Pega Call

7.31

Developer's Guide for Server-side Integration

Trademarks

For information about the third-party software that is delivered with the product, refer to the third-party license file on your installation media that is specific to your release.

Notices

This document is the property of:

Pegasystems Inc.
One Rogers Street
Cambridge, MA 02142-1209
USA
Phone: 617-374-9600
Fax: (617) 374-9620
www.pega.com

DOCUMENT: Pega Call Developer's Guide for Server-side Integration
SOFTWARE VERSION: 7.31
UPDATED: November 2017

CONTENTS

Overview

Pega Call provides seamless integration with third-party computer-telephony integration (CTI) systems. Using OpenCTI architecture, you can create integrations between Pega Call and telephony systems.

Note: If you want to integrate CTI server using client-side integration based on web technologies such as JavaScript, XMPP, see the latest version of OpenCTI Desktop Developer's Guide on PDN.

Prerequisites

- Before you start, make sure you understand the following information:
  - Telephony and contact center terminology and architecture. For an overview, visit the Pega Call page on PDN.
  - Pega Call integration.
  - Knowledge on Pega Integration techniques. For more information, see Integration Connectors landing page.
- Knowledge of the third party telephony system and the API used to integrate with it.
- Install the latest version of Pega Call in your system environment. For more information, see the Pega Call Installation guides on PDN.
- Verify that you have appropriate access to the CTI server that you need to connect to with the following:
  - Network Connectivity
  - Lab environment with telephony and contact center configured, for example, Phones, Users, and queues.

Getting started with OpenCTI

The OpenCTI interface in Pega Call provides a means to create adapters that can be used to integrate with Computer Telephony Integration (CTI) systems. This adapter plugs into the existing Pega Call framework and provides a bridge between the Pega Call framework and the third party CTI server.

OpenCTI adapter

Use OpenCTI to integrate Pega Call with additional CTI servers.

The OpenCTI adapter is a modular component that consists of a collection of rules that are used to send requests to the CTI server and receive events from the CTI server.

An OpenCTI adapter consists of two layers:

- OpenCTI Interface
- OpenCTI Implementation

OpenCTI Interface

This layer defines the integration required and acts as an interface between Pega Call and the CTI server. The interface layer is a template for creating implementations and includes some helper methods and an event framework to facilitate the processing of events.

OpenCTI implementation:

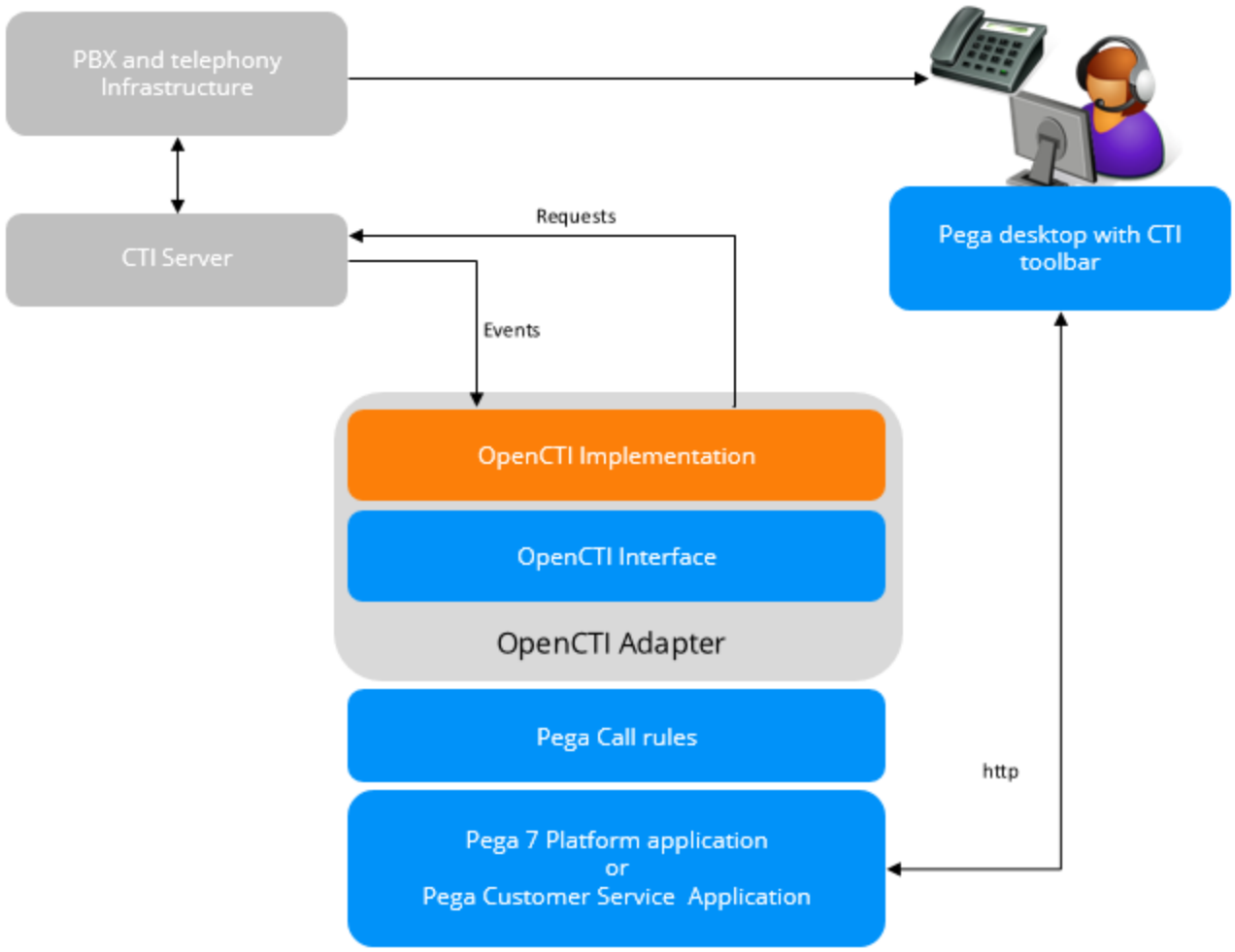
This layer implements the OpenCTI interface to communicate with the CTI server. In this layer, the OpenCTI developer creates or extends the rules to enable Pega Call to send requests and receive events.

