.get('eventServiceWrapper');   
 var agentServiceWrapper = localInjector.get('agentServiceWrapper');   
 var dialogService = localInjector.get('dialogService');   
   
 // Array to store the received Wrap Up data   
 var fixedWrapupItems = [];   
   
 // Subscribe to the required events   
 eventServiceWrapper.subscribe("GetWrapUpCodesSuccess", null, function   
(eventData) {   
   
 // Ensure that the wrap up item is treated as an array even if is   
 // actually a single object   
 fixedWrapupItems = fixArray(eventData.response.ConfigData.Item);   
 });   
   
 eventServiceWrapper.subscribe("wrapupWindowEnded", null, function   
(wrapupInfo) {   
 // Get the wrap up item that is identified by the selected wrapup code   
 var requiredItem = getItemByKey(fixedWrapupItems,   
wrapupInfo.selectedWrapupCode);   
 if (requiredItem !== null) {   
   
 // Get the disposition code from the item

CCSP UI Integrations Guide - 7.3 and 7.4   
124   
 var dispositionCode = getValueByKey(requiredItem,   
"AgentPDDispositionCode\_ID");   
 var description = getValueByKey(requiredItem, "Description");   
 dialogService.showMessageBox   
 ("ReadConfigItem Output", "Disposition code is " +   
dispositionCode +   
 ", Description is " + description,   
 \_root.enghouse.enums.MessageBoxButtons.OK);   
 }   
 });   
   
 // Call the API function on the Agent object to get the Wrap Up info   
 agentServiceWrapper.agent.GetWrapUpCodes();   
}   
ReadConfigItemExample();   
//# sourceURL=readConfigItemExample.js   
The second last line calls the function that contains all the functionality and is called when the   
functional gadget is loaded. The last line is simply to ensure that the browser debugger tool can   
see the file.   
To test this, ensure that Wrap Up codes with meaningful descriptions and Disposition codes have   
been defined for the agent and add the gadget. The ReadConfigItemExample() function is called   
upon login and when the an agent completes Wrap Up after a call, the dialog pops up with the   
Disposition Code and Description.   
Note   
The example subscribes to the wrapupWindowEnded event, which is only sent when the Wrap Up   
window is closed after a call.   
When the window is closed while the call is still active, wrapupWindowSaving is sent, which has   
the same structure.

CCSP UI Integrations Guide - 7.3 and 7.4   
125   
Standard gadget properties   
The code in a gadget can access any property of the gadget that is visible for the gadget in CCSP   
AdminPoint. This enables custom properties to be added to the gadget that can be referenced in   
the code to set default values, maximum and minimum limits, timeout periods, ping intervals, and   
URLs to load.   
Gadgets also contain various "live" properties that describe the current state of the gadget and the   
specific instance of the gadget. These are also accessible from the gadget code itself.   
Accessing properties of a standard gadget   
In a standard gadget, the gadget's window has a property called windowObject. This property in   
turn has a Gadget property. The Gadget property contains all the properties that are editable for   
the gadget through CCSP AdminPoint and other properties that describe its current state.   
In turn, each gadget can have multiple active instances up to the limit specified by the Allowed   
Instances property. These instances are held in the Windows property of the gadget and each   
object contains information relating to the current state and appearance.   
The top level window of the gadget is referenced in pure JavaScript code as either this or window. In   
an AngularJS controller, the window can be referenced as $window if that has been injected as a   
parameter of the controller definition. The Demo CRM gadget has $window injected into   
demoCRMController, therefore it can use any of these references.   
This hierarchy is summarized below.   
   
The Demo CRM gadget contains a check box for enabling and disabling automatic (blind) transfers.   
The initial value of this check box is currently hard-coded in the DemoCRM.js file. We now set this   
using a property of the DemoCRM gadget.

CCSP UI Integrations Guide - 7.3 and 7.4   
126   
Adding an example custom property   
In CCSP AdminPoint, edit the system-level version of the Demo CRM gadget.   
Click Add Properties and select Boolean from the drop down menu.   
In Display Name, type Auto Transfer Enabled and leave the Default Value as False.   
Save the changes.   
Accessing the custom property in the code   
In the DemoCRM.js code, change the declaration of $scope.autoTransferEnabled to the following.   
$scope.autoTransferEnabled = $window.windowObject.Gadget.AutoTransferEnabled;   
Relogin to the user and open the Demo CRM gadget. The Auto Transfer check box is unchecked.   
Overriding a custom property   
In CCSP AdminPoint, edit the customization of the Demo CRM gadget (role, group, or tenant) that is   
relevant to the user, enable the Auto Transfer Enabled property and save the changes. Relogin to   
the Agent and open the Demo CRM gadget. The Auto Transfer check box is now selected.   
Accessing the custom property object in normal JavaScript   
To show that the this object can be used directly in non-AngularJS code, add the following   
declaration before the controller definition in DemoCRM.js.   
var autoTransferEnabled = this.windowObject.Gadget.AutoTransferEnabled;   
Change the $scope.autoTransferEnabled definition as follows.   
$scope.autoTransferEnabled = autoTransferEnabled;   
Reload the gadget. You don’t need to log out again. This produces the same result of the Auto   
Transfer check box being selected.   
windowObject.Gadget properties   
The following table lists the minimum set of properties that are present in the   
windowObject.Gadget object. Any custom properties added in CCSP AdminPoint are also present.   
Name   
Type   
Description   
AllowedInstances   
integer   
Corresponds to default property Allowed Instances   
defined in CCSP AdminPoint (see Table 2: Standard   
gadget default properties).   
Behavior   
string   
This can be used to specify a CSS class for the display   
of the gadget in the gadgetory, which can then change   
its appearance.

CCSP UI Integrations Guide - 7.3 and 7.4   
127   
Name   
Type   
Description   
CanClose   
Boolean   
Corresponds to default property Can Close defined in   
CCSP AdminPoint (see Table 2: Standard gadget   
default properties).   
CanExpand   
Boolean   
Corresponds to default property Can Expand defined   
in CCSP AdminPoint (see Table 2: Standard gadget   
default properties).   
CanFloat   
Boolean   
Corresponds to default property Can Float defined in   
CCSP AdminPoint (see Table 2: Standard gadget   
default properties).   
CanFold   
Boolean   
Corresponds to default property Can Fold defined in   
CCSP AdminPoint (see Table 2: Standard gadget   
default properties).   
CanHide   
Boolean   
Corresponds to default property Can Hide defined in   
CCSP AdminPoint (see Table 2: Standard gadget   
default properties).   
CanMaximize   
Boolean   
Corresponds to default property Can Maximize   
defined in CCSP AdminPoint (see Table 2: Standard   
gadget default properties).   
CanPopout   
Boolean   
Corresponds to default property Can Popout defined   
in CCSP AdminPoint (see Table 2: Standard gadget   
default properties).   
CanTab   
Boolean   
Corresponds to default property Can Tab defined in   
CCSP AdminPoint (see Table 2: Standard gadget   
default properties).   
ControllerURL   
string   
Corresponds to default property Controller URL   
defined in CCP AdminPoint (see Table 2: Standard   
gadget default properties).   
DefaultHeight   
integer   
Corresponds to the H component of the default   
property Float Default Size defined in CCSP   
AdminPoint (see Table 2: Standard gadget default   
properties).   
DefaultPosition   
string   
String representation of the Float Default Position   
property defined in CCSP AdminPoint (see Table 2:   
Standard gadget default properties). Format is "(X,Y)"

CCSP UI Integrations Guide - 7.3 and 7.4   
128   
Name   
Type   
Description   
DefaultPositionX   
integer   
Corresponds to the X component of the default   
property Float Default Position defined in CCSP   
AdminPoint (see Table 2: Standard gadget default   
properties).   
DefaultPositionY   
integer   
Corresponds to the Y component of the default   
property Float Default Position defined in CCSP   
AdminPoint (see Table 2: Standard gadget default   
properties).   
DefaultSize   
string   
String representation of the Float Default Size   
property defined in CCSP AdminPoint (see Table 2:   
Standard gadget default properties). Format is "(W,H)"   
DefaultState   
string   
Corresponds to default property Default State   
defined in CCSP AdminPoint (see Table 2: Standard   
gadget default properties).   
DefaultWidth   
integer   
Corresponds to the W component of the default   
property Float Default Size defined in CCSP   
AdminPoint (see Table 2: Standard gadget default   
properties).   
Disabled   
Boolean   
Corresponds to default property Disabled defined in   
CCSP AdminPoint (see Table 2: Standard gadget   
default properties).   
DisplayIcon   
String   
This is the path to the currently displayed icon for the   
gadget in the gadgetory. It has the same value as   
either the Icon or SelectedIcon property depending   
on whether or not the mouse is hovered over the   
gadget's position.   
DisplayName   
string   
The name of the gadget as seen in CCSP AdminPoint,   
displayed in the tool tip for the gadget in the   
gadgetory and used for the gadget's window title.   
FolderParent   
Gadget   
The parent folder in the gadgetory that contains the   
gadget when it has been "folded" due to the CCSP UI   
window size shrinking to a point where not all gadgets   
can be shown in the gadgetory. The folder is itself a   
gadget.   
Gadgets   
Array   
Only defined for folder gadgets that appear when the   
CCSP UI window size is shrunk to a point where not all   
gadgets can be shown in the gadgetory.

CCSP UI Integrations Guide - 7.3 and 7.4   
129   
Name   
Type   
Description   
HasFocus   
Boolean   
Flag indicating that at least one of the gadget's   
instances has the input focus.   
Hide   
Boolean   
Flag indicating that the gadget is not to be displayed in   
the gadgetory. Use this to hide system gadgets rather   
than the NoGadgetoryIcon property.   
Icon   
string   
The relative path to the icon file used for the gadget   
representation in the gadgetory.   
IsBottom   
Boolean   
Corresponds to default property Is Bottom defined in   
CCSP AdminPoint (see Table 2: Standard gadget   
default properties).   
IsSupervisorGadget   
Boolean   
Corresponds to default property Is Supervisor Gadget   
defined in CCSP AdminPoint (see Table 2: Standard   
gadget default properties).   
Name   
string   
The identifier of the gadget used within the CCSP UI   
code, with any spaces removed.   
NoGadgetoryIcon   
Boolean   
Corresponds to the default property No Icon in   
Gadgetory defined in CCSP AdminPoint Flag (see   
Table 2: Standard gadget default properties). This   
property is only defined for standard gadgets.   
Notifications   
integer   
The current number of notifications for this gadget.   
When non-zero, this number appears against the   
gadget's icon in the gadgetory.   
OpenWindowsCount   
integer   
The number of instances that have been opened for   
this gadget (including hidden ones)   
PopoutDefaultHeight   
integer   
Corresponds to the H component of the default   
property Popout Default Size defined in CCSP   
AdminPoint (see Table 2: Standard gadget default   
properties).   
PopoutDefaultPosition   
string   
String representation of the Popout Default Position   
property defined in CCSP AdminPoint (see Table 2:   
Standard gadget default properties). Format is "(X,Y)"   
PopoutDefaultPositionX   
integer   
Corresponds to the X component of the default   
property Popout Default Position defined in CCSP   
AdminPoint (see Table 2: Standard gadget default   
properties).

CCSP UI Integrations Guide - 7.3 and 7.4   
130   
Name   
Type   
Description   
PopoutDefaultPositionY   
integer   
Corresponds to the Y component of the default   
property Popout Default Position defined in CCSP   
AdminPoint (see Table 2: Standard gadget default   
properties).   
PopoutDefaultSize   
string   
String representation of the Popout Default Size   
property defined in CCSP AdminPoint (see Table 2:   
Standard gadget default properties). Format is "(W,H)"   
PopoutDefaultWidth   
integer   
Corresponds to the W component of the default   
property Popout Default Size defined in CCSP   
AdminPoint (see Table 2: Standard gadget default   
properties).   
SelectedIcon   
string   
The relative path to the icon file used for the   
highlighted gadget representation in the gadgetory.   
Stylesheet   
string   
Corresponds to default property Stylesheet defined in   
CCSP AdminPoint (see Table 2: Standard gadget   
default properties).   
TargetUrl   
string   
Corresponds to default property Target URL (HTML)   
defined in CCSP AdminPoint (see Table 2: Standard   
gadget default properties).   
Type   
string   
The gadget type either system, standard, or functional   
Windows   
Array   
The gadget instances that have been created.   
   
Table 4: windowObject.Gadget properties

CCSP UI Integrations Guide - 7.3 and 7.4   
131   
Gadget instance properties   
The following table lists the set of properties that are present in each of the gadget instances.   
Note   
Custom properties added in CCSP AdminPoint are not present in the gadget instance.   
   
Name   
Type   
Description   
ComponentData   
Object   
User specified data that is passed to the gadget instance on   
its creation. This can be anything that the instance requires to   
function, for example, a Call object.   
CurrentSelectedTab   
Boolean   
Flag that indicates if this instance is the currently selected tab.   
Only valid if WindowState value is Tab.   
DefaultState   
String   
Corresponds to the gadget's Default State property.   
Draggable   
Boolean   
Flag that indicates if the gadget instance can be dragged, that   
is, is not fixed in position.   
Expanded   
Boolean   
Flag that indicates if the gadget instance is currently in the   
expanded state.   
FrameHasFocus   
Boolean   
Flag that indicates if the gadget instance currently has the   
focus when it is in the Tab state   
HasFocus   
Boolean   
Flag that indicates if the gadget instance currently has the   
focus when it is in the Tab state.   
Height   
int   
The current height of the gadget instance when in the Float   
state.   
Id   
string   
The unique GUID identifier for the gadget instance.   
IsDragging   
Boolean   
Flag indicating that the gadget instance is being moved when   
in the Float state.   
IsHidden   
Boolean   
Flag indicating that the gadget instance has been minimized.   
PositionX   
integer   
The X component of the gadget instance's current position   
when in the Float state.   
PositionY   
integer   
The Y component of the gadget instance's current position   
when in the Float state.   
Resizable   
Boolean   
Flag indicating that the gadget instance can be resized.   
TargetUrl   
String   
Corresponds to the gadget's Target URL (HTML) property.   
Title   
String   
The name displayed in the title bar of the gadget instance.

CCSP UI Integrations Guide - 7.3 and 7.4   
132   
Name   
Type   
Description   
Width   
integer   
The current width of the gadget instance when in the Float   
state.   
WindowState   
String   
The current state of the gadget instance. See Default State in   
Table 2: Standard gadget default properties for possible   
values.   
WindowType   
String   
The type of the gadget instance. Each default CCSP UI gadget   
has its own type, for example: WrapUp or CallBack, but   
standard gadgets are always assigned the Gadget type.   
Zindex   
integer   
Current zIndex of the gadget instance element.   
   
Table 5: Gadget instance properties   
Standard gadget properties example   
Simple Gadget   
A standard gadget can access the property values of the gadget and the gadget instance. The   
Simple Gadget located in app\gadgets\Examples\Simple provides an example where all these   
property values are displayed and demonstrates how some of them can be changed   
programmatically. It consists of the SimpleGadget.html file that contains label elements for the   
property values and buttons to change some of the values. The HTML file loads a SimpleGadget.js   
script which is written in standard javascript with no AngularJS functionality.   
To add the pre-defined Simple Gadget standard gadget:   
1.   
Log in to CCSP AdminPoint using the appropriate credentials.   
2.   
On the Gadgetory tab, on the Gadgets subtab, click   
 Add and then click Standard   
Gadget.   
3.   
Type Simple Gadget as the name of the new gadget.   
4.   
In Target URL (HTML), type the following URL:   
app/gadgets/Examples/Simple/SimpleGadget.html   
5.   
Change the Allowed Instances value to 5.   
6.   
Click Save.   
   
A standard gadget can be produced to display the values of the gadget and the gadget instance.   
The Simple Gadget located in app\gadgets\Examples\Simple provides an example where all the   
property values are displayed.

CCSP UI Integrations Guide - 7.3 and 7.4   
133   
To get the gadget and the instance, the SimpleGadget.js uses the following code.   
var \_root = GetRootWindow();   
var \_gadget;   
var \_gadgetInstance;   
var defaultPropertiesHidden = false;   
angular.element(document).ready(function () {   
   
 \_gadget = this.windowObject.Gadget;   
 \_gadgetInstance = \_root.app.GadgetComm.getWindowById(this.windowId);   
   
 //Population of the HTML elements   
 ...   
};   
   
Variables global to the window are declared, then the ready() function stores references to the   
gadget and this particular instance of it. The HTML elements can then be populated from the   
properties.   
Note   
Although the gadget instance is referred to as a window and it shares its Id with its top level   
window, they are not the same object. The top level window is the HTML object, whereas the   
gadget instance is a construct specific to CCSP.   
Changing the default values of the gadget properties   
The default values of a gadget can be changed programmatically although caution is advised and   
attention must be paid to the following caveats:   
•   
The changes only affect the current login session and a new session reverts to using the values   
set for the gadget either at the system level or in the customization.   
•   
Changes only take effect the next time an instance of the gadget is opened.   
•   
Properties should remain consistent with one another where required, for example, the   
corresponding flag that allows a DefaultState value to be used (for example, CanTab for "Tab")   
should be set to true.   
•   
Only properties that have been defined in CCSP AdminPoint should be changed. All other   
properties visible in the gadget are for internal use only and may cause unexpected side   
effects if tampered with.

