



Cross-Server Sign-In, Transfers, and Live Reply in HTTPS Networking

- [Cross-Server Sign-In, Transfers, and Live Reply in HTTPS Networking, on page 1](#)

Cross-Server Sign-In, Transfers, and Live Reply in HTTPS Networking

Introduction

This chapter describes the cross-server sign-in, transfer, and live reply features that are available between Cisco Unity Connection locations in an HTTPS network. Also included in this chapter are the procedures for turning on the cross-server features.

Overview of Cross-Server Sign-In, Transfer, and Live Reply

In order to limit replication traffic and keep the directory size manageable, only a subset of user information is replicated from the home location of the user to other networked locations. For this reason, only the home location of the user has information about call transfer settings, greetings, and other specific details for the user. In order for the location to properly handle calls destined for a user on a different location, it must hand off the call to the home location of the user. The purpose of the cross-server features is to make the user experience in a networked environment almost the same as in a single server environment, as shown in [Table 1: Cross-Server Features](#).

Table 1: Cross-Server Features

Feature	Description
Cross-server sign-in	Cross-server sign-in allows administrators to provide users who are homed on different locations with one phone number that they can call to sign in. When calling from outside the organization, users—no matter which is their home server—call the same number and are transferred to the applicable home server to sign in.

Feature	Description
Cross-server transfer	Cross-server transfer enables calls from the automated attendant or from a directory handler of one location to be transferred to a user on another location, according to the call transfer and screening settings of the called user.
Cross-server live reply	Cross-server live reply allows users who listen to their messages by phone to reply to a message from a user on another location by transferring to the user (according to the call transfer and screening settings of the called user).

Although the cross-server features are distinct features, they all use the same underlying functionality—an enhanced supervised call transfer:

1. The location on which a sign-in, transfer, or live reply originates puts the caller on hold and calls the receiving location by dialing a phone number designated as the cross-server dial string for the receiving location.
2. When the receiving location answers, the originating location sends a sequence of DTMF tones that identify the call as a handoff request.
3. The receiving location responds with a sequence of DTMF tones, and the originating location hands off the call to the receiving location for processing.

At this point the functionality is the same as if the call had originated on the receiving location.

In this chapter, an originating location is defined as a server (or cluster) that calls other locations. A receiving location is defined as a server (or cluster) that answers a cross-server call.

Cross-server dial strings are not synchronized between locations. Each originating location can be configured with a dial string for each receiving location. Note that if an originating location is configured for multiple phone system integrations, you must choose a dial string that all phone system integrations can use to reach the receiving location.

In case of a video call, when two Unity Connection locations are linked by an HTTPS link, then if a user from one Unity Connection location attempts to sign-in to another Unity Connection location, the call is downgraded to audio.

Search Space Considerations for Cross-Server Sign-In, Transfers, and Live Reply

When a user dials the pilot number of a Unity Connection location that is not the home server of the user, the answering location processes the call according to its call management plan. A search space is assigned to the call by the first call routing rule that the call matches. At each subsequent processing step, the search scope of the call may change. Unity Connection uses the search space that is assigned to the call at the point that the call reaches the Attempt Sign-In conversation to identify which user is trying to sign in. If a user calls from an extension that is in a partition that is not a member of this search space, the call is not identified as coming from the user. If the extension of the user overlaps with an extension in another partition that also appears in this search space, the call is identified as coming from the first object that Unity Connection finds when searching the partitions in the order in which they appear in the search space. Check the direct routing rules on each Unity Connection location that handles incoming sign-in calls from remote users to determine the search space that is set by the rule or other call management object that sends calls to the Attempt Sign-In

conversation. If the partitions that contain remote users are not a part of this search space, cross-server sign-in does not work, even if it is enabled.

Also note that for cross-server calls from one Unity Connection location to another Unity Connection location, a mismatch between the search space that is applied to the call on the originating location and the search space that is applied on the receiving location can cause problems for cross-server sign-ins and cross-server transfers. A match could be made on the search scope on the originating location that cannot be made on a different search scope on the receiving location. For this reason, we recommend that you verify that the same search scope is configured on both originating and receiving locations. For example, call routing rules can be used to direct cross-server calls on the receiving location to the appropriate search space based on the cross-server dial string that is used to reach that location.