In the given architecture diagram, the following components are provided by Pegasystems:

- Pega Call rules
- Pega 7 Platform application or Pega Customer Service application

- Pega desktop with a CTI toolbar

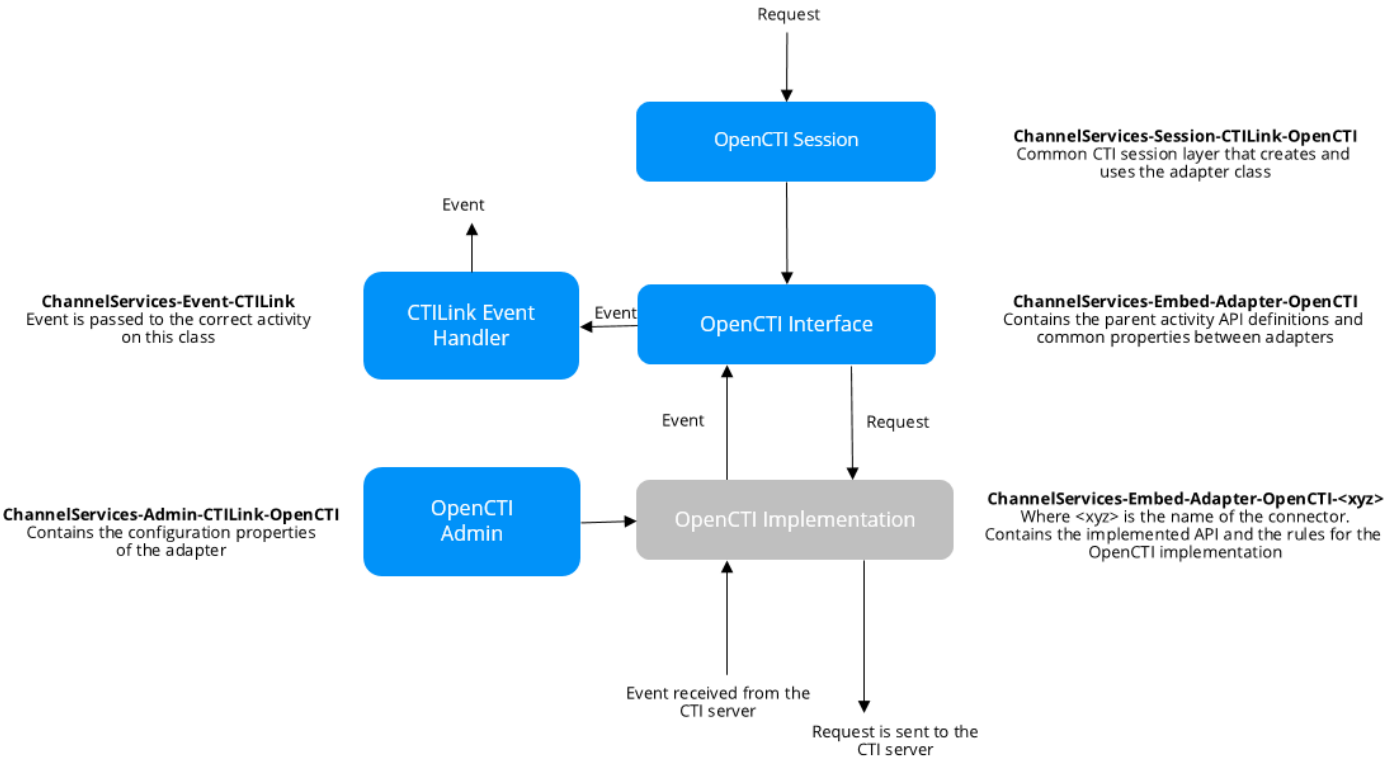
OpenCTI adapter architecture



Understanding the OpenCTI class structure

Each specific OpenCTI implementation requires a separate OpenCTI adapter class. The implementation class is *ChannelServices-Embed-Adapter-OpenCTI-<xyz>* where <xyz> is the unique name for the connector. This class contains Pega 7 Platform Integration Connectors, rules and properties that can used to implement the connector. To create a new OpenCTI adapter, start by creating a new class that inherits the *ChannelServices-Embed-Adapter-OpenCTI* parent class.

The classes in the OpenCTI architecture are explained diagrammatically as follows.



The following classes are a part of the OpenCTI architecture.

OpenCTI Session - ChannelServices-Session-CTILink-OpenCTI

This class provides the request interface from the CTI toolbar and creates and uses the adapter implementation class for example, *ChannelServices-Embed-Adapter-OpenCTI-<xyz>*.

OpenCTI Adapter - ChannelServices-Embed-Adapter-OpenCTI

This class acts as the base class (Interface) for OpenCTI adapters providing the common API activities. It also provides common properties and rules for the adapters and performs event handling to normalize the event for the CTI toolbar.

CTILink Event Handler - ChannelServices-Event-CTILink

This class formats and sends the event to the CTI toolbar.

