HTTPS Networking Guide for Cisco Unity Connection Release 12.x

First Published: 2019-01-24

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

© 2019 Cisco Systems, Inc. All rights reserved.

CONTENTS

CHAPTER 1 **Overview of HTTPS Networking** 1

Overview of HTTPS Networking 1

Overview 1

About Unity Connection HTTPS Links 1

About VPIM Networking 2

Directory Synchronization 3

Replication Within HTTPS Network 3

Overview of High Availability in HTTPS Networking 6

Behavior of Cluster in Standard mode 6

Behavior of Cluster in Alert mode 6

Directory Size Limits 7

Messaging 8

Handling the Messages to System Distribution Lists Within an HTTPS Network 8

Cross-Server Sign-In, Transfers, and Live Reply 9

Addressing and Dial Plan Considerations 9

Addressing Options for Non-Networked Phone Systems 9

Identified User Messaging 10

CHAPTER 2 **Setting Up an HTTPS Network** 13

Setting Up an HTTPS Network 13

Setting Up an HTTPS Network 13

Prerequisites 13

Task List for Setting Up an HTTPS Network 14

Procedures for Setting Up an HTTPS Network 15

Notable Behavior in Networked Unity Connection Servers 30

Networked Broadcast Messages are Not Supported 30

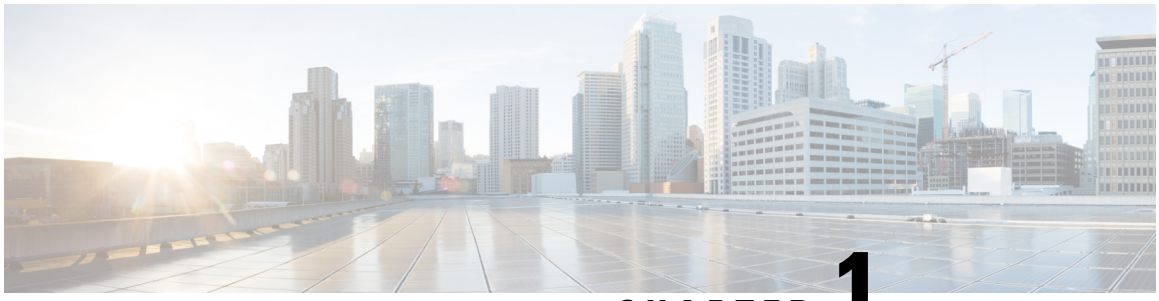
Networked Dispatch Messages are Not Supported 30
Manual Synchronization and Resynchronization Runs Both Directory and Voice Name Synchronization Tasks 30
Adding Remote Users as Private Distribution List Members 30

CHAPTER 3 **Migration from Legacy Network to HTTPS Network 33**
Migration from Legacy Network to HTTPS Network 33
Migration from Legacy Network to HTTPS Network 33
Prerequisites for Migration from Legacy to HTTPS Network 33
Task List to Migrate from Legacy Network to HTTPS Network using Flash Cut Approach 34

CHAPTER 4 **Making Changes to HTTPS Networking Configuration 37**
Making Changes to HTTPS Networking Configuration 37
Migration from Legacy Network to HTTPS Network 37
Removing an HTTPS Link Between Two Unity Connection Locations 38
Removing an HTTPS Link 38
Removing a Location from an HTTPS Network 39
Using “Remove Self from Site” Option 40
Making Changes to a Unity Connection Location 41
Updating Directly Connected Nodes in HTTPS Networking for IP or Hostname Change 41

CHAPTER 5 **Cross-Server Sign-In, Transfers, and Live Reply in HTTPS Networking 43**
Cross-Server Sign-In, Transfers, and Live Reply in HTTPS Networking 43
Introduction 43
Overview of Cross-Server Sign-In, Transfer, and Live Reply 43
Search Space Considerations for Cross-Server Sign-In, Transfers, and Live Reply 44
Cross-Server Sign-In 45
Task List for Enabling Cross-Server Sign-In 46
Procedures for Enabling Cross-Server Sign-In 46
Cross-Server Transfers 48
Task List for Enabling Cross-Server Transfers 48
Procedures for Enabling Cross-Server Transfers 49
Cross-Server Live Reply 52
Task List for Enabling Cross-Server Live Reply 52