For cross-server live reply, as with any live reply attempt, a Unity Connection user can only call the sender if the sender is in a partition that is a member of the search space configured for the user.

Cross-Server Sign-In

The cross-server sign-in feature enables users who are calling from outside the organization to call the same number regardless of which server they are homed on, and they are transferred to the applicable home server to sign in. If you do not enable cross-server sign-in, users need to call the phone number of their home server to sign in.

The process for a cross-server sign-in call is as follows:

1. A user calls the server configured for cross-server sign-in. The user is identified by the calling number or is asked to enter his or her ID.
2. The server looks up the caller ID in the database to determine whether the account is homed on the local server or on a networked server.
 - If the user account is homed on the local server, the sign-in proceeds as usual.
 - If the user account is homed on another server, the conversation plays a “One moment please” prompt (if configured to do so), puts the user on hold, and calls the user home server using the same port that the user called in on. Note that if the user is calling from his or her primary or alternate extension, the “One moment please” prompt is typically the first prompt that the user hears.

When the receiving server answers, the originating server sends a sequence of DTMF tones that identifies the call as a cross-server sign-in.

3. The receiving location responds with a sequence of DTMF tones.
4. The originating location hands off the call to the receiving server for processing. The conversation on the receiving server prompts for the user password. At this point, the behavior is as though the user had called the receiving server directly.

The intended use of this feature is limited to users calling in from outside your organization. Although cross-server sign-in transfers internal calls to the home server, doing so for a large number of users increases the load on the servers. Therefore, user phones should always be configured so that the “Messages” or voicemail speed-dial button calls the home server of the user directly. When moving a user account from one server to another, update the phone system configuration for the user accordingly.

Task List for Enabling Cross-Server Sign-In

When you are configuring an HTTPS network, use the following task list to enable cross-server sign-in. The cross references take you to detailed procedures.

1. Determine which locations are the originating locations and which are the receiving locations for cross-server sign-in. Often a single location is designated as the originating location that all users call into from outside the organization, and all other location are designated as receiving locations; however, this does not have to be the case. A single location also may be both an originating location and a receiving location.
2. For each originating location, make a list of the phone numbers that the location must dial to reach the receiving location servers.



Note

You can enter only one dial string for each receiving location. If the originating location is configured for multiple phone system integrations, you need a dial string that all phone system integrations can use to reach the receiving location.

3. Configure each receiving location so that it can handle incoming cross-server handoff requests.
 - If the receiving location is a Unity Connection server, see the [Configuring a Unity Connection Receiving Location to Accept Cross-Server Handoff Requests](#).
4. For each originating location, enable the cross-server sign-in feature and enter the pilot numbers of the receiving locations from the list that you created in Task 2
 - If the location is a Unity Connection server, see the .
5. Test the cross-server sign-in functionality. See the [Testing Cross-Server Sign-In](#).

Procedures for Enabling Cross-Server Sign-In

See the following sections:

Configuring a Unity Connection Receiving Location to Accept Cross-Server Handoff Requests

By default, each Unity Connection server is configured to ignore cross-server handoff requests. To enable cross-server features, you must configure the receiving location to accept requests and also verify that the location routes incoming cross-server calls to a call handler. Do the following two procedures to configure each receiving Unity Connection location to accept handoffs. (Doing so allows the location to receive handoffs of all types—sign-in, transfer, and live reply.)

- [Steps to Configure a Cisco Unity Connection Receiving Location to Accept Cross-Server Handoff Requests](#)
- [Verifying Call Routing Rules are Set to Route Calls to a Call Handler Greeting](#)

Steps to Configure a Unity Connection Receiving Location to Accept Cross-Server Handoff Requests

- Step 1** In Cisco Unity Connection Administration, on a location that accepts cross-server handoffs for users who are homed on that location (the receiving location), expand **System Settings** > **Advanced**, then select **Conversations**.