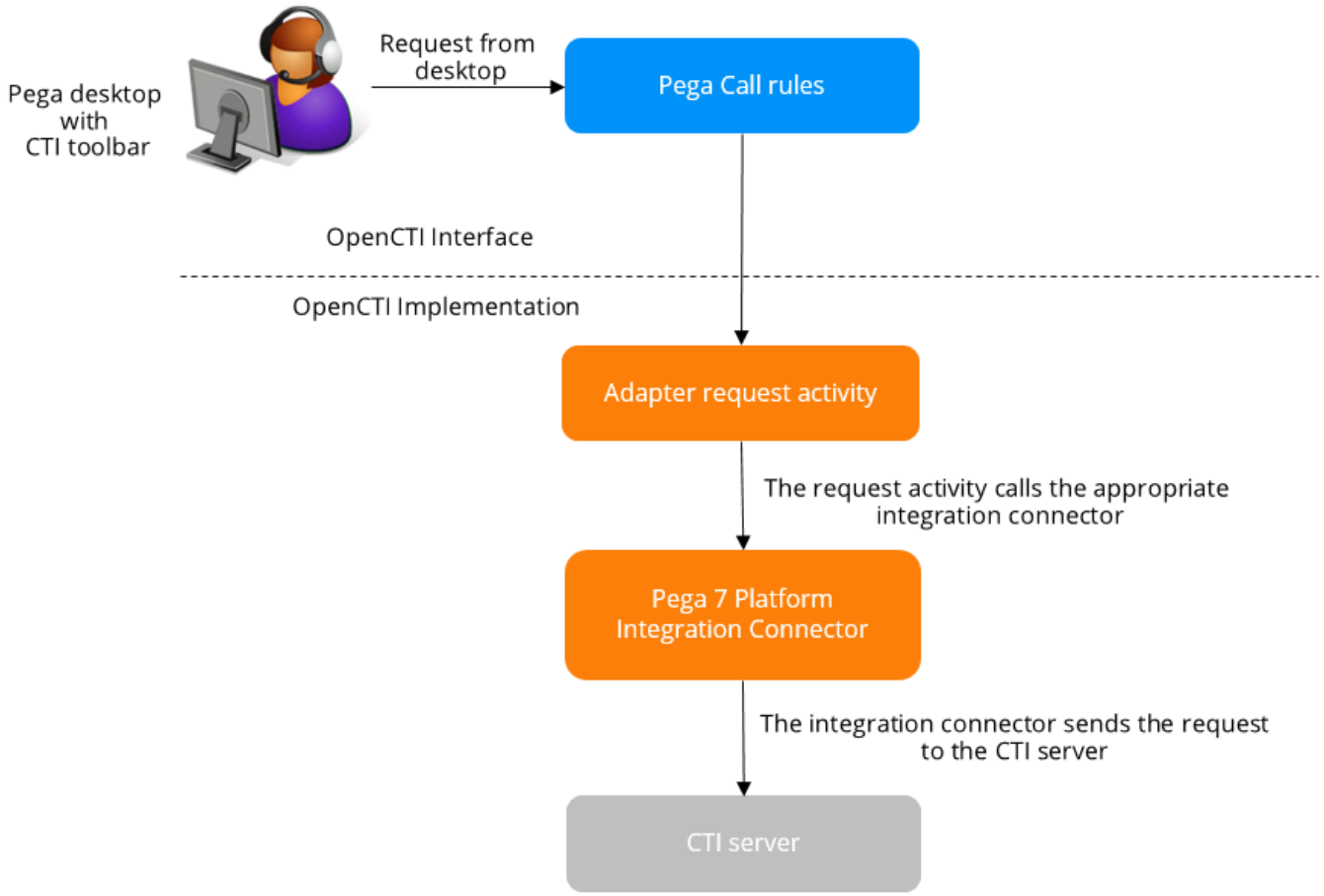
OpenCTI Admin - ChannelServices-Admin-CTILink-OpenCTI

Provides the configuration properties required by the adapter.

Understanding how Requests flow to the CTI Server

Requests are actions that the user performs on the CTI toolbar such as answer a call or hang up a call. For OpenCTI, requests are sent from the CTI toolbar to the OpenCTI interface layer which passes the request to the OpenCTI adapter layer. The adapter implementation is responsible for utilizing a Pega Integration Connector that sends the request to the CTI server.

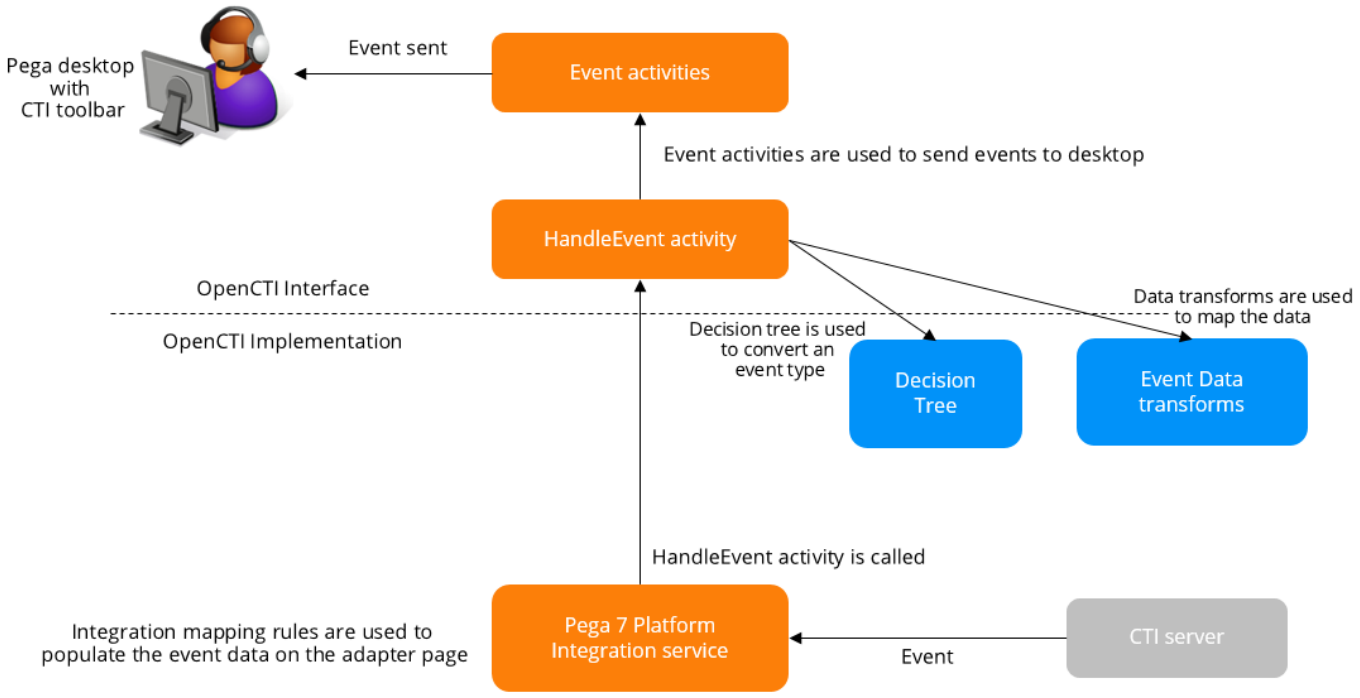
The following flow chart gives you a high level overview of the request flow.



