CSTA Connector for BroadSoft BroadWorks Deployment Guide

T-Servers 8.0.0

1/15/2022

Table of Contents

CSTA Connector for BroadSoft BroadWorks Deployment Guide

Genesys CSTA Connector for Broadsoft BroadWorks provides a standard CSTA (computer-supported telecommunications applications) Phase III ASN.1 interface for CTI applications such as Genesys T-Server for CSTA Connector. This documentation set should give you most of the information you need in order to install CSTA Connector for BroadSoft Broadworks. See the summary of the highlighted topics below:

About CSTA Connector

Find out about the CSTA Connector:

Overview

New in This Release

Feature Configuration

Find out about the supported features:

Hot Desking

Redundancy Support

Hiding Data in Logs

all topics>>

Configuration Options

Find out about the configuration options:

Connector Section

link-%s Section

License Section

all topics>>

Deploying CSTA Connector

Find procedures to configure and install the CSTA Connector that include these topics:

High Availability Deployment

Overview

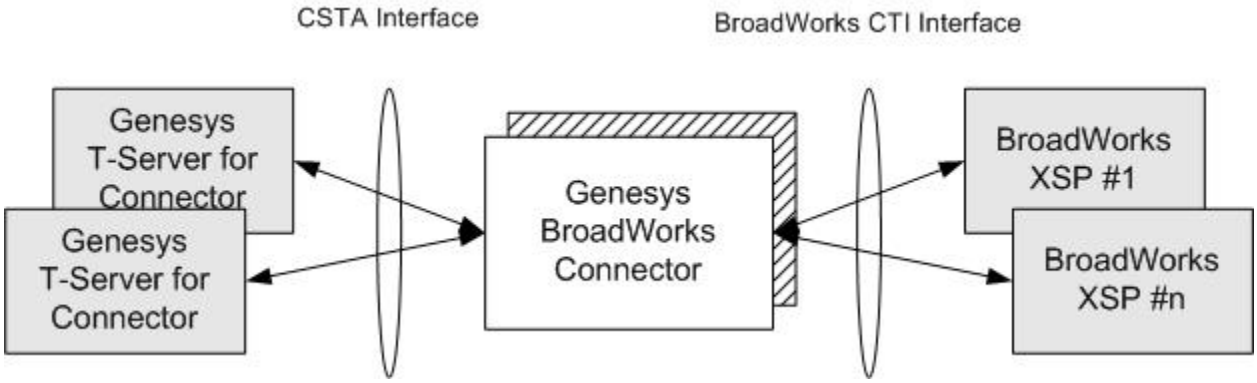
Welcome to the *CSTA Connector for BroadSoft BroadWorks Deployment Guide*. These topics introduce you to the concepts, terminology, and procedures that are relevant to CSTA Connector for BroadSoft BroadWorks deployment.

About CSTA Connector

Genesys CSTA Connector for BroadSoft BroadWorks is an application designed to provide a CSTA Phase III interface used by T-Server for CSTA Connector. Connector is installed between the BroadWorks XSI Interface and T-Server to allow the translation of CSTA requests and events into BroadWorks XSI requests and events.

Genesys CSTA Connector for BroadSoft BroadWorks is also responsible for authentication, session management, and data type translation between the BroadWorks XSI interface and the CSTA Phase III Interface.

An example of the solution architecture is shown in the following figure:



Genesys CSTA Connector for BroadSoft BroadWorks

New in This Release

This is the initial release of the CSTA Connector for BroadSoft BroadWorks.

Feature Configuration

Supported Features

The following is a summary of the functionality supported by CSTA Connector for BroadSoft BroadWorks:

Call Distribution Model

The types of call distribution are as follows:

- Automatic (ACD Queue): ACD and the Agent Model
- Host Controlled (Routing Point): CSTA Routing Profile

ACD and the Agent Model

Agent assignment to a BroadWorks Contact Center device is provisioned in BroadWorks configuration and can not be controlled through computer telephony integration (CTI); That means that which ACD Queue the agent takes calls from cannot be controlled through CTI. The call is then automatically distributed to agents in the `Available` state. The agent state can be changed using CTI requests. No state transition restrictions are applied by BroadWorks Connector, this is, a transition from any agent state is permissible to any state at any time.

Note: BroadWorks CTI has a limitation regarding ACD Queue assignments and does not provide a request to enable agents to modify their queue assignment. Agent assignment must be accomplished through BroadWorks configuration.

Agent Model

BroadWorks implements a simple agent model where all agent state changes are permitted. The following table provides a list of Broadsoft agent states along with BroadWorks Connector's mapping into the CSTA agent state model.

Broadsoft Agent States Mapping

BroadWorks Agent State	CSTA Agent State	Comments
`Agent Sign-in`	`agentNotReady`	This state is a temporary state while logging in.
`Agent Available`	`agentReady`	
`Agent Unavailable`	`agentNotReady`	
`Agent Wrapping-up`	`agentWorkingAfterCall`	
`Agent Signed-out`	`agentNull`	

Agent State Synchronization

BroadWorks CTI issues an `ACDAgentJoinUpdate` event whenever an agent joins or leaves a queue as well as initial synchronization events when the BroadWorks Connector subscribes for these events. The BroadWorks Connector uses this information in these events to allow CSTA client's agent objects to remain synchronized with the switch status.

BroadWorks Connector sends the following CSTA Agent State event to its client whenever it receives an `ACDAgentJoinUpdate` event from the BroadWorks XSP server.

CSTA AgentState Event Sent to Clients

CSTA	BroadWorks CTI
`AgentState` event:	`ACDAgentJoinUpdate` event:
• `device`: acd subscriber id	• `subscriberId`: agent subscriber id

• agent ID: subscriber id	• `ACDAgentJoinInfo`:
	  • `ACDUserId`: acd subscriber id
	  • `agentACDState`: `agent_state`

• logged on state: `true`, if signed in	
• acd group: acd subscriber id	
• agent state: `agent_state`	

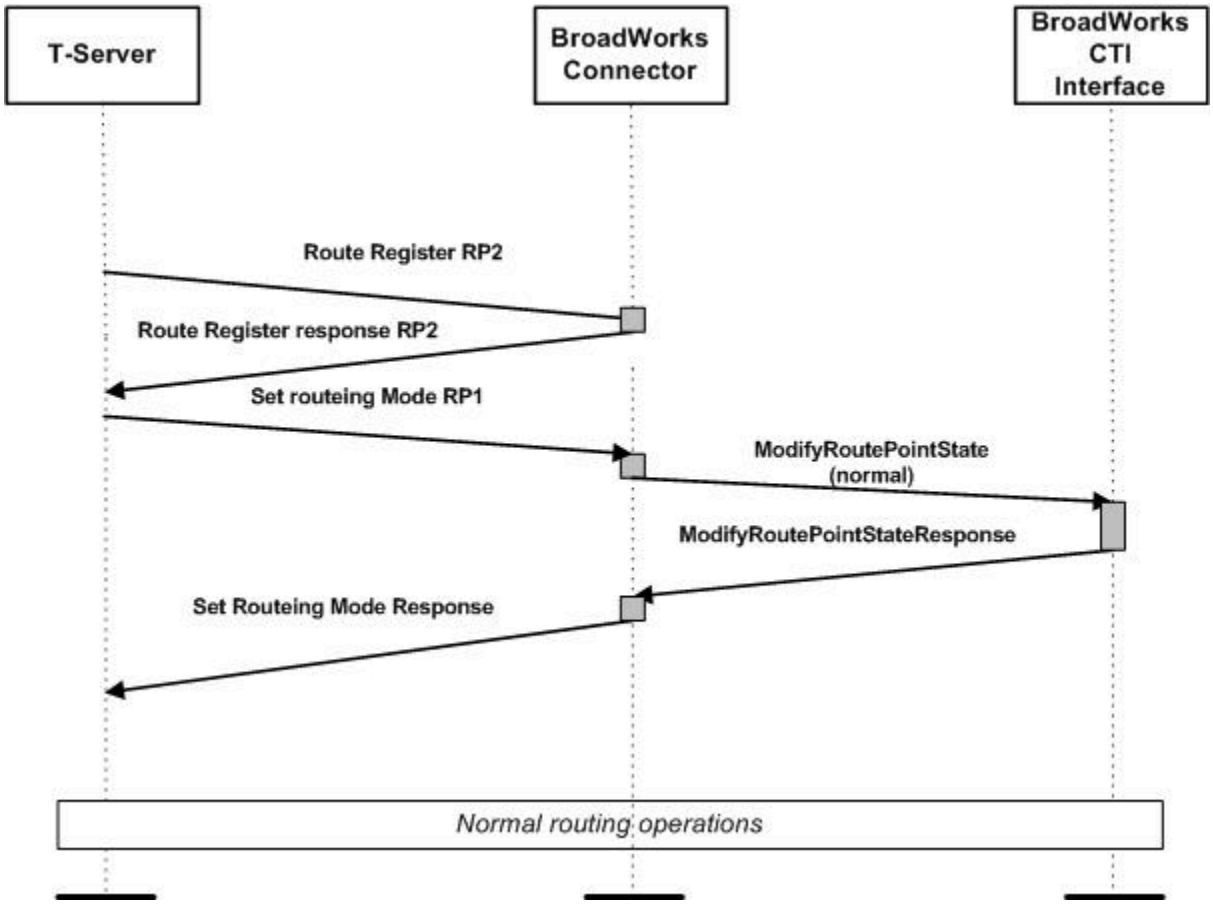
CSTA Routing Profile

BroadWorks Connector supports the CSTA Routing Profile. A BroadWorks Routing Point subscriber is used as a routing device.

The CSTA Routing profile consists of the following topics:

- Treatments
- CSTA PlayEvent Message
- CSTA StopEvent Message
- BroadWorks Event Cause Translation
- Configuring PlayAnnouncement Treatments

Note: Only routing devices support `TMakePredictiveCall` requests.



Routing Registration.

Treatments

The CSTA Play Message request is mapped into a BroadWorks Treatment request as described in the following table:

Mapping of the BroadWorks Treatment Request to the CSTA PlayMessage Request

BroadWorks Request	Parameters	CSTA PlayMessage Parameter
RoutePointPlayBusyRequest		Message ID = 0
	duration	Duration
	busyPattern	Extension busyPattern
RoutePointPlayMOHRequest		Message ID = 1
	duration	Duration
	digitMap	Extension collectDigits
RoutePointPlayRingbackRequest		Message ID = 2
	duration	Duration
	digitMap	Extension collectDigits
RoutePointPlaySilenceRequest		Message ID = 3
	duration	Duration
	digitMap	Extension collectDigits
RoutePointPlayTreatmentRequest		Message ID = 4
	duration	Duration

BroadWorks Request	Parameters	CSTA PlayMessage Parameter
	digitMap	Extension collectDigits
	audioUrlList	Extension audioUrl
	videoUrlList	Extension videoUrl
NoMatch	CauseInvalidNumberFormat	
Timeout	CauseTimeout	
EOF	CauseTerminationCharReceived	
Stopped-OutOfQueue	CauseSwitchTerminated	
Stopped-NewPlayCollect	CauseNextMessage	
Stopped-OfferedWithRingback	CauseDistributed	
Error	CauseReserved	There is no specific error cause in CSTA.