Procedures for Enabling Cross-Server Live Reply 53
Notable Behavior for Cross-Server Sign-In, Transfers, and Live Reply 55
Cross-Server Sign-In Not Providing User Workstation Client Sign-In Access 55
Factors Causing Delays During Cross-Server Handoff 55
Increased Port Usage with Cross-Server Features 55
Transfer Overrides on Cross-Server Transfers 55
Using Cross-Server Features with the Display Original Calling Number on Transfer Parameter 55



CHAPTER 1

Overview of HTTPS Networking

• Overview of HTTPS Networking, on page 1

Overview of HTTPS Networking

Overview

When the messaging needs of your organization require more than one Cisco Unity Connection server or cluster, you need a way to combine multiple Unity Connection directories or to ensure that the connected servers can communicate with each other. The concept of HTTPS Networking, connects different Unity Connection servers and clusters in a network.

Note

- In Unity Connection, legacy networking is also supported to connect multiple Unity Connection servers in a network. However, it is recommended to deploy a new network as per HTTPS networking. Legacy networking includes both intrasite (digital) and intersite networking.
- The legacy and HTTPS networking are not supported simultaneously in the same network.

The main objective of HTTPS networking is to increase the scalability of Unity Connection deployments. The architecture of HTTPS networking is scalable both in terms of number of Unity Connection locations and the total directory size.

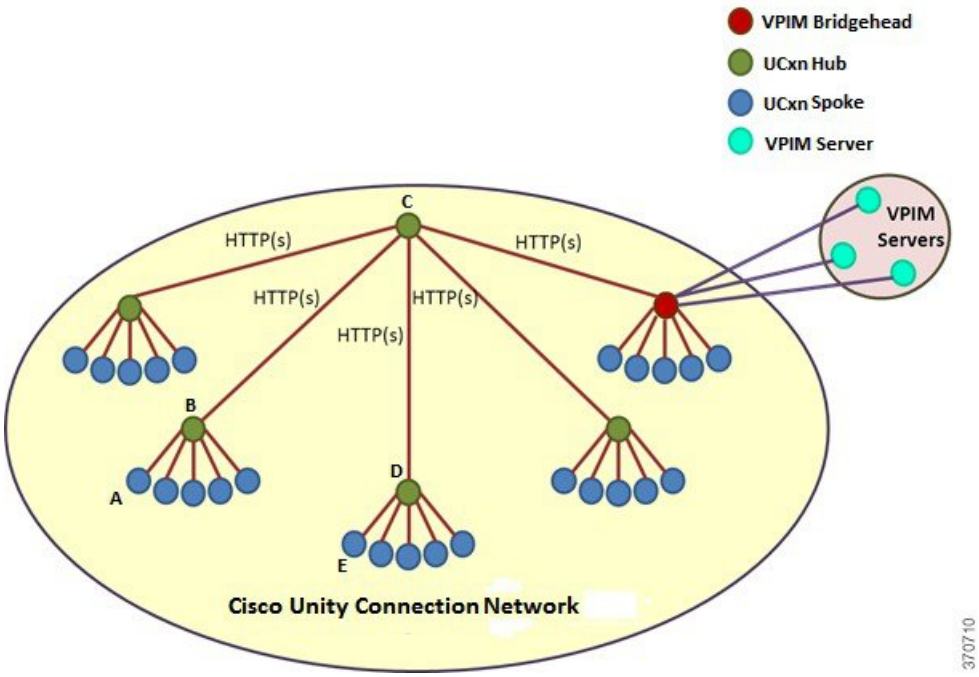
About Unity Connection HTTPS Links

You can join two or more Unity Connection servers or clusters to form a well-connected network, referred to as an HTTPS network. The servers that are joined to the network are referred to as locations. (When a Unity Connection cluster is configured, the cluster counts as one location in the network.) Within a network, each location uses HTTPS protocol to exchange directory information and SMTP protocol to exchange voice messages with each other.