- Step 2** Check the **Respond to Cross-Server Handoff Requests** check box.
- Step 3** Repeat the procedure on all remaining Unity Connection receiving locations.
-

Verifying Call Routing Rules are Set to Route Calls to a Call Handler Greeting

- Step 1** In Cisco Unity Connection Administration, on a location that accepts cross-server handoffs, expand **Call Management** > **Call Routing** and select **Direct Routing Rules**.
- Step 2** Select the display name of the routing rule that applies to incoming cross-server calls from originating locations.
- Step 3** Verify that calls that match the rule are routed to a call handler.
- Step 4** Repeat the procedure on all remaining Unity Connection receiving locations.
-

Configuring a Unity Connection Originating Location to Perform Cross-Server Sign-In Requests

By default, a Unity Connection location do not attempt to perform a cross-server sign-in for users homed on any other locations. Do the following procedure to enable cross-server sign-in on any Unity Connection originating locations.

Steps to Configure a Unity Connection Originating Location to Perform Cross-Server Sign-In Requests

- Step 1** In Cisco Unity Connection Administration, on a location that handles sign-in calls from remote users (the originating location), expand **Networking** and select **Locations**.
- Step 2** On the Search Locations page, select the Display Name of a remote location that accepts cross-server sign-in requests for users who are homed on this location (the receiving location).
- Step 3** On the Edit Location page for the receiving location, do the following to initiate cross-server features to this receiving location:
- To enable cross-server sign-in to the remote location, check the **Allow Cross-Server Sign-In to this Remote Location** check box.
 - Enter the dial string that this location used to call the receiving location when performing the handoff (for example, the pilot number of the home server).
- Note** You can enter only one dial string for each receiving location. If the originating location is configured for multiple phone system integrations, enter a dial string that all phone system integrations can use to reach the receiving location.
- Step 4** Repeat [Step 2](#) and [Step 3](#) to configure each receiving location that accepts cross-server sign-in handoffs from this location.
- Tip** After you have saved the changes on a page, use the **Next** and **Previous** buttons to quickly navigate through each location in the organization.
- Step 5** Repeat the procedure on all remaining Unity Connection originating locations.
-

Testing Cross-Server Sign-In

We recommend that you test cross-server sign-in before allowing users to use the feature.

For failover systems, first test that the primary destination servers answer cross-server calls. Then manually fail over the destination servers to verify that the secondary server answers cross-server calls. If the destination

servers are properly configured for failover, the secondary server should answer cross-server calls when the primary server is unavailable.

Do the following steps to test cross-server sign-in :

-
- Step 1** Create a new user account (or use an existing account) on each of the destination servers for testing purposes. Be sure to verify that the user account information has replicated to all of the servers that you are testing. The time that it takes for the user data to replicate depends on your network configuration and replication schedule.
- Step 2** For each user account, call the pilot number for the server configured for cross-server sign-in, and attempt to sign in. Verify that:
- The “One moment please” prompt is played (if configured to do so).
 - You successfully sign in.
-

Cross-Server Transfers

A cross-server transfer is a special kind of supervised transfer that passes control of a call from the automated attendant or a directory handler to the home server of the called user.

1. A caller calls a Unity Connection server on which an audio text application has been configured.
2. The caller does one of the following:
 - In a call handler (such as the opening greeting), enters the extension of a user on another server, or
 - In a directory handler, spells the name of a user on another server.
3. The server that is handling the call puts the caller on hold, and calls the home server of the user.
4. When the receiving server answers, the originating server sends a sequence of DTMF tones that identify the call as a cross-server transfer.
5. The receiving server responds with a sequence of DTMF tones.
6. The originating server hands off the call to the receiving server for processing. At this point, the behavior is as though the caller had directly called the automated attendant or directory handler on the receiving server.
7. In case of a video call, when two Unity Connection locations are linked by an HTTPS link, then if a user from one Unity Connection location attempts to cross-server transfer, the call is downgraded to audio.