CSTA PlayEvent Message

The CSTA `PlayEvent` message is generated upon receiving the appropriate *Treatment started* BroadWorks message as described in the following table:

BroadWorks Events and the Associated CSTA PlayEvent Parameters

BroadWorks Events	Parameters	CSTA PlayEvent Parameters
`RoutePointBusyStartedEvent`		Message ID = `0`
`RoutePointMOHStartedEvent`		Message ID = 1
`RoutePointRingbackStartedEvent`		Message ID = 2
`RoutePointSilenceStartedEvent`		Message ID = 3
`RoutePointTreatmentStartedEvent`		Message ID = 4

CSTA StopEvent Message

CSTA `StopEvent` message is generated upon receiving appropriate *Treatment completed* BroadWorks message as described in the following table:

BroadWorks Events and the Associated CSTA StopEvent Parameters

BroadWorks Events	Parameters	CSTA StopEvent Parameters
`RoutePointBusyCompletedEvent`		Message ID = `0` (zero)
	`completionReason`	Cause
	`errorReason`	N/A (presented as string in BroadWorks)
	`digits`	Extension `collectDigits`
`RoutePointMOHCompletedEvent`		Message ID = 1
	`completionReason`	Cause
	`errorReason`	N/A (presented as string in BroadWorks)
	`digits`	Extension `collectDigits`
`RoutePointRingbackCompletedEvent`		Message ID = 2
	`completionReason`	Cause
	`errorReason`	N/A (presented as string in BroadWorks)
	`digits`	Extension `collectDigits`
`RoutePointSilenceCompletedEvent`		Message ID = 3
	`completionReason`	Cause

BroadWorks Events	Parameters	CSTA StopEvent Parameters
	errorReason	N/A (presented as string in BroadWorks)
	digits	Extension collectDigits
RoutePointTreatmentCompletedEvent		Message ID = 4
	completionReason	Cause
	errorReason	N/A (presented as string in BroadWorks)
	digits	Extension collectDigits

BroadWorks Event Cause Translation

The following table describes the BroadWorks event cause translation for the StopEvent message.

Event Cause Translation for the StopEvent Message

BroadWorks Event Cause	CSTA Event Cause	Comments
Match	CauseCharacterCountReached	
NoMatch	CauseInvalidNumberFormat	
Timeout	CauseTimeout	
EOF	CauseTerminationCharReceived	
Stopped-OutOfQueue	CauseSwitchTerminated	
Stopped-NewPlayCollect	CauseNextMessage	
Stopped-OfferedWithRingback	CauseDistributed	
Error	CauseReserved	There is no specific error cause in CSTA.

Configuring PlayAnnouncement Treatments

Configuring PlayAnnouncement Treatments

Currently, Interaction Routing Designer (IRD) can only pass integers when using the standard `Treatment` block. However, the BroadWorks switch must receive a string value in order to activate a treatment: to specify either a URL or the location of a file stored on the web server.

The BroadWorks Connector looks for the `USER_ID` value in the treatment request in order to provide the treatment details to the switch. Starting with release 8.1.2, Interaction Routing Designer supports this functionality within the standard `Treatment` block. However, for older versions of IRD, you must use one of the following two approaches to achieve this functionality:

1. the SendRequest Function
2. the Treatment Function

The following topics describe these approaches in more detail:

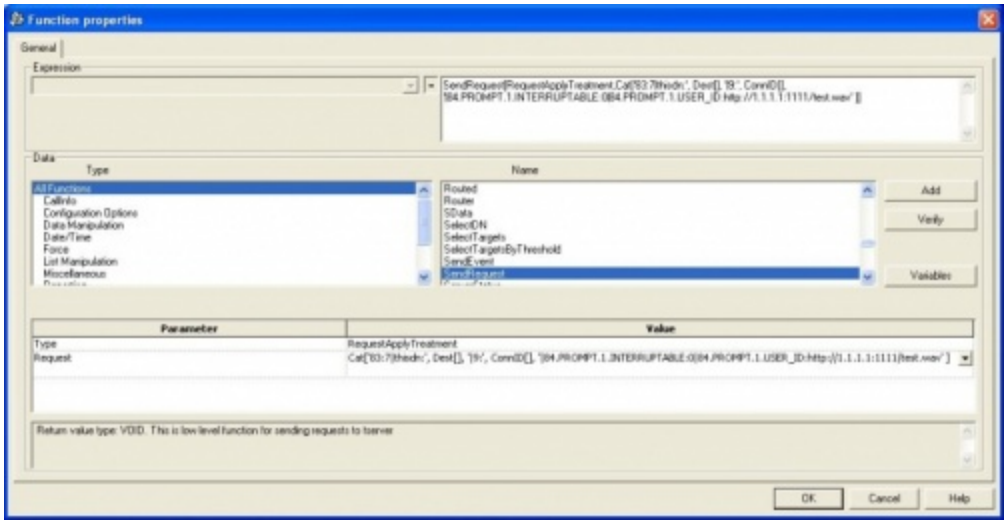
Using the SendRequest Function

Using the SendRequest Function

To use this approach, add a `Function` block to the routing strategy, select `SendRequest` and provide the following information:

```
SendRequest[RequestApplyTreatment,Cat['83:7|thisdn:', Dest[], '|9:',
ConnID[],'|84.PROMPT.1.INTERRUPTABLE:0|84.PROMPT.1.USER_ID:http://1.1.1.1:111/
test.wav' ]]
```

The clickable image below displays how to use the `SendRequest` function:



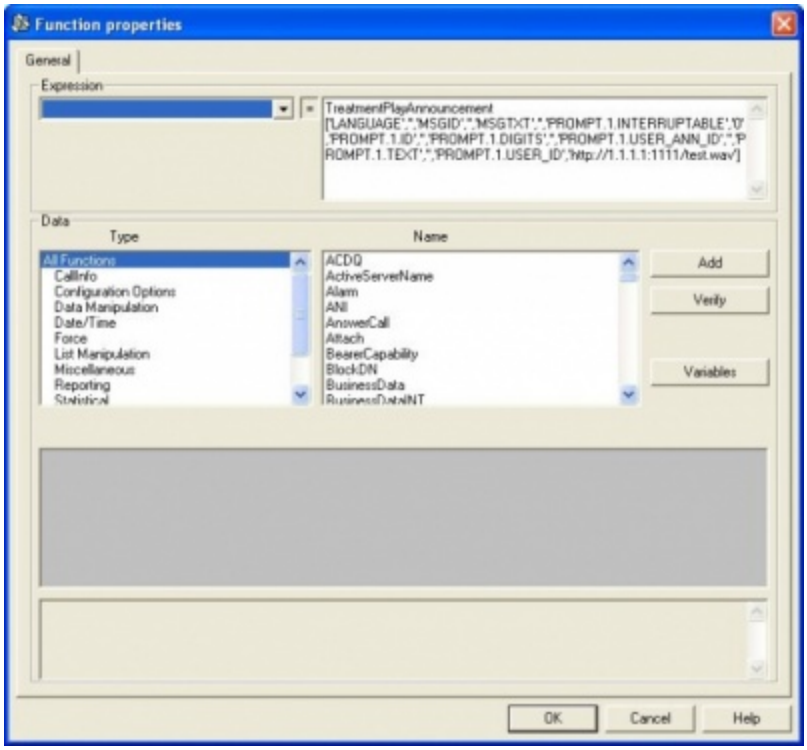
Using the Function Block

Using the Function Block

You can also enter the information typically provided by the `Treatment` block directly into a `Function` block. In the `Function` block, provide the following information:

```
TreatmentPlayAnnouncement['LANGUAGE','','MSGID','','MSGTXT','','PROMPT.1.INTERRUPTABLE',
'0','PROMPT.1.ID','','PROMPT.1.DIGITS','','PROMPT.1.USER_ANN_ID','','PROMPT.1.TEXT','',
'PROMPT.1.USER_ID','http://1.1.1.1:1111/test.wav']
```

The clickable image below displays how to enter information directly into a `Function` block.:



Treatment Request Results

Treatment Request Results

Either of these methods result in the Universal Routing Server (URS) sending the following `TApplyTreatment` request:

```
message RequestApplyTreatment
   AttributeThisDN '54201'
   AttributeConnID 009a01fd7f522011
   AttributeTreatmentType 7 (TreatmentPlayAnnouncement)
   AttributeTreatmentParms [89] 00 01 03 00..
      'PROMPT'(list) '1'(list) 'INTERRUPTABLE' 0
      'USER_ID' 'http://1.1.1.1:1111/test.wav'
   AttributeReferenceID 12
```

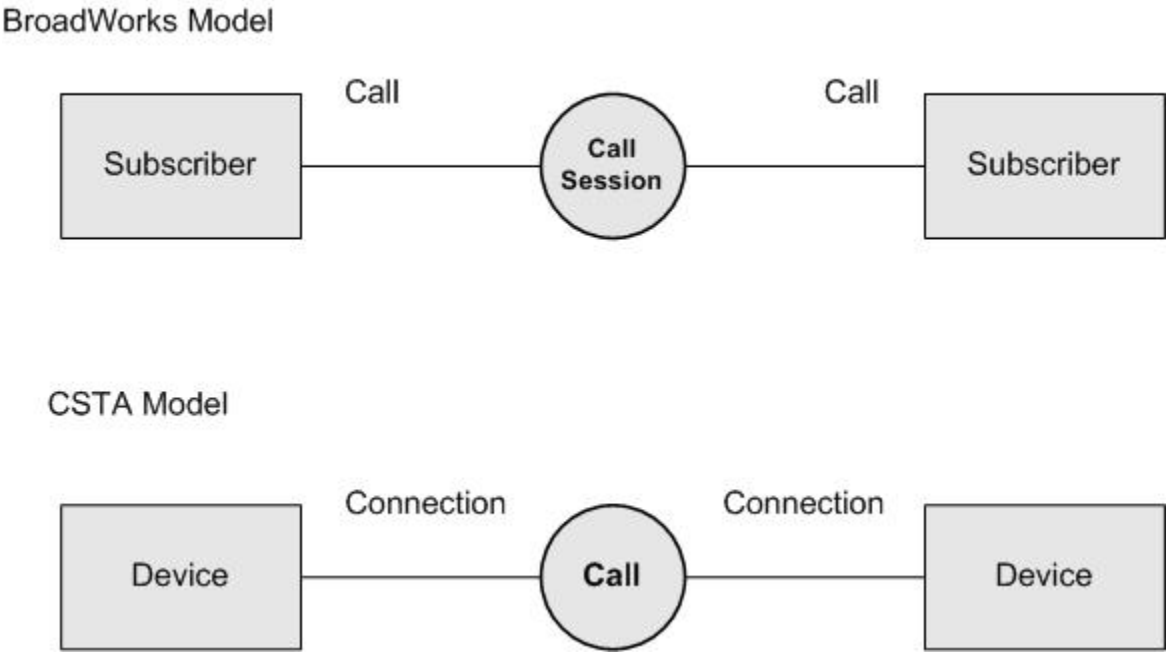
From this request, the BroadWorks Connector takes the information provided by the `USER_ID` parameter and sends it to the switch in the `audioUrlList`:

```
12:30:55.704 +++ XMLToSwitch +++
12:30:55.704 <?xml version="1.0" encoding="UTF-8"?>
<xsp:request version="17.0"xmlns:xsp="http://schema.broadsoft.com/XspXMLInterface">
<requestId>2228</requestId>
  <sessionId>10</sessionId>
```

```
  <credentials>UUFBUlRfRW52M19hZG1pbkBhcHBzZXJ2MDE6YWRtaW4=</credentials>
  <xsp:RoutePointPlayTreatmentRequest>
    <uri>/com.broadsoft.cti-actions/v2.0/
routepoint/%routepointId%/calls/%callId%/playtreatment</uri>
    <method>PUT</method>
    <version>17.0</version>
    <params>
      <routepointId>441290554201</routepointId>
      <callId>callhalf-4258123:0</callId>
    </params>
    <xsp:payload>
      <xsi:RoutePointPlayTreatment xmlns:xsi="http://schema.broadsoft.com/xsi">
      <xsi:audioUrlList>
        <xsi:uri>http://1.1.1.1:1111/test.wav</xsi:uri>
      </xsi:audioUrlList>
      <xsi:numberOfPlay>1</xsi:numberOfPlay>
    </xsi:RoutePointPlayTreatment>
    </xsp:payload>
  </xsp:RoutePointPlayTreatmentRequest>
</xsp:request>
```

Call Models

BroadWorks Connector uses the CSTA call model for reporting. BroadWorks CTI uses a call model which is similar to the CSTA call model. Genesys BroadWorks Connector translates BroadWorks CTI objects to CSTA objects.