CCSP UI Integrations Guide - 7.3 and 7.4   
134   
Changing the gadget's default size and position as a floating window   
One useful application of this capability is the reset of the default position and size of future gadget   
instances to the current position and size when an instance is closed. The window.unload() function   
can be implemented as below.   
window.onunload = function () {   
 \_gadget.DefaultState = \_gadgetInstance.WindowState;   
 if (\_gadget.DefaultState == "Float") {   
 \_gadget.DefaultPositionX = \_gadgetInstance.PositionX;   
 \_gadget.DefaultPositionY = \_gadgetInstance.PositionY;   
 \_gadget.DefaultHeight = \_gadgetInstance.Height;   
 \_gadget.DefaultWidth = \_gadgetInstance.Width;   
 \_gadget.DefaultSize = "(" + \_gadgetInstance.Width + "," +   
\_gadgetInstance.Height + ")";   
 \_gadget.DefaultPosition = "(" + \_gadgetInstance.PositionX + "," +   
\_gadgetInstance.PositionY + ")";   
 }   
};   
   
Subsequent instances of this gadget now open with the same state as the last one. If the gadget is   
a floating window, it occurs in the last position with the same size as the one just closed.   
Alternatively a positional offset can be introduced for floating gadgets in the   
angular.element(document).ready() function to achieve a cascade of gadget instances.   
angular.element(document).ready(function () {   
 if (\_gadget.DefaultState == "Float") {   
 \_gadget.DefaultPositionX = \_gadgetInstance.PositionX + 10;   
 \_gadget.DefaultPositionY = \_gadgetInstance.PositionY + 10;   
 }

CCSP UI Integrations Guide - 7.3 and 7.4   
135   
Changing the gadget's default size and position as a popout window   
The same method can be applied to the reset of the popout window's default size and position,   
using the "Popout" prefixed properties as below.   
 else if (\_gadget.DefaultState == "Popout") {   
 \_gadget.PopoutDefaultPositionX = this.screenLeft;   
 \_gadget.PopoutDefaultPositionY = this.screenTop;   
 \_gadget.PopoutDefaultHeight = this.parent.innerHeight;   
 \_gadget.PopoutDefaultWidth = this.parent.innerWidth;   
 \_gadget.PopoutDefaultSize = "(" + \_gadget.PopoutDefaultWidth + "," +   
\_gadget.PopoutDefaultHeight + ")";   
 \_gadget.PopoutDefaultPosition = "(" + \_gadget.PopoutDefaultPositionX +   
"," + \_gadget.PopoutDefaultPositionY + ")";   
 }   
   
When subsequent instances of this gadget are opened in the Popout state, they occur in the same   
size and position as the last closed instance. Note that in this case, the size and position values are   
being taken directly from the browser window object.   
Changing the gadget's appearance   
Note   
When updating any display property of the gadget itself, the root scope's $apply() method must   
be called after the property change to refresh the gadgetory element. This requires the use of   
the GadgetComm's scopeApply() method that is available in CCSP 7.2.2 and later. For earlier   
CCSP versions use the following statement:   
   
GetRootWindow().angular.element("[ng-controller]").scope().$apply();   
   
The following properties affect the appearance of the gadget in the gadgetory:   
•   
Behavior   
•   
DisplayName   
•   
Icon   
•   
SelectedIcon   
•   
Notifications   
•   
NoGadgetoryIcon

CCSP UI Integrations Guide - 7.3 and 7.4   
136   
Changing the DisplayName property   
The Display Name is the name that appears as the tool tip for the gadget in the gadgetory as   
below.   
   
The Simple Gadget has an Update Display Name button which demonstrates the changing of the   
DisplayName property. Its onclick event handler is defined as below.   
updateDisplayName = function () {   
 \_gadget.DisplayName = "Renamed Gadget";   
 \_root.angular.element("[ng-controller]").scope().$apply();   
};   
   
After clicking the Update Display Name button in the Simple Gadget, the gadget's appearance is   
updated as below.   
   
Changing the Icon and SelectedIcon properties   
The Icon and SelectedIcon properties are strings that define the URL for their respective icon files.   
The URL must be relative to the TouchPoint virtual directory. The Simple Gadget example has an   
Update Icon button with its onclick event handler defined as below.   
 updateIcon = function () {   
 if (\_gadget.MyProps.SavedIcon == undefined || \_gadget.MyProps.SavedIcon ==   
"") {   
 //Store the original icons   
 \_gadget.MyProps.SavedIcon = \_gadget.Icon;   
 \_gadget.MyProps.SavedSelectedIcon = \_gadget.SelectedIcon;   
 \_gadget.Icon = "app/gadgets/Examples/Simple/SimpleGadgetIcon1.ico";   
 \_gadget.SelectedIcon =   
"app/gadgets/Examples/Simple/SimpleGadgetIcon1Selected.ico";   
 \_gadget.DisplayIcon = \_gadget.Icon;   
 } else {

CCSP UI Integrations Guide - 7.3 and 7.4   
137   
//Reset the icons   
 \_gadget.Icon = \_gadget.MyProps.SavedIcon;   
 \_gadget.SelectedIcon = \_gadget.MyProps.SavedSelectedIcon;   
 \_gadget.DisplayIcon = \_gadget.Icon;   
 \_gadget.MyProps.SavedIcon = "";   
 \_gadget.MyProps.SavedSelectedIcon = "";   
 }   
 \_root.angular.element("[ng-controller]").scope().$apply();   
};   
   
This toggles the icons between those specified in the customization and the ones located in the   
gadget's folder.   
Normal:   
   
After update:   
   
Highlighted:   
   
Note   
When changing the Icon property to a new value, the DisplayIcon property must also be updated   
otherwise the icon's appearance only changes after it has been highlighted again.

CCSP UI Integrations Guide - 7.3 and 7.4   
138   
The original icon values are stored in the MyProps object which is a property on the gadget itself.   
This is done so that the values are persistent between gadget instances. The   
angular.element(document).ready() function sets up the MyProps object as below.   
if (\_gadget.MyProps == null) {   
 \_gadget.MyProps = {   
 SavedIcon: "",   
 SavedSelectedIcon: "",   
 BehaviorCycle: 0,   
 TitleNum: 0   
 }   
}   
Changing the Behavior property   
The Behaviour property is the name of a class that can be applied to the gadget's representation in   
the gadgetory. Because the gadgetory is part of the top level CCSP UI window, the class has to be   
part of the CCSP UI application's stylesheet – a style from the gadget's stylesheet is not recognized.   
To avoid changing a CCSP UI CSS file, you can use a functional gadget to add the style dynamically.   
To demonstrate this, use AdminPoint to create a new functional gadget called Demo Library Func   
with the Script URL property set to app/gadgets/Examples/DemoCRM/DemoLibraryFunctions.js   
and add it to the same customization that contains Simple Gadget.   
DemoLibraryFunctions.js contains the following code.   
function createClass(name, rules) {   
 var style = document.createElement('style');   
 style.type = 'text/css';   
 document.getElementsByTagName('head')[0].appendChild(style);   
 style.sheet.insertRule(name + "{" + rules + "}", 0);   
}   
createClass('.mySimpleClassGreen', "background-color: green;");   
createClass('.mySimpleClassRed', "background-color: red;");   
createClass('.mySimpleClassNone', "background-color: none;");   
   
When the gadget is loaded at login, the function adds a class with the given name and rules as a   
new style to the CCSP UI document. The function is then called to add styles that define different   
background colors.

CCSP UI Integrations Guide - 7.3 and 7.4   
139   
In the Simple Gadget, there is an Update Behavior button, which has an onclick event handler   
defined as below.   
updateBehavior = function () {   
 switch (\_gadget.MyProps.BehaviorCycle) {   
 case 0:   
 \_gadget.Behavior = "mySimpleClassGreen";   
 break;   
 case 1:   
 \_gadget.Behavior = "mySimpleClassRed";   
 break;   
 case 2:   
 \_gadget.Behavior = "mySimpleClassNone";   
 break;   
 default:   
 break;   
 }   
   
 \_gadget.MyProps.BehaviorCycle++;   
 if (\_gadget.MyProps.BehaviorCycle > 2) \_gadget.MyProps.BehaviorCycle = 0;   
 \_root.angular.element("[ng-controller]").scope().$apply();   
};   
   
This just cycles through the different classes with the effect of changing the gadget's background   
with activation of the Update behaviour button as below.   
First click:   
   
Second click:   
   
The third click resets the background to none.

CCSP UI Integrations Guide - 7.3 and 7.4   
140   
One useful application of the Behavior property, is to dynamically hide the icon for a standard   
gadget in the gadgetory. This may be needed if a set of gadgets should be hidden during a call,   
while allowing another set to appear. An example of how to do this is given in Hiding the Demo   
Chat Prompt gadget icon.   
Note   
The same result can also be achieved by enabling the NoGadgetoryIcon property for the gadget   
in CCSP AdminPoint. When this property is enabled, the gadget icon is always hidden from the   
agent, regardless of the Behaviour property value.   
Changing the Notifications property   
The value of the Notifications property is displayed in the gadget's icon when non-zero up to a   
maximum value of 99. In the Simple Gadget example there is an Update Notifications button with a   
text field for the number to which to set the Notifications. The button's onclick event handler is   
defined as below.   
updateNotifications = function () {   
 \_gadget.Notifications = document.getElementById('notificationsVal').value;   
 \_root.angular.element("[ng-controller]").scope().$apply();   
};   
   
Setting the value to 50 for example gives the following:   
   
Note   
The Notifications value can also be incremented and decremented by the GadgetComm's   
sendNotifications() method (see GadgetComm functions).

CCSP UI Integrations Guide - 7.3 and 7.4   
141   
Changing the title of a gadget instance   
Caution   
Changes to any other property apart from the Title of a gadget instance should never be   
performed programmatically. There is no established interface for changing these properties   
and just changing a property value - for example, changing WindowState from "Tab" to   
"Popout" - usually has no effect on the appearance of the gadget instance, but may put it in an   
indeterminate state.   
   
   
The Title property of a gadget instance can be changed which is useful for distinguishing between   
multiple gadgets when they are tabbed or minimized. It also updates the text displayed in the title   
bar of gadget when it is a floating window. The Simple Gadget example has an Update Title button   
which has the onclick event handler defined as follows.   
updateTitle = function () {   
 \_gadgetInstance.Title = "A New Title " + ++\_gadget.MyProps.TitleNum;   
 \_root.angular.element("[ng-controller]").scope().$apply();   
};   
   
This increments the title number on every activation and because it is part of the gadget's MyProps   
object, every instance has a different title. The screenshot below shows how this appears in the   
gadgetory when three instances of the Simple Gadget have been opened, two of which have had   
the Update Title button activated.

CCSP UI Integrations Guide - 7.3 and 7.4   
142   
Functional gadget properties   
For functional gadgets, the properties are stored in the top level window object for the CCSP UI   
application and are accessed using the gadget's name as a property index. The object that contains   
all the properties is functionalGadgetProperties and the usage is as follows.   
rootWindow.enghouse.functionalGadgetsProperties[<gadgetName>].<propertyName>   
   
To demonstrate this in the Demo Chat Prompt Func gadget, the questions and responses that are   
currently hard-coded in the JavaScript file are replaced by the use of gadget properties.   
Using custom properties in a functional gadget   
In CCSP AdminPoint, edit the system-level version of the Demo Chat Prompt Func gadget.   
Click Add Properties and select String from the drop down menu.   
In Display Name, type Question. In Default Value, type Please answer the following question with   
yes or no. Do you have an existing policy with us?   
Repeat for the following String values:   
•   
In Display Name, type Response Is Yes. In Default Value, type Thank you, please wait a   
moment while I retrieve your information.   
•   
In Display Name, type Response Is No. In Default Value, type Thank you, would you like to   
open a new policy with us?   
•   
In Display Name, type Response Is Neither. In Default Value, type Sorry, I did not understand   
that response.   
•   
In Display Name, type Give Up Goodbye. In Default Value, type Sorry, I'm not understanding   
you, so am terminating this call. Goodbye.   
   
Add an integer property with the Display Name of Maximum Retries and the Default Value of 5.   
Save the changes.   
Accessing the custom properties in the code   
In the DemoChatPromptFunc.js code, add the following declaration before the question variable.   
var thisGadget = null;   
if (\_root.enghouse.functionalGadgetsProperties &&   
\_root.enghouse.functionalGadgetsProperties["DemoChatPromptFunc"]) {   
 thisGadget =   
\_root.enghouse.functionalGadgetsProperties["DemoChatPromptFunc"];   
}   
   
This now provides a reference to the gadget properties that can be used throughout the code. Edit   
the question and response variables in the code to use the appropriate properties from the gadget.   
The recommended implementation is below.

CCSP UI Integrations Guide - 7.3 and 7.4   
143   
var question = thisGadget && thisGadget.Question || "Question";   
var responseIsYes = thisGadget && thisGadget.ResponseIsYes || "Response if   
yes";   
var responseIsNo = thisGadget && thisGadget.ResponseIsNo || "Response if no";   
var responseIsNeither = thisGadget && thisGadget.ResponseIsNeither ||   
"Response if Neither";   
var giveUpGoodbye = thisGadget && thisGadget.GiveUpGoodbye || "Goodbye";   
var maximumRetries = thisGadget && thisGadget.MaximumRetries || 1;   
   
The properties are now available in the gadget and when the gadget is reloaded through a new   
login session of the user, the question and acknowledgements are shown as they were before.   
Before testing the gadget, you can use the MaximumRetries property to control the number of   
invalid responses the user accepts from the caller, before abandoning the call. The GiveUpGoodbye   
message is sent to notify the caller.   
Edit the onNewChatItem() function as below.   
if (response != null && response.length > 0) {   
…   
} else {   
 if (chatCalls[x].retries < maximumRetries) {   
 chatCalls[x].call.SendChatMessage(responseIsNeither + question);   
 chatCalls[x].retries++;   
 } else {   
 chatCalls[x].call.SendChatMessage(giveUpGoodbye);   
 setTimeout(function (index) {   
 chatCalls[index].call.HangUp();   
 }, 1000, x);   
 }   
}   
   
Note   
After sending the goodbye message, the gadget uses the setTimeout() function to delay the   
calling of the HangUp() method for the call. This is to ensure that the message is fully sent before   
the call enters the wrap up state or is destroyed.   
   
To test the gadget, relogin the user and make a chat call to the Agent. The question and response   
text should be the same as before. If a call is made and 5 incorrect responses to the question are   
given by the caller, the call is terminated with the goodbye message sent to the caller.

CCSP UI Integrations Guide - 7.3 and 7.4   
144   
Overriding the custom properties   
In CCSP AdminPoint, edit the customization of the Demo Chat Prompt gadget (role, group, or   
tenant) that is relevant to the user.   
Edit the properties to change the question and response values as follows:   
•   
Question: Please answer the following question with yes or no. Is your vehicle insured with us?   
•   
Response Is Yes: Thank you, I am loading your vehicle policy details.   
•   
Response Is No: Thank you, would you like to insure your vehicle with us?   
•   
Response Is Neither: Sorry, the response didn't match an valid answer.   
   
Change the Maximum Retries value to 3.   
Relogin the user and make a chat call to the Agent. The question and responses have now been   
updated to the ones entered for the customization. The Agent also now only accepts three invalid   
responses before ending the call.   
Disabling a functional gadget   
Functional gadgets can be disabled by enabling the default Disabled property in CCSP AdminPoint.   
Edit the customization for the Demo Chat Prompt Func gadget, enable the Disabled property and   
save the change.   
Relogin the user and make a chat call to it. The call is no longer answered automatically because   
the Demo Chat Prompt Func gadget is not loaded. In the Client log file, there is no debug output for   
demoChatPromptFunc.

CCSP UI Integrations Guide - 7.3 and 7.4   
145   
Accessing other gadgets and windows   
Every customized gadget is added to the main CCSP UI application. How it can be accessed is   
dependent on whether or not it is a functional or standard gadget. Standard gadgets need to be   
accessed through the top level application, using either the ngController scope, or through the   
GadgetComm object.   
If its a functional gadget, it is actually part of the CCSP UI top level code. This is why we strongly   
recommend that all properties and functions for a functional gadget are wrapped in a single   
function or object with a name unique to the provider. This then enables them to be used as   
properties and functions of declared objects in other gadget code.   
Using the GadgetComm object   
The GadgetComm object provides access to standard gadgets and their instances. It exposes   
methods to open, close, and hide gadgets and send them notifications. It is a property of the CCSP   
UI top level app object and its usage is one of the following.   
window.app.GadgetComm   
GetRootWindow().app.GadgetComm   
GadgetComm functions   
This section lists only those functions that are suitable for custom gadgets to use. Other functions   
contained in the object are for internal use only.   
clearGadgetNotifications   
Purpose   
Resets the Notifications for a gadget to 0   
Signature   
clearGadgetNotifications(gadgetName)   
Parameter   
Type   
Description   
gadgetName   
string   
Name of the gadget as entered in CCSP AdminPoint with no   
spaces   
Returns   
-   
-   
closeComponentWindow   
Purpose   
Closes the window identified by the windowId parameter.   
Note   
This ignores the gadget's CanClose property value.   
   