The locations in an HTTPS network are linked together through an HTTPS link. The topology used in HTTPS networking is hub and spoke topology, which plays an important role in increasing scalability of directory size and number of Unity Connection locations. In hub-spoke topology, there are two types of locations: hub location and spoke location. The Unity Connection location which has more than one HTTPS links is known

as hub location. However, the Unity Connection location which has only one HTTPS link is known as spoke location. Figure 1 illustrates a network of multiple Unity Connection locations joined by HTTPS links.

Figure 1: A Cisco Unity Connection 11.x Network Joined by HTTPS Links Among All Locations



In hub-spoke topology, all the directory information among the spokes is shared through the hub(s) connecting the spokes. For example, in the above figure, if spoke A needs to synchronize directory information with spoke E, the directory information flows from spoke A to hub B, hub B to hub C, hub C to hub D, and then from hub D to spoke E.

Each Unity Connection server (or cluster) is represented in the network as a single Unity Connection location, which is created locally during installation and which cannot be deleted from the server itself. When you join the server (or cluster) to an existing location in a network, a Unity Connection location is automatically created for the server (or cluster).

Note HTTPS networking supports single site networks only. You cannot connect multiple HTTPS networks or single site networks together to form a larger network. The maximum number of Unity Connection locations that you can connect in an HTTPS network is 25.

About VPIM Networking

Unity Connection supports the Voice Profile for Internet Mail (VPIM) protocol, which is an industry standard that allows different voice messaging systems to exchange voice and text messages over the Internet or any TCP/IP network. VPIM is based on the Simple Mail Transfer Protocol (SMTP) and the Multi-Purpose Internet Mail Extension (MIME) protocol.

VPIM Networking is supported for use with Cisco Business Edition.

Unity Connection supports up to 100 VPIM locations and 150,000 VPIM and System contacts in the Connection directory. These limits apply either to the directory of a single Unity Connection server or cluster pair, or to the global directory in a network.

Note To support 150K contacts on a single Unity Connection server, you need to dedicate the server as VPIM bridgehead only.

If you deploy VPIM in an HTTPS network, you can designate one or more Unity Connection locations in the network as VPIM bridgehead(s) to handle the configuration of VPIM locations and contacts depending upon your requirements. The VPIM location data and all contacts at the VPIM location (including automatically created contacts) are replicated from the bridgehead to other locations within the network. When a VPIM message is sent by a user who is homed on a Unity Connection location other than the bridgehead, the message first passes to the bridgehead, which handles forwarding the message to the destination server. Similarly, the messages from VPIM contacts are received by the bridgehead and relayed to the home server of the Unity Connection recipient.

For more information on VPIM Networking, design considerations, and configuration details, see the "VPIM Networking" chapter of the *Networking Guide for Cisco Unity Connection Release 12.x*, at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/networking/guide/b_12xcucnetx.html.

Directory Synchronization

Each location in an HTTPS network has its own directory of users and other objects that are created on the location and are said to be "homed" on that location. The collection of objects and object properties that are replicated among locations is referred to as the global directory.

See the following section for details on the specific objects and object properties that are replicated during directory synchronization:

- Replication Within HTTPS Network

Replication Within HTTPS Network

Within HTTPS network, the objects and object properties that are replicated during directory synchronization are shown in Table 1: Replicated Objects in a Cisco Unity Connection HTTP(S) Network:

Table 1: Replicated Objects in a Cisco Unity Connection HTTP(S) Network

Replicated Object	Replicated Properties
Users with mailboxes	• Alias • First name, last name, display name, alternate names • Extension, cross-server transfer extension, and alternate extensions • Directory listing status • Partition • Recorded name • SMTP address and SMTP proxy addresses • Phone Numbers
System and VPIM contacts	All properties
System distribution lists	All properties, including list membership
Partitions	All properties
Search spaces	All properties, including membership
Unity Connection location	• Display name • Host address • SMTP domain name • Unity Connection version • Encryption Key • Subscriber's Base feed URL • Maximum DL count • Maximum Users count • Destination Type • Maximum Contact Count

Replicated Object	Replicated Properties
VPIM location	• Display name • Dial ID • Partition • Search Scope • SMTP Domain Name • IP Address • Recorded Names • Remote Phone Prefix

In most cases, you can use replicated objects just as you would use local objects; for example, you can assign a remote user to be the message recipient of a system call handler, or configure the search scope of a user to use a remote search space. Note the following exceptions:

- System call handler owners must be local users.
- Objects that belong to a partition (users, contacts, handlers, system distribution lists, and VPIM locations) can only belong to local partitions. You can, however, add a remote partition to a local search space.