The implemented request activity formats the request data and uses a Pega 7 platform Integration Connector rule to send the request to the CTI server. For example, if the CTI server provides a REST API, the implementation request activity formats the request data and calls the REST connector to forward the request to the CTI server.

Understanding how to handle Events from the CTI server

Events are sent from the CTI server to indicate a change in status either of the extension or of the agent. The received CTI server events drive the CTI toolbar and the received events are converted into a format that Pega Call understands, and then sent to the desktop.



The OpenCTI developer builds integration services to process events received from the CTI Server and to convert them into Pega Call events. Once the CTI server receives the event, the Integration Service uses the Integration Mapping rules to parse the data from the event into the corresponding properties on the adapter class.

For example, a Pega web service notification from a CTI server notifies Pega that a call is now connected on the operator's phone. The Integration Service is a point of entry for all events into the system.

Implementing OpenCTI

The Pega Customer Service interaction portal includes a telephony toolbar with three modes of configuration. The modes refer to the set of telephony capabilities presented to the user.

Simple telephony mode

This mode is intended for users (CSR) who use a different tool for call control operations (for example, placing calls on hold, transferring calls) and agent-state management (making themselves Available or Unavailable for customer calls). The user receives new call notifications (screen pop or automatic interaction start) when calls arrive at their phone. Agent state management and call control features are not available on the Pega desktop.

Simple telephony with Agent state management mode

This mode is intended for users who use a separate phone (hardware or software) for call control operations, but need to perform agent-state management from their Pega desktop. When this mode is selected, the user can manage their agent state (Available or Unavailable) from the Pega desktop. Call control features are not available from the Pega desktop.

Full telephony mode

This mode is intended for users who perform all telephony functions from the Pega desktop. When this mode is selected, the user can control calls (for example, Hold, Retrieve, Transfer, Hangup) on their phone and manage agent state from their Pega desktop. In addition, they receive call notifications the same way as simple telephony mode.

Follow these steps to implement OpenCTI based on your mode preferences.

- Implementing Full telephony

Creating your application context

Before you start configuring the OpenCTI adapter, you must create the following items to use the OpenCTI adapter.

1. Create a new application using the New Application wizard. For more information, see Creating an application.
2. Add the *PegaFW-CTI* ruleset to your application ruleset to use the Pega Call rules. If you have built your implementation application on the Pega Call application you can skip this step.
3. Create an access group that points to your implementation application.
4. In the **Explorer** panel, click **Records**, and then click **Integration-Resources > Service Package**.
5. Search for **OpenCTI** service package and select to open the record.
6. In the **OpenCTI service package** page, modify the **Service access group** to your access group.

Configuring the OpenCTI adapter

To use the OpenCTI adapter, you must perform the following steps:

1. Create the OpenCTI adapter infrastructure for the implementation class.
2. Create the OpenCTI link definition to point to the implementation class that you created.

Creating the OpenCTI implementation class

Create a new OpenCTI Implementation class extending the ChannelServices-Embed-Adapter-OpenCTI class provided by Pega Call. This class contains properties and activities required for the adapter implementation.

Creating the OpenCTI link

Create an OpenCTI definition to connect your adapter to the Pega CTI toolbar and to provide configuration data for the adapter.

1. In Designer Studio, from the **App explorer**, open the **ChannelServices-Admin-CTILink-OpenCTI** class.
2. Click **Create** to create a new instance of this class.
3. Enter the Short description and the Link Definition Name the In the Create OpenCTI CTI Link page.
4. Click Create and open.
5. On the **LinkConfiguration** tab, select the newly created class from the Implementation Class list.
6. Add the properties as key value pairs that you need for configuring your instance.

Initializing the adapter

Use Pega 7 Platform Integration resources such as REST or SOAP or JAVA connectors and services to send requests to the CTI server to authenticate and subscribe for events.

Note: The *OpenAdapter* and the *CloseAdapter* activities are called only once per user to allow the adapter to do any adapter specific setup or shutdown.

OpenAdapter activity

The OpenAdapter activity is invoked when the user (CSR) attempts to log in to the toolbar. It is used to perform any initialization that is required. You can override and implement this activity to perform any initialization required for your adapter. However, if you do not require initialization for each user then you need not implement this activity.

1. In Designer Studio, search for the **OpenAdapter** activity in the OpenCTI adapter class.
2. Copy the activity into the implementation class created for the new adapter. This activity will be called when the adapter is first used.
3. In the activity page, add steps that is required for the adapter.

CloseAdapter activity

Optionally, you can use CloseAdapter activity to perform any clean up activity on the adapter.

1. In Designer Studio, search for the **CloseAdapter** activity in the OpenCTI adapter class.
2. Copy the activity into the implementation class created for the new adapter.
3. In the activity page, add any clean-up or shut-down steps that is required for the adapter.

Sending Requests to the CTI server

All requests from the CTI Toolbar are sent to the CTI Server using the Pega 7 Platform integration connectors that can be invoked from the activities based on the type of request. These connectors are implemented by an OpenCTI developer to pass the request information in a CTI Server specific format.

For new adapter implementations, copy the request API activities from the parent OpenCTI adapter class into the implementation class.

For example, if you want to implement an Answer request, copy the AnswerCall activity in the ChannelServices-Embed-Apapter-OpenCTI class into the adapter implementation class.

Subscribing to events

Events are sent from the CTI server to indicate a change in the status either of the extension or of the agent.

Implement the following activities in the OpenCTI Implementation class to determine the action you want to perform on the event.

SubscribeForEvents activity

The SubscribeForEvents activity is called to send a request to start an event subscription for a device when a user tries to monitor an extension.

1. In Designer Studio, search for the **SubscribeForEvents** activity in the OpenCTI adapter class.
2. Copy the activity into the implementation class add the Pega 7 Platform integration connector to subscribe for the event for a device.
3. In the activity page, add any steps that is required for the adapter to subscribe for events for the user identified. For more information on the properties, see the property table.
4. If the subscription of the device fails, then an error should be returned. For more information, see Handling Errors.

The following are the properties available to the SubscribeForEvents activity. You can see these properties on the adapter step page of the clipboard.