Signature   
closeComponentWindow(windowId)   
Parameter   
Type   
Description   
windowId   
   
string   
GUID of the window   
Returns   
-   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
146   
getGadget   
Purpose   
Returns the gadget object whose name matches the given parameter.   
Note   
This is new for CCSP 7.2.2. To access a gadget using earlier versions   
use this instead.   
findObjByKey(angular.element("[ng-  
controller]").scope().viewmodel.Gadgets, "Name",   
gadgetName);   
   
Signature   
GetGadget(gadgetName)   
Parameter   
Type   
Description   
gadgetName   
string   
Name of the gadget as entered in CCSP AdminPoint with no   
spaces   
Returns   
Object   
The gadget object if found, otherwise null   
getWindowById   
Purpose   
Returns the window object that is identified by the windowId parameter.   
Signature   
getWindowById(windowId)   
Parameter   
Type   
Description   
windowId   
   
string   
GUID of the window   
Returns   
Object   
The window if found or null if not   
hideWindow   
Purpose   
Hides or minimizes the gadget window given by the parameter. New for   
CCSP 7.2.2.   
Signature   
hideWindow(windowId)   
Parameter   
Type   
Description   
windowId   
   
string   
GUID of the window   
Returns   
-   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
147   
OpenGadgetInstance   
Purpose   
Opens an instance of the gadget and adds an optional method with the   
given args parameter to the queue array that is called when the window is   
opened.   
Signature   
OpenGadgetInstance(gadgetName, method, args)   
Parameter   
Type   
Description   
gadgetName   
string   
Name of the gadget as entered in CCSP AdminPoint with no   
spaces   
method   
string   
Name of a method that is exposed by the gadget that is to be   
called when the gadget is opened.   
args   
Object   
Contains any arguments for the method that are called or null if   
not required.   
Returns   
string   
The GUID of the new gadget window or null if one could not be   
opened. New for CCSP 7.2.2: earlier versions raise an exception   
if a window cannot be opened and do not return anything even   
if it is opened.   
openSingleComponentWindow   
Purpose   
Shows the window identified by the winId parameter if it exists, or opens a   
new window with the given parameters and returns its new GUID.   
Note   
We recommend that custom code uses the OpenGadgetInstance and   
showWindow functions rather than this one.   
   
Signature   
openSingleComponentWindow(winId, name, data, bringToFront)   
Parameter   
Type   
Description   
winId   
string   
GUID of the window   
name   
string   
Display title of the window   
data   
Object   
Any data to be passed to the new window as its ComponentData   
property   
bringToFront   
Boolean   
Flag that indicates whether or not to bring the window to the   
front   
Returns   
string   
The window Id (either a new one or the existing one)

CCSP UI Integrations Guide - 7.3 and 7.4   
148   
sendGadgetNotification   
Purpose   
Increases the gadget Notifications by the given value.   
Signature   
sendGadgetNotification(gadetName, number)   
Parameter   
Type   
Description   
gadgetName   
string   
Name of the gadget as entered in CCSP AdminPoint with no   
spaces   
number   
integer   
Amount by which to increase the current Notifications value. To   
decrease the Notifications, specify a negative value.   
Returns   
-   
-   
scopeApply   
Purpose   
Updates the root window, typically to refresh the gadgetory or display   
parameters of the gadget.   
Gadgets should only use this after updating a gadget display property.   
Signature   
scopeApply()   
Parameter   
Type   
Description   
-   
-   
-   
Returns   
-   
-   
showWindow   
Purpose   
Shows the window identified by the windowId parameter.   
Signature   
showWindow(windowId)   
Parameter   
Type   
Description   
windowId   
string   
GUID of the window   
bringToFront   
Boolean   
Flag that indicates whether or not to bring the window to the   
front   
Returns   
-   
-

CCSP UI Integrations Guide - 7.3 and 7.4   
149   
windowReady   
Purpose   
When the gadget window has been opened using the   
OpenGadgetInstance() function, this invokes the gadget method that was   
passed to that function with any specified arguments.   
Note   
This should normally be called from the   
angular.element(document).ready event handler in the gadget.   
   
Signature   
windowReady(window)   
Parameter   
Type   
Description   
window   
Object   
Top level window of the gadget.   
Returns   
-   
-   
Inter-gadget access example   
In this example, the Demo Chat Prompt functional gadget is extended to open a standard gadget   
instead of a URL when it receives a negative response to the opening question. The standard   
gadget uses another functional gadget to display different prompts to the agent, depending on the   
messages received from the caller. Finally, when the call is completed, the Demo Chat Prompt   
functional gadget closes the standard gadget window.   
Add new gadgets to CCSP AdminPoint   
In CCSP AdminPoint, add the following two gadgets:   
•   
Demo Chat Prompt (standard gadget)   
•   
Demo Chat Prompt DB (functional gadget)   
   
For the Demo Chat Prompt standard gadget, use the following default values:   
•   
Allowed Instances: 3   
•   
Default State: Float   
•   
Float Default Size W: 300, H: 320   
•   
Float Default Position X: 0, Y: 450   
•   
Target URL (HTML): app/gadgets/DemoChatPrompt/DemoChatPrompt.html   
   
For the Demo Chat Prompt DB functional gadget, use the default value:   
•   
Script Url: app/gadgets/DemoChatPrompt/DemoChatPromptDB.js   
   
Also, add a Boolean Disabled property with the default value of False.   
Add both gadgets to the customization for the test user and ensure that the Demo Chat Prompt   
Func gadget in the same customization has any Disabled property set to False.

CCSP UI Integrations Guide - 7.3 and 7.4   
150   
Create Demo Chat Prompt DB JavaScript   
The Demo Chat Prompt DB functional gadget simulates an interface to a repository of prompts that   
are referenced by a keyword. This could be accomplished by any number of means, but for this   
example, it just contains a single function GetPromptTexts() that takes in a string and returns any   
prompts that have been defined for it.   
Create the file DemoChatPromptDB.js in the DemoChatPrompt location and add the following   
code.   
GetPromptTexts = function (searchText) {   
 var promptDB =   
 [   
 {   
 matchText: "car",   
 promptText:   
 [   
 "Would you like comprehensive or third party fire and   
theft cover?"   
 ]   
 },   
 {   
 matchText: "van",   
 promptText:   
 [   
 "Would you like business use cover?"   
 ]   
 },   
 {   
 matchText: "bike",   
 promptText:   
 [   
 "Would you like life insurance on top of the normal   
cover"   
 ]   
 }   
 ];   
   
 var returnPrompts = [];   
 for (var x = 0; x < promptDB.length; x++) {   
 var searchStr = new RegExp('\\b' + promptDB[x].matchText + '\\b',   
'i');

CCSP UI Integrations Guide - 7.3 and 7.4   
151   
 if (searchText.match(searchStr) != null) {   
 returnPrompts = returnPrompts.concat(promptDB[x].promptText);   
 }   
 }   
 return returnPrompts;   
};   
   
The GetPromptTexts() function is now a function of the root window, so can now be referenced as   
follows.   
var \_root = GetRootWindow();   
\_root.GetPromptTexts();   
   
This is a very simplistic definition of the function and it is entirely up to the developer as to how to   
declare functions in a gadget for use in other gadgets. It could be declared in the global scope, in   
which case the name should be something that is unlikely to override another function, or as part   
of an object so that it is prefixed with a namespace. The important aspect is that it is available for   
use by other code.   
Create Demo Chat Prompt HTML   
The Demo Chat Prompt gadget handles one call only and displays the caller name, the call state,   
last received message and its sender and the relevant prompts. Create the file   
DemoChatPrompt.html in the DemoChatPrompt folder and add the basic HTML code for this   
example.   
<!DOCTYPE html>   
<html ng-app="demoChatPromptApp" xmlns="http://www.w3.org/1999/xhtml">   
<head>   
 <title>Demo Chat Prompt</title>   
 <script type="text/javascript" src="/TouchPoint/Scripts/jquery-  
2.0.1.js"></script>   
 <script type="text/javascript"   
src="/TouchPoint/Scripts/angular.js"></script>   
 <script type="text/javascript"   
src="/TouchPoint/App/common/gadgetInit.js"></script>   
 <script type="text/javascript" src="/TouchPoint/Ajax/helpers.js"></script>   
 <script type="text/javascript"   
src="/TouchPoint/App/common/directives/baseClasses/subscriberBase.js"></script  
>   
 <script type="text/javascript" src="DemoChatPrompt.js"></script>   
</head>   
<body id="demoChatPrompt" ng-controller="demoChatPromptController">

CCSP UI Integrations Guide - 7.3 and 7.4   
152   
 <h3>Prompts for: </h3><div>{{   
callWrapper.call.CallInfo.Caller.Name}}</div>   
 <div>   
 <p>   
 <label for="enablePromptsChecked">Display prompts</label>   
 <input type="checkbox" ng-model="enablePrompts"   
id="enablePromptsChecked" />   
 </p>   
 </div>   
 <div>   
 <div><label>Call state: </label>{{ callWrapper.call.State }}</div>   
 <br />   
 <div><label>Last message from </label>>{{ lastSender }}</div>   
 <div style="height: 80px; width: 100%; background-color: azure">   
 {{ lastMessage }}   
 </div>   
 <br />   
 <div>   
 <label>Possible questions:</label>   
 <div style="background-color: azure; height: 100px; width: 100%">   
 <ul>   
 <li ng-repeat="prompt in prompts" style="list-style:   
none">   
 {{ prompt }}<br />   
 </li>   
 </ul>   
 </div>   
 </div>   
 </div>   
</body>   
</html>   
   
The HTML ngApplication is demoChatPromptApp and the body ngController is   
demoChatPromptController. The body element has also been given the id demoChatPrompt for   
use in the JavaScript code later.   
The script includes are the normal set for a gadget that are AngularJS-based plus the   
DemoChatPrompt.js. There is also the additional inclusion of subscriberBase.js which is required for   
using the SubscriberBase object.

CCSP UI Integrations Guide - 7.3 and 7.4   
153   
Create Demo Chat Prompt JavaScript   
Create the file DemoChatPrompt.js in the DemoChatPrompt folder. In the file, insert the code to   
create the demoChatPromptApp and add the required services, which for this gadget are just   
loggerService and eventServiceWrapper.   
//get reference to root window - the one containing our main modules and   
services   
var \_root = GetRootWindow();   
   
//create module   
var mainModule = angular.module("demoChatPromptApp", []);   
//declare the services you wish to inject to your module   
var directServices = ["loggerService", "eventServiceWrapper"];   
//use this method to inject all the services above to your module   
\_root.appendServiceProxiesToModule(mainModule, directServices);   
   
Create the demoChatPromptController with these dependencies.   
//declare your controller with its dependencies - angular or custom ones.   
mainModule.controller("demoChatPromptController", ["$scope", "$window",   
"loggerService", "eventServiceWrapper",   
 function ($scope, $window, loggerService, eventServiceWrapper) {   
   
Inside the demoChatPromptController, add the following.   
loggerService.WriteLine("Demo Chat Prompt standard gadget has been opened",   
\_root.enghouse.enums.Severity.Debug, "demoChatPrompt");   
   
 $scope.callWrapper = null;   
 $scope.enablePrompts = true;   
 $scope.chatEventId = null;   
 $scope.messageCount = 0;   
 $scope.thisInstance =   
\_root.app.GadgetComm.getWindowById($window.windowId);   
 //Add the SubscriberBase to this controller   
 var self = this;   
 var base = new SubscriberBase(eventServiceWrapper);   
 $.extend(true, self, base);   
   
The inclusion of subscriberBase.js in the HTML allows for the use of the SubscriberBase object   
which makes the management of subscriptions easier, particularly if there is no need to publish   
events from this gadget.   
The gadget instance is being obtained here to update the Title property later in the code.

CCSP UI Integrations Guide - 7.3 and 7.4   
154   
Next define the $scope function that subscribes to the call object that is passed into the gadget.   
$scope.SubscribeToCall = function (call) {   
 self.addSubscription("CallStateChange", call.CallId, onCallStateChange);   
 self.addSubscription("NewChatItem", call.CallId, onNewChatItem);   
 $scope.callWrapper = { callId: call.CallId, call: call };   
 loggerService.WriteLine("Demo Chat Prompt has subscribed to " +   
call.CallId, \_root.enghouse.enums.Severity.Debug, "demoChatPrompt");   
};   
   
The function uses the addSubscription() method to subscribe to the call events in place of the   
eventServiceWrapper's subscribe() method.   
Define the call event handlers as below.   
onNewChatItem = function (chatEvent) {   
 //Use the function provided by the ChatPromptDB functional gadget if it   
exists   
 if (chatEvent.chatItem.Who == "Caller" && $scope.enablePrompts &&   
\_root.GetPromptTexts) {   
 $scope.prompts = \_root.GetPromptTexts(chatEvent.chatItem.Text);   
 }   
 $scope.lastMessage = chatEvent.chatItem.Text;   
 $scope.lastSender = chatEvent.chatItem.Who;   
 $scope.thisInstance.Title = $scope.callWrapper.call.CallInfo.Caller.Name +   
" " + $scope.messageCount++ + " messages received";   
 $scope.$apply();   
};   
   
onCallStateChange = function (stateEvent) {   
 //Just need to update the displayed call information   
 $scope.$apply();   
};   
   
The onNewChatItem() function uses the GetPromptTexts() method if it is defined to populate the   
prompts text. To distinguish between multiple instances of the gadget for different calls, the Title   
property of each instance is being updated to display the caller and the number of messages   
exchanged, updated on every NewChatItem event.

CCSP UI Integrations Guide - 7.3 and 7.4   
155   
To tidy up the subscriptions when the window is closed, you can now use the clearSubscriptions()   
function.   
 $window.onunload = function () {   
 loggerService.WriteLine("Demo Chat Prompt has been closed",   
\_root.enghouse.enums.Severity.Debug, "demoChatPrompt");   
 self.clearSubscriptions();   
 };   
 }]);   
   
Outside of the controller, you need to add the functionality that initializes the gadget with the call   
object when it is opened. Define the angular.element(document).ready() function as below.   
angular.element(document).ready(function () {   
 \_root.app.GadgetComm.windowReady(window);   
};   
   
This calls the GagdetComm object's windowReady() function that fires the window method to   
initialize the gadget. In this case, that method is InitDemoChatPrompt(), which is defined as below.   
function InitDemoChatPrompt(call) {   
 if (call) {   
 var scope =   
angular.element(document.getElementById('demoChatPrompt')).scope();   
 scope.$apply(function () {   
 scope.SubscribeToCall(call);   
 });   
 }   
};   
   
The function obtains the angularJS scope property of the demoChatPrompt element and calls the   
SubscribeToCall() method to initialize the gadget. The call must be wrapped in a $apply() function   
call to ensure that any bindings in the gadget HTML are updated.

CCSP UI Integrations Guide - 7.3 and 7.4   
156   
Opening the standard gadget from DemoChatPrompt functional gadget   
The original DemoChatPrompt functional gadget can now be modified to open and close the   
DemoChatPromptWindow standard gadget.   
In DemoChatPromptFunc.js, add the following line after the statements to obtain the gadget's   
properties.   
var demoChatPromptGadget = \_root.app.GadgetComm.getGadget("DemoChatPrompt");   
   
Note   
If using CCSP versions prior to 7.2.2, see the getGadget function definition for the statement to   
use.   
   
In the onNewChatItem, change the handling of the question response to do the following.   
var responseText = response[0].toUpperCase();   
if (responseText == "YES") {   
 chatCalls[x].call.SendChatMessage(responseIsYes);   
 var infoURL = \_root.constants.rootPath +   
 "app/gadgets/ChatPrompt/chatanswer.html?CallerName=" +   
chatCalls[x].call.CallInfo.Caller.Name + "&answer=" + responseText;   
 window.open(infoURL);   
} else {   
 chatCalls[x].call.SendChatMessage(responseIsNo);   
 if (demoChatPromptGadget) {   
 if (demoChatPromptGadget.OpenWindowsCount <   
demoChatPromptGadget.AllowedInstances) {   
 chatCalls[x].promptWindowId =   
\_root.app.GadgetComm.OpenGadgetInstance("DemoChatPrompt",   
"InitDemoChatPrompt", chatCalls[x].call);   
 }   
 }   
}   
   
Now, the URL is only invoked if the caller response is "Yes". Otherwise a check is made for the   
existance of the standard gadget and then its properties are checked to ensure that the number of   
opened instances hasn't exceeded the limit. If these checks pass, the   
OpenGadgetInstance()function is called with the name of the initialization function and the call   
object.