In HTTPS networking, the directory replication is accomplished by means of a Feeder service and a Reader service running on each location in the network. The Reader service periodically polls the remote location for any directory changes since the last poll interval. The Feeder service checks the change tracking database for directory changes and responds to poll requests with the necessary information.

On each location in a network, you can configure the schedule on which the Reader polls the remote Feeder for directory data, and the schedule on which it polls for recorded names. In Cisco Unity Connection Administration interface of a Unity Connection location, you can access the schedules on the **Tools > Task Management** page by selecting either the **Synchronize Directory With Local Network** task or the **Synchronize Voice Names With Local Network** task.

Note The tasks are not available unless HTTPS networking is configured on the system.

When the **Synchronize Voice Names With Local Network** task is enabled, the Reader processes recorded name files for remote users, contacts, VPIM locations, and system distribution lists (if applicable). Once a recorded name is created for a remote object on the Unity Connection location, it is updated only if the remote and local filenames for the recorded name differ. For example, if you change the outgoing codec for recorded names on the remote location, the local Unity Connection location do not update its files because the change does not affect filenames. In order to pull updated copies of recorded names in this case, you must clear all existing recorded names from the local Unity Connection location and then do a full resynchronization using the **Clear Recorded Names** button and the **Resync All** button on the **Search HTTPS Links** page in Unity Connection Administration.

Overview of High Availability in HTTPS Networking

This section describes the concept of high availability of Unity Connection in terms of directory synchronization. Unity Connection can be configured as a cluster node comprising of the publisher and subscriber servers. In the HTTPS networking, when the publisher server of a cluster location is up and running, it is responsible for the synchronization of directory information. However, if the publisher server is down, the subscriber server takes the role of synchronizing directory information. This concept of maintaining a backup Unity Connection server for the situations when the primary server is down is known as high availability.

Depending upon the component of a cluster (publisher or subscriber) with which the directory synchronization is being performed, the directory synchronization can be of the following types:

- Standard - Specifies that the directory synchronization is done by the publisher server with the connected locations.
- Alert - Specifies that the publisher server is unreachable and subscriber server is responsible for providing directory information to the connected locations. However, the subscriber server has the directory information stored that is last synchronized with the publisher server when it was running.

Behavior of Cluster in Standard mode

The following is the behavior of Unity Connection cluster and Reader/Feeder services on the cluster in Standard mode:

- By default, a Unity Connection cluster is in Standard mode
- The Reader service remains running on the publisher server.
- The Feeder service runs on the publisher as well as subscriber server. By default the Feeder service of the publisher server responds to the directory synchronization request.

Behavior of Cluster in Alert mode

The following is the behavior of Unity Connection cluster and Reader/Feeder services on the cluster in Alert mode:

- The publisher server is inaccessible.
- The Reader service remains inactive on the publisher as well as subscriber server.
- The Feeder remains running on the subscriber server and responds to the directory synchronization requests.

Usually the publisher is down for a very short period of time and the directory synchronization occurs in the Alert mode. During the Alert mode, the connected nodes have limited access to directory synchronization with the subscriber. The limited access means that the connected nodes can fetch only the directory information that was last synchronized with the publisher when it was running. When the publisher comes up, the nodes that are directly connected to the publisher synchronize the updated directory information through the publisher. Therefore, the key benefit of the Alert mode is that the connected nodes remain synchronized with the subscriber server even when the publisher is down.

Note In an HTTPS network, if the publisher server of a Unity Connection cluster is down, then the Unity Connection cluster moves to "Alert" mode. However, this mode is reflected on the Cisco Unity Connection Administration interface of the connected nodes only after the completion of directory synchronization on the connected nodes.