BroadWorks vs CSTA Call Model.

Note: The BroadWorks conference call model does not fit the CSTA call model. See, the BroadWorks Conference Model topic for more details.

Device

The Subscriber ID in BroadWorks uniquely identifies an addressable telephony system user – a subscriber. Subscribers are normally assigned E.164 number. For compatibility with CSTA Device Identifier, the E.164 phone number format is used for Subscriber ID in BroadWorks CTI and CSTA. Refer to ITU E.164 Numbering Plan, http://www.itu.int/rec/T-REC-E.164/en for further details.

BroadWorks Connector translates the Public Directory Number used by BroadWorks for identifying Subscribers into a Private Directory Number used in CSTA switching and computing functions. The CSTA Device Monitoring service is translated to a BroadWorks User Subscription on all available event packages for the specified device type.

Connector supports a subscription for the following BroadWorks resources:

- User - Translates to a virtual or real endpoint.

- Call Center - Translates to a CSTA ACD device.
- Routing Point - Translates to a CSTA Routing device.

Call

A call session in BroadWorks is identified by a BroadWorks External Tracking ID. An External Tracking ID is translated into a CSTA Call ID. All External Tracking ID changes are reflected in the appropriate CSTA event reporting.

Connection

A call in the BroadWorks CTI call model is a *logical connection between a subscriber and a call session* Therefore, a *BroadWorks call* is translated into a CSTA Connection. However, the BroadWorks Call ID is not equivalent to the CSTA Call ID because two or more parties on a call have different BroadWorks Call IDs. The BroadWorks Call ID is actually a party ID. The BroadWorks external tracking ID is equivalent to a CSTA Call ID. Due to this characteristic, the BroadWorks Call ID is mapped to a CSTA dynamic device ID element of the CSTA Connection ID.

Configuring CSTA Connector

With the 8.1 Management Layer, an application should be configured as type `CSTA Connector`, and with a pre 8.1 Management Layer, an application should be configured as type `High Availability Proxy`.

T-Server Specific Connection Configuration

T-Server for Connector initiates a session with a Connector as follows:

- If there is at least one application with the `CSTA Connector` type in the list of `Connections`, T-Server reads the Connector's connectivity options (including secure port information) and Connector is handled as a server with normal connectivity options.
- T-Server looks for `CSTA Connector` or `HA Proxy` application types (a pre-8.1 Management layer cannot be used with a CSTA Connector type).

Note: At the moment, T-Server can only have one CSTA Connector in its list of connections.

- If there is no `CSTA Connector` application object in the list of `Connections`, T-Server uses the normal link section configuration.

Device Configuration in the Configuration Layer

The following table displays the device configuration in the Configuration Layer:

Switch Device Type	DN Type	Switch-Specific Type	Association
End Point	Extension	Not applicable	Not applicable
Call Center	ACD Queue	Not applicable	Not applicable
Route Point	• Routing Point • Routing Queue • External Routing Point	• Not applicable • Not applicable • Not applicable	• Not applicable • Not applicable • Not applicable

Switch Terminology

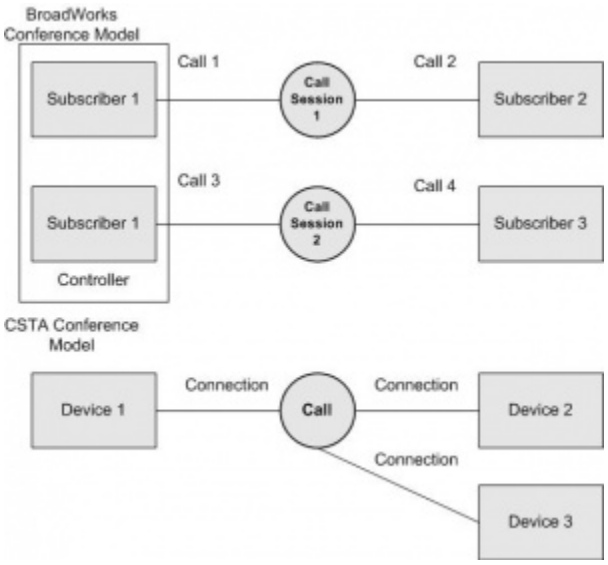
The following table compares the relevant switch terminology with the Genesys terminology:

Genesys Term	Switch Term
ACD Queue	Call Center
Agent ID	User
Extension	End Point
Routing Point	Route Point
Logon	Sign In
Logoff	Sign Out
Ready	Available
NotReady	Unavailable
AfterCallWork	ACD Wrap-up Timer
Account Code	Call Disposition Code
Reason Code	Agent Unavailable Code

Conference Model

In BroadWorks, a conference bridge is logically established by the conference controller device. BroadWorks CTI messaging does not include any messages that inform other conference participants that they are included in the conference. All call sessions remain the same before and after establishing a conference.

In the CSTA model, all conference members (including the conference controller) are connected to the same CSTA call, which is illustrated in the figure below:



CSTA Conference Model Flow.

BroadWorks Compared to the CSTA Conference Model

In a conference scenario, BroadWorks Connector uses BroadWorks Events so the conference controller device can link multiple `CallSessions` into a single CSTA Call. In general, the CSTA conference model does not apply additional restrictions on the conference controller device. Due to BroadWorks' specific way of reporting conference calls, the following restrictions are applied at the CSTA-level to the conference controller device (Device 1 in the figure above):

- The conference controller can not be deleted from the conference call without ending the entire conference call.
- The consultation call from the conference controller is supported, however the `Reconnect Call` and `Transfer Call` CSTA operations are not available for the conference controller.

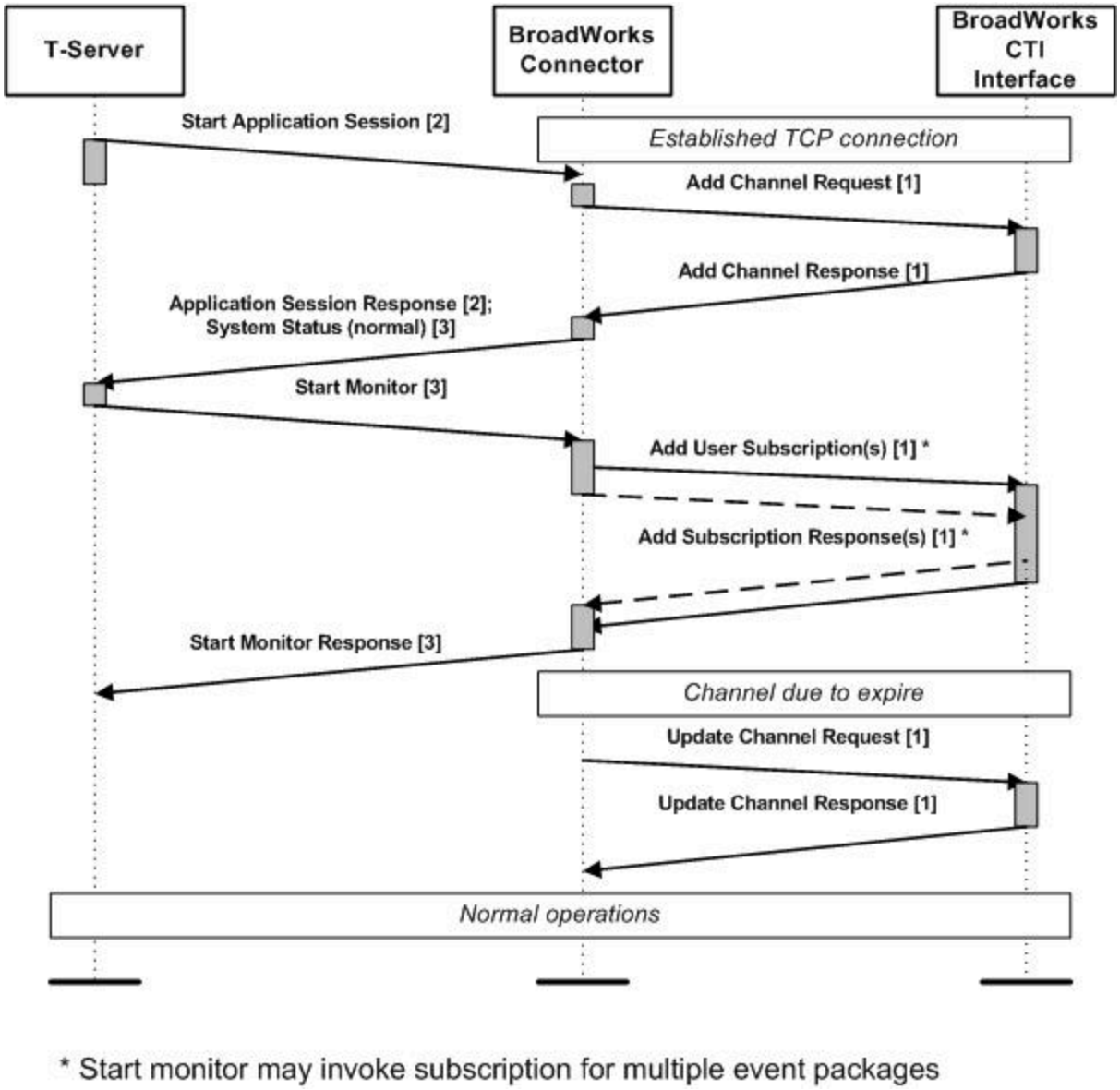
CTI Connection

Establishing Connection to CTI Link

Upon startup, CSTA Connector for BroadSoft BroadWorks opens a Transmission Control Protocol (TCP) connection to the BroadWorks XSP server. All requests (including channel-related requests for event reporting) are sent over the connection. Responses and events are received from the BroadWorks CTI interface through the same connection.

Connector supports connections to multiple XSP servers. CSTA Connector for BroadSoft BroadWorks creates one event channel per TCP connection. See, Redundancy Support for more information.