CCSP UI Integrations Guide - 7.3 and 7.4   
157   
Note   
The code stores the return value from OpenGadgetInstance() as the windowId for the call's   
prompt window. This return value is not implemented in CCSP versions prior to 7.2.2, so the only   
way to obtain this is to use the Windows property of the gadget object, for example:   
chatCalls[x].promptWindowId =   
demoChatPromptGadget.Windows[demoChatPromptGadget.Windows.length - 1].Id;.   
Closing the standard gadget from DemoChatPrompt functional gadget   
To close the Demo Chat Prompt gadget after the call has ended, add the following code to the   
"Destructed" case statement in DemoChatPromptFunc.js.   
if (chatCalls[x].promptWindowId) {   
 \_root.app.GadgetComm.closeComponentWindow(chatCalls[x].promptWindowId);   
}   
Testing the gadgets   
With all three of the Demo Chat gadgets enabled, login to the Agent and send in a chat call. The call   
is still automatically answered and the initial question sent. From the caller, send a "No" response.   
The Demo Chat Prompt gadget should now be displayed with the call details. From the caller, send   
the text "van insurance". The Demo Chat Prompt gadget should now display the message "van   
insurance" and the prompts associated with that keyword.   
Exchange a few more messages and then hang up the call. When the call is destructed, the Demo   
Chat Prompt gadget window is removed.   
Hiding the Demo Chat Prompt gadget icon   
In a real system, the Demo Chat Prompt gadget should only be activated in response to a new chat   
message and the agent should not be able to activate it from the gadgetory itself. Using the   
Behavior property of the gadget, you can hide the gadget icon from the agent, while still having the   
gadgtet appear when needed.   
In the Demo Library Functions gadget, add another createClass statement as below.   
createClass('.hiddenGadget', "display: none;");   
...   
The Demo Chat Prompt gadget's Behavior property can now be set to "hiddenGadget" by the   
DemoChatPrompt functional gadget.   
var demoChatPromptGadget = \_root.app.GadgetComm.getGadget("DemoChatPrompt");   
demoChatPromptGadget.Behavior = 'hiddenGadget';   
Log in to the Agent again and the DemoChatPrompt gadget icon no longer appears in the   
gadgetory. However, when the chat call is sent in and the no response made to the initial question,   
the gadget itself still appears.

CCSP UI Integrations Guide - 7.3 and 7.4   
158   
Using the Controller URL property   
The Controller URL property of a standard gadget can replace the need for a functional gadget to   
subscribe to events before the standard gadget is opened. The script specified as the Controller   
URL is loaded on login just like a functional gadget, so it can perform the same subscriptions.   
This can be demonstrated using the MiniCRM gadget that is installed as part of CCSP UI. If it is not   
already part of the gadget set, add the Mini CRM as a new standard gadget in CCSP AdminPoint   
and set the following parameters:   
•   
Target URL (HTML): app/gadgets/MiniCRM/MiniCRM.html   
•   
Controller URL: app/gadgets/MiniCRM/init.js   
   
The init.js script subscribes to the NewCall event and pops up the gadget with the call details. Add   
the gadget to the appropriate customization in CCSP AdminPoint and then log in to the test agent.   
Now when you make or receive a call in CCSP UI, the Mini CRM gadget appears with the call details.   
Persisting data between gadgets and logins   
Data can be shared between gadgets and gadget instances using objects and functions provided by   
functional gadgets (see Inter-gadget access example) and programmatically adding properties to   
standard gadgets (see Changing the Icon and SelectedIcon properties). In addition, data can be   
persisted between different gadgets and between login sessions using the following services   
provided by CCSP UI:   
•   
viewModelStoreService   
•   
localSettingsService   
•   
DALService   
   
For details on each service, see the individual descriptions in the section 4: CCSP UI AngularJS   
services. In summary:   
•   
viewModelStoreService and localSettingsService allow for data to be shared with different   
gadgets in single login session   
•   
localSettingsServices and DALService allow for data to be persisted between different login   
sessions for an Agent.

CCSP UI Integrations Guide - 7.3 and 7.4   
159   
CCSP UI and web storage   
Web storage is a mechanism by which data is stored as a key-value pair for a client browser. It   
consists of session storage which persists data only for the lifetime of the client browser window and   
local storage which persists data across multiple instances of the same browser.   
CCSP UI uses local storage for persisting data in web storage. The storage is performed for each   
user, for each client browser, meaning that data saved in a Chrome CCSP UI session cannot be   
accessed by the same user when logged onto CCSP UI using Edge.   
The availability of local storage can be affected by settings in the browser. Local storage is not   
maintained in specific browsers under the following conditions:   
•   
Edge: enabling Clear browsing history on close with the Cookies and other site data option   
enabled in Settings → Privacy, search, and services.   
•   
Chrome: enabling Keep local data only until you quit your browser in Settings → Advanced   
Settings → Content Settings.   
•   
Firefox: enabling Clear history when Firefox closes in Privacy with Cookies enabled.   
Using viewModelStoreService   
A view model in the context of CCSP UI is an object that describes a business logic entity such as a   
call or application window. The viewModelStoreService is used by the application code to store a   
single instance of an email call, the chat app, the assist window and other entities that only occur   
once and are used by different components of the application. Gadgets can create their own View   
models to represent specific business logic and store them in the viewModelStoreService so that   
state data can be maintained between gadget instances. The view model object itself usually   
subscribes to events to maintain its own data and can update the application scope, that is, the   
display, if the scope has been injected as a constructor parameter or property.   
This is demonstrated using a View Model to maintain a log of all messages exchanged in a chat call.   
This is set up by the Demo Chat Prompt functional gadget and used by the standard gadget.

CCSP UI Integrations Guide - 7.3 and 7.4   
160   
Chat Log View Model   
This is a simple function added to the DemoChatPromptFunc.js file. It registers for the   
NewChatItem event of a specific call and record all messages with their sender. The code is as   
follows.   
DemoChatLogVM = function ($scope, callId, eventServiceWrapper) {   
   
 var self = this;   
 $.extend(true, self, new SubscriberBase(eventServiceWrapper));   
 self.chatMessages = [];   
 self.scope = $scope;   
   
 self.addSubscription("NewChatItem", callId, function(chatEvent){   
 self.chatMessages.push({ "Sender": chatEvent.chatItem.Who, "Message":   
chatEvent.chatItem.Text });   
 self.scope.$apply();   
 });   
   
 self.destroy = function ()   
 {   
 self.clearSubscriptions();   
 }   
}   
Creating the Chat Log View Model   
A view model is created for each call by the Demo Chat Prompt functional gadget when it receives a   
NewCall event. During the initialization of the function itself in DemoChatPromptFunc.js, it needs to   
get the reference to the viewModelStoreService.   
var vmStoreService = localInjector.get('viewModelStoreService');   
   
Then in the NewCall event handler, the Chat Log view model is created after first ensuring that an   
instance does not already exist.   
if (vmStoreService.checkOut("demoChatLog\_" + callEvent.call.CallId) == null) {   
 chatLogVM = new DemoChatLogVM(\_root.angular.element("[ng-  
controller]").scope(),   
 callEvent.call.CallId, eventServiceWrapper);   
 vmStoreService.checkIn("demoChatLog\_" + callEvent.call.CallId, chatLogVM);   
}   
   
The reference for the View Model includes the CallId to ensure uniqueness.

CCSP UI Integrations Guide - 7.3 and 7.4   
161   
Updating the Demo Chat Prompt standard gadget   
The Demo Chat Prompt standard gadget currently displays the last message received, but this is   
only the first message that has been received after the gadget has been displayed and subscribed   
to NewChatItem. However, this would only be the count of messages that it is aware of during the   
gadget instance's existence.   
Using the Chat Log View Model, the Demo Chat Prompt gadget can now always be displayed with   
the up-to-date number of messages and latest message.   
Add the viewModelStoreService to the list of services used by the demoChatPromptApp and   
injected into the demoChatPromptController. Within the SubscribeToCall() function, add the   
following code to obtain the view model and populate the gadget's data.   
$scope.chatLogVM = viewModelStoreService.checkOut("demoChatLog\_" +   
call.CallId);   
if ($scope.chatLogVM != null) {   
 $scope.messageCount = $scope.chatLogVM.chatMessages.length;   
 $scope.lastMessage =   
$scope.chatLogVM.chatMessages[$scope.chatLogVM.chatMessages.length -   
1].Message;   
 $scope.lastSender =   
$scope.chatLogVM.chatMessages[$scope.chatLogVM.chatMessages.length -   
1].Sender;   
 $scope.thisInstance.Title = $scope.callWrapper.call.CallInfo.Caller.Name +   
" " + $scope.messageCount + " messages received";   
}   
   
The messageCount, lastMessage and lastSender variables now take their values from the view   
model's state so the initial gadget display is now consistent with the state of the conversation.   
To test this, log in to the test Agent and send in a chat call. The call is answered as before, but   
before entering the "no" response to the initial question, enter a couple of invalid responses. When   
you enter the "no" response, the Demo Chat Prompt gadget appears with the latest message,   
which should be the second question from the agent, and the total number of messages   
exchanged in the gadget's title bar.

CCSP UI Integrations Guide - 7.3 and 7.4   
162   
Using DALService and localSettingsService   
DALService uses the web browser's local storage for saving and retrieving data. The data saved is   
then persistent across different CCSP UI login sessions for the agent, so it can be useful for saving a   
user's gadget preferences. The localSettingsService object also uses the web browser storage (it   
does in fact use the DALService functions) but also offers the option of just storing the data in   
memory.   
The use of the two services is demonstrated by adding functionality to the Demo CRM gadget to   
store, retrieve, and delete data using both methods.   
Add data controls to Demo CRM gadget   
In the DemoCRM.html file, add a new div element that contains text fields to enter a key name and   
simple string value for the data and a textarea element to display retrieved data. The example code   
is as follows.   
<div>   
 <form title="DALService Example">   
 <label for="saveDALKeyInput">Enter DAL Key name</label>   
 <input type="text" id="saveDALKeyInput" placeholder="e.g.   
MyDALKeyName" />   
 <label for="saveDALDataInput">Enter DAL Data value</label>   
 <input type="text" id="saveDALDataInput" placeholder="e.g. SomeValue"   
/>   
 <button ng-click="saveDALData()">Save DAL Data</button>   
 <br />   
 <span>   
 <label for="getDALKeyInput">Enter DAL Key name</label>   
 <input type="text" id="getDALKeyInput" placeholder="e.g.   
MyDALKeyName" />   
 <button ng-click="getDALData()">Get DAL Data</button>   
 <label id="DALresult">Result</label><textarea disabled   
style="width:250px; height: 15px; vertical-align: middle">{{ retrievedDALData   
}}</textarea>   
 </span>   
 <br />   
 <label for="removeDALKeyInput">Enter DAL Key name</label>   
 <input type="text" id="removeDALKeyInput" placeholder="e.g.   
MyDALKeyName" />   
 <button ng-click="removeDALData()">Remove DAL Data</button>   
 </form>   
</div>

CCSP UI Integrations Guide - 7.3 and 7.4   
163   
This sets up the controls for the DALService example. Create a similar div element for the   
localSettingsService example.   
<div>   
 <form title="localSettingsService Example">   
 <label>Use DAL in localSettingsService</label><input type="checkbox"   
ng-model="useDALInLocal"/><br />   
 <label for="saveLocalKeyInput">Enter Key name</label>   
 <input type="text" id="saveLocalKeyInput" placeholder="e.g. MyKeyName"   
/>   
 <label for="saveLocalDataInput">Enter Data value</label>   
 <input type="text" id="saveLocalDataInput" placeholder="e.g.   
SomeValue" />   
 <button ng-click="saveLocalData()">Save Local Data</button>   
 <br />   
 <span>   
 <label for="getLocalKeyInput">Enter Key name</label>   
 <input type="text" id="getLocalKeyInput" placeholder="e.g.   
MyKeyName" />   
 <button ng-click="getLocalData()">Get Local Data</button>   
 <label id="localResult">Result</label><textarea disabled   
style="width:250px; height: 15px; vertical-align: middle">{{   
retrievedLocalData }}</textarea>   
 </span>   
 <br />   
 <label for="removeLocalKeyInput">Enter Key Name</label>   
 <input type="text" id="removeLocalKeyInput" placeholder="e.g.   
MyKeyName" />   
 <button ng-click="removeLocalData()">Remove Local Data</button>   
 </form>   
</div>   
   
Apart from the change in variable names, the main difference between the two div elements is the   
addition of the Use DAL in localSettingsService check box.

CCSP UI Integrations Guide - 7.3 and 7.4   
164   
Update the Demo CRM code   
In DemoCRM.js, declare DALService and localSettingsService as part of the directServices array and   
add them to the demoCRMController declaration. Also declare the scope variables used in the   
ngModel directives in the controller:   
$scope.retrievedDALData = "";   
$scope.retrievedLocalData = "";   
$scope.useDALInLocal = false;   
   
For the DALService example, the saveDALData, getDALData, and removeDALData functions are   
implemented as below.   
$scope.saveDALData = function () {   
 var saveDALKeyInput = document.getElementById("saveDALKeyInput");   
 var saveDALDataInput = document.getElementById("saveDALDataInput");   
 if (saveDALDataInput && saveDALDataInput.value != "" && saveDALKeyInput &&   
saveDALKeyInput.value != "") {   
 DALService.SetValue(saveDALKeyInput.value, saveDALDataInput.value);   
 }   
};   
   
$scope.getDALData = function () {   
 var getDALKeyInput = document.getElementById("getDALKeyInput");   
 if (getDALKeyInput && getDALKeyInput.value !== "") {   
 $scope.retrievedDALData = DALService.GetValue(getDALKeyInput.value);   
 if ($scope.retrievedDALData != null) {   
 document.getElementById('DALresult').innerHTML = "Found!";   
 } else {   
 document.getElementById('DALresult').innerHTML = "Not found!";   
 }   
 }   
};   
   
$scope.removeDALData = function () {   
 var removeDALKeyInput = document.getElementById("removeDALKeyInput");   
 if (removeDALKeyInput && removeDALKeyInput.value !== "") {   
 DALService.DeleteValue(removeDALKeyInput.value);   
 }   
};

CCSP UI Integrations Guide - 7.3 and 7.4   
165   
The implementation of the equivalent functions for the localSettingsService example is as follows.   
$scope.saveLocalData = function () {   
 var saveLocalKeyInput = document.getElementById("saveLocalKeyInput");   
 var saveLocalDataInput = document.getElementById("saveLocalDataInput");   
 if (saveLocalDataInput && saveLocalDataInput.value != "" &&   
saveLocalKeyInput && saveLocalKeyInput.value != "") {   
 localSettingsService.SetValue(saveLocalKeyInput.value,   
saveLocalDataInput.value, $scope.useDALInLocal);   
 }   
};   
   
$scope.getLocalData = function () {   
 var getLocalKeyInput = document.getElementById("getLocalKeyInput");   
 if (getLocalKeyInput && getLocalKeyInput.value !== "") {   
 $scope.retrievedLocalData =   
localSettingsService.GetValue(getLocalKeyInput.value, $scope.useDALInLocal);   
 if ($scope.retrievedLocalData != null) {   
 document.getElementById('localResult').innerHTML = "Found!";   
 } else {   
 document.getElementById('localResult').innerHTML = "Not found!";   
 }   
 }   
};   
   
$scope.removeLocalData = function () {   
 var removeLocalKeyInput = document.getElementById("removeLocalKeyInput");   
 if (removeLocalKeyInput && removeLocalKeyInput.value !== "") {   
 localSettingsService.DeleteValue(removeLocalKeyInput.value,   
$scope.useDALInLocal);   
 }   
};

CCSP UI Integrations Guide - 7.3 and 7.4   
166   
Saving, retrieving and deleting the data   
With the changes made, reload the gadget. Use the new controls to save and retrieve data using   
both methods and with the Use DAL in localSettingsService button checked and unchecked. The   
following table summarizes how the different storage, retrieval, and deletion methods affect the   
retrieval of data using the same key name in all cases.   
Save method   
Removal method   
Retrieval method   
Retrieval outcome   
Using   
"DemoKey1000"   
   
   
   
DALService   
   
DALService   
Found   
DALService   
   
localSettingsService no DAL   
Not found   
DALService   
   
localSettingsService with   
DAL   
Found   
DALService   
   
localSettingsService no DAL   
Found (key now   
present in   
localSettingsService)   
Log out and   
back in   
   
   
   
n/a   
   
DALService   
Found   
n/a   
   
localSettingsService no DAL   
Not found   
n/a   
   
localSettingsService with   
DAL   
Found   
   
   
localSettingsService no DAL   
Found (key now   
present in   
localSettingsService)   
n/a   
localSettingsService   
no DAL   
DALService   
Found   
n/a   
n/a   
localSettingsService no DAL   
Not found   
n/a   
n/a   
localSettingsService with   
DAL   
Found   
   