When cross-server transfers have been configured, user call transfer, call screening, call holding, and announce features are available.

Task List for Enabling Cross-Server Transfers

When you are configuring an HTTPS network, use the following task list to enable cross-server transfers. The cross references take you to detailed procedures.

1. Determine whether each location is an originating location, a receiving location, or both.

2. For each originating location, make a list of the phone numbers the location must dial to reach the receiving location servers.



Note You can enter only one dial string for each receiving location. If the originating location is configured for multiple phone system integrations, you need a dial string that all phone system integrations can use to reach the receiving location.

3. Configure each receiving location so that it can handle incoming cross-server handoff requests.
 - If the receiving location is a Cisco Unity Connection server, see the [Configuring a Unity Connection Receiving Location to Accept Cross-Server Handoff Requests](#).
4. For each originating location, enable the cross-server transfer feature and enter the pilot numbers of the receiving locations from the list that you created in Task 2
 - If the location is a Cisco Unity Connection server, see the [Configuring a Unity Connection Originating Location to Perform Cross-Server Transfer Requests](#).
5. Test the cross-server transfer functionality. See the [Testing Cross-Server Transfer](#).

Procedures for Enabling Cross-Server Transfers

See the following sections:

Configuring Cross-Server Transfers during Call Forward to Cisco Unity Connection

Do the following procedure to configure cross-server transfers during call forward to Cisco Unity Connection:

Step 1 To view the configuration of cross-server transfers during call forward, execute the following command:

```
run cuc dbquery unitydirdb select fullname,name,parentid,valuebool,value from vw_Configuration where
name like 'HandoffForwardRemoteForward'
```

If the command results a configured table entry, it means the feature is configured on Cisco Unity Connection. Otherwise, go to [Step 2](#) to create a configuration entry.

In configured table entry, check the value of "valuebool" parameter. If valuebool is one, it means the feature is enabled for Cisco Unity Connection. Otherwise, go to [Step 3](#) to enable the feature.

Step 2 *(Applicable for Unity Connection 11.5(1) SU5 and earlier releases)* Create the configuration entry using the following command:

```
run cuc dbquery unitydirdb execute procedure
csp_ConfigurationCreate(pName='HandoffForwardRemoteForward'::lvarchar,
pParentFullName='System.Conversations.CrossBox'::lvarchar, pType=11, pValueBool=0, pRequiresRestart=1)
```

Step 3 Enable the cross-server transfers during call forward using the following command:

```
run cuc dbquery unitydirdb execute procedure
csp_ConfigurationModify(pName='HandoffForwardRemoteForward'::lvarchar,
pParentFullName='System.Conversations.CrossBox'::lvarchar, pValueBool=1)
```

Step 4 Disable the cross-server transfers during call forward using the following command:

```
run cuc dbquery unitydirdb execute procedure
csp_ConfigurationModify(pName='HandoffForwardRemoteForward'::lvarchar,
pParentFullName='System.Conversations.CrossBox'::lvarchar, pValueBool=0)
```

Note

- In case of a cluster, execute the commands only on publisher server and make sure that database replication is working fine for the cluster.
- Service restart is not required after executing the above commands.

Note With Unity Connection 11.5(1) SU6 and later, you can also enable the cross-server transfers during call forward through Cisco Unity Connection Administration. To enable the feature, navigate **System Setting > Advanced > Conversations** and check the **Cisco Unity Cross-Server Handoff During Call Forward** check box on Conversation Configuration page of Cisco Unity Connection Administration.

Configuring a Unity Connection Receiving Location to Accept Cross-Server Handoff Requests

By default, each Unity Connection server is configured to ignore cross-server handoff requests. To enable cross-server features, you must configure the receiving location to accept requests and also verify that the location routes incoming calls to a call handler. Do the following two procedures to configure each receiving Unity Connection location to accept handoffs. (Doing so allows the location to receive handoffs of all types—sign-in, transfer, and live reply.)