The figure below displays the initialization of the connection between the Connector and the BroadWorks XSP server:



Initialization Procedure.

Failure Detection and CTI Link Recovery

BroadWorks Connector uses the keep-alive functionality on the TCP connection level. In addition, channel integrity is checked using the `Get Application Controller State` request.

If Connector detects a channel failure, CSTA clients are sent system status and/or monitor stop messages indicating a failed CTI link(s). Connector then restores the TCP socket after the failure and re-creates the Event channel. Resource re-subscription must be re-initiated by the CSTA client.

Hiding Data in Logs

The following topics describe the data hiding functionality for the CSTA Connector:

Hiding Data in Logs Overview

CSTA Connector provides data filtering capabilities based on the Perl-Compatible Regular Expressions (PCRE) library. Sensitive data is defined by a set of regular expressions provisioned in the application configuration though Genesys Management Framework. All data to be hidden is overwritten with an asterisk (*) symbol.

CSTA Connector supports two major modes of hiding sensitive data:

1. **Simple:** Data identified by a matching prefix is hidden up to the end of line.
2. **Complex:** If the regular expression contains one or more *capture* subexpressions, the data matching the subexpressions is hidden while the rest of the text is not. This allows partial hiding of data, such as displaying the last four digits of a credit card number, and preservation of syntactic elements such as parentheses and quotes.

The technical definition of the behaviour of complex patterns is as follows:

- If the regular expression positively matches, and there are no captured sub-strings, the remainder of the text line after the match is hidden.
- If the regular expression matches and there are one or more captured sub-strings, all captured sub-strings are hidden.

Multiple regular expressions can be provided that are arranged in ASCII order by their corresponding option name and applied sequentially. Each subsequent expression is applied to a string that could have been already modified by the preceding expression. It is thus essential that preceding regular expressions do not hide the keyword part of the succeeding expressions. It is recommended that the expressions do not overlap in their matches and that multiple sensitive data chunks are handled in a single expression.

The `non-capture` subexpressions denoted by `(?:)` is not hidden and can be used for grouping the keyword expressions. For example, the expression:

`User (?:PIN|account)` has no `capture` strings and thus the remainder of the string after the match is hidden. If the expression `User (PIN|account)` is used, the words `PIN` or `account` are hidden, while the following data is not, because `(PIN|account)` forms a `capture` sub-expression.

The `non-capture` and `capture` sub-expressions can be combined in one expression with the expected results. For example, the expression:

`[Aa](?:uthentication|ccess)(?: code)?: *"([^"]).{3}"` has two non-capture and one capture strings; the latter, which is the leading characters of the code enclosed between double quotes, is hidden; the last three characters of the code and the closing double quotes are displayed. (The `non-capture` part matches words `authentication` or `authorization` with optionally capitalized initial a

and optionally followed by the word code).

Note: The use of optional capture strings can lead to obscure results. For example, the expression:

`User( name)?`, which matches `User` or `User name`, results in hiding the trailing string in the former case, but the result is hiding the word `name` in the latter case, because `( name)` is a capture subexpression. In this instance, a non-capture string should be used, such as `User(?: name)?`.

A valid example of optional capture string is a code that can come in one or two parts, for example: `Code ([A-Z]{3-5} )? ([0-9]{10})`. This expression matches the word Code followed by a code that is composed of an optional alphanumeric prefix of three to five symbols, and a whitespace, followed by mandatory 10-digit code. Both the prefix and the digital code are masked.

Limits and Constraints

This feature allows you to use arbitrarily complex expressions that affect the central processing unit (CPU) usage according to the number and the complexity of the expressions.

Note: Test the CPU usage impact of your regular expressions before deploying the solution in the production environment.

HA Considerations

Redundant instances of Connectors have independent sensitive data settings that are not automatically synchronized. Use the Application Template to set a generic option synchronization mechanism. See, *The Framework Deployment Guide* for more information.

Related Configuration Options

The log-filter-x section contains the sensitive data definitions. The definitions comprise of a set of Perl-Compatible Regular Expression (PCRE) libraries provisioned as arbitrary options in that section. The configuration option names are lexicographically compared to determine the order of operation. Changes to the contents of the section, or any option, take effect immediately.

Note: Previously written logs are not processed retroactively; the data that was already written remains in the log.

Hot Desking

BroadWorks Connector makes this feature available to T-Server by using the Broadsoft hoteling feature.

Hoteling allows subscribers to associate their service profiles with a different device. Hoteling functionality is delivered through two separate subscribers, Hoteling Host and Hoteling Guest. To provide Hoteling Host functionality, subscriber accounts need to have the Hoteling Host service assigned to them in the BroadWorks configuration. Subscribers that have Hoteling Guest assigned can be associated with a Hoteling Host subscriber and use the Hoteling Host subscriber's device with their service profiles.

Once a guest subscriber has established a hoteling association with a host, it is assigned the host's endpoint (physical device), leaving the host without it. The guest subscriber's id is used in BroadWorks call reporting such as when a call is sent directly to the guest or when an ACD Queue diverts a call to them.

As part of establishing a hoteling session, the Connector sets unconditional forwarding from the Host subsriber to the Guest subscriber. When the hoteling session is cancelled, the Connector cancels forwarding.

T-Server CTI clients initiate a hoteling guest/host association for Hot Desking by sending a `PrivateService` request.

Call Centre Operation

Hoteling is used to support call centre operations by providing a pool of physical devices configured as hoteling hosts that can be used by a potentially larger number of agents configured as hoteling guests. These entities are configured in the Genesys Configuration Layer as Extensions and Agent Logins respectively. It is important that the Hoteling Guests are configured as Agent Logins only and that no Extension is configured for the Hoteling Guest.

T-Server CTI clients initiate a hoteling guest/host association by sending an `AgentLogin` request. The hoteling association is reported as the agent being logged in on the host DN. T-Server uses agent substitution to report all subsequent events using the host as the device id. T-Server CTI clients terminates a hoteling guest/host association by sending an `AgentLogout` request.

Note: The `AgentLogin` request does not assign the agent to a queue as BroadWorks CTI does not support this feature. T-Server CTI clients can still manipulate the agent's queue state using the `AgentReady` and `AgentNotReady` requests.

Event Subscription

BroadWorks Connector identifies hoteling guest subscribers by using the `MonitorDeviceType` private extension in the CSTA `StartMonitor` request; they have a `MonitorDeviceType` of `acdAgent`.

Initially, BroadWorks Connector only subscribes for `HotelingGuest` events on behalf of guest subscribers as follows:

- When BroadWorks Connector receives a `HotelingGuest` event indicating that the guest subscriber is associated with a hoteling host, it subscribes for all event types.
- When BroadWorks Connector receives a `HotelingGuest` event indicating that the guest subscriber is no longer associated with a hoteling host, the Connector deletes all event subscriptions except for Hoteling Guest.

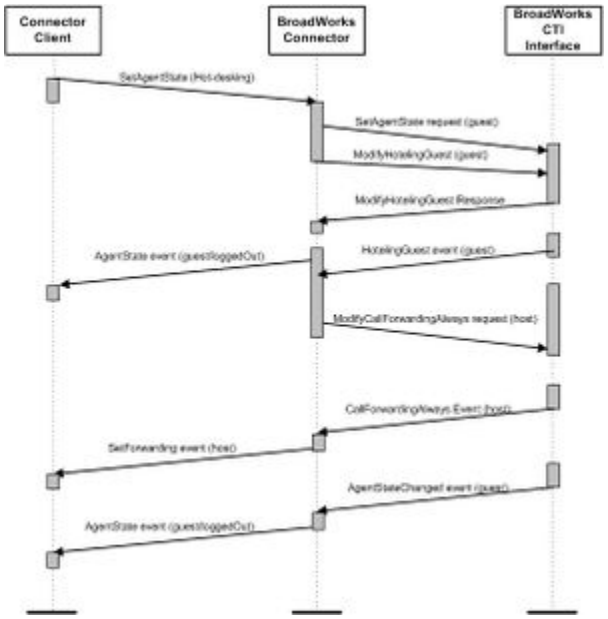
Initiating a Hoteling Guest/Host Association

The BroadWorks Connector performs the following actions when it initiates a hoteling association:

1. Sends a request to the BroadWorks XSP server to set the host subscriber's agent status to signed out, if there is an agent signed in;
2. Sends a request to the BroadWorks XSP server to set unconditional call-forward on the host's subscriber id to the guest's subscriber id;
3. Sends a request to the BroadWorks XSP server to begin the hoteling guest/host association;
4. Sends a request to the BroadWorks XSP server to set the guest subscriber's agent state to unavailable.

BroadWorks Connector initiates a hoteling association when it receives a CSTA `SetAgentState` indicating an agent login. The host subscriber id is taken from the request's device id and the guest subscriber id is taken from the request's agent id.

The following diagram and table shows the sequence of requests and events that are exchanged between BroadWorks Connector and the BroadWorks XSP server:



Initiating the Hoteling Association Message Flow.

Requests and Events Exchanged Between BroadWorks Connector and the BroadWorks XSP Server During Initiation

CSTA	BroadWorks CTI	Comments
`SetAgentState` request:	`SetAgentState` request:	BroadWorks Connector determines the

CSTA	BroadWorks CTI	Comments
• device: host subscriber id • agent state: state • agent ID: guest subscriber id	• subscriberId: host subscriber id • state: signed out	following: the semantics of the request and if the `SetAgentState` request is an agent login from the current agent state. If the `HotelingGuest` event is matched with a Connector request, then these requests are sent to the XSP in response to receiving the `HotelingGuest` event. **Note:** BroadWorks Connector always initialises the agent state as unavailable to remove any dependency between this request and the initial CSTA `SetAgentState` request.
group: n/a	`ModifyCallForwardingAlways` request: • subscriberId: host subscriber id • address: guest subscriber address • active: true	
	`ModifyHotelingGuest` request: • subscriberId: guest subscriber id • hotelingGuest: see notes • hotelingGuest.isActive: true • hotelingGuest.hostSubscriberId: host subscriber id	The HotelingGuest information is provided in the Hoteling Guest events from the initial events subscription.
`AgentState` event: • device: host subscriber id • agent ID: guest subscriber id • logged on state: true • acd group: n/a • agent state: logged in	`HotelingGuest` event: • subscriberId: guest subscriber id • hotelingGuest: from event • hotelingGuest.isActive: true • hotelingGuest.hostSubscriberId: host subscriber id	BroadWorks Connector may also receive a `HotelGuest` event when an external application modifies the hoteling guest association of a monitored subscriber id.
	`SetAgentState` request: • subscriberId: guest subscriber id • state: unavailable	

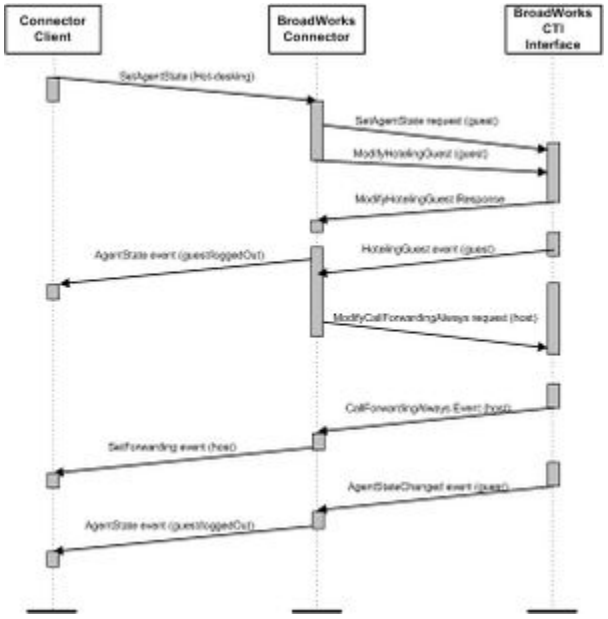
CSTA	BroadWorks CTI	Comments
`AgentState` event: • device: subscriber id • agent ID: subscriber id • logged on state: agent_state dependent • acd group: n/a • agent state: agent_state	`AgentStateChanged` event: • subscriberId: subscriber id • agentACDState: agent_state	• BroadWorks Connector's handling of `AgentStateChanged` events is independent of whether the request was sent as part of the initial hoteling sequence, or in response to any subsequent CSTA `SetAgentState` request. • BroadWorks Connector sends the `AgentState` event to the client in response to receiving the `AgentStateChanged` event.
`SetForwarding` event • device: host subscriber id • type: immediate • activate: true • forwardDN: guest subscriber id	`CallForwardingAlways` event • subscriberId: host subscriber id • address: guest subscriber address • active: true	• BroadWorks Connector's handling of `CallForwardingAlways` events is independent of whether the request was sent as part of the initial hoteling sequence, or in response to any other CSTA `SetForwarding` request. • BroadWorks Connector sends the `SetForwarding` event to the Connector client in response to receiving the `CallForwardingAlways` event.