DALService   
localSettingsService with   
DAL   
Not found   
   
n/a   
DALService   
Not found   
   
Table 6: DALService and localSettingService usage summary

CCSP UI Integrations Guide - 7.3 and 7.4   
167   
Saving and loading gadget properties   
The use of DALService can be combined with the update of a gadget's properties from within the   
gadget or another gadget to persist gadget properties and data between login sessions. This is   
demonstrated in the SimpleGadget example which uses an object from the Demo Library Functions   
gadget to save and load properties in local storage.   
Simple gadget example   
The SimpleGadgetProperties() function is defined in DemoLibfraryFunctions.js and obtains a   
references to the DALService using the Angular injector method and the SimpleGadget object. It   
then provides, saves, loads, and removes functions to manage the storage and retrieval of the   
SimpleGadget's default properties and its MyProps object.   
var dalService = localInjector.get('DALService');   
var simpleGadget = \_root.app.GadgetComm.getGadget("SimpleGadget");   
   
self.save = function () {   
 var props = {   
 MyProps: simpleGadget.MyProps,   
 DefaultState: simpleGadget.DefaultState,   
 DefaultPositionX: simpleGadget.DefaultPositionX   
 ...   
 };   
 dalService.SetValue("CCSPSimpleGadgetExampleProperties", props);   
};   
   
self.load = function () {   
 var props = dalService.GetValue("CCSPSimpleGadgetExampleProperties");   
 if (props) {   
 simpleGadget.DefaultState = props.DefaultState ||   
simpleGadget.DefaultState;   
 simpleGadget.DefaultPositionY = props.DefaultPositionY ||   
simpleGadget.DefaultPositionY;   
 ...   
 }   
};   
For this example, the properties are loaded from the DAL on login, so after the function declaration,   
a simpleGadgetProps variable is created to load the properties.   
var simpleGadgetProps = new SimpleGadgetProperties();   
simpleGadgetProps.load()

CCSP UI Integrations Guide - 7.3 and 7.4   
168   
This simpleGadgetProps variable is then referenced as a property of the root window by the   
SimpleGadget.js code. The Simple Gadget window provides two controls for DAL management:   
•   
Store and Retrieve Properties in DAL check box   
•   
Delete Properties in DAL button   
   
When the check box is enabled, the window.onunload() function in SimpleGadget.js saves the   
properties using the simpleGadgetProps object.   
window.onunload = function () {   
 ...   
 \_gadget.MyProps.useDAL = document.getElementById('useDAL').checked;   
 if (\_gadget.MyProps.useDAL && \_root.simpleGadgetProps) {   
 \_root.simpleGadgetProps.save();   
 }   
};   
   
The handler for the Delete Properties in DAL button click event simply calls the remove() function   
of simpleGadgetProps.   
deleteDAL = function () {   
 if (\_root.simpleGadgetProps) {   
 \_root.simpleGadgetProps.remove();   
 }   
};   
   
The only change in the angular.element(document).ready function, is to set the check box status   
according to the MyProps value and increment the title.   
if (\_gadget.MyProps.useDAL) {   
 updateTitle();   
 document.getElementById('useDAL').checked = \_gadget.MyProps.useDAL;   
}   
   
There is nothing else to do in angular.element(document).ready, because all the properties have   
already been loaded and used within the gadget instance.   
When using the gadget, you can see that changes to the Icon, Behavior, and Title fields and when   
displayed as a float, changes in the gadget's size and position, persist not only between gadget   
instances but between browser sessions.   
The SimpleGadgetProperties code in DemoLibraryFunctions.js can be improved and refactored to   
enable it to be used by any gadget.

CCSP UI Integrations Guide - 7.3 and 7.4   
169   
6: Migrating 6.1.X Web Agent   
integrations   
Legacy Integrations.js in CCSP UI   
The 6.1.X Web Agent enabled customers to integrate applications into the Web Agent through an   
Integration tab that could display a web site and perform screen pops on new calls or other Web   
Agent actions. Such integrations can be migrated to CCSP UI.   
The 6.1.X Web Agent integration was made possible by the use of the Integrations.js (plural) file that   
was notified of Web Agent events and in turn raised asynchronous events to be consumed by the   
customer-supplied Integration.js (singular) file. Integrations.js has been incorporated into separate   
functional and standard gadgets that enable the activation and display of the content that is   
associated with Integration.js. The gadgets subscribe to CCSP UI and AJAX API events, then pass the   
event data to the Integration.js event handler functions using the same mechanism developed for   
the 6.1.X Web Agent.   
Note   
Because Integrations.js was developed for the 6.1.X Web Agent, the gadget is only written and   
supported for Microsoft Edge in IE mode or Google Chrome with the IE Tab extension.   
If an integration is required to run in other browsers such as Chrome and Firefox, we   
recommend that customers create an entirely new gadget.   
Integrations gadget structure   
Figure 2 shows the components of the Integrations.js gadget and the relationship with the CCSP UI   
application and the AJAX API.

CCSP UI Integrations Guide - 7.3 and 7.4   
170   
   
Figure 2: Legacy integrations structure in CCSP UI   
   
AsyncEventManager.js: ported from 6.1.X Web Agent to transmit the events that are consumed by   
Integrations.js functions to the Integration.js event handlers.   
Integrations.js: defines the 6.1.X Integrations\_\* event handlers that are called by the   
IntegrationsViewModel. These functions in turn call the AsyncEventManager to pass on the event   
notifications.   
IntegrationsViewModel.js: subscribes to the various CCSP UI events and calls the equivalent   
Integrations\_\* event handlers to pass on the notification.   
IntegrationsFunc.js: loads the Integrations script files including the custom Integration.js and   
creates the IntegrationsViewModel object. This is the script that is specified as the Target URL for   
the Integrations functional gadget.   
IntegrationsView.html: a simple page that is the Target URL for the Integrations standard gadget   
and includes the IntegrationsFunc script. It provides the equivalent of the 6.1.X Web Agent   
Integrations tab, allowing customisations to display web pages there.   
IntegrationsInit.js: the default Controller script file for the Integrations standard gadget. It can be   
used to register with events on agent login so that it can display the Integrations standard gadget   
in response to specific events. This contains commented out example code.

CCSP UI Integrations Guide - 7.3 and 7.4   
171   
Adding a 6.1.X integration to CCSP UI   
The Integrations functional and standard gadgets are included as part of the CCSP UI build and are   
part of the Default Gadgets available in CCSP AdminPoint. When added to the tenant, the   
customized integration file can be specified individually for tenant/group/role for each gadget.   
Note   
The use of the Integrations functional and standard gadgets is mutually exclusive within a single   
login session. This is because only one integration is supported (as with the 6.1.X Web Agent)   
and there is a possibility of the Integration\_\* event handler being called more than twice for the   
same event if both gadgets are included.   
The gadgets can both be present in a tenant/group/role, but one or the other must have its   
Disabled property set to True. If the standard gadget needs to register with events or do other   
processing prior to it being displayed, the functionality should be added to the IntegrationsInit.js   
script.   
Adding the Integrations functional gadget   
   
1.   
Add the Integrations functional gadget to the tenant in CCSP AdminPoint as follows:   
a.   
Log onto CCSP AdminPoint with the appropriate privileges   
b.   
Edit the gadgets for the tenant and add the Integrations functional gadget.   
c.   
Edit the gadget properties and change the IntegrationURL to point to the script that   
defines the Integration\_ event handlers (usually Integration.js). This needs to be relative to   
the TouchPoint virtual folder, for example:   
app/gadgets/Integrations/Functional/Integration.js.   
d.   
Save the changes.   
2.   
Migrate the Integration.js file. There are a number of changes that need to be made to the   
Integration.js script for it to work with CCSP UI. The table below lists the major changes   
required, which can mostly be performed by a search and replace of the 6.1.X values with their   
7.2 equivalents.   
Note   
The Integration.js must define the function Integration\_AgentStarted() to be loaded.   
3.   
Test the Integration. Log onto CCSP UI. The expected outcome is that the Integrations   
functional gadget registers for CCSP UI events when the main page has loaded and sends the   
Integrations\_AgentStarted event to the integration.

CCSP UI Integrations Guide - 7.3 and 7.4   
172   
Adding the Integrations standard gadget   
   
1.   
Add the Integrations standard gadget to the tenant in CCSP AdminPoint as follows:   
a.   
Log onto CCSP AdminPoint with the appropriate privileges   
b.   
Edit the gadgets for the tenant and add the Integrations standard gadget.   
c.   
Edit the gadget properties and change the IntegrationURL to point to the script that   
defines the Integration\_ event handlers (usually Integration.js). This needs to be relative to   
the TouchPoint virtual folder, for example:   
app/gadgets/Integrations/AgentTestDriver/Integration.js.   
d.   
If a Controller script other than IntegrationsInit.js is required, edit the Script URL property   
to point to script to be used.   
e.   
Edit the other properties as appropriate.   
f.   
Save the changes.   
2.   
Migrate the Integration.js file. There are a number of changes that need to be made to the   
Integration.js script for it to work with CCSP UI. The table below lists the major changes   
required, which can mostly be performed by a search and replace of the 6.1.X values with their   
7.2 equivalents.   
Note   
The Integration.js must define the function Integration\_AgentStarted() to be loaded.   
   
3.   
Add supporting JS and CSS files. If specific files from the 6.1.X Web Agent are required to   
support the integration such as AgentStyles.css and Library.vbs, include them in the   
IntegrationsView.html file although the actual file locations should still be in the Integration   
folder. For example.   
<script type="text/javascript" src="Integration/Library.js"></script>   
4.   
Test the Integration. Log onto CCSP UI. Opening the Integrations gadget should cause it to   
register with the CCSP UI events and send the Integrations\_AgentStarted event to the   
integration.   
After the Agent has access to the Integrations standard gadget, changes can be made to   
Integration.js and its associated files, with the results seen in CCSP UI without having to relogin.   
The Integrations gadget is refreshed every time it is closed and re-opened.   
6.1.X Integration.js script migration   
These are the changes in the Agent AJAX API and HTML document handling that need to be made   
in order for a 6.1.X Integration.js script to be migrated.   
In some cases, fields that were object properties in 6.1.X are now array elements in 7.X and require   
the use of convenience functions such as those in helpers.js to extract their values. The most useful   
functions from helpers.js are as follows.   
Note   
This list may not cover all changes required due to customers making use of other functions and   
properties.

CCSP UI Integrations Guide - 7.3 and 7.4   
173   
Object   
6.1.X   
7.X Equivalent   
Agent   
Property: WrapUpCodes   
WrapupCodesList   
Call   
Property: ParentAgent   
agent   
Call   
Property: MediaType   
CallInfo.Type   
Call   
Function: PickUpEnsureMedia()   
Removed, use PickUp()   
Call   
Function:   
Call.Line.WrapUpDialog.contentWind  
ow.WrapUpCall()   
Removed, use Call.FinishWrapUp('0', "")   
Call   
Property: Direction   
CallInfo.Direction   
Call   
Property: IsAgentToAgent   
CallInfo.IsAgentToAgent   
Call   
Property: AgentFirstName   
CallInfo.CallFromAgent.FirstName (for   
inbound A2A call)   
CallInfo.CallToAgent.FirstName (for outbound   
A2A call)   
Call   
Property: AgentLastName   
CallInfo.CallFromAgent.LastName (for   
inbound A2A call)   
CallInfo.CallToAgent.LastName (for outbound   
A2A call)   
Call   
Property: EnteredNumber   
StreamInfo.PhoneNumber   
Call   
Property: CallerANI   
CallInfo.Caller.ANI   
Call   
Property: CallerDNIS   
CallInfo.Caller.DNIS   
Call   
Property: CallInfo.IsInvitation   
CallInfo.IsTransferred   
Call   
Property:CallInfo.MediaInfo.Property   
Use getItemByKey() to get the MediaInfo   
ConfigItem, for example:   
var item =   
getItemByKey(CallInfo.Item,   
"MediaInfo")   
Use getValueByKey() to get the individual   
properties from the MediaInfo ConfigItem, for   
example:   
var prop = getValueByKey(item,   
"VCSAgentLocalAddress")

CCSP UI Integrations Guide - 7.3 and 7.4   
174   
Object   
6.1.X   
7.X Equivalent   
Call   
Property: OptPar.Property   
Use getValueByKey() functionon relevant   
object.   
Chat and Voice   
var value =   
getValueByKey(CallInfo.OptionalParam  
s, "Property")   
Email   
var value =   
getValueByKey(CallInfo.EmailInfo.Opt  
ionalParameters, "Property")   
Callback   
See Note below   
Browse  
r frame   
Property: document, for example,   
Browser.document   
Use contentWindow.document, for example,   
Browser.contentWindow.document   
Browse  
r frame   
Function: navigate()   
Use src property, for example, browser.src =   
<url>   
   
Table 7: 7.X equivalents of 6.1.X object properties   
   
Note   
For Callback objects, the CallbackInfo.OptionalParameters property contains an array called   
OptionalParameterClass. This is an array of objects with the properties of {Key, Value} rather   
than {Id, Value} as is the case for the other Call types. Therefore, the getValueByKey function will   
not work with this object. To get a specific optional parameter, do a search on the Key property   
of the CallbackInfo.OptionalParameters.OptionalParameterClass elements. For example:   
   
var key = "CID";   
var value = "";   
var params =   
call.CallInfo.CallbackInfo.OptionalParameters.OptionalParameterClass;   
for (var i = 0; i < params.length; i++) {   
 if (key === params[i].Key) {   
 value = params[i].Value;   
 break;   
 }   
}   
   
This will be fixed in a later version of CCSP so that the OptionalParameters property can be used   
in the same manner as the one in the EmailCall object.

CCSP UI Integrations Guide - 7.3 and 7.4   
175   
Examples   
CCSP UI 7.2.2 includes two examples of integrations that have been migrated from 6.1.X. These are   
the integrations that were distributed as part of the 6.1.X Web Agent in the   
CosmoAgent\integrationExamples folder:   
•   
Agent Test Driver   
•   
Profiler   
   
The integrations are located in folders under TouchPoint\App\gadgets\Integrations. To incorporate   
an integration, change the IntegrationURL property of the required Integrations gadget.   
Note   
The functionality is unchanged from the 6.1.X version: they are simply to provide working   
examples of migrated integrations.   
Troubleshooting   
Customer integration doesn't load:   
•   
Ensure that the IntegrationURL points to the correct file and location.   
•   
Open Developer Tools in the browser and check the console log for errors.   
•   
Ensure that at a minimum, the Integration\_AgentStarted() method is defined in the   
IntegrationURL file. If this function is not defined, an attempt to display the Integrations   
standard gadget just shows the message "No Integration loaded".   
   
Customer integration raises undefined property/function errors:   
•   
Ensure that all replacements have been performed for each table.   
•   
If there are still undefined errors on agent or call functions/properties, check the CCSP Agent   
AJAX API Reference for differences.   
   
Stylesheets not found:   
•   
Ensure that any references to stylesheets and other files are updated to point to the file in the   
Integration folder, not the parent Integrations one. The Integrations folder itself should not   
contain any integration-specific files.   
   
Permission Denied error when trying to change the document domain:   
•   
There should be no need to change the document domain so lines that do so should be   
removed. If this is required for some reason, consider creating an iframe within the document   
and changing the location of that instead.

CCSP UI Integrations Guide - 7.3 and 7.4   
176   
7: Co-browsing integration   
Overview   
This section describes the framework in CCSP UI that enables you to integrate with a third-party co-  
browsing solution of your choice, instead of the default co-browsing solution supplied as part of   
CCSP. It describes the API provided and how to write the integration code to create your co-  
browsing sessions.   
Concepts   
Co-browsing is the ability for CCSP UI agents to see the screen or the current browser web page of   
a caller that they are in conversation with. It can be achieved in different ways:   
•   
Embedded code in Web page (direct co-browsing)   
•   
Web page shared via proxy server   
•   
Screen sharing using WebRTC   
   
Many companies provide co-browsing solutions of one of the above types, often as a feature to be   
integrated into larger customer service solutions. The solution that is packaged with CCSP is based   
on togetherJS, which is an open-source embedded code solution, originally developed by Mozilla.   
We recognize that this will not be suitable for some service providers and tenants, therefore CCSP   
UI provides the facility to integrate with other third-party solutions.   
Co-browsing session lifecycle   
Regardless of which co-browsing technology is used or who develops it, a co-browsing session has   
a definite lifecycle and each step can be classed as mainly being the responsibility of either the   
CCSP UI domain or third-party provider domain.   
The lifecycle of a co-browsing session can be broken down as follows:   
1.   
Initiate the co-browsing session: either from the agent chat window or from a chat call   
initiated by a customer browsing the tenant's web site. (CCSP UI domain)   
2.   
Establish the co-browsing session: perform the handshaking, resource allocation, hosting,   
and rendering to set up the co-browsing session. (Third-party provider domain)   
3.   
Connect to the co-browsing session: on both the agent and customer side, the shared   
content is displayed by accessing the URL that defines the session. (CCSP UI domain)   
4.   
Work in the co-browsing session: navigate both sides to different pages and sites, highlight   
areas, display cursor on the other side. Also switch between agent and customer control of the   
shared session. (Third-party provider domain)

CCSP UI Integrations Guide - 7.3 and 7.4   
177   
5.   
End the co-browsing session: can be done from either the agent or customer side,   
independently of closing the call, but when the call is closed, so is the co-browsing session if it   
is still open. (Either domain)   
   
Figure 3 shows the co-browsing session lifecycle.   
CCSP TouchPoint  
Third Party Provider  
1. Initiate Browsing   
Session  
2. Establish 3rd party   
specific co-browsing   
session  
3. Connect caller to   
co-browsing session  
4. Work in co-browsing   
session  
5. End call/End co-  
browsing session  
5. End co-browsing   
session  
Call data, HTML display element  
Session URL  
Display element for session  
Session exit initiator must  
notify the other domain  
Change page, change controller  
   
Figure 3: Co-browsing session lifecycle   
   
From the above stages, the responsibilities of the third-party provider solution are therefore:   
•   
Establish the co-browsing session   
•   
Work in the co-browsing session   
•   
End the co-browsing session

CCSP UI Integrations Guide - 7.3 and 7.4   
178   
CCSP co-browsing support   
CCSP UI provides a client-side only framework that is illustrated in Figure 4. The main element of   
this framework is the Co-Browsing gadget. This is a standard gadget that is part of the CCSP UI   
installation and is included in the System gadgets template within CCSP AdminPoint. It is not part of   
the default template for a tenant and you must add it to each tenant or group that requires it.   
The other main component provided by CCSP UI is the CoBrowsingAPI object, which is explained in   
further detail later in this document. The ThirdPartyCode.js represents the script that you must   
provide to integrate with your chosen co-browsing solution.   
   