Steps to Configure a Unity Connection Receiving Location to Accept Cross-Server Handoff Requests

- Step 1** In Cisco Unity Connection Administration, on a location that accepts cross-server handoffs for users who are homed on that location (the receiving location), expand **System Settings > Advanced** then select **Conversations**.
- Step 2** Check the **Respond to Cross-Server Handoff Requests** check box.
- Step 3** Repeat the procedure on all remaining Unity Connection receiving locations.

Verifying Call Routing Rules are Set to Route Calls to a Call Handler Greeting

- Step 1** In Cisco Unity Connection Administration, on a location that accepts cross-server handoffs, expand **Call Management > Call Routing** and select **Direct Routing Rules**.
- Step 2** Select the display name of the routing rule that applies to incoming cross-server calls from originating locations.
- Step 3** Verify that calls that match the rule are routed to a call handler.
- Step 4** Repeat the procedure on all remaining Unity Connection receiving locations.

Configuring a Unity Connection Originating Location to Perform Cross-Server Transfer Requests

By default, a Unity Connection location do not attempt to perform a cross-server transfer. Note that when you enable cross-server transfers on Unity Connection, cross-server live reply is automatically enabled. Do the following procedure to enable cross-server transfer and live reply on any Unity Connection originating locations.

Steps to Configure a Unity Connection Originating Location to Perform Cross-Server Transfer Requests

-
- Step 1** In Cisco Unity Connection Administration, on a location that transfers calls to remote users (the originating location), expand **Networking**, then select **Locations**.
- Step 2** On the Search Locations page, select the Display Name of a remote location that accepts cross-server transfer handoffs for users who are homed on this location (the receiving location).
- Step 3** On the Edit Location page for the receiving location, do the following to initiate cross-server features to this receiving location:
- To enable cross-server transfer and live reply to the remote location, check the **Allow Cross-Server Transfer to this Remote Location** check box.
 - Enter the dial string that this location used to call the receiving location when performing the handoff (for example, the pilot number of the receiving location).
- Note** You can enter only one dial string for each receiving location. If the originating location is configured for multiple phone system integrations, enter a dial string that all phone system integrations can use to reach the receiving location.
- Step 4** Repeat [Step 2](#) and [Step 3](#) for each receiving location that accepts cross-server transfer handoffs from this location.
- Tip** After you have saved the changes on a page, use the **Next** and **Previous** buttons to quickly navigate through each location in the organization.
- Step 5** Repeat the procedure on all remaining Unity Connection originating locations.
-

Testing Cross-Server Transfer

We recommend that you test cross-server transfers before allowing callers to use the feature.

For failover systems, first test that the primary destination servers answer cross-server calls. Then manually fail over the destination servers to verify that the secondary server answers cross-server calls. If the destination servers are properly configured for failover, the secondary server should answer cross-server calls when the primary server is unavailable.

Do the following to test cross-server transfer

-
- Step 1** Create a new user account (or use an existing account) on each of the destination servers for testing purposes. Be sure to verify that the user account information has replicated to all of the servers that you are testing. The time that it takes for the user data to replicate depends on your network configuration and replication schedule.
- Step 2** For each user account, call the pilot number for the server configured for cross-server transfer, and enter the user extension at the opening greeting. Verify that:
- The “One moment please” prompt is played (if configured to do so).
 - The call is transferred to the user phone or the greeting, according to the call transfer settings of the called user.
-

Cross-Server Live Reply

Live reply, when enabled, allows a user who is listening to messages by phone to reply to a message from another user by transferring to the user. Note that whether users have access to live reply is controlled by the class of service.

When cross-server live reply is enabled:

1. After listening to a message from a user on another networked location, the message recipient chooses to call the user who left the message.

Note that if identified subscriber messaging (ISM) is disabled on the location that recorded the message, the cross-server live reply option is available only for messages that are sent by users who sign in and address and send the message from their mailboxes.

2. The originating location puts the user on hold and looks up the extension in the database to determine whether the user who is being replied to is on the same server or is on another networked location. If the user is on the same server, processing proceeds as usual.