Property Name	Description

DN	The device number on which the event is subscribed
AgentID	The agent ID logged in to the extension
AgentPassword	The password configured for that agent ID

Note: Some CTI servers do not need **AgentID** and **AgentPassword** for monitoring. In this case, the OpenCTI developer can ignore these properties when subscribing for events.

UnSubscribeForEvents activity

This activity is called to send the request to stop the event subscription for a device.

1. In Designer Studio, search for the **UnSubscribeForEvents** activity in the OpenCTI adapter class.
2. Copy this activity to the implementation class and add the Pega 7 Platform integration connector to unsubscribe for the event for a device.
3. If the unsubscribe event of the device fails, then an error should be returned. For more information, see Handling Errors.

The following are the properties included in the UnSubscribeForEvents activity. You can see these properties on the adapter step page of the clipboard.

Property Name	Description
DN	The device number on which the event is subscribed
AgentID	The agent ID that is logged in to the extension

Handling events from the CTI server

The **HandleEvent** activity converts the incoming event data to Pega format. As part of the OpenCTI framework, Pega provides data transforms, decision trees, and activities to achieve this conversion. Override these rules in the adapter layer and modify them according to the needs of your implementing adapter.

When you receive events from a service, the event data must be mapped to a clipboard page using a parse rule such as XML or JSON parser.

After you map the event data to the clipboard page, call the **HandleEvent** activity of *ChannelServices-Embed-Adapter-OpenCTI*, where these properties are mapped to the CTIEvent (*ChannelServices-Event-CTILink*) clipboard page.

The **HandleEvent** activity starts the processing of an event. The processing of an event consists of the following functions:

1. Retrieving the user's device presence data using the RetrieveDevicePresence activity.
2. Identify the corresponding Pega provided OpenCTI event using the SetEventType decision tree.
3. Filtering the event based on the telephony mode used, and that are not rendered by the CTI toolbar using the FilterEvent activity activity.
4. Filtering the event based on device capabilities using the FilterEventsOnDeviceCapabilities decision tree.
5. Mapping events to a form recognized by the Pega Call framework using theConvertEvent activity.
6. Sending the event to the desktop.
7. Specialized processing of the event after sending the event to desktop using the EventPostProcessing activity.

The activities listed in the following section are used to process the events. These activities use the OpenCTI adapter class rules to receive any format of event data and modify it into the CTI toolbar's required data structure.

Copy the following rules from the OpenCTI base adapter class and implement them in your implementation class.

RetrieveDevicePresence activity

This activity is used in the retrieval of the user's workstation ID and client handle from the user's presence object. The contextual data retrieved is used in building the event in a Pega format and to deliver the event to the CTI toolbar. Override the **PopulatePresenceKeys** data transform to populate the OpenCTI adapter implementation presence keys if the implementation is using a different key from the one provided by the OpenCTI interface.

IdentifyEvent activity

This activity identifies which Pega Call event corresponds with the event received. This activity uses the decision tree **SetEventType** to identify the event. This should only be overridden if a different decision rule type is needed.

SetEventType decision tree

This rule normalizes the CTI server event agent state or call state into the agent or call state used by the CTI toolbar. You can copy this decision tree into the adapter class and modify it to identify the corresponding Pega Call event.

For example, if the CTI server event agent state is Talking then add an if condition to this rule `if AgentState = "Talking" then return "Busy"`. When modifying existing if conditions,

do not change the return value. The return values listed in the decision tree are the states used by the CTI toolbar.

FilterEvent activity

This activity is used to filter any events that are not required by the Pega Call implementation. This activity uses the *FilterEvent* when rule to update the **SendEvent** property. Override this when rule to set the **SendEvent** property to either True or False depending on the event type. By default, this when rule returns False.

FilterEventsOnDeviceCapabilities decision tree

This decision tree is used to filter call or agent state events based on the CTI toolbar capabilities.

ConvertEvent activity

This activity maps properties to the Pega Call event page with data for each Pega Call event using **PopulateEventData** data transforms. These data transforms are called from the **ConvertEvent** activity.

PopulateEvent data transforms

These are a collection of data transforms that represent the events that are recognized by the Pega Call framework. These data transforms copy the data from the OpenCTI adapter implementation page to the CTI event page. Copy these data transforms from the OpenCTI interface class into the OpenCTI implementation adapter class and map the values of the properties on the source side of the data transform with the corresponding property from the adapter.

Caution: Do not change the contents of the event name property.

There may be instances where the event data model from the CTI server system does not fit into the OpenCTI event model when using the Decision Tree and Data Transforms listed. In these cases, there are two activities which can be copied and overridden.

EventPostProcessingactivity

This activity allows any processing that might be required after the event is sent.

Handling Errors

If an error occurs during a CTI toolbar request, such as hold, answer, and make call, an error message is displayed on the user's desktop.

When an error occurs on the request in the OpenCTI adapter implementation:

1. Using the Activity-Set-Status method in the activity, handle the error message.
2. Copy the error message to **pyStatusMessage** and **pyPegaCTIError**.
3. Set the **Severity** to Fail and enter the error message in the **Message** field. The error message in the **Message** field is displayed on the user's desktop.
4. Exit the request activity after the error is encountered and the Activity-Set-Status is performed. Any extra error information can be sent to the logs.

For events received from the CTI server, errors are written to the Pega 7 Platform logs and the event processing is aborted.

Implementing Simple Telephony

Simple telephony mode is intended for users who wish to receive screenpops, but use a different tool for call control operations (for example, placing calls on hold, transferring calls) and agent-state management (making themselves Available or Unavailable for customer calls). For Simple Telephony, an OpenCTI developer needs to handle the below request.

- Handling Login requests
- Handling incoming calls and call treatment
- Implementing call answering

Handling Login requests

Login requests enables monitoring a device and to subscribe for events for a device. You must create the Pega 7 Platform integration connectors to subscribe for events from the CTI server.

You can use **LoginAgent** activity based on the implementation and the CTI server. However, for Simple Telephony with Agent state management mode, you must use **LoginAgent** activity. For more information on subscribing to events, see Subscribing to events.

To handle an agent login request in the adapter:

1. In Designer Studio, search for the **LoginAgent** activity in the OpenCTI adapter class.
2. Copy the **LoginAgent** activity into the adapter class.