   
Co-Browsing Gadget  
Co-Browsing.html  
• UI Controls  
• Shared View element  
CoBrowsingViewModel.js  
• UI functions  
• Event subscriptions  
• CoBrowsingStarted event   
publish  
CoBrowsingController.js  
• StartCoBrowsing event   
subscription  
• Gadget Display  
CoBrowsingAPI  
Third Party  
[ThirdPartyCode].js  
• CoBrowsingAPI function   
implementations  
• Code specific to Co-  
browse solution  
CoBrowsingAPI.js  
• Function definitions  
CoBrowsing.js  
• View Model retrieval  
• Gadget initialisation  
   
Figure 4: CCSP UI co-browsing framework

CCSP UI Integrations Guide - 7.3 and 7.4   
179   
Co-bowsing gadget components   
CoBrowsingController.js   
This component is the ControllerURL script for the gadget that is loaded on agent logon. It is   
responsible for opening the gadget in response to the StartCoBrowsing event that is published   
when the agent clicks the Co-browse button in the call card or chat window.   
CoBrowsing.js   
This component provides the angular controller for the gadget and creates the view model object.   
CoBrowsing.html   
This is the template HTML for the gadget.   
CoBrowsingViewModel.js   
This component contains the CCSP UI domain business logic for the co-browsing session and   
provides the functionality for the session controls exposed in the HTML. This object is created when   
the gadget is first created for a call and exists until the gadget is closed via the Exit Session button,   
window close button, or when the call enters wrap up or is destructed.

CCSP UI Integrations Guide - 7.3 and 7.4   
180   
CoBrowsingAPI interface   
The CoBrowsingAPI javascript object provides the interface definitions that the third-party code   
must satisfy for CCSP UI to interact with the co-browsing solution. It defines a set of functions with   
null or default implementations that you must then override as required.   
This section gives an overview of the functions and events that the CoBrowsingAPI object defines. A   
complete reference for the CoBrowsingAPI object is provided in the online CCSP AJAX API Reference.   
Session functions   
These functions should be overridden in order to create, manage, and end the co-browsing session.   
Note   
If either of the CreateSession or ExitSession functions are not overridden, the Co-Browsing   
gadget is not loaded and an error message is displayed to the user.   
CreateSession   
This function is a mandatory override that must create the co-browsing session using whatever   
facilities are supplied by the third-party solution based on the given parameters.   
GoToURL   
This function is an optional override that should navigate the session based on the input provided.   
SwitchController   
This function is an optional override that should attempt to change the controller or presenter of   
the session to the given participant.   
ExitSession   
This function is a mandatory override that must cleanly exit the co-browsing session, including   
destroying the session if this is required.   
Gadget controls functions   
These functions should be overridden to control the availability of the session controls in the   
gadget.   
Note   
You cannot change the availability of the Stop Session button.

CCSP UI Integrations Guide - 7.3 and 7.4   
181   
IsSessionAutoCreated   
When this function returns true, the gadget calls CreateSession as soon as it is displayed and the   
Start Session button is not displayed, regardless of what the IsStartSessionAvailable method   
returns.   
IsSessionControllerAvailable   
When this function returns true, the gadget displays the current controller or presenter.   
IsStartSessionAvailable   
When this function returns true, the gadget displays the Start Session button. Note that if   
IsSessionAutoCreated is true, the return value from this method has no effect and the button is   
hidden regardless.   
IsSwitchControllerAvailable   
When this function returns true, the gadget allows the current controller or presenter to be   
changed.   
IsURLEntryAvailable   
When this function returns true, the gadget displays the URL input field.   
Initialization functions   
Thes functions are used to initialize the gadget according to the co-browsing session parameters.   
GetControlType   
This function specifies what type of control the session allows. This must be one of the predefined   
types that are defined in the ControlType enumeration object:   
•   
Open: any participant can control the session, for example, togetherJS co-browsing   
•   
Presenter: one party presents, the other parties view only, for example, screen sharing   
•   
Controller: one party presents and controls, the other parties view only, for example, Surfly®   
co-browsing   
GetDefaultController   
This function specifies which participant has control or is presenting when the session is created.   
This must be one of the predefined types that are defined in the ControllerType enumeration   
object:   
•   
None: no controller has been selected or can be defined   
•   
Agent: represents the CCSP UI agent   
•   
Caller: represents the external caller

CCSP UI Integrations Guide - 7.3 and 7.4   
182   
GetDisplayType   
This function specifies which kind of HTML element is to be used for the display of the co-browsing   
session. This must be one of the predefined types that are defined in the DisplayType enumeration   
object:   
•   
None: default value when no DisplayType is selected   
•   
Iframe: the iframe element   
•   
Video: the video element   
Events   
The CoBrowsingAPI API exposes events that you can then raise to inform the gadget of changes to   
the session that have been generated from the third-party solution.   
OnControllerChanged   
This event should be raised when the controlling or presenting participant in the session has been   
changed. It updates the selected controller in the gadget.   
OnParticipantAdded   
This event should be raised when a new participant has been joined the session. The participant is   
added to the list of participants shown in the Switch Controller drop-down menu, if it is displayed.   
OnParticipantRemoved   
This event should be raised when a participant has left the session. The participant is removed from   
the list of participants shown in the Switch Controller drop-down menu, if it is displayed. If there is   
only one participant left in the session after removal, the session is automatically closed.   
OnSessionExited   
This event should be raised when the agent's participation in the session has been ended by the   
third-party solution. The gadget is closed and a modal dialog is displayed to indicate that the   
session has been remotely closed.   
OnURLChanged   
This event should be raised when the URL displayed in the session is changed. This is not the same   
as the URL of the session itself, rather it is equivalent to a change in the current website location of   
the caller.

CCSP UI Integrations Guide - 7.3 and 7.4   
183   
Writing co-browsing integrations   
This section shows you how to integrate your own implementation of the CoBrowsingAPI as a   
replacement for the togetherJS implementation that comes with CCSP UI.   
Configuring the gadget   
In CCSP AdminPoint, select the tenant or group for which to change the co-browsing   
implementation and edit the Co-Browsing gadget. Scroll down to the bottom of the properties list   
and edit the API Implementation URL property to point to a new Javascript file.   
   
Figure 5: Edit Co-browsing gadget implementation file   
   
The URL can be either relative to the TouchPoint folder or an absolute URL. Save the change and   
then create the file in the location specified.   
CoBrowsingImplementation object   
The third-party code must define a CoBrowsingImplementation object that inherits from   
CoBrowsingAPI as below.   
CoBrowsingImplementation.prototype = new CoBrowsingAPI();   
This is the object that the CoBrowsingViewModel tries to create when it loads the third-party code.   
All code for this object needs to be in the file specified as the API Implementation URL property of   
the Co-browsing gadget.   
The constructor for the CoBrowsingImplementation needs to call the CoBrowsingAPI constructor   
as in the example below.   
function CoBrowsingImplementation(thisWindow, timeoutFunc, intervalFunc) {   
 CoBrowsingAPI.call(this, thisWindow, timeoutFunc, intervalFunc);

CCSP UI Integrations Guide - 7.3 and 7.4   
184   
The functions from CoBrowsingAPI should then be overridden by defining the function in the   
prototype for the CoBrowsingImplementation object as shown in the example below.   
CoBrowsingImplementation.prototype.CreateSession = function (creationData,   
successHandler, errorHandler) {   
...   
}   
Implementing the session functions   
Asynchronous implementation   
The session control functions must be implemented by the third-party solution to avoid locking the   
single javascript execution thread and thereby locking the GUI. Each session control function   
accepts successHandler and errorhandler callback functions in its parameter list. The appropriate   
function must be called by the third-party code in each case for the Co-Browsing gadget to verify   
the operation as complete, regardless of success or failure.   
After making a call to a session control function, the Co-Browsing gadget waits a set amount of   
time for one of the callbacks to be invoked by the third-party code. If neither are invoked within this   
timeout period, an error is displayed to the agent. If it is the ExitSession function that has timed   
out, the gadget is automatically closed.   
The timeout period is the value of the API Request Timeout property for the Co-Browsing gadget,   
which can be set in CCSP AdminPoint.   
Exception handling   
The third-party code should ensure that all possible exceptions that can be thrown during the   
course of processing a session function are handled and that the error or success callback is still   
invoked. If the third-party code throws an unhandled exception, the Co-Browsing gadget displays   
an error to the user.   
Note that the session functions deliberately use callbacks instead of promises for the asynchronous   
processing as the functionality on the CCSP UI side is fairly simple after a function has completed.   
However, the third-party code can make use of promises itself as required.   
CreateSession implementation   
Note   
Co-browsing is currently only applicable to chat calls.   
   
The third-party code should be capable of handling the following scenarios in the CreateSession   
function.   
•   
Creating a new Session   
•   
Joining a session   
•   
Continuing a session

CCSP UI Integrations Guide - 7.3 and 7.4   
185   
   
The function prototype is:   
function (creationData, successHandler, errorHandler)   
   
Co-browsing can only occur in the context of a call and the creationData parameter to this   
function includes the associated call object. This object is described in the online CCSP AJAX API   
Reference. The other properties of the creationData object are set according to the scenario in   
which the CreateSession function is being called.   
The full list of properties in the creationData object parameter is:   
•   
call   
•   
continueSession   
•   
currentUrl   
•   
elementId   
•   
joinSession   
•   
sessionID   
•   
windowState   
   
The properties are documented in the CoBrowsingAPI reference.   
Creating a new session   
This function is called when the agent clicks on the co-browsing button in the Chat window or on   
the call card in CCSP UI to initiate a new session with the caller. The currentUrl property provides   
the current website location of the caller and therefore the initial page at which the co-browsing   
session should start, if this is applicable to the third-party solution.   
The implementation should create the session using whatever API is provided by the co-browsing   
solution. It must then call the successHandler with the following properties set in the object passed   
to this function:   
•   
customerSessionUrl: string that represents a valid URL to send to the caller for them to join   
the created session   
•   
currentController: integer that represents the participant currently in control of or presenting   
the session. This should be a value from ControllerType   
   
If currentController is omitted, then the Co-Browsing gadget sets the controller to the value   
returned by the GetDefaultController function.

CCSP UI Integrations Guide - 7.3 and 7.4   
186   
Joining a session   
This function is called when the caller has initiated a session on the website page that they are   
currently browsing and has sent a message via the CCSP Chat API that they want the agent to join   
that session. In this scenario, the joinSession flag is set and the sessionID property contains the   
URL of the session or some relevant ID that identifies the session to to be joined.   
The implementation still needs to call the successHandler but the customerSessionUrl value must   
be null or undefined. If the customerSessionUrl is defined, this is automatically sent to the caller.   
Continuing a session   
This function is called when the agent changes the window state of the Co-Browsing gadget by   
clicking the Pop out menu item or button when the gadget is in the Tab or Float state. It is also   
called when performing the reverse operation. In this scenario, the continueSession flag is set.   
If the session was originally joined by an agent instead of being created, the joinSession and   
sessionID properties may be set but the third-party code should not rely on these to rejoin the   
session. The third-party code should instead maintain its own state to determine at what point to   
continue the session.   
The same applies when the session was created from the agent side. The creationData’s currentUrl   
value should not be used as the continuation point for the session in the new window. The third-  
party code itself should keep track of the session’s current location.   
The implementation still needs to call the successHandler but the customerSessionUrl value must   
be null or undefined. If the customerSessionUrl is defined, this is automatically sent to the caller.   
Exiting a session   
The ExitSession function is called by the Co-browsing gadget in the following scenarios:   
•   
When the Exit session button is clicked in the gadget.   
•   
When the gadget window’s close (  
) button is clicked.   
•   
When the call is hung up (enters the Wrap Up or Destructed state)   
   
The function prototype is:   
function (successHandler, errorHandler)   
   
The third-party code must perform whatever actions are necessary to gracefully exit or close the   
co-browsing session and call the successHandler function. Any errors that occur during the exit   
processing should be handled and if necessary, the third-party code should invoke the   
errorHandler with a descriptive message to raise as an alert to the agent.   
If the errorHandler is invoked by the third party code, the gadget is still closed unless a   
keepWindowOpen flag is set to true in the object passed to the errorHandler. If this flag is set, the   
alert is still displayed but the gadget remains open.

CCSP UI Integrations Guide - 7.3 and 7.4   
187   
Caution   
The keepWindowOpen flag should only be set if it is essential that the gadget remain open. This   
should only be the case when the agent is required to complete some action in the session   
before it can be closed.   
   
If an unhandled exception occurs in the exit processing, the gadget displays an error message and   
the window is always closed.   
SwitchController implementation   
The SwitchController function is called by the Co-Browsing gadget when the agent chooses a   
different controller or presenter from the Controller drop down menu. The function prototype is:   
function (participantId, successHandler, errorHandler)   
   
The participantId is the numeric identifier of the participant to switch control to. This is either a   
value from ControllerType or an integer from 0 upwards if there are other participants.   
GoToURL implementation   
The GoToURL function is called by the Co-Browsing gadget when the agent types text into the URL   
entry field and either presses the Enter key or clicks the arrow button. The function prototype is:   
function (url, successHandler, errorHandler)   
   
The url parameter does not necessarily have to be a formatted URL string, and the gadget will pass   
on any non-zero length text to this function. It is the responsibility of the third-party code to   
validate the input according to what the co-browsing solution requires.   
Implementing the gadget control functions   
Synchronous, simple implementation   
The gadget control functions are used to hide/show the various display elements of the gadget and   
must therefore be implemented synchronously by the third-party code. Whatever critieria are used   
to determine the return value for a function, it should involve minimal processing. It should not use   
callbacks, calls to server–side code, or anything else that involves asynchronous processing   
because control functions are called directly by the UI layer.

CCSP UI Integrations Guide - 7.3 and 7.4   
188   
Raising events   
The Co-Browsing gadget should be notified of changes to the session caused by the third-party   
solution that affect the gadget controls or display. Such changes are:   
•   
Change in the session location   
•   
Change to the controlling participant   
•   
Participant has joined the session   
•   
Participant has left the session   
•   
Termination of the session   
Change in the session location   
Raise the event OnURLChanged. The event data object should contain the property:   
•   
newURL: a properly formatted URL string.   
   
This event should not be raised as part of the processing for the GoToURL function.   
Participant has joined the session   
Raise the event OnParticipantAdded. The event data object must contain the following properties:   
•   
name: name of the participant to be displayed in the SwitchController drop down menu   
•   
id: numeric identifier of the participant starting from 0   
•   
canControl: boolean that indicates if the participant is able to control or present the session   
   
This event must not be raised when the caller joins the session as the caller is already a participant   
upon session creation.   
Participant has left the session   
Raise the event OnParticipantRemoved. The event data object must contain the property:   
•   
participantId: numeric identifier of the participant corresponding to either a ControllerType   
value or identifier of another participant starting from 0.   
   
This event should not be raised when the caller leaves the session. The OnSessionExited event   
should be used instead.   
Change in session controller   
Raise the event OnControllerChanged. The event data object muct contain the property:   
•   
participantId: numeric identifier of the participant corresponding to either a ControllerType   
value or the identifier of another participant starting from 0.   
   
This event should not be raised as part of the processing for the SwitchController function.