However, if the user who is being replied to is on another location, the originating location calls the applicable receiving location.

3. When the receiving location answers, the originating location sends a sequence of DTMF tones that identify the call as a cross-server live reply.
4. The receiving location responds with a sequence of DTMF tones.
5. The originating location hands off the call to the receiving location for processing.
6. In case of a video call, when two Unity Connection locations are linked by an HTTPS link, then if a user from a Unity Connection location attempts to live reply, the call is downgraded to audio.

Task List for Enabling Cross-Server Live Reply

**Note**

In Unity Connection, cross-server live reply is automatically supported (for users whose class of service allows it) when cross-server transfer is enabled. If you have previously configured a Unity Connection location as an originating or receiving location for cross-server transfers, the location also originate or receive cross-server live reply requests.

Use the following task list to enable cross-server transfers and live reply between Unity Connection locations in an HTTPS network. The cross references take you to detailed procedures.

1. Determine whether each location is an originating location, a receiving location, or both.
2. For each originating location, make a list of the phone numbers the location must dial to reach the receiving location servers.

**Note**

You can enter only one dial string for each receiving location. If the originating location is configured for multiple phone system integrations, you need a dial string that all phone system integrations can use to reach the receiving location.

3. Configure each receiving location so that it can handle incoming cross-server handoff requests.
 - If the receiving location is a Cisco Unity Connection server, see the [Configuring a Unity Connection Receiving Location to Accept Cross-Server Handoff Requests](#).
4. For each originating location, enable the applicable cross-server features and enter the pilot numbers of the receiving locations from the list that you created in Task 2
 - If the location is a Cisco Unity Connection server, see the [Testing Cross-Server Sign-In](#).
5. Test the cross-server live reply functionality. See the [Testing Cross-Server Sign-In](#).

Procedures for Enabling Cross-Server Live Reply

Configuring a Unity Connection Receiving Location to Accept Cross-Server Handoff Requests

By default, each Unity Connection server is configured to ignore cross-server handoff requests. To enable cross-server features, you must configure the receiving location to accept requests and also verify that the location routes incoming calls to a call handler. Do the following two procedures to configure each receiving Unity Connection location to accept handoffs. (Doing so allows the location to receive handoffs of all types—sign-in, transfer, and live reply.)

Steps to Configure a Cisco Unity Connection Receiving Location to Accept Cross-Server Handoff Requests

-
- | | |
|---------------|--|
| Step 1 | In Cisco Unity Connection Administration, on a location that accepts cross-server handoffs for users who are homed on that location (the receiving location), expand System Settings > Advanced , then select Conversations . |
| Step 2 | Check the Respond to Cross-Server Handoff Requests check box. |
| Step 3 | Repeat the procedure on all remaining Unity Connection receiving locations. |
-

Verifying Call Routing Rules are Set to Route Calls to a Call Handler Greeting

-
- | | |
|---------------|--|
| Step 1 | In Cisco Unity Connection Administration, on a location that accepts cross-server handoffs, expand Call Management > Call Routing and select Direct Routing Rules . |
| Step 2 | Select the display name of the routing rule that applies to incoming cross-server calls from originating locations. |
| Step 3 | Verify that calls that match the rule are routed to a call handler. |
| Step 4 | Repeat the procedure on all remaining Unity Connection receiving locations. |
-

Configuring a Unity Connection Originating Location to Perform Cross-Server Live Reply and Transfer Requests

By default, a Unity Connection location do not attempt to perform a cross-server live reply. Note that when you enable cross-server live reply on Unity Connection, cross-server transfer is automatically enabled. Do the following procedure to enable cross-server transfer and live reply in on any Unity Connection originating locations.