In the Alert mode, the Feeder service running on the subscriber server of a cluster node has the capability to provide directory information to the directly connected nodes. In addition, the Reader service running on the nodes that are directly connected to a cluster node has the capability to fetch directory information from the subscriber server when the publisher is down.

Note When a cluster node is in split brain condition, the Reader service installed on the cluster node remains in inactive state. For more information see *Install, Upgrade, and Maintenance Guide for Cisco Unity Connection, Release 12.x*, available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/install_upgrade/guide/b_12xcuciumg.html.

Directory Size Limits

The Unity Connection global directory (the entire collection of local and replicated objects) is subject to certain size limits. However, it also generates an RTMT alert so that administrator can take appropriate action. In Unity Connection, there are separate limits on the number of users, the number of contacts, and the number of system distribution lists.

The size limits are:

- 100,000 Users
- 150,000 Contacts
- 100,000 system distribution lists
- 25,000 users per system distribution list
    - 1.5 million total list members across all system distribution lists
    - 20 levels of nesting (where one system distribution list is included as a member of another list)

Note Additional directory object limits exist, and the directory object limits may have been updated since the time of release. For detailed and up-to-date limit information, see the *System Requirements for Cisco Unity Connection Release 12.x* at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/requirements/b_12xcucsysreqs.html.

In HTTPS network, you are allowed to link up to 25 Unity Connection locations. However, whenever the Reader service installed on the Unity Connection locations runs, it checks the limit for the replication objects, such as users or contacts. If the number of replication objects exceeds its maximum limit, the Reader service moves to "Warning" mode. In the Warning mode, you can still create new directory objects for remote location objects but you might face performance issues. Therefore, it is recommended to get the Reader service out of

Warning mode to avoid such issues. In order to move the Reader out of "Warning" mode, you must remove enough number of objects of the appropriate type to bring the number less than the maximum limit.

Consider the following example of the user limit check. Unity Connection location A has 50,000 users, and Unity Connection location B has 60,000 users. Now, after joining the two locations, when the Reader service on Unity Connection A and Unity Connection B runs, both the locations move to the "Warning" mode as the total number of users becomes 110,000 (maximum limit is 100,000). To bring the two locations to the normal mode, you need to remove some users either from Unity Connection A or from Unity Connection B.

Messaging

See the following section for details on how messaging is handled in specific networking situations:

- Handling the Messages to System Distribution Lists Within an HTTPS Network

Handling the Messages to System Distribution Lists Within an HTTPS Network

Because system distribution lists are replicated among locations in a Cisco Unity Connection network, a user can address messages to any system distribution list at any location, as long as the list is reachable in the user search scope.

When a user addresses a message to a system distribution list, the local Cisco Unity Connection location parses the distribution list membership. The sending location delivers the message directly to local users on the list. If there are remote Unity Connection users in the distribution list, the sending location delivers the message to each location that homes the remote users. If there are VPIM users in the list, the sending server either delivers the message to the VPIM destination if the VPIM location is homed locally, or passes it to the server on which the location is homed and that server handles forwarding the message to the destination server.

Unity Connection includes the following predefined system distribution lists: **All Voicemail Users, Undeliverable Messages,** and **All Voicemail-Enabled Contacts**. Each Unity Connection server in your organization has a distinct version of each of these lists. If you have not changed the names of these lists to be unique, during initial replication each server automatically adds the remote server name to the display name of any remote lists whose names overlap with local list names.

By default, the predefined lists on each Unity Connection location have the same recorded name, and the **All Voicemail Users** and **All Voicemail-Enabled Contacts** lists have the same extension at each location (the Undeliverable Messages list by default is not assigned an extension, because users do not typically address messages to this list). When setting up an HTTPS Unity Connection network, you should consider modifying the recorded name of each **All Voicemail Users** list and each **All Voicemail-Enabled Contacts** list; if you do not, users can hear a confusing list of choices when they address messages by name to one of these lists. When users address by extension to a list whose extension overlaps that of another list, they reach the first list that is located when Unity Connection searches the partitions of the user search space in order.