CCSP UI Integrations Guide - 7.3 and 7.4   
189   
Session terminated   
Raise the event OnSessionExited. This event has no parameters. The Co-Browsing gadget exits   
immediately upon receiving this event without calling the ExitSession function and displays an alert   
to the agent indicating the reason for the session termination.   
This event should not be raised as part of the processing for the ExitSession function.   
Co-Browsing gadget properties   
The Co-Browsing gadget defines the usual properties for a standard gadget, which are listed in   
windowObject.Gadget properties. However, not all of these properties are accessible using CCSP   
AdminPoint. For information about which properties can be edited in CCSP AdminPoint, see CCSP   
AdminPoint Help.   
In addition to the standard properties, the Co-Browsing gadget defines further properties to   
configure the gadget behavior, which are described in Table 8: Co-Browsing gadget properties.   
CCSP AdminPoint   
name   
Gadget property name   
Usage   
API Request   
Timeout   
cobAPIRequestTimeout   
The maximum length of time in milliseconds   
that the gadget should wait for a response to   
an API call on the implementation before   
returning an error to Agent.   
Dock Chat in   
Session   
cobDockChat   
When set, the chat window is automatically   
docked in the co-browsing gadget when the   
gadget is opened.   
One Session per   
Call   
cobOneSessionPerCall   
When set, each chat call has its own co-  
browsing session, regardless of the number   
of concurrent chat calls. If it is not, the   
number of co-browsing sessions is limited by   
the Allowed Instances property.   
API   
Implementation   
URL   
cobThirdPartySourceCode  
URL   
Relative (to TouchPoint folder) or absolute   
location of the Javascript file containing the   
implementation to be used.   
   
Table 8: Co-Browsing gadget properties

CCSP UI Integrations Guide - 7.3 and 7.4   
190   
Accessing the gadget properties   
The parent CoBrowsingAPI object contains the property localWindow. This is the current window   
object that the Co-Browsing gadget, and therefore the third party code, is operating in. The   
localWindow property can therefore be used as the window object and the third party code can use   
this to access all the properties of the gadget as described in windowObject.Gadget properties. This   
means that the system or tenant administrator can add custom properties to the Co-Browsing   
gadget in CCSP AdminPoint and these can then be accessed in the third-party code. An example of   
this is the addition of an API Key property. Co-browsing solutions that provide Javascript APIs often   
require a numeric identifier that is obtained through a subscription to initiate a co-browsing   
session. This can be held in the gadget’s property instead of hard-coding it into the third-party   
code.   
Note   
Changing the value of a gadget property from the third-party code only takes effect the next   
time the gadget is opened.   
Examples   
The CCSP UI installation includes a co-browsing solution based on the open source TogetherJS   
project developed by Mozilla. This is described in https://togetherjs.com/. In addition, the   
app/gadgets/Examples/CoBrowsing folder in the CCSP UI installation contains test implementations   
and working examples using other co-browsing solutions.   
Template   
This is the most basic example that implements only the CreateSession and ExitSession functions   
and displays an HTML page as the co-browsing “session”. The associated files are:   
•   
templateImplementation.js   
•   
templateCoBrowsingPage.html   
   
All gadget control functions are implemented but return false. This template provides a good basis   
to develop your own implementation.   
Test   
This is used for testing the various aspects of the CoBrowsingAPI and returns true for most of the   
session control functions. It also uses a session testing HTML page that allows for the automated   
injection of CoBrowsingAPI events and testing of error handling. The associated files are:   
•   
testThirdPartyCode.js   
•   
testCoBrowsingpage.html

CCSP UI Integrations Guide - 7.3 and 7.4   
191   
Surfly   
This uses the API provided by the Surfly® co-browsing solution (www.surfly.com). It can create a   
session or join one provided that the creationData’s sessionID parameter contains a valid   
surfly.com URL. The Surfly solution is proxy server-based and it provides the ability to co-browse   
any website without having specific code embedded in it.   
The implementation enables the URL entry field, which allows you to navigate the session to   
different sites. It also uses the OnURLChanged event to update this field. The SwitchController   
function and OnControllerChanged event are also fully functional.   
The associated file is:   
•   
surflyTestImplementation.js   
   
You need to set up a Surfly account, which can be obtained on a trial basis by registering on their   
website. On registration, you obtain an API key that must be used in the place indicated in the file   
to create and join Surfly sessions. In addition, CCSP UI must run in a domain that is visible on the   
internet for Surfly to accept its connection to the session. For more details, see www.surfly.com.   
When the CreateSession function is called for a new session, the implementation starts a Surfly   
session as leader and returns a follower link to this session as the customerSessionUrl through the   
successHandler.   
Temasys Skylink   
Temasys Skylink uses WebRTC to exchange audio, video, and data between client browsers without   
using embedded code in a website. The example implementation uses the screen sharing capability   
that can be added to browsers such as Chrome and Firefox via plugins and uses Temasys Skylink to   
set up the WebRTC connection between the agent and the caller.   
The associated files are:   
•   
skylinkImplementation.js   
•   
skylinkscreensharing.html   
   
Note   
The implementation file expects the html file to be hosted in an application called /skylinkPOC on   
the same server. If it is hosted anywhere else, change the reference in the implementation file as   
appropriate.   
   
You need to register at https://skylink.io/ for an App key that must then be used in the places   
indicated in the two files.   
The implementation can only be used to start a session, not to join one. When the CreateSession   
function is called for a new session, the implementation starts a Skylink “room” and returns a link   
to the HTML page with the room ID as its query, in the customerSessionUrl. When the caller uses   
this link to connect to the room and enables screen sharing, the stream of their screen or window is   
displyed in the video element of the Co-Browsing gadget.   
This implementation does not display the URL or Controller controls in the gadget.

CCSP UI Integrations Guide - 7.3 and 7.4   
192   
CCSP UI must be run using HTTPS and the html file must also be hosted on HTTPS. On the caller   
side, the Chrome browser requires a screen sharing extension to be added which can be obtained   
from the Chrome Web Store.   
Points to note   
•   
The Co-Browsing gadget loads the implementation file each time it is activated, so it is not   
necessary to log out and back in again to pick up changes in the file In Chrome and Firefox, it   
may be necessary to have Dev tools open and caching disabled.   
•   
The Co-Browsing gadget is only activated in response to clicking the button on the chat call   
card or window. It cannot be opened from the gadgetory panel.   
•   
When modifying the example implementations, be sure to save them under new names to   
avoid them being overwritten in the event of an upgrade to the CCSP UI component.

CCSP UI Integrations Guide - 7.3 and 7.4   
193   
8: Screen Recording gadget   
implementation override   
Overview   
This section describes the framework in CCSP UI that enables you override the Screen Recording   
(SR) gadget behavior implementation. It describes the API provided and how to override the   
implementation of the default screen recording behavior.   
Note   
The SR gadget and its implementation, and the multi recording feature are supported only with   
the ffmpeg encoder. EE4 uses the legacy behavior of one call recording only and the SR gadget   
does not appear on the gadgetory.   
   
The SR gadget enables agents to control for which calls their screen is recorded. They can start   
screen recording by clicking the SR gadget while the call is ringing, however, they cannot start   
screen recording when there are no calls.   
The SR gadget:   
1.   
Supports parallel screen recording of several calls.   
2.   
Decouples screen and media recordings. Screen recording can be initiated from the SR gadget,   
which also shows if screen recording is active. Media recording can be initiated from the call   
card.   
3.   
Has a default implementation for multi recording. This this behavior can be changed by   
overriding the screen recording gadget implementation.   
Screen Recording gadget architecture   
ScreenRecrodingController.js   
This component manages the screen recording logic and calls the screenRecordingService.js, which   
calls CCSPScreenRecordingService (the actual screen recording service) to capture the screen. This   
functioanlility was previously in CCSP UI.   
ScreenRecordingRules.js   
This component is the interface for the screen recording implementation. This interface expects the   
implementation to return a list of calls with the required recording status (start\stop).

CCSP UI Integrations Guide - 7.3 and 7.4   
194   
ScreenRecordingRulesImplementation.js   
This component is the implementation for the ScreenRecordingRules.js interface, based on the   
implementor's screen recording rules.   
screenRecordingService.js   
This component is the javascript service that makes the actual requests to the   
CCSPScreenRecordingService (windows service on agent machine)   
CCSPScreenRecordingService   
This component is the .Net application that captures the desktop user's screen.   
   
Figure 6: Screen Recording gadget architecture

CCSP UI Integrations Guide - 7.3 and 7.4   
195   
Screen Recording gadget interface   
The ScreenRecordingRules.js interface expects the implementation to return a list of calls with the   
required recording status (start\stop).   
Functions   
These functions potentially get a call for the screen recording status change request and return a   
list of calls and the new recording status.   
GetStatusesAfterNewCall (call)   
This event occurs after a new call is arrived. This function implementation should go over the list of   
current calls and return the required recording statuses.   
GetStatusesAfterResumeCall (call)   
This event occurs after a call is returned to in call state after call was put on hold. This function   
implementation should go over the list of current calls and return the required recording statuses.   
GetStatusesAfterDestructedCall (call)   
This event occurs after a call was ended. This function implementation should go over the list of   
current calls and return the required recording statuses.   
GetStatusesAgentRequestedStopCall (call)   
This event occurs after after agent requeseted to stop that specific call (by clicking on recording   
button in call card). This function implementation should go over the list of current calls and return   
the required recording statuses.   
GetStatusesAgentRequestedStartCall (call)   
This event occurs after after agent requeseted to start that specific call (by clicking on recording   
button in call card). This function implementation should go over the list of current calls and return   
the required recording statuses.   
GetStatusesAgentRequestedStopCalls   
This event occurs after after agent requeseted to stop calls (by clicking screen recording gadget).   
This function implementation should go over the list of current calls and return the required   
recording statuses.

CCSP UI Integrations Guide - 7.3 and 7.4   
196   
GetStatusesAgentRequestedStartCalls   
This event occurs after after agent requested to start calls (by clicking screen recording gadget).   
This function implementation should go over the list of current calls and return the required   
recording statuses.   
OnScreenRecordingStatus   
Callback for the event ScreenRecordingStatus published by CCSPScreenRecordingService (process   
that captures the screen). This callback is invoked by the original event in the   
ScreenRecordingController.js, which updates the screenRecordingStatus for each call (Started or   
Stopped) that was received by the CCSPScreenRecordingService. This callback can add a specific   
implementation for this time frame of updating the screen recording statuses that were changed   
for each call.   
For example, in the default implementation we want to check if all calls were stopped by the agent   
and to set a flag because the rules are that if the agent has stopped the screen recording, then new   
calls with script ON are not affected.   
Constants   
The following interface constants are used to set a desired recording status in the result of each   
implementation and determine the request owner.   
RequiredRecordingStatus   
This constant defines a desire of new screen recording status: either Start or Stop.   
ScreenRecordingRules.RequiredRecordingStatus =   
 {   
 Start: "Start",   
 Stop: "Stop",   
 };   
RequestOwner   
This constant defines the request owner for the recording status change: either Agent or System   
(script)   
ScreenRecordingRules.RequestOwner =   
 {   
 Agent: "Agent",   
 System: "System",   
 };

CCSP UI Integrations Guide - 7.3 and 7.4   
197   
Example   
The following function is run on the active calls and updates their required recording statuses   
before sending the request to the CCSPScreenRecordingService.   
ScreenRecordingRulesImplementation.prototype.OnScreenRecordingStatus =   
function (status) {   
 var self = this;   
 var recordingStatuses = [];   
 for (callId in self.callList) {   
 var c = self.callList[callId];   
 c.RequiredRecordingStatus = status;   
 recordingStatuses.push(c);   
 self.loggerService.WriteLine("Call id: " + c.CallId + ", Screen recording   
status request: " + status, window.enghouse.enums.Severity.Debug, "Screen   
Recording Implementation");   
 }   
 return recordingStatuses;   
}   
Screen recording sequence   
The screen recording sequence is as follows:   
1.   
CCSP UI receives an event that screen recording should start\stop or that an agent requested   
to start\stop screen recording.   
2.   
CCSP UI requests the SR gadget (ScreenRecordingController.js) to start\stop screen recording.   
3.   
ScreenRecordingController.js receives these events and handles the screen recording logic for   
both the FFmpeg encoder and the EE4 encoder. Note that EE4 has the legacy behavior of   
recording one call at a time. Also, the SR gadget does not appear when EE4 is set as the   
encoder.   
4.   
ScreenRecordingController.js requests from ScreenRecordingRulesImplementations.js a list of   
calls with the required recording status (Start or Stop). Depending on the rules, this list can be   
a combination of several calls to Start and Stop.   
5.   
ScreenRecordingController.js sends the list with the required statuses to   
screenRecordingService.js.   
6.   
screenRecordingService.js calls the CCSPScreenRecordingService API for start\stop recording   
with the lists of the calls and their required statuses.   
7.   
CCSPScreenRecordingService starts\stops screen recording for the specified calls and sends   
back the ScreenRecordingStatus event with a list of calls and their new recording status,   
Eventually this is an Ack\Nack for the recording start\stop of the calls for the request.

CCSP UI Integrations Guide - 7.3 and 7.4   
198   
   
Figure 7: Screen recording sequence   
Screen recording default implementation   
In the default behavior:   
•   
Media recording belongs to the call.   
o   
When transferring or getting a call:   
▪   
If there is a recording script, it should be considered.   
▪   
If there is no script, the last recording status of the call should be taken into account.   
•   
Screen recording belongs to the agent.   
o   
In multi-recording situations, by default the screen should be most secure.   
o   
When there is one call, the call should consider the recording script.   
o   
When there are multiple calls, the following should be considered:   
▪   
If the first call has script OFF and the second call has script ON, the screen recording   
should not start.   
▪   
If the first call has script ON and the second call has screen OFF, the screen recording   
should stop for all calls when this call starts.   
▪   
Screens will be recorded when:   
- All scripts of calls have screen ON.   
- Agents start the recording by themselves. However, script OFF will stop the   
 recording.   
o   
If agents stop the recording by themselves and there is screen ON, the script will not start   
the recording.

CCSP UI Integrations Guide - 7.3 and 7.4   
199   
The following table shows these rules.   
Action\SR script ON or   
OFF   
Call 1   
Call 2   
Call N   
Result   
Agent control   
Agent clicks on recording   
button in call card   
on\off   
on\off   
on\off   
SR starts\stops for all calls   
regardless of the script   
configuration   
Agent clicks on screen   
recording gadget   
on\off   
on\off   
on\off   
SR starts\stops for all calls   
regardless of the script   
configuration   
Incoming call   
Incoming call answered   
when agent does not   
have control   
off   
on   
on   
SR stops for all (at least one script of   
a call has off, SR is stopped for all   
calls)   
on   
on   
on   
SR starts for all (all call scripts must   
have SR script on to start screen   
recording)   
Incoming call answered   
after agent stopped   
manually   
on\off   
on\off   
on\off   
SR does not continue until agent   
requests to start recording again or   
there are no more calls   
Incoming call answered   
after agent started   
manually   
on\off   
on\off   
on\off   
Incoming call + call state change   
rules applied   
Call state change   
Call 1 ended   
off   
on   
on   
SR stops for call 1 + SR starts for   
others that have SR script on (since   
call 1 is destructed, its off   
configuration does not apply)   
Call 2 ended   
off   
on   
on   
SR is already stopped for all due to   
call 1 off so SR stays stopped   
Call 1 moved to on hold   
status   
on\off   
on\off   
on\off   
SR continue as usual (recording or   
not)   
Call 1 returns to active   
from on hold   
on   
on   
on   
SR stops call 1 and immediately   
starts call 1 to keep sync with audio

CCSP UI Integrations Guide - 7.3 and 7.4   
200   
Action\SR script ON or   
OFF   
Call 1   
Call 2   
Call N   
Result   
Call 1 returns to active   
from onhold   
off   
on   
on   
SR is not affected since it was not   
recording due to incoming call rules   
Outgoing call   
Outgoing call   
on\off   
on\off   
on\off   
Similar to incoming call rules only if   
there is outgoing call script   
No recording script   
Incoming\outgoing call   
answered without script   
-   
-   
-   
Acts as script off   
SR = Screen recording   
Script is configured via CCSP Designer application.   
   
Table 9: Screen recording rules   
Overiding the screen recording implementation   
If the encoder is set to ffmpeg, you can override the default behavior by writing your own rules and   
loading them via AdminPoint. The implementation can be different for each Tenant, Group, or role.   
In AdminPoint, you enter the URL of the script that implements the screen recording rules in the   
Screen Recording gadget properties, in Screen Recording Rules Implementation URL. The default   
is app/gadgets/ScreenRecording/ScreenRecordingRulesImplementation.js.   
Use the default script ScreenRecordingRulesImplementation.js as an example of how to write your   
own rules.   
The new script should implement the interface (by inheriting that class) represented in   
ScreenRecordingRules.js, described in Screen Recording gadget interface.   
The ScreenRecordingRules.js has a reference to the active calls list that CCSP UI holds and interacts   
with:   
ScreenRecordingRules = function (callList) {   
 var self = this;   
 self.callList = callList;   
}   
   
The implementation should return a new list of calls that are required to change their recording   
status, based on your rules, with the property of RequiredRecordingStatus.