Terminating a Hoteling Guest/Host Association

The BroadWorks Connector performs the following actions when it terminates a hoteling association:

1. Sends a request to the BroadWorks XSP server to set the guest subscriber's agent status to signed out;
2. Sends a request to the BroadWorks XSP server to cancel unconditional call-forward on the host's subscriber id;
3. Sends a request to the BroadWorks XSP server to end the hoteling guest/host association.

The following diagram and table shows the sequence of requests and events that are passed between BroadWorks Connector and the BroadWorks XSP server:



Terminating the Hoteling Association Call Flow.

Requests and Events Passed Between BroadWorks Connector and BroadWorks XSP Server During Termination

CSTA	BroadWorks CTI	Comments
`SetAgentState` request: • device: host subscriber id • agent state: logged out	`SetAgentState` request: • subscriberId: guest subscriber id	• The `ModifyHotelingGuest` request is sent by BroadWorks Connector to the XSP server in response to

CSTA	BroadWorks CTI	Comments
• agent ID: guest subscriber id • group: n/a	• state: signed out `ModifyCallForwardingAlways` request: • subscriberId: host subscriber id • address: guest subscriber address • active: false `ModifyHotelingGuest` request • subscriberId: guest subscriber id • hotelingGuest: see notes • hotelingGuest.isActive: false • hotelingGuest.hostSubscriberId:	receving a `SetAgentState` request from Connector's client application. • The `HotelingGuest` information is provided in the Hoteling Guest event(s).
`AgentState` event: • device: guest subscriber id • agent ID: guest subscriber id • logged on state: false • acd group: n/a • agent state: logged out	`HotelingGuest` event • subscriberId: guest subscriber id • hotelingGuest: from event • hotelingGuest.isActive: false • hotelingGuest.hostSubscriberId:	• BroadWorks Connector may also receive a `HotelGuest` event when an external application modifies the hoteling guest association of a monitored subscriber id. *BroadWorks Connector sends the `AgentState` event to the Connector client to indicate that the hoteling guest/host association has ended. **Note:** The CSTA `AgentState` event uses the guest subscriber id as it has no information about the former host subscriber.
`AgentState` event: • device: subscriber id • agent ID: subscriber id • logged on state: true • acd group: n/a • agent state: logged in	`AgentStateChanged` event: • subscriberId: guest subscriber id • agentACDState: agent_state	• BroadWorks Connector's handling of `AgentStateChanged` events is independent of whether the request was sent as part of the terminating hoteling sequence, or in response to any subsequent CSTA `SetAgentState` requests. • BroadWorks Connector sends the `AgentState` event to the T-Server in response to receiving the

CSTA	BroadWorks CTI	Comments
		`AgentStateChanged` event.
`SetForwarding` event • device: host subscriber id • type: immediate • activate: false • forwardDN: guest subscriber id	`CallForwardingAlways` event • subscriberId: host subscriber id • address: guest subscriber address • active: true	• BroadWorks Connector's handling of `CallForwardingAlways` events is independent of whether the request was sent as part of the initial hoteling sequence, or in response to any other CSTA `SetForwarding` request. • BroadWorks Connector sends the `SetForwarding` event to the Connector client in response to receiving the `CallForwardingAlways` event.

Interoperability

Interoperability testing confirms that T-Server for CSTA Connector connects and communicates with the BroadWorks Connector.

The following section is a series of tables that presents the T-Library functionality supported by T-Server for CSTA Connector and provides the information on the compatibility of the T-Server for CSTA Connector with the BroadWorks Connector.

The table entries use the following notations:

- **N**—Not supported
- **Y**—Supported
- **I**—Supported, but reserved for Genesys Engineering
- **E**—Event only supported

Call Control Services

Call Control Services

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
TAlternateCall	Alternate Call	TalkRequest	Y	
TAnswerCall	Answer Call	TalkRequest	Y	
TAttachUserData				Internal service, obsolete
TClearCall	Clear Call	ConferenceReleaseRequest	Y	Applicable for conferences only.
TCompleteConference	Conference Call	ConferenceStartRequest	Y	
		ConferenceAddParticipantRequest		Only applicable to the conference controller. (See, BroadSoft BroadWorks Conference Model)
TCompleteTransfer	Transfer Call	ConsultativeTransferRequest	Y	
TDeleteAllUserData			Y	Internal service
TDeleteFromConference	Connection Cleared	ReleaseRequest	Y	The conference controller device can not be deleted from the

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
				conference. (See, BroadSoft BroadWorks Conference Model)
TDeleteUserData			Y	Internal service
THoldCall	Hold Call	HoldRequest	Y	
		ConferenceHoldRequest		Only applicable to the conference controller. (See, BroadSoft BroadWorks Conference Model)
TInitiateConference	Consultation Call	DialRequest	Y	
TInitiateTransfer	Consultation Call	DialRequest	Y	
TListenDisconnect	Speaker Mute	ConferenceDeafRequest	Y	
TListenReconnect	Speaker Mute	ConferenceUnDeafRequest	Y	
TMakeCall	Make Call	DialRequest	Y	
TMakePredictiveCall	Make Predictive Call	RoutePointOutgoingDialRequest		Available from BroadWorks Route Point only
TMergeCalls,			Y	
—MergeForTransfer		Join Call	N	
—MergeForConference		Transfer Call	N	
TMuteTransfer	Single Step Transfer	MuteTransferRequest	Y	Private extension, `MuteTransfer`, in CSTA requests.
TQueryAddress,				Internal service
AddressInfo —AddressStatus			Y	
AddressInfo			Y	
—MessageWaiting Status			Y	
—AddressInfoQueue Status			N	
—AddressInfo AssociationStatus				
AddressInfo —CallForwardingStatus			Y	
AddressInfo			Y	

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
—AgentStatus				
AddressInfo —NumberOfAgentsIn Queue			Y	
AddressInfo —NumberOfAvailable Agents			Y	
InQueue,			Y	
AddressInfo —NumberOfCallsIn Queue			Y	
AddressInfo —AddressType			Y	
AddressInfo —CallsQuery			Y	
AddressInfo —SendAllCallsStatus			N	
AddressInfo —QueueLoginAudit			Y	
AddressInfo —NumberOfIdleTrunks			N	
AddressInfo —NumberOfTrunksInUse			N	
AddressInfo —DatabaseValue			N	
AddressInfo —DNStatus			Y	
TQueryCall,	Snapshot Call		Y	
—CallInfoPartiesQuery —CallInfoStatusQuery			Y	Internal Service
TReconnectCall	Reconnect Call	ReconnectRequest	Y	
TRedirectCall	Deflect Call	BlindTransferRequest	Y	
TReleaseCall	Connection Cleared	ReleaseRequest	Y	

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
TRetrieveCall	RetrieveCall	TalkRequest	Y	
		ConferenceRetrieveRequest		Only applicable to the conference controller. (See, BroadSoft BroadWorks Conference Model)
TSendDTMF	Generate Digits	TransmitDTMFRequest	Y	
TSetCallAttributes			Y	Internal service
TSingleStepConference	Single Step Conference	MonitorCallRequest	Y	
TSingleStepTransfer	Single Step Transfer	BlindTransferRequest	Y	
TUpdateUserData			Y	Internal service
TMonitorNextCall	MonitorStart (call)	MonitorNextCallRequest	Y	
TCancelMonitoring	MonitorStop (call)	ReleaseRequest	Y	

Capability Exchange Services

Capability Exchange Services

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
TQueryServer			Y	Internal service
TSwitchQuery			N	

Private and Special Services

Private and Special Services

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
Set Account Code	Escape Service		Y	Where supported by CSTA Connector
Associate Guest with host	Escape Service		Y	Where supported by CSTA Connector
Disassociate Guest with host	Escape Service		Y	Where supported by CSTA Connector

Registration Services

Registration Services

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
TRegisterAddress	Monitor Start	AddUserSubscriptionRequest	Y	Multiple subscriptions may be created for single subscriber
TRegisterAll			Y	Internal service, restricted
TSetInputMask			Y	Internal service, restricted
TUnregisterAddress	Monitor Stop	DeleteUserSubscriptionRequest	Y	Multiple subscriptions may be deleted for single subscriber
TUnregisterAll			Y	Internal service, restricted

Routing and Treatment Services

Routing and Treatment Services

T-Library	CSTA	BroadWorks CTI	T-Server	Com
TRouteCall,				
—RouteTypeUnknown	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	
—RouteTypeDefault	Route End service	RoutePointFailCall	Y	
—RouteTypeLabel	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	Processed as unknown
—RouteTypeOverwriteDNIS	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	
—RouteTypeDDD	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	Processed as unknown
—RouteTypeIDDD	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	Processed as unknown
—RouteTypeDirect	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	Processed as unknown