3. In the activity page, add the required steps to call the Pega 7 Platform integration connector to send the login request to the CTI Server. The data required for the login request such as **Agent ID** and **Agent Password** can be retrieved from the adapter clipboard.

Handling log out requests

The log out request unsubscribes for event from the CTI server for the device logging out, so that OpenCTI interface will stop receiving any events from the server. When you send a log out request, the following two activities are called.

CloseAdapter activity

This activity is called when a user logs out of the telephony device. For more information, see CloseAdapter activity under Initializing the adapter chapter.

UnSubscribeForEvents activity

This activity is called to send the request to stop the event subscription for a device. For more information, see UnSubscribeForEvents activity under Initializing the adapter chapter.

Handling incoming calls and call treatment

The incoming call event from a CTI server should be passed to the Pega 7 Platform. To achieve this, you must first create a Pega 7 Platform integration service which can be invoked by the CTI server or by a third party connector.

For more information, see Handling events from the CTI server.

1. In Designer Studio, search for **SetEventType** decision tree.
2. Copy the decision tree to the OpenCTI adapter class.
3. Modify the decision steps to map the event names appropriately.

An event is required to signify a new call arrival and is called alerting, ringing, or offering.

For an incoming call, the **EventName** is mapped to 'Offering'. This mapping is done using **SetEventType** decision tree. For Simple Telephony mode, we require only Alerting or Ringing or Offering or Delivered events based on the CTI server which indicates a new inbound call.

Passing data to Pega clipboard

Any Call state event received by the CTI Server is passed using a web service like SOAP or REST services and are mapped to an event page. This event page is the OpenCTI Implementing class type.

Map the incoming data from the CTI Server to the clipboard page using a service activity. Use the Service activity to copy the incoming data to the corresponding property on a clipboard page of the Service rule. This page acts as primary page for Service activity.

Passing the basic call Info details

You can pass the basic call info details to the clipboard page using the service activity. The following are the basic CTI properties that needs to passed with a call event. You can see these properties on the clipboard.

Property Name	Description
pyCallState	Stores the call state such as hold, retrieved.
pyCallType	Stores the call type like Internal or Consult used in invoking call treatment rules from the adapter page to the clipboard. For more information in the call types, see Call type details.
pyCallID	Unique ID of a call.
pyDNIS	Stores the extension to where the call was connected.
pyANI	Stores the extension from where the call arrived.
pyOtherDN	Stores the values of other devices from which the call is being placed.
pyEventName	Stores the Offering and normalizes the Pega Call event name based on the call state.

Passing the call variables

The call variables received from the CTI server are stored as property-value pair in *pyCallVariables* (ChannelServices-Event-CTILink) in the class **ChannelServices-Request-CallOptions-OpenCTI**. You can see this property on the clipboard.

Property Name	Description
pyCallVariables	Set the call variables passed in any call state event to pyCallVariables

Call type details

Pega Call supports a call type that can be used to circumstance call treatment rules. The **pyCallType** property on the call page specifies call types, which include:

INBOUND

For inbound calls.

CONSULT

For Consultation and transferred calls.

OUTBOUND

For calls placed from the contact center or the CSR's phone to an external phone number.

INTERNAL

For calls that are internal to the call center, such as calls from one CSR to another.

Implementing call answering

With Pega Call you can initiate an automatic call process to answer calls on behalf of the customer service representative (CSR).

This feature can be used only when OpenCTI implementation configures Call Treatment rules to automatically answer the call when an interaction is presented or started. The following are the required steps to implement automatic call answering.

1. In Designer Studio, search for the **AnswerCall** activity in the OpenCTI adapter class. This activity is used to answer the specified call on the CSR's extension.
2. Copy the **AnswerCall** activity into the OpenCTI adapter implementation class.
3. In the activity page, add the required steps to call the Pega 7 Platform integration connector to send the answer call request to the CTI server.
4. If the call to the CTI server results in an error, handle it as discussed in the Handling Errors.

You can see this property on the primary page of the clipboard.

Property Name	Description
CallId	The identifier used to track calls and handle call features.

Pega Call agent states

When a CSR logs into an automatic call distributor (ACD) they need to indicate if they are available or unavailable to answer calls from customers. Pega Call enables the CSRs to perform this operation from their CTI toolbar.

Note: Implementing agent state management along with Simple Telephony allows you to support Simple Telephony with Agent state management mode.

Pega Call supports the following major Agent states:

Login

Agent is logged into Automatic Call Distributor (ACD) and the agent can move to either Available or Unavailable.

Available

Agent is ready to accept calls from the Automatic Call Distributor (ACD) or Call Queue.

After Call Work

This state is used to finish up any work related to the last call agent attended. No queued calls are routed to the agent while in this state.

Unavailable

While in this state, the agent does not receive any calls from the ACD/Call queue.

Logout

Agent is logged out of the ACD/Call Queue and does not receive calls from the ACD/Call Queue

Note: The Unavailable and Logout requests support the use of reason codes.

Sending Agent-state requests from the desktop

Copy the following activities from the *ChannelServices-Embed-Adpter-OpenCTI* class to OpenCTI implementation class, so that Pega Call can invoke these activities to send request to CTI Server.

- LoginAgent activity
- SetAgentReady activity
- SetAgentNotReady activity
- LogoutAgent activity

The **AgentOptions** page stores options such as reason code that is used while setting agent state to **NotReady** or **Logout**. If other agent options are required to send a request to CTI server, they can be added using a relevant data transform (with name same as agent state change request) in the *ChannelServices-Request-AgentOptions* class.

The OpenCTI developer must translate the OpenCTI interface's agent state (Pega Call supported agent state) to the equivalent Agent State supported by the CTI Server.

If there is an error in the request, handle it in the activities using *Activity-Set-Status* method by changing the **Severity** as Fail. The OpenCTI interface passes this error to the Agent's desktop.

LoginAgent activity

This activity is used to log in the agent into a queue. When this activity is enabled, the agent receives calls from the queue and can manage their agent state to either Ready or Not Ready state. In this activity you can also specify the queue to which you want the agent to log on.

1. Use the **AgentId** and **AgentPassword** properties on the adapter page. The OpenCTI adapter copies the extension details to the device number.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI Server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

Unlike in other Pega OpenCTI activities, when this log in request is successful, no statuses or properties would be updated. When there is an error on the request, the status is set to Fail and the status message contains an error message.