CCSP UI Integrations Guide - 7.3 and 7.4   
201   
Example   
The following implementation example get a new incoming call and sets its recording status to   
Start if it has ScreenRecording script set to ON in its CCSP Designer application.   
ScreenRecordingRulesImplementation.prototype.GetStatusesAfterNewCall =   
function (call) {   
 var self = this;   
 var recordingStatuses = [];   
 if (call.CallInfo.Recording && call.CallInfo.Recording.ScreenRecording)   
 {   
 Call. RequiredRecordingStatus =   
ScreenRecordingRules.RequiredRecordingStatus.Start;   
 recordingStatuses.push(call);   
 }   
 return recordingStatuses   
}   
   
After the implementation returns a list with one call with the required status of Start, the   
ScreenRecordingController gets this event and calls to start screen recording for that call.   
Points to note   
•   
The Screen Recording gadget loads the implementation file on login. When changing the   
implementation script, agents must log out and log in again.   
•   
The implementation is flexible. Even if only one call is received, you can decide to stop all other   
screen recording. For example, in the default implementation, if threee calls are answered   
with the script set to ON, and the fourth call has the script set to OFF, screen recording is   
stopped for all calls.

CCSP UI Integrations Guide - 7.3 and 7.4   
202   
9: Single Sign On Client development   
Overview   
This section describes how to add Single Sign On (SSO) capability to a custom gadget. It describes   
the main components of the SSO interface and provides a worked example of the code that is   
required for a gadget to support SSO.   
Relevance   
This section applies to CCSP releases 7.4 and above only. The HF74-61362 hotfix is a prerequisite for   
using CCSP UI SSO.   
Scope   
This document only discusses CCSP SSO in terms of developing a functional or standard gadget   
that is hosted inside CCSP UI. For full details on the concepts and workings of CCSP SSO, refer to   
the CCSP Single Sign On Guide.   
General SSO concepts   
This section provides a summary of those main concepts behind the CCSP implementation of SSO   
that are relevant to the development of CCSP UI gadgets.   
Single Sign On (SSO)   
This is the means by which a user only has to enter their credentials once in order to access   
multiple applications.   
Auth Server   
Application that provides authentication and authorization services to Resource Clients seeking to   
access a Resource Server, using the OAuth or OpenID Connect protocol.   
Resource Client   
Application that uses services provided by Resource Servers and requires authorization to use   
those services.

CCSP UI Integrations Guide - 7.3 and 7.4   
203   
Resource Server   
Application that provides services that can only be accessed by authorized users and client   
applications.   
Access token   
Token that authorizes a user and Resource Client to use services provided by a Resource Server.   
Refresh token   
Used in lieu of a user’s username and password credentials to get a new Access token from an   
Auth Server.   
OpenID Connect (OIDC)   
This is a widely used protocol for authentication and authorization of users via tokens exchanged   
between Resource Clients, Resource Servers, and Auth Servers. The protocol defines various flows   
to describe how tokens are obtained and exchanged between the different components.   
Authorization code grant   
OpenID Connect flow where Resource Client gets an authorization code from Auth Server that it   
then exchanges for the required tokens to access the Resource Server.   
Refresh token grant   
OpenID Connect flow where Resource Client sends a previously received refresh token to the Auth   
Server to acquire new tokens in return.

CCSP UI Integrations Guide - 7.3 and 7.4   
204   
CCSP UI SSO Concepts   
This section defines concepts specific to the architecture of the CCSP UI SSO implementation.   
Basic architecture   
The concepts previously presented are shown in the following representation of OIDC   
Authorization code flow in CCSP UI.   
   
Figure 8: Basic OIDC Authorization Code flow in CCSP SSO   
1.   
The user navigates the Browser to the URL of the Resource Client.   
2.   
The Resource Client sends a “Login” GET request to the Auth Server. This request must   
contain certain data for it to be successful, including the Resource Client’s redirect URL.   
3.   
If the Resource Client data is valid and the user is authenticated, the Auth Server generates   
an authorization code and invokes the Resource Client’s redirect URL with this code in the   
query.   
4.   
The Browser navigates to the redirect URL of the Resource Client.   
5.   
The Resource Client detects the presence of the authorization code in its URL and sends a “Get   
Tokens” POST request to the Auth Server.   
6.   
If the Authorization code is valid, the Auth Server determines if the user is authorized to access   
the requested Resource Server. If the user is authorized, an access token is returned to the   
Resource Client. If specified in the request, a refresh token is also returned.   
7.   
Both the user and Resource Client are now validated, so Resource Client can use the access   
token in all its API calls to the Resource Server.

CCSP UI Integrations Guide - 7.3 and 7.4   
205   
Detailed architecture   
For the purposes of CCSP UI gadget SSO development, the following objects map to the basic   
components of the OIDC Authorization code flow:   
•   
Auth Server => CCSP Auth Server   
•   
Resource Client => CCSP UI gadget   
•   
Resource Server => Web application that the CCSP UI gadget needs to access   
   
To facilitate the development of CCSP SSO, two additional objects are used:   
•   
oidcClient: An internal pure Javascript class that handles all aspect of the OIDC authorization   
code and refresh flows.   
•   
oidcService: A thin wrapper for the oidcClient that is Javascript framework-specific. CCSP UI   
uses the angularJS version, but there is also a VueJS version. For full details of the methods   
exposed by this service, see oidcService in 4: CCSP UI AngularJS services.   
   
The usage of these objects in the flow is seen in Figure 9, which expands on the basic flow shown in   
Figure 8.   
   
Figure 9: Detailed OIDC flow in CCSP UI SSO   
The flow details are:   
1.   
The user navigates the Browser to the URL of the Resource Client.   
a.   
The gadget creates an instance of the oidcClient by calling the CCSP UI’s   
oidcService.addClient() function. The paramers include a client name to uniquely identify   
the gadget within the CCSP UI.

CCSP UI Integrations Guide - 7.3 and 7.4   
206   
2.   
The Resource Client sends a “Login” GET request to the Auth Server. This request must   
contain certain data for it to be successful, including the Resource Client’s redirect URL.   
a.   
oidcService.login() is called for the client name with the redirect URL to be used.   
b.   
oidcClient.login() is called on the oidcClient object for this client name.   
c.   
The /authorize method is called on the CCSP Auth Server.   
3.   
If the Resource Client data is valid and the user is authenticated, the Auth Server generates   
an authorization code and invokes the Resource Client’s redirect URL with this code in the   
query.   
4.   
The Browser navigates to the redirect URL of the Resource Client.   
a.   
The gadget retrieves the oidcClient previously created.   
5.   
The Resource Client detects the presence of the authorization code in its URL and sends a “Get   
Tokens” POST request to the Auth Server.   
a.   
oidcService.processCodeCallback() is called for the client name.   
b.   
oidcClient.loginCallback is called on the oidcClient object for this client name.   
c.   
The /token POST method is called on the CCSP Auth Server.   
6.   
If the Authorization code is valid, the Auth Server determines if the user is authorized to access   
the requested Resource Server. If the user is authorized, an access token is returned to the   
Resource Client. If specified in the request, a refresh token is also returned.   
a.   
The token data is returned to the oidcClient.   
b.   
oidcClient stores the token data in internal memory.   
c.   
oidcClient returns success to the oidcService.   
d.   
oidService returns success to the gadget.   
7.   
Both the user and Resource Client are now validated, so Resource Client can use the access   
token in all its API calls to the Resource Server.   
a.   
The gadget calls getAccessToken() on oidcService to retrieve the access token.   
b.   
oidcService calls getAccessToken() on the oidcClient object for this client name.   
c.   
The access token is returned back to the gadget, which can then use it for the API call.   
   
Note   
If the user is not authenticated, the Auth Server issues a 302 challenge in response to the   
/authorize request. that normally causes the Browser to attempt to redirect to a login page for   
the user to enter their credentials. This should never happen to a gadget hosted in CCSP UI   
because the architecture requires that a user must already have been authenticated before the   
gadget makes the request. In the event that this does happen, the gadget receives an error in   
response to the login() request.

CCSP UI Integrations Guide - 7.3 and 7.4   
207   
CCSP UI Gadget with SSO   
The recommended structure of a CCSP UI gadget to use SSO is shown below.   
   
Figure 10: CCSP UI gadget with SSO architecture   
This structure:   
•   
Keeps the OIDC functionality in separate files and classes from the main gadget code.   
•   
Uses a simple html page to contain the gadget’s OIDC class and provide the redirect URL.

CCSP UI Integrations Guide - 7.3 and 7.4   
208   
Example gadget with OIDC   
The following sections present extracts from a demo functional gadget that illustrates the above   
components. The code for this example is in in the app/gadgets/Examples/DemoSSO folder in the   
CCSP UI installation and comprises three files:   
•   
demoOIdc.js – Gadget OIDC class   
•   
demoOidc.html – Gadget OIDC HTML page   
•   
demoSSOGadget.js – Main gadget script   
Main gadget script (demoSSOGadget.js)   
The CCSP Auth Server needs a redirect URL to:   
•   
Validate that the Resource client is coming from a whitelisted source.   
•   
Know where to send the authorization code.   
   
The gadget therefore needs to define the page within CCSP UI that will provide this redirect URL   
and process the authorization code. To facilitate this, CCSP UI defines a container div in the DOM   
that is identified by the id “backgroundGadgetContainer”. The gadget can then create a hidden   
iframe that is contained within this element containing the page.   
The following code shows the creation of this element and its addition to the page.   
 // Open the oidc window in a hidden iframe   
 const oidcElement = document.createElement("iframe");   
 oidcElement.src =   
"app/gadgets/Examples/DemoSSO/demoOidc.html?op=login";   
 oidcElement.width = "0px";   
 oidcElement.height = "0px";   
 oidcElement.id = "demoOidc";   
 const backgroundContainer =   
document.getElementById("backgroundGadgetContainer");   
 if (backgroundContainer) {   
 if (backgroundContainer.appendChild(oidcElement)) {   
 logger.log("Added demoOidc window to document");   
 } else {   
 logger.log("Failed to add demoOidc window to document");   
 }   
 }

CCSP UI Integrations Guide - 7.3 and 7.4   
209   
Note   
Always assign an id to the new element.   
We recommend that you always use an iframe within backgroundGadgetContainer for both   
functional and standard gadgets, to maintain separation of the SSO functionality from the main   
gadget code.   
   
Notice that the oidcElement src attribute has a query parameter “op=login”. This is used to allow   
the page to know that it has been invoked by the gadget and not as a result of a redirect.   
For the gadget to get notifications of access token events, use the eventServiceWrapper to   
subscribe to them.   
 eventServiceWrapper.subscribe("DemoAccessTokenError", null,   
onDemoAccessTokenError);   
 eventServiceWrapper.subscribe("DemoAccessTokenReceived", null,   
onDemoAccessTokenReceived);   
Subscribe to the events before adding the iframe.   
Gadget OIDC Page (demoOIdc.html)   
This should be a simple page with minimal functionality. It should just contain the gadget’s OIDC   
javascript function.   
Example code   
<!DOCTYPE html>   
<html>   
<head>   
 <meta charset="utf-8" />   
 <title></title>   
 <script type="text/javascript" src="../../../../Ajax/helpers.js"></script>   
 <script type="text/javascript"   
src="../../../../Scripts/angular.js"></script>   
 <script type="text/javascript"   
src="../../../common/gadgetInit.js"></script>   
 <script type="text/javascript" src="demoOidc.js"></script>   
</head>   
<body onload="startDemoOidc()">   
</body>   
</html>   
   
Note   
Any script references should be relative to the gadget folder and not include the expected CCSP   
UI top-level folder name, as this is subject to change.

CCSP UI Integrations Guide - 7.3 and 7.4   
210   
Gadget OIDC Class (demoOidc.js)   
This handles the interaction with oidcService. All the functionality should be contained with a   
function as is generally recommended for all gadget scripts.   
startDemoOidc = function () {   
...   
}   
Setting up the variables   
Initialize a local angular injector as usual then get the required services.   
 var eventServiceWrapper = localInjector.get("eventServiceWrapper");   
 var oidcService = localInjector.get('oidcService');   
 var loggerService = localInjector.get('loggerService');   
   
Define the values required for the oidcService addClient parameters.   
 const CLIENT\_ID = "my-demo-sso-gadget";   
 const CLIENT\_NAME = "DemoSSO";   
 const redirectUri =   
`${root.enghouse.globals.appBasePath}app/gadgets/Examples/DemoSSO/${FUNC\_NAME}  
.html`   
   
The client\_ID value must match the client ID that is assigned to your gadget in the Configuraton   
Database. This corresponds to the Audience that the Resource Server will accept access tokens   
from. It can be any string, including a GUID, but it is recommended that special characters like &   
and ^ are avoided. Underscores and dashes are fine.   
Note how the base path for the CCSP UI application can be retrieved using the   
root.enghouse.globals object.   
Add an oidcClient for the gadget to the oidcService.   
 if (!oidcService.hasClient(CLIENT\_NAME)) {   
 oidcService.addClient({   
 client: CLIENT\_NAME,   
 clientId: CLIENT\_ID,   
 redirectUri: redirectUri,   
 refreshSlidingTime: 40 \* 1000,   
 refreshTokenCallback: function (data) {   
 if (data?.accessToken) {   
 eventServiceWrapper.publish("DemoAccessTokenReceived",   
null, data.accessToken };   
 } else {   
 eventServiceWrapper.publish("DemoAccessTokenError");

CCSP UI Integrations Guide - 7.3 and 7.4   
211   
 }   
 },   
 refreshTokenRequired: true,   
 responseModePostRequired: true   
 });   
   
This code first checks if the client has already been added using the hasClient() method and if not,   
adds a new one. For this gadget, access token renewal using the refresh token is enabled so a   
callback function is added. In this example, the callback function publishes the …Received event, or   
the error event depending upon the success of the request. It could equally publish a single event   
with a flag indicating success or failure.   
The refresh sliding time of 40 seconds means that 40 seconds before the access token expires, a   
new token request will be made. Note that each successful token request will replace both the   
access and refresh tokens for the client.   
Making the Login request   
It is important to remember that the page hosting this function and therefore the function itself will   
be called twice during the authentication process:   
•   
First time to perform the login of the client   
•   
Second time with the authorization code from the CCSP Auth Server   
   
The function must therefore know why it has been invoked. The easiest way to do this is using the   
URL query parameters. Define a query parameter and value that can identify a login invocation of   
the page and check for this in the function. For this example, the query parameter “op” is used,   
where a value of “login” indicates that the function should perform a login call on the oidcService.   
 const searchParams = new URLSearchParams(window.location.search);   
 const operation = searchParams.get("op")   
 if (operation === "login") {   
 (async () => {   
 await oidcService.login(CLIENT\_NAME, null, null, window.location);   
 })();   
 }   
   
Notice how the async call to the login() method is enclosed in a function to avoid the entire   
startupDemoOidc() method being declared async and then having to be awaited by the hosting   
page.   
In practice, the await call should be surrounded by a try-catch so that any errors (for example a   
redirect) are handled appropriately.   
If the authentication succeeds, the page and function are invoked again.

CCSP UI Integrations Guide - 7.3 and 7.4   
212   
Getting the tokens   
When the page and the function are invoked for the second time with the authorization code, use   
the oidcService.isRedirectCallbackCode() method to verify that this is the case. For this example, the   
“op” query parameter is no longer defined so it drops to the else branch below where this check   
can be performed.   
(async () => {   
 const isRedirectCallbackCode = await   
oidcService.isRedirectCallbackCode(CLIENT\_NAME, window.location);   
 if (   
 try {   
 const result = await oidcService.processCodeCallback(CLIENT\_NAME,   
null, window.location);   
 if (result?.tokenReceived) {   
eventServiceWrapper.publish("DemoAccessTokenReceived", null,   
oidcService.getAccessToken(CLIENT\_NAME));   
 }   
 else {   
 eventServiceWrapper.publish("DemoAccessTokenError");   
 }   
 }   
 catch (error) {   
 eventServiceWrapper.publish("DemoAccessTokenError");   
 }   
 }   
})();   
   
In the above code, the processCodeCallback() method is awaited for the result of the POST request   
to the CCSP Auth Server API for a token. If the result indicates that an access token has been   
received the token is published in the success event, otherwise an error event is published. The use   
of an event means that the main gadget code can get the access token from this event, without   
knowing anything about oidcService, token refresh etc.

CCSP UI Integrations Guide - 7.3 and 7.4   
213   
Removing the client   
To tidy up the oidcService when the gadget is no longer used, call the removeClient() method of the   
oidcService. This should be done when:   
•   
The gadget is closed if it is a standard gadget.   
•   
The agent logs out of CCSP UI.   
   
To detect agent logout, subscribe to the AgentStateChange event and call removeClient() when the   
state is “LoggedOut”.   
 eventServiceWrapper.subscribe("AgentStateChange", null, function (state) {   
 if (state === "LoggedOut") {   
 try {   
 oidcService.removeClient(CLIENT\_NAME);   
 }   
 catch (ex) {   
 // Don't care about any exceptions here   
 }   
 finally {   
 window.close();   
 }   
 }   
 })