T-Library	CSTA	BroadWorks CTI	T-Server	Com
—RouteTypeReject	Route End	RoutePointReleaseCallRequest	Y	
—RouteTypeAnnouncement	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	Processed as unknown
—RouteTypePostFeature	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	Processed as unknown
—RouteTypeDirectAgent	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	Processed as unknown
—RouteTypePriority	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	Processed as unknown
—RouteTypeDirectPriority	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	Processed as unknown
—RouteTypeAgentID	Route Select service	RoutePointDistributeCallRequest, RoutePointBlindTransferRequest	Y	Processed as unknown
—RouteTypeCallDisconnect	Route Reject	RoutePointReleaseCallRequest	Y	
TGiveMusicTreatment	Play Message	PlayTreatmentRequest	Y	
TGiveSilenceTreatment	GenerateTelephonyTones	PlaySilenceRequest	Y	
TGiveRingBackTreatment	GenerateTelephonyTones	PlayRingbackRequest	Y	
TApplyTreatment,				
—TreatmentUnknown			N	
—TreatmentIVR			N	
—TreatmentMusic	Play Message	PlayMusicOnHoldRequest	Y	
—TreatmentRingBack	GenerateTelephonyTones	PlayRingbackRequest	Y	
—TreatmentSilence	GenerateTelephonyTones	PlaySilenceRequest	Y	
—TreatmentBusy	GenerateTelephonyTones	PlayBusyRequest	Y	
—TreatmentCollectDigits	Start Data Collection	PlaySilenceRequest	Y	Collect digit added to req
—Treatment	GenerateTelephonyTones	PlayTreatmentRequest	Y	
PlayAnnouncement,				
—Treatment	Play Message	PlayTreatmentRequest	Y	
PlayAnnouncementAndDigits,				
—TreatmentVerifyDigits	Start Data Collection		Y	
—Treatment	Start Data Collection		Y	
RecordUserAnnouncement,			N	
—Treatment			N	
DeleteUserAnnouncement,			N	
—TreatmentCancelCall	Route End Service	ReleaseCallRequest	Y	

T-Library	CSTA	BroadWorks CTI	T-Server	Com
—TreatmentPlayApplication			N	
—TreatmentSetDefault Route			N	
—TreatmentTextToSpeech			N	
TextToSpeechAndDigits,			N	
—TreatmentFastBusy			N	
—TreatmentRAN			N	

Set Feature Services

Set Feature Services

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
TSetMessageWaitingOn	Set Message Warning		N	
TSetMessageWaitingOff	Set Message Warning		N	
TSetDNDOn	Set Do Not Disturb	ModifyDoNotDisturb	Y	
TSetDNDOff	Set Do Not Disturb	ModifyDoNotDisturb	Y	
TCallSetForwardOn,				
—ForwardModeNone	Set Forwarding	ModifyCallForwardingAlways	Y	Processed as unconditional
—ForwardMode	Set Forwarding	ModifyCallForwardingAlways	Y	
Unconditional,				
—ForwardMode	Set Forwarding	ModifyCallForwardingBusy	Y	
OnBusy,				
—ForwardMode	Set Forwarding	ModifyCallForwardingNoAnswer	Y	
OnNoAnswer,				
—ForwardMode			N	
OnBusyAndNoAnswer,				
—ForwardMode			N	
SendAllCalls				
TCallCancelForward,				
—ForwardModeNone	Set Forwarding	ModifyCallForwardingAlways		Processed as unconditional
—ForwardMode	Set Forwarding	ModifyCallForwardingAlways	Y	

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
Unconditional,				
—ForwardModeOnBusy	Set Forwarding	ModifyCallForwardingBusy	Y	
—ForwardMode	Set Forwarding	ModifyCallForwardingNoAnswer	Y	
OnNoAnswer,				
—ForwardMode			N	
OnBusyAndNoAnswer,				
—ForwardMode			N	
SendAllCalls				
TAgentLogin,				
—AgentWorkMode Unknown	Set Agent State	SetAgentStateRequest(Sign-In),	Y	Full support for emulated agents only, used for hoteling functionality with guest subscribers
—AgentManualIn	Set Agent State	SetAgentStateRequest(Sign-In),	Y	
—AgentAutoIn	Set Agent State	SetAgentStateRequest(Sign-In),	Y	
—AgentAfterCallWork			N	
—AgentAuxWork			N	
—AgentNoCall Disconnect			N	
TAgentLogout	Set Agent State	SetAgentStateRequest(Sign-Out)	Y	Full support for emulated agents only, used for hoteling functionality with guest subscribers
		ModifyHotelingGuest		
TAgentSetReady	Set Agent State	SetAgentStateRequest(Available)		
TAgentSetNotReady,				
—AgentWorkMode Unknown	Set Agent State	SetAgentStateRequest(Unavailable)		
—AgentManualIn	Set Agent State	SetAgentStateRequest(Unavailable)		
—AgentAutoIn	Set Agent State	SetAgentStateRequest(Unavailable)		
—AgentAfterCallWork	Set Agent State	SetAgentStateRequest(Wrap-Up)	Y	
—AgentAuxWork	Set Agent State	SetAgentStateRequest(Unavailable)		
—AgentNoCall Disconnect			N	
TAgentSetIdleReason			N	
TSetMuteOff			N	

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
TSetMuteOn			N	
TListenDisconnect			N	
TListenReconnect			N	

Supported Events

Supported Events

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
EventAbandoned	Connection Cleared	CallReleased,	Y	
		ACDCallAbandoned,		
		RoutePointCallAbandoned		
EventAddressInfo			Y	Internal T-Server event
EventAgentIdleReasonSet			N	
EventAgent Login				
--AgentWorkModeUnknown,	Agent Logged On	AgentState	Y	
--AgentManualIn	Agent Logged On	AgentState	Y	
--AgentAutoIn	Agent Logged On	AgentState	Y	
--AgentAfterCallWork			N	
--AgentAuxWork			N	
--AgentNoCallDisconnect			N	
EventAgentLogout	Agent Logged Off	AgentState	Y	
EventAgentNotReady				
--AgentWorkModeUnknown,	Agent Not Ready	AgentState	Y	
--AgentManualIn	Agent Not Ready	AgentState	Y	
--AgentAutoIn	Agent Not Ready	AgentState	Y	
--AgentAfterCallWork	Agent Working After Call	AgentState	Y	
--AgentAuxWork	Agent Not Ready	AgentState	Y	
--AgentNoCallDisconnect			N	
EventAgentReady		AgentState	Y	

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
EventAttachedData Changed			Y	Internal T-Server Event
EventDestinationBusy	Failed	CallReleasingEvent	Y	
EventDialing	Originated, Delivered	CallOriginated	Y	
EventDigitsCollected	Stop Event (with collectDigits extension)	RoutePointTreatmentCompleted,	Y	Distributed when collected data is available
		RoutePointMOHCompleted,		
		RoutePointRingbackCompleted,		
		RoutePointBusyCompleted,		
		RoutePointSilenceCompleted		
EventDiverted	Diverted	RoutePointCallForwarded,	Y	
		RoutePointCallFailed,		
		RoutePointCallAnsweredByAgent.,		
		ACDCallForwarded,		
		ACDCallAnsweredByAgent,		
		ACDCallBounced (when “redirect” attribute present)		
EventDNDOff	Do Not Disturb	DoNotDisturb	Y	
EventDNDOn	Do Not Disturb	DoNotDisturb	Y	
EventDTMFSent	Digits Generated	Response on TransmitDTMFRequest	Y	Response on Request
EventError			Y	Negative response on request
EventEstablished	Established	CallAnswered	Y	
EventForwardCancel	Forwarding	CallForwardingAlways,	Y	
		CallForwardingNoAnswer,		
		CallForwardingBusy		
EventForwardSet	Forwarding	CallForwardingAlways,	Y	
		CallForwardingNoAnswer,		
		CallForwardingBusy		
EventHeld	Held	CallHeld,	Y	
		ConferenceHeld		
EventHardwareError			N	
EventLinkConnected			Y	Internal T-Server event
EventLinkDisconnected			Y	Internal T-Server event
EventMailBoxLogin			N	

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
EventMailBoxLogout			N	
EventMessageWaitingOff			N	
EventMessageWaitingOn			N	
EventNetworkReached	NetworkReached	CallUpdated	Y	
EventOffHook	ServiceInitiated	CallOriginating	Y	
EventOnHook	ConnectionCleared	CallReleased	Y	
EventPartyAdded	Conferenced	ConferenceStarted	Y	
EventPartyChanged	Transferred,	CallTransferred	Y	
	Conferenced			
EventPartyDeleted	ConnectionCleared	CallReleased	Y	
EventQueued	Queued,	ACDCallAdded,	Y	
	Delivered	RoutePointCallAdded		
EventRegistered	Response on Monitor Start Request	Initial Subscription Event	Y	Internal T-Server event for subsequent registartions
EventRegisteredAll			Y	Internal T-Server Event
EventReleased	ConnectionCleared	CallReleased		
EventRemoteConnectionFailed			Y	Internal T-Server Event
EventRemoteConnectionSuccess			Y	Internal T-Server Event
Event Retrieved	Retrieved	CallRetrieved,	Y	
		ConferenceRetrieved		
Event Ringing	Delivered	CallReceived	Y	
Event Route Request		RoutePointCallAdded	Y	
Event Route Used		RoutePointCallOfferedToAgent,	Y	
		RoutePointCallAnsweredByAgent		
Event Server Connected			Y	Internal T-Server Event
Event Server Disconnected			Y	Internal T-Server Event
Event Server Info			Y	Internal T-Server Event
Event Unregistered		SubscriptionTerminated	Y	
Event Unregistered All			Y	Internal T-Server Event
Event User Event			Y	Internal T-Server Event
Event Voice File Closed			N	

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
Event Voice File End Play			N	
Event Voice File Opened			N	
EventResourceAllocated			N	
EventResourceFreed			N	
EventListenDisconnected		ConferenceCallMadeDeaf	Y	
EventListenReconnected		ConferenceCallMadeUndeaf	Y	
EventPartyInfo			Y	Internal T-Server Event
EventCallInfoChanged			Y	Internal T-Server Event
EventTreatmentApplied		RoutePointTreatmentStarted,	Y	
		RoutePointMOHStarted,		
		RoutePointRingbackStarted,		
		RoutePointBusyStarted,		
		RoutePointSilenceStarted		
EventTreatmentNotApplied		Error Response	Y	
EventTreatmentEnd		RoutePointTreatmentCompleted,	Y	
		RoutePointMOHCompleted,		
		RoutePointRingbackCompleted,		
		RoutePointBusyCompleted,		
		RoutePointSilenceCompleted		
EventTreatmentRequired			N	
EventSwitchInfo			N	
Event DN Out Of Service			TBD	
Event DN Back In Service			TBD	
Event Private Info			Y	

Voice Unit Services

Voice Unit Services

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
TPlayVoiceFile			N	
TLoginMailBox			N	

T-Library	CSTA	BroadWorks CTI	T-Server	Comments
TLogoutMailBox			N	
TOpenVoiceFile			N	
TCloseVoiceFile			N	
TCollectDigits			N	

Messaging

Requests

Connector processes incoming CSTA requests according to device types specified in the request. The result of a CSTA request invocation varies according to different types of targets.

Request Groups Requirements

BroadWorks CTI interface offers different subset of requests for different types of resources. See, BroadWorks Requests and *CTI Interface Specification, v 1.6. BroadSoft* for more information on BroadWorks request applicability.

Request Response

BroadWorks Connector uses responses received from BroadWorks CTI to generate appropriate response(s) to send to CSTA clients.

Events

Event Subscription

BroadWorks Connector subscribes to specific types of notifications called Event Packages. Different types of event packages are required for different device types. For example, Routing Points require Route Point Queue event package subscription. Refer to *CTI Interface Specification, v 1.6. BroadSoft* for further details.