Steps to Configure a Unity Connection Originating Location to Perform Cross-Server Live Reply and Transfer Handoff Requests

-
- Step 1** In Cisco Unity Connection Administration, on a location that transfers calls to remote users (the originating location), expand **Networking**, then select **Locations**.
- Step 2** On the Search Locations page, select the Display Name of a remote location that accepts cross-server live reply and transfer handoffs for users who are homed on this location (the receiving location).
- Step 3** On the Edit Location page for the receiving location, do the following to initiate cross-server features to this receiving location:
- To enable cross-server transfer and live reply to the remote location, check the **Allow Cross-Server Transfer to this Remote Location** check box.
 - Enter the dial string that this location used to call the receiving location when performing the handoff (for example, the pilot number of the receiving location).
- Note** You can enter only one dial string for each receiving location. If the originating location is configured for multiple phone system integrations, enter a dial string that all phone system integrations can use to reach the receiving location.
- Step 4** Repeat [Step 2](#) and [Step 3](#) for each receiving location that accepts cross-server transfer handoffs from this location.
- Tip** After you have saved the changes on a page, use the **Next** and **Previous** buttons to quickly navigate through each location in the organization.
- Step 5** Repeat the procedure on all remaining Unity Connection originating locations.
-

Testing Cross-Server Live Reply

We recommend that you test cross-server live reply before allowing callers to use the feature.

For failover systems, first test that the primary destination servers answer cross-server calls. Then manually fail over the destination servers to verify that the secondary server answers cross-server calls. If the destination servers are properly configured for failover, the secondary server should answer cross-server calls when the primary server is unavailable.

Do the following steps to test cross-server live reply

-
- Step 1** Create a new user account (or use an existing account) on each location for testing purposes. Verify that users belong to a class of service in which live reply is enabled. Also verify that the user account information has replicated to all of the servers that are testing. The time that it takes for the user data to replicate depends on your network configuration and replication schedule.
- Step 2** Sign in as a user on an originating location and send a message to the test users on other locations.
- Step 3** For each user that receives the test message, sign in, listen to the message, and choose to call the sender. Verify that:
- The “One moment please” prompt is played (if configured to do so).
 - The call is transferred to the user phone or the greeting, according to the call transfer settings of the called user.
-

Notable Behavior for Cross-Server Sign-In, Transfers, and Live Reply

This section provides information about notable expected behavior associated with cross server sign-in, transfers and live reply.

See the following sections:

Cross-Server Sign-In Not Providing User Workstation Client Sign-In Access

Users must access their home server (or cluster) when using client applications such as the Cisco Personal Communications Assistant (Cisco PCA) and IMAP clients. The phone interface is the only client that provides cross-server sign-in capability.

Factors Causing Delays During Cross-Server Handoff

The following factors can contribute significantly to delays in cross-server call handoff:

- Longer user extensions. A four-digit extension does not take as long to dial during the handoff as a ten-digit extension.
- Longer dialing strings to reach the receiving location. A four-digit dialing string does not take as long to dial as a ten-digit dialing string.
- Multiple elements (such as PIMG/TIMG units, voice gateways, TDM trunks, and PSTN interfaces) in the call path between the originating location and the receiving location. More elements in the call path require more processing time for handing off cross-server calls.

In your environment, these factors can create delays that may cause the cross-server features to be unusable or unfeasible for callers. You must test your cross-server configuration on a representative call path in your environment to determine whether the delays that callers experience are acceptable.

Increased Port Usage with Cross-Server Features

The cross-server features require the use of ports on both the originating and receiving locations. Depending on how busy your servers are, you may need to add more ports or an additional server before enabling these features. You may also need to adjust how ports are configured. For example, you may need to enable more ports to accept incoming calls.

After enabling the cross-server features, we recommend that you monitor activity on the servers closely until you are confident that the servers can handle the increased load. For Cisco Unity Connection servers, you can use the Port Activity report in Cisco Unity Connection Serviceability to monitor port usage.

Transfer Overrides on Cross-Server Transfers

When a caller enters an extension in the automated attendant followed by the digits “#2,” the caller is routed directly to the greeting for the extension entered without a transfer being attempted. This is known as the transfer override digit sequence. Cisco Unity Connection 11.x automatically supports the transfer override sequence between networked locations.