The following are the properties of this activity that can be seen on the primary page of the clipboard.

Property Name	Description
Queue	Specifies which queue to log the agent into. You can configure to agents to log in and log out of specific queues. If this property is left blank, then the agent will be logged into the queues specified by the CTI server or Automatic Call Distributor (ACD) queue. This is an optional property
AgentOptions	Property page that contains the reason code for logging the agent out. This is an optional property.
AgentID	The agent ID on the ACD or Queue. This is a required property that is populated on the page during login.
AgentPassword	The password for the Agent on the ACD or Queue. This is a required property that is populated on the page during login.

SetAgentReady activity

This activity is used to set the agent state to Ready. When the state is set to Ready, then the agent starts receiving calls from the queue. However, the agent must be logged into at least one queue before this method is invoked.

1. On the **AgentState** property page, set the Ready equivalent state of the CTI Server.
2. Set other CTI server specific properties such as a Queue or a work mode that are to be sent with the Agent State change request.
3. Invoke the appropriate request to the CTI Server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

Unlike in other Pega OpenCTI activities, when this request is successful, no statuses or properties would be updated. When there is an error on the request, the status is set to Fail and the status message contains an error message.

You can see these properties on the primary page of the clipboard.

Property Name	Description
Queue	Specifies which queue to log the agent into. You can configure to agents to log in and log out of specific queues. If this property is left blank, then the agent will be logged into the queues specified by the CTI server or Automatic Call Distributor (ACD) queue. This is an optional property
AgentID	The agent ID on the ACD or Queue. This is a required property that is populated on the page during login.

SetAgentNotReady activity

This activity is used to set the agent state to a Not Ready state. When the state is Not Ready, then the agent does not receive calls from the queue. However, the agent must be logged into at least one queue before this method is invoked.

1. On the **AgentState** property page, set the **NotReady** equivalent state of the CTI Server. If the agent selects a reason code, then the OpenCTI adapter sets this value on the AgentOptions embed page on *pyOpenCTI.pyReasonCode* property.

   In case of After Call Work, the WorkMode property on Adapter page will be set to AFTER_CALL_WORK by OpenCTI adapter. OpenCTI Implementation should set equivalent state of CTI Server on AgentState property
2. Set other CTI Server specific properties that need to be sent in Agent State change request.
3. Build a request appropriate for your connector.
4. Invoke the appropriate request to the CTI Server using Pega 7 Platform integration connector.
5. Identify errors and handle it as discussed in Handling Errors.

When this request to make the agent not ready is successful, no status and no properties are updated. When there is an error on the request, the status is set to Fail and the status message contains an error message.

You can see these properties on the primary page of the clipboard.

Property Name	Description
Queue	Specifies which queue to log the agent into. You can configure to agents to log in and log out of specific queues. If this property is left blank, then the agent will be logged into the queues specified by the CTI server or Automatic Call Distributor (ACD) queue. This is an optional property
AgentID	The agent ID on the ACD or Queue. This is a required property that is populated on the page during login.
AgentOptions	Property page that contains the reason code for logging the agent out. This is an optional property.

LogoutAgent activity

This activity is used to set the agent state into a logged out state. When the state is Log out, then the agent no longer receive calls from the queue and will not be able to change to any other states. However, the agent must be logged into at least one queue before this method is invoked.

1. On the **AgentState** property page, set the Logout equivalent state of the CTI Server. If the agent selects a reason code for log out, then OpenCTI adapter will set this value on the AgentOptions embed page on pyOpenCTI.pyReasonCode property.
2. Set other CTI Server specific properties that need to be sent with the Agent State change request.
3. Build a request appropriate for your connector.
4. Invoke the appropriate request to the CTI Server using Pega 7 Platform integration connector.
5. Identify errors and handle it as discussed in Handling Errors.

When this request to log the agent out is successful, no status and no properties are updated. When there is an error on the request, the status is set to Fail and the status message contains an error message.

You can see these properties on the primary page of the clipboard.

Property Name	Description
Queue	Specifies which queue to log the agent into. You can configure to agents to log in and log

	out of specific queues. If this property is left blank, then the agent will be logged into the queues specified by the CTI server or Automatic Call Distributor (ACD) queue. This is an optional property
AgentOptions	Property page that contains the reason code for logging the agent out. This is an optional property.
AgentID	The agent ID on the ACD or Queue. This is a required property that is populated on the page during login.

Handling Agent state events

Map the incoming data from the CTI Server to the clipboard using service activity. The Service activity copies the incoming data to corresponding property on the clipboard page mentioned in Service rule. This page acts as primary page for the Service activity.

After mapping event data to the clipboard page, call the **HandleEvent** activity of *ChannelServices-Embed-Adapter-OpenCTI*, where the properties are mapped to CTIEvent (ChannelServices-Event-CTILink) clipboard page. HandleEvent activity converts the incoming event data to Pega understandable format. As part of the OpenCTI framework, Pega provides Data Transforms, Decision Trees, and Activities. You can override them in the Adapter layer and modify them according to the needs of your implementing adapter.

For more information see, Handling events from the CTI server.

Implementing Full telephony

Full telephony mode is intended for users who perform all telephony functions from the Pega desktop. When this mode is selected the users can control calls on their telephony device and manage agent state from their Pega desktop. In addition, they receive call notifications the same way as simple telephony mode.

To implement full telephony you must:

- Configure simple call control
- Configure agent state
- Configure complex call control

Simple call control functions

With simple call control activities, you can perform the basic call control functions such as answering a call, putting a call on hold, and making a call. The CTI server receives the simple call request, applies the appropriate settings, and returns the event that instructs the CTI toolbar on how to handle the call and any additional call treatment to apply.

The simple call controls provide the following functions and each simple call state request calls an activity to send the request to the CTI server.

Answering a call

This function uses the AnswerCall activity to answer an incoming call from the line which is in an altering or ringing state. CTI server uses the *CallId* and *DN* (DeviceNumber) properties on the adapter page to answer the call. Use the OpenCTI framework to set the *CallId* to the adapter page. If the *CallId* property is -1, then the OpenCTI implementation must search for the call which is in the ALERTING state from the CTI Server and map its to the *CallID* of the adapter page.