Event Processing

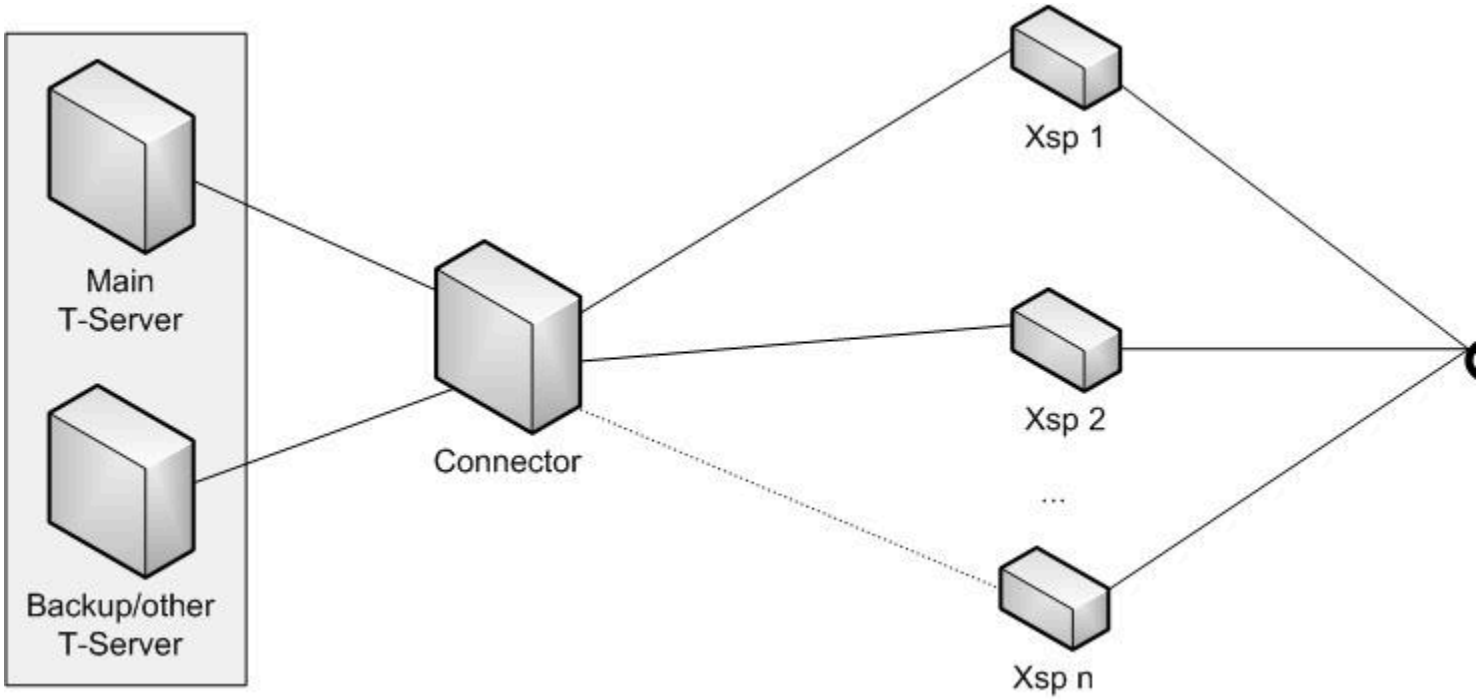
BroadWorks CTI events are translated into a common set of CSTA events, which are independant (generally) from the resource type. See, BroadWorks Events for more information.

Redundancy Support

The following topics describe the redundancy support for the CSTA Connector:

Redundancy Support Feature Overview

BroadWorks Connector operates in unison with the generic Connector T-Server in order to provide a high availability (HA) solution. The HA deployment configurations are shown below.



Multiple Links from CSTA Client to Connector.

The main features of this solution are:

- Connector T-Servers runs as a standard Genesys HA pair.
- BroadWorks Connector operates independently of other Connectors, it does not synchronise information in the same manner as the T-Server HA pair.
- BroadWorks Connector maintains redundant links to the BroadWorks switch independently of the Connector T-Server.
- BroadWorks Connector is able to support multiple T-Server connections.

BroadWorks Connector Operation

BroadWorks Connector supports connections to multiple XSP servers. BroadWorks Connector uses multiple Event Channels, and each Event Channel is given equal weight. BroadWorks CTI balances the event reporting across all the Event Channels. If one of the Event Channels fails, BroadWorks CTI continues to serve the event subscription over the remaining Event Channels adjusting the load accordingly. BroadWorks Connector initiates its link recovery procedure to re-establish a failed connection.

Connector supports multiple T-Server connections. T-Servers may or may not start monitors for the same set of devices, Connector filters the events and requests for each connection against their respective start monitor requests.

Connector responds to `SystemStatusRequests` from T-Server indicating overall status of links, and reports a failure if all links are down.

Supported BroadWorks Requests and Events

BroadWorks supports the following requests and events:

- BroadWorks Requests
- BroadWorks Events

BroadWorks Requests

The following tables represent the supported BroadWorks requests:

Call Requests

Call Requests

BroadWorks CTI	CSTA	Comments
BlindTransferRequest	Single Step Transfer Call	
ConferenceAddParticipantRequest	Conference Call	
ConferenceHoldRequest	Hold Call	
ConferenceReleaseRequest	Clear Call	
ConferenceRetrieveRequest	Retrieve Call	
ConferenceStartRequest	Conference Call	
ConsultativeTransferRequest	Transfer Call	
DialRequest	Make Call, Consultation Call	

BroadWorks CTI	CSTA	Comments
DirectedCallPickupRequest	Directed PickUp Call	
DirectedCallPickupWithBargeInRequest	Join Call	With the `BargeIn` `Extensions` attribute.
EscalateToSupervisorRequest	Consultation Call	Agent request
EmergencyCallToSupervisorRequest	Single Step Conference	Agent request
GetACDRequest	Snapshot Device	ACD request
GetRoutePointRequest	Snapshot Device	Route Point request
GetRoutePointStateRequest	GetRouteingMode	Route Point request
HoldRequest	Hold Call	
ModifyRoutePointStateRequest	SetRouteingMode	Route Point request
MonitorCallRequest	Join Call	
MonitorNextCallRequest	Make Connection	
MuteTransferRequest	Consultation Call	Private `MuteTransfer` `Extensions` attribute in request.

BroadWorks CTI	CSTA	Comments
ReconnectRequest	Reconnect Call	
ReleaseRequest	Clear Connection	
RetrieveCallIdInfoListRequest	Snapshot Device	
RoutePointBounceCallRequest	Route Reject / Re-Route	Route Point request
RoutePointBlindTransferRequest	Route Select	Route Point request
RoutePointDistributeCallRequest	Route Select	Route Point request (CSTA extensions)
RoutePointFailCallRequest	Route End	Route Point request
RoutePointOutgoingDialRequest	Make Predictive Call	Route Point request
RoutePointReleaseCallRequest	Route Reject	Route Point request
TalkRequest	Answer Call, Retrieve Call, Alternate Call	
TransmitDTMFRequest	Generate Digits	

Device Requests

Device Requests

BroadWorks CTI	CSTA	Comments
GetCallForwardingAlwaysRequest	Get Forwarding	
GetCallForwardingBusyRequest	Get Forwarding	
GetCallForwardingNoAnswerRequest	Get Forwarding	
GetDoNotDisturbRequest	Get Do Not Disturb (DND)	
GetHotelingGuestRequest	Escape service	
GetHotelingHostListRequest	N/A	Internal Connector service
ModifyCallForwardingAlwaysRequest	Set Forwarding	
ModifyCallForwardingBusyRequest	Set Forwarding	
ModifyCallForwardingNoAnswerRequest	Set Forwarding	
ModifyDoNotDisturbRequest	Set Do Not Disturb (DND)	
ModifyHotelingGuestRequest	Set Agent State or Escape Service	

BroadWorks CTI	CSTA	Comments
SetAgentStateRequest	Set Agent State	
TagDispositionCodeRequest	Escape service	
TagDispositionCodeToCallRequest	Escape service	

Media Requests

Media Requests

BroadWorks CTI	CSTA
RoutePointPlayBusyRequest	Play Message
RoutePointPlayMOHRequest	Play Message
RoutePointPlayRingbackRequest	Play Message
RoutePointPlaySilenceRequest	Play Message, StartDataCollection
RoutePointPlayTreatmentRequest	Play Message, StartDataCollection

System Requests

System Requests

BroadWorks CTI	CSTA	Comments
AddChannelRequest	Not required	Internal Connector service
AddEnterpriseGroupSubscriptionRequest	Not required	
AddEnterpriseSubscriptionRequest	Not required	
AddServiceProviderGroupSubscriptionRequest	Not required	
AddServiceProviderSubscriptionRequest	Not required	
AddSystemSubscriptionRequest	Not required	
AddUserSubscriptionRequest	Start Monitor	
DeleteChannelRequest	Not required	Internal Connector service
DeleteSubscriptionRequest	Stop Monitor	Internal Connector service
GetApplicationControllerStateRequest	Not required	Internal Connector KPL service
GetChannelSetRequest	Not required	Internal Connector

BroadWorks CTI	CSTA	Comments
		service
GetSingleSubscriptionRequest	Not required	Internal Connector service
GetSubscriptionsRequest	Not required	Internal Connector service
SetApplicationControlledRouteRegisterRequest	RouteRegisterReq	
UpdateChannelRequest	Not required	Internal Connector service
UpdateSubscriptionRequest	Not required	Internal Connector service

BroadWorks Events

The following tables represent the supported BroadWorks events:

Call Events

Call Events

BroadWorks CTI	CSTA	Comments
ACDCallAbandonedEvent	Connection Cleared	
ACDCallAddedEvent	Queued	
ACDCallAnsweredByAgentEvent	Diverted	
ACDCallBouncedEvent	Delivered, Diverted	Translated to a diverted event if the `redirect` field is provided.
ACDCallEscapedEvent	Connection Cleared	
ACDCallForwardedEvent	Diverted	
ACDCallOfferedToAgentEvent	Delivered	
ACDCallOverflowedEvent	Diverted	

BroadWorks CTI	CSTA	Comments
ACDCallPromotedEvent	Call Information	
ACDCallReleasedEvent	Connection Cleared	
ACDCallReorderedEvent	Call Information	
ACDCallStrandedEvent	Connection Cleared	
ACDCallTransferredEvent	Diverted	
ACDCallUpdatedEvent	Transferred	
ACDHolidayPolicyAppliedEvent	Diverted	
ACDNightPolicyAppliedEvent	Diverted	
ACDSubscriptionEvent	N/A	Internal Connector functionality.
CallAnsweredEvent	Established	
CallBargedInEvent	Conferenced	
CallCollectingEvent	Service Initiated	With private data.