Make sure to synchronize the distribution list immediately after changing the membership of the distribution list to avoid facing issues, for example NDR, and in sending voice messages to the changed distribution list.

Tip Distribution lists can be nested such that a distribution list contains other lists. You can create one master **All Voicemail Users** distribution list for a network that contains the **All Voicemail Users** list of each Unity Connection location.

Cross-Server Sign-In, Transfers, 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 locations. For this reason, only the user home location has information about call transfer settings, greetings, and other specific details for the user. In order for a location to properly handle calls destined for a user on a different location, the location that receives the call 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 2: Cross-Server Features.

Table 2: 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.
Cross-server transfer	Cross-server transfer enables calls from the automated attendant or from a directory handler of one server to be transferred to a user on another server, 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 server by calling the user (according to the call transfer and screening settings of the called user).

For more information and instructions on enabling the cross-server features, see the Cross-Server Sign-In, Transfers, and Live Reply in HTTPS Networking chapter.

Addressing and Dial Plan Considerations

Addressing Options for Non-Networked Phone Systems

If your organization has a separate phone system for each location, users at one location dial a complete phone number, not just an extension, when calling someone at another location. When users sign in to send messages to users on another networked location, the number that they enter when addressing a message by extension depends on whether the numbering plans overlap across locations.

When user extensions on one location overlap with user extensions on another location, you can provide unique extensions for each user by setting up alternate extensions for each user account. For each user, enter a number for the alternate extension that is the same as the full phone number for the user, and make sure that the alternate extension is in a partition that is a member of the search spaces that users at other locations use. Once this has been set up, when users sign in to send messages, the number that they enter when addressing messages is the same number that they use when calling.

When numbering plans do not overlap across networked locations-that is, when user extensions are unique across locations-users can enter an extension when addressing a message to a user who is associated with another location. Optionally, as a convenience for users in this circumstance, you may select to add alternate extensions to each user account, so that users do not need to remember two different numbers-one for calling a user directly, and one for addressing a message. However, if you do not set up alternate extensions, be sure to tell users to use the extension instead of the full phone number when addressing messages to users who are associated with another location.

Note that alternate extensions have other purposes beyond their use in networking, such as handling multiple line appearances on user phones.

Identified User Messaging

When a user calls another user, and the call is forwarded to the greeting of the called user, the ability of Unity Connection to identify that it is a user who is leaving a message is referred to as identified user messaging. Because Unity Connection is able to identify the caller as a user:

- Unity Connection plays the internal greeting of the called user when the caller leaves a message.
- Unity Connection plays the recorded name of the user who left the message when the recipient listens to the message.
- Unity Connection allows the recipient to record a reply.

It is important to note the difference between the following two circumstances:

- A user signs in to Unity Connection and then records and sends a message. In this circumstance, when the user has signed in, Unity Connection can identify the message as being from the user, regardless of which location the message recipient is homed on. In this case, the phone system is not involved and the recipient phone does not ring. Instead, the message is sent via networking message exchange (using SMTP).
- A user places a phone call to another user, and then leaves a message. This circumstance is the basis of identified user messaging.

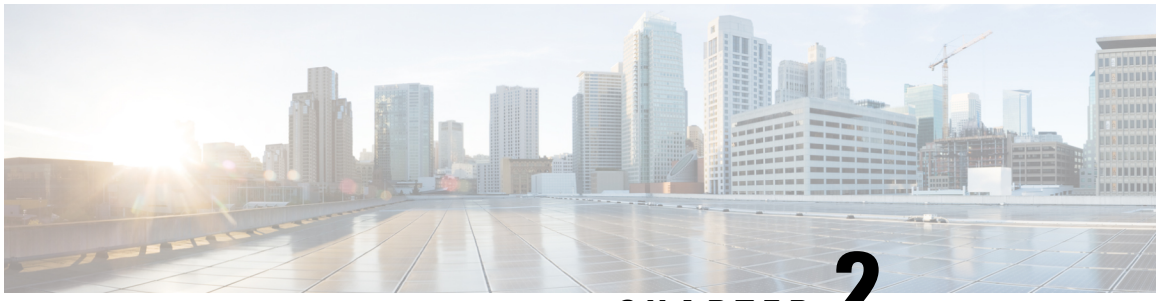
As long as identified user messaging is enabled on a Unity Connection location, Unity Connection is able to identify both local and remote users. Note, however, that for identified user messaging to work in both cases, the initial search scope of the call must be set to a search space that locates the correct user based on the calling extension, regardless of whether the caller is a local or remote user.