Using Cross-Server Features with the Display Original Calling Number on Transfer Parameter

Do the following tasks so that cross-server handoffs complete properly between locations when this service parameter is set in Cisco Unified CM. In the task list, you create a special directory number for each receiving

location that is used only during cross-server handoffs, so that the receiving location recognizes the call as a handoff.

Task List for Configuring a Cross-Server Directory Number for Cross-Server Features

1. In Cisco Unified Communications Manager Administration, create a new directory number (for example, on a CTI route point) for each location that receives cross-server sign-in, transfer, or live reply calls. Configure the new directory number to always forward calls to the pilot number for the location. See the “Directory Number Configuration” chapter of the applicable *Cisco Unified Communications Manager Administration Guide* for your release of Cisco Unified CM, at http://www.cisco.com/en/US/products/sw/voicesw/ps556/prod_maintenance_guides_list.html.
2. Configure each receiving location with a forwarded call routing rule that sends calls in which the forwarding station equals the location’s new cross-server directory number to the Opening Greeting call handler. See the [Adding Forwarded Call Routing Rules to Destination Locations for Cross-Server Calls](#).
3. Update each originating location to dial the cross-server directory number of the receiving location during cross-server calls, rather than the pilot number. See the [Configuring Cross-Server Directory Number as the Dial String on Originating Locations](#).

Adding Forwarded Call Routing Rules to Destination Locations for Cross-Server Calls

Do the following to add a forwarded call routing rule :

-
- Step 1** In Cisco Unity Connection Administration on any one of the Unity Connection receiving locations, create the new forwarded routing rule:
- a) Expand **Call Management**, then expand **Call Routing**.
 - b) Select **Forwarded Routing Rules**.
 - c) On the Forwarded Routing Rules page, select **Add New**.
 - d) On the New Forwarded Rule page, enter the name of the new rule in the Display Name field.
 - e) Select **Save**.
 - f) On the Edit Forwarded Routing Rule page, for Send Call To, select **Call Handler**. From the call handler drop-down list, select **Opening Greeting**.
 - g) Select **Save**.
 - h) On the Edit Forwarded Routing Rule page, under Routing Rule Conditions, select **Add New**.
 - i) On the New Forwarded Routing Rule Condition page, select **Forwarding Station**. From the forwarding station drop-down list, select **Equals**. In the text box, enter the new cross-server directory number for this location.
 - j) Select **Save**.
- Step 2** Return to the Forwarded Routing Rules page by selecting **Forwarded Routing Rules > Forwarded Routing Rules**, or by navigating to **Call Management > Call Routing > Forwarded Routing Rules**.
- Step 3** Check the order of forwarded routing rules on the page. If the new routing rule that you created in [Step 1](#) is not at the top of the table (in order of descending precedence) do the following substeps to move the new routing rule to the top of the forwarded routing rules table:
- a) On the Forwarded Routing Rules page, select **Change Order**.
 - b) On the Edit Forwarded Routing Rule Order page, select the Display Name of the new routing rule that you created in [Step 1](#).
 - c) Select the up arrow icon below the table to move the rule to the top position. (You may need to select the icon multiple times.)
 - d) Select **Save**.

Step 4 Repeat the procedure for each remaining Unity Connection receiving location.

Configuring Cross-Server Directory Number as the Dial String on Originating Locations

Step 1 In Cisco Unity Connection Administration, on any one of the Unity Connection locations that originate cross-server calls, expand **Networking > Locations**.

Step 2 On the Search Locations page, select the Display Name of a receiving location).

Step 3 On the Edit Location page for the receiving location, change the dial string that this location used to call the receiving location to the new cross-server directory number of the receiving location.

Step 4 Repeat [Step 2](#) and [Step 3](#) to configure each receiving location that accepts cross-server handoffs from this location.

Tip After you have saved the changes on a page, use the **Next** and **Previous** buttons to quickly navigate through each location in the organization.

Step 5 Repeat the procedure on all remaining Unity Connection originating locations.