BroadWorks CTI	CSTA	Comments
CallForwardedEvent	Diverted	
CallHeldEvent	Held	
CallMonitoredEvent	Conferenced	
CallOriginatedEvent	Originated	
CallOriginatingEvent	Service Initiated	
CallParkRetrievedEvent	Retrieved	
CallPicked-UpEvent	Diverted	
CallReceivedEvent	Delivered	
CallRedirectedEvent	Diverted	
CallReleasedEvent	Connection Cleared	
CallReleasingEvent	Failed, Connection Cleared	
CallRetrievedEvent	Retrieved	

BroadWorks CTI	CSTA	Comments
CallSubscriptionEvent	---	Event may be translated into various CSTA Events
CallTransferredEvent	Transferred	
CallUpdatedEvent	---	Depends on call topology.
ConferenceHeldEvent	Held	
ConferenceReleasedEvent	Connection Cleared	
ConferenceRetrievedEvent	Retrieved	
ConferenceStartedEvent	Conferenced	Depends on call topology.
ConferenceUpdatedEvent	Conferenced	
RoutePointCallAbandonedEvent	Connection Cleared	Route End is sent if the Routing dialog is opened
RoutePointCallAddedEvent	Queued; Route Request	
RoutePointCallAnsweredByAgentEvent	Diverted; Route End	

BroadWorks CTI	CSTA	Comments
RoutePointCallBouncedEvent	Queued; Route Request	
RoutePointCallFailedEvent	Diverted; Route End	Route End is sent if the `Routing` dialog is opened.
RoutePointCallForwardedEvent	Diverted; Route End	Route End is sent if the `Routing` dialog is opened.
RoutePointCallOfferedToAgentEvent	Route End	
RoutePointCallOverflowedEvent	Diverted; Route End	Route End is sent if the `Routing` dialog is opened.
RoutePointCallReleasedEvent	Connection Cleared, Route End	Route End is sent if the `Routing` dialog is opened.
RoutePointCallTransferredEvent	Diverted; Route End	
RoutePointCallUpdatedEvent	Transferred	
RoutePointFailedEvent	Connection Cleared; Route End	Route End is sent if the `Routing` dialog is opened.

BroadWorks CTI	CSTA	Comments
RoutePointHolidayPolicyAppliedEvent	Connection Cleared; Route End	Route End is sent if the `Routing` dialog is opened.
RoutePointNightPolicyAppliedEvent	Connection Cleared; Route End	Route End is sent if the `Routing` dialog is opened.
RoutePointOutgoingCallAnsweredEvent	Queued; Route Request	
RoutePointOutgoingCallOriginatedEvent	Originated; Delivered	
RoutePointRecoveredEvent	Routeing Mode	
RoutePointSubscriptionEvent	N/A	Internal Connector functionality
RoutePointWhisperStartedEvent	N/A	

Device Events

Device Events

BroadWorks CTI	CSTA	Comments
`ACDAgentJoinUpdateEvent`	• Agent Logged On • Agent Logged Off	

BroadWorks CTI	CSTA	Comments
AgentSubscriptionEvent	Agent (varies)	A combination of agent events to provide the matching target agent state.
AgentDispositionCodeAddedEvent	Call Information	
AgentStateEvent	Agent (varies)	An appropriate CSTA event to provide the matching target agent state.
CallForwardingAlwaysEvent	Forwarding	
CallForwardingBusyEvent	Forwarding	
CallForwardingNoAnswerEvent	Forwarding	
DoNotDisturbEvent	Do Not Disturb	
HotelingGuestEvent	• Agent Logged On • Agent Logged Off • Private Event	Distinguished from ACDAgentJoinUpdateEvent by the omission of the ACD Queue group.

Media Events

Media Events

BroadWorks CTI	CSTA
RoutePointBusyCompletedEvent	Stop
RoutePointBusyStartedEvent	TelephonyTones generated
RoutePointMOHCompletedEvent	Stop
RoutePointMOHStartedEvent	TelephonyTones generated
RoutePointRingbackCompletedEvent	Stop
RoutePointRingbackStartedEvent	TelephonyTones generated
RoutePointSilenceCompletedEvent	Stop
RoutePointSilenceStartedEvent	TelephonyTones generated
RoutePointTreatmentCompletedEvent	Stop
RoutePointTreatmentStartedEvent	TelephonyTones generated

System Events

System Events

BroadWorks CTI	CSTA	Comments
`ChannelTerminatedEvent`	System Status, Monitor Stop	Monitor Stop sent for all subscriptions created over the terminated channel.
`SubscriptionTerminatedEvent`	Monitor Stop	

Configuration Options

You must configure the configuration objects and options described in the topics below in the Framework Configuration Layer:

CSTA Connector Application-Level Options

The configuration options specific to the CSTA Connector functionality are set in Configuration Manager, in the corresponding sections of the `Options` tab of the CSTA Connector `Application` object. The configuration options are configured in the following sections:

- Connector Section Options
- Link-%s Section Options
- License Section Options
- log-filter-x Section Options

Connector Section

This section must be called `Connector`.

app-controller

app-controller

Default Value: Mandatory field. No default value.
Valid Value: Any string
Changes Take Effect: Immediately

Required identifier used by BroadWorks routing functionality. Refer to the Route Point Failure Policy in the BroadWorks CTI documentation.

application-name

application-name

Default Value: `GenesysConnector`
Valid Value: Any string
Changes Take Effect: Immediately

Identifier used in BroadWorks event reporting to represent the Connector application.

channel-set-name

channel-set-name

Default Value: Mandatory field. No default value
Valid Values: Any string
Changes take effect: When the link is started or restarted

The `ChannelSet` name specifies a logical name for a CTI link to BroadWorks. Multiple TCP/IP channels opened to BroadWorks CTI server(s) with the same `ChannelSet` name are treated as a single CTI link. The Connector generates a unique `ChannelSet` name to ensure that there is no unwanted interaction between multiple Connectors connected to the same switch.

default-domain

default-domain

Default Value: None
Valid Value: Any string
Changes Take Effect: Immediately

> **Important**
> Genesys does not recommend the use of this option.

BroadWorks require a fully qualified device identifier in requests. A default domain name can be appended to a `UserID` attribute in requests for convenience. In the BroadWorks switch web configuration, the list of assigned domains can be seen by browsing to the Enterprise branch, selecting the Resources section, and then selecting the Assign Domains sub-section. See, the `dn-prefix` option.

def-treatment-uri

def-treatment-uri

Default Value: None
Valid Value: Any valid URI path
Changes Take Effect: Immediately

The uniform resource identifier (URI) to a default treatment to apply in treatment requests. Refer to the vendor's configuration documentation for further information.

def-whisper-uri

def-whisper-uri

Default Value: None
Valid Value: Any valid URI path
Changes Take Effect: Immediately

The uniform resource identifier (URI) to the default whisper message to apply in routing requests with supervision. Refer to the vendor's configuration documentation for further information.

def-whisper-video-uri

def-whisper-video-uri

Default Value: None
Valid Value: Any valid URI path
Changes Take Effect: Immediately

The uniform resource identifier (URI) to the default whisper video message to apply in routing requests with supervision. Refer to the vendor's configuration documentation for further information.

dn-prefix

dn-prefix

Default Value: Mandatory field. No default value.
Valid Value: Any string
Changes Take Effect: Immediately

Numbers (DNs) are provisioned in BroadWorks and Configuration Manager using a shortened number format.
However, CTI requests require the fully qualified E.164 number. The DN Prefix is prepended to the short DN to provide the full E.164 number.

max-outstanding

max-outstanding

Default Value: 16
Valid Value: 1-1000
Changes Take Effect: Immediately

Specifies the maximum number of outstanding unacknowledged requests sent to the switch at any given time.

password

password

Default Value: Mandatory field. No default value.
Valid Value: The password configured on the BroadWorks switch.
Changes Take Effect: When the link is started, or restarted.

Configures the password provisioned in the BroadWorks Administrator, which is required for request authentication on the XSP CTI interface.

restart-period

restart-period

Default Value: `10`
Valid Values: Any positive integer between `3-300`
Changes Take Effect: When the link is started or restarted.
Specifies the interval (in seconds) that the Connector waits between attempts to reconnect on all configured connections.

rq-gap

rq-gap

Default Value: `0` (zero)
Valid Value: `0-1000`
Changes Take Effect: Immediately
Specifies the minimum interval, in milliseconds, between successive requests sent over the link to the PBX. The value can be adjusted to meet the CTI-link load and performance requirements.

username

username

Default Value: Mandatory field. No default value.
Valid Value: The username configured on the BroadWorks switch.
Changes Take Effect: When the link is started, or restarted.

This option configures the username provisioned in the BroadWorks Administrator, which is required

for request authentication on the XSP CTI interface.

link-%s Section

This section name represents a specific Connector link to the switch and is user-customisable with `%s` being replaced by any string.

hostname

hostname

Default Value: Mandatory field. No default value.
Valid Value: Any valid host name
Changes Take Effect: When the link is started, or restarted.

Specifies the hostname/IP address of the switch/XSP Server. You must specify a value for this option.

port

port

Default Value: `8011`
Valid Value: Any valid TCP/IP port
Changes Take Effect: When the link is started or restarted.

Specifies the port address of the switch interface. This option must be specified.

priority

priority

Default Value: `0` (zero)
Valid Values: `0-100`
Changes Take Effect: When the link is started or restarted.

Specifies the priority of the connection. Higher numbers have higher priority.

License Section

Connector is a licensed product and can only be operational if a valid license is applied. Licensing uses the FlexLM library in common with T-Server and is configured in Configuration Manager.

num-sdn-licenses

Default Value: `max`
Valid Value: Any positive integer or `max`
Changes Take Effect: Immediately

Specifies how many seat-related licenses Connector checks out. A value of `0` (zero) means that Connector does not grant control of seat-related DNs to any client, and it does not look for seat-related DN licenses at all. When set to `Max`, all available licenses are checked out.

log-filter-x Section

Use the `log-filter-x` section configuration options to hide sensitive data from log(s). The sensitive data is masked by asterisks (*) in the log file. See, Hiding Data in Logs for further details.

Connector also uses the `log` section which is common to T-Server. See, The CSTA T-Server for Connector Deployment Guide wiki for full details of these options.

Any

(Anything can be used for the option name(s) in this section.)

Default Value: None
Valid Value: Perl-compatible regular expression (PCRE)
Changes Take Effect: Immediately

For detailed information about PCRE syntax, please refer to http://pcre.org.

A PCRE with or without string sub-patterns. If the expression contains a parenthesis-delimited `capture` sub-expressions, these sub-string matches are masked. Otherwise, the remainder of the string after the matching expression are masked. To define a sub-string that is a part of matching expression, but is not a part of the sensitive data, a `non-capturing` sub-expression can be used and is denoted by `(?:)`.

For example, the pattern: `User (?:PIN|pin|account)` can be used to mask everything after the full match of the word `User` followed by `PIN`, `pin` or `account`, but these words themselves are not masked. A case insensitive match can be forced by prefixing the expression with the following PCRE option: `(?i)`.

Examples:

```
[Pp]assword:? *
```

Masks the following: everything after the word `password`, the possible colon and any whitespace up to the end of the line. There are no sub-patterns, so everything is masked after a greedy algorithm match of the whole pattern.

For example:

```
User password (syspw) becomes
User password *******
```

```
[dD]igits.*"([^"])"
```

Masks everything between the double quotes that is preceded by `digits`.

For example:

```
collectedDigits: numberString "773645" becomes
collectedDigits: numberString "******"
```

```
(?i)account.*(\d{4})\d{2}-(\d{3-5})\d+
```

The account consists of two parts:

1. a 6-digit sort code
2. a dash and a variable-length number which is four digits or more

The pattern inhibits the first four digits of the sort code (first sub-pattern), up to the five first digits of the account number (second sub-pattern) but shows the last two digits of the sort code and at least one last number of the account number, or all of the last numbers after the initial five digits that are suppressed.

Matching of the word, `account`, is case-insensitive due to the preceding option.