Putting a call on hold

This function uses the HoldCall activity to hold and activate a call. CTI server uses the *CallId* property on the adapter page to identify and place the call on hold.

Initiating a call (Make call)

This function uses the MakeCall activity to place or initiate a call. The CTI server uses the *DestinationNumber* property to set the destination number on the adapter page.

Concluding a call (Hang up)

This function uses the HangUpCall activity to end a call.

Retrieving a call from a hold state

This function uses the RetrieveCall activity to release a call from a hold state to active state.

Sending a DTMF (Dual tone multiple frequencies) tone when the call is on hold

This function uses the SendDTMF activity to send DTMF tones while the call is put on hold.

AnswerCall activity

The AnswerCall activity is called to initiate the call answer state to an incoming call while in ringing state. This activity uses the CallID and the DN (Device Number) properties to place an answer call request to the CTI server. Before initiating this activity, ensure that you set the CallID and DN properties on the adapter page. If the CallID is not set prior to initiating this activity, then the CTI

server will be queried to find a call in the ringing or alerting state. If no calls are found in the ringing state, then an error is returned to the CTI toolbar.

Follow the given steps to implement this activity.

1. Map the *CallID* and *DN* properties of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

HoldCall activity

The HoldCall activity is used to put an ongoing call on hold. This activity uses the HeldCallId and the CallId properties to place the hold call request to the CTI server. Before initiating this activity, ensure that you set the HeldCallId and CallId properties in the primary page of the clipboard. If the CallID is not set prior to initiating this activity, then the CTI server will return an error to the CTI toolbar.

Follow the given steps to implement this activity.

1. Map the *HeldCallID* and *DN* properties of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

MakeCall activity

The MakeCall activity is used to place a new call to a destination number specified on the adapter page. The call variables such as DestinationNumber are used in sending the new call request to the CTI server. If the call is successfully placed, then no data is returned from the CTI server. If the make call request is unsuccessful, then the CTI server will return an error to the CTI toolbar.

Follow the given steps to implement this activity.

1. Map the *DestinationNumber* property of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

HangUpCall activity

The HangUpCall activity is invoked to release an active call from an established state. This activity uses the DN (Device number) property to place the hang call request to the CTI server. If the call is successfully released, then the state of the call is changed to released state. If the hangup call request is unsuccessful, then the CTI server will return an error to the CTI toolbar.

Follow the given steps to implement this activity.

1. Map the *DN* property of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

RetrieveCall activity

The RetrieveCall activity is called to retrieve the call in a hold state to an active state. This activity uses the CallID and the DN (Device Number) properties to perform a retrieve call request to the CTI server. If the call retrieval is successful released, then the state of the call is changed to active state. If the request is unsuccessful, then the CTI server will return an error to the CTI toolbar.

Follow the given steps to implement this activity.

1. Map the *CallID* and *DN* properties of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.

3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

SendDTMF activity

The SendDTMF activity is used to send DTMF tones while the call is put on hold. If the request is unsuccessful, then the CTI server will return an error to the CTI toolbar.

Follow the given steps to implement this activity.

1. Map the *CallID* and *DN* properties of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

Complex call control functions

Using complex call control activities, you can perform the complex call control functions such as conferences, warm transfer, and blind transfer. The CTI server receives the requests, applies the appropriate settings, and returns the event that instructs the CTI toolbar to handle the call.

Conferences

Conference call is a call between a three parties, a customer and two customer service representatives. This function uses ConferenceInitiate activity, ConferenceCall activity and DropParty activity to initiate, set up, and drop a party from a conference call.

Warm Transfer

Warm transfer is a call between the customer and the CSR, where one CSR transfers an active call to another CSR. The second CSR gets the an alert, if accepted, the first CSR clicks complete transfer on the Call menu list, then the call is completely transferred to the second CSR. This function uses TransferInitiate activity and TransferCall activity to initiate a warm transfer call.

Blind Transfer

Blind transfer is a call transferred from one CSR to another CSR without alerting or waiting for acceptance. This function uses the BlindTransfer activity to allow the CSR to immediately transfer a call to another CSR.

ConferenceInitiate activity

The ConferenceInitiate activity is used to initiate a conference call. This activity requests a consultation call to another party which places the active call on hold. This activity uses the **CallID** and the **DestinationNumber** properties to initiate the request to the CTI server.

Follow the given steps to implement this activity.

1. Map the *CallID* and *DestinationNumber* properties of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

ConferenceCall activity

The ConferenceCall activity is used to place and complete a conference call. This activity uses the **CallID** and the **HeldCallId** properties to complete the request.

Follow the given steps to implement this activity.

1. Map the *HeldCallID* and *CallID* properties of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

DropParty activity

The DropParty activity is used if any one of the party in a conference call decides to drop off. This activity uses the **CallID** and the **Party** properties to initiate the request to the CTI server. Before initiating this activity, ensure that you have the **CallID** and **PartyToRemove** properties on the adapter page.

Follow the given steps to implement this activity.

1. Map the *CallID* and *Party* properties of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

TransferInitiate activity

The TransferInitiate activity is used to initiate a warm transfer call. This activity requests a consultation call to another party which places the active call on hold. This activity uses the **CallID** and the **DestinationNumber** properties to complete the request.

Follow the given steps to implement this activity.

1. Map the *CallID* and *HeldCallID* properties of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

TransferCall activity

The TransferCall activity is used to complete the call transfer from one CSR to another in a warm transfer call. This activity uses the **CallID** property to identify the call that is put on hold and the

ConsultCallId property is used to temporarily hold the caller details of the consultation call.

Follow the given steps to implement this activity.

1. Map the *CallID* and *HeldCallID* properties of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.

BlindTransfer activity

The BlindTransfer activity is used to transfer a call in the single step. This activity uses the **CallID** and the **DestinationNumber** properties to complete the request.

Follow the given steps to implement this activity.

1. Map the *CallID* and *DestinationNumber* properties of the OpenCTI adapter page to the corresponding request parameters of the Pega 7 Platform integration connectors.
2. Build a request appropriate for your connector.
3. Invoke the appropriate request to the CTI server using Pega 7 Platform integration connector.
4. Identify errors and handle it as discussed in Handling Errors.

For more information, see Integration Connectors landing page.