If a user calls from an extension that is in a partition that is not a member of the search space that was set as the initial search scope for the call, 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 that they appear in the search space.

In situations where numbering plans overlap across locations, it is therefore possible to have a user leave a message that is incorrectly identified as coming from another user with the same extension in a different partition. Because the initial search scope of the call is based on call routing rules, to avoid this situation, use the following configuration guidelines:

- Maintain a separate search space for each location in which the partition containing its users appears first in the search space. (By default, each Unity Connection server uses its own default partition and default search space, which are replicated to other locations when the server is networked.)

- On each location, set up forwarded call routing rules specific to every other location by specifying a routing rule condition that applies only to calls from that location (for example, based on the port or phone system of the incoming call). Configure the rule to set the search scope of the call to the search space in which the partition containing users at the location appears first.



CHAPTER 2

Setting Up an HTTPS Network

• Setting Up an HTTPS Network, on page 13

Setting Up an HTTPS Network

Setting Up an HTTPS Network

This section describes the prerequisites for setting up an HTTPS network of Unity Connection servers, and provides a high-level task list of all of the tasks that you need to complete for the setup, and the order in which they should be completed. If you are unfamiliar with HTTPS networking concepts, you should first read the Overview of HTTPS Networking chapter and then review the task list and procedures before beginning the setup.

Prerequisites

Before starting the setup, verify that the following prerequisites have been met on each server that join the HTTPS network (for clusters, verify these prerequisites for the publisher server):

- The server meets the requirements listed in the " Requirements for HTTPS Networking " section of the *System Requirements for Cisco Unity Connection Release 12.x*, available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/requirements/b_12xcucsysreqs.html.
- Unity Connection is already installed.
- The servers networked together are directly accessible through TCP/IP port 25 (SMTP), or SMTP messages are routable through an SMTP smart host. In addition, both locations must be able to route to each other via HTTP on port 8081 or HTTPS on port 8444.
- In order for directory synchronization and message exchange to occur between the two locations in a HTTPS network, the locations must have the following connectivity with each other:
  - HTTPS (if you choose to encrypt the connection) or HTTP connectivity, for directory synchronization.
  - SMTP connectivity, for voice message exchange.
- For Unity Connection clusters, you must have a smart host available to resolve the SMTP domain of the cluster to both the publisher and subscriber servers in order for message traffic to reach the cluster subscriber server in the event that the publisher server is down.

In addition, before setting up an HTTPS network of Unity Connection servers, you should be familiar with the concepts in the " Dial Plan " section of the "Call Management" chapter in the *System Administration Guide for Cisco Unity Connection Release 12.x*, available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/administration/guide/b_12xcucsag.html.

Task List for Setting Up an HTTPS Network

Use this task list to set up an HTTPS network between Unity Connection servers or clusters. The cross-references take you to detailed procedures.

If you have a Unity Connection cluster, do the following tasks only on the publisher server.

1. Determine the network topology and the arrangement of the locations in the network depending upon the number of Cisco Unity Connection servers. See the "Deciding the Network Topology" section.
2. Make decisions about your networking deployment approach and gather information needed to configure the network. See the "Making Deployment Decisions and Gathering Needed Information for Setting Up an HTTPS Network section.
3. Determine how messages are routed between the locations. See the "Determining SMTP Routing Between Locations" section.
4. Check the display name of each server that you are joining to the network, and modify it if it is not unique, or if you want to select a more descriptive name. Also check the SMTP domain of each server that you are joining to the network, and modify it if it is not unique. See the "Verifying Each Unity Connection Server has a Unique Display Name and SMTP Domain" section.

Caution If the display name of a server matches the display name of another server in the network, the server cannot join the network. Similarly, if the SMTP domain matches the SMTP domain of another server in the network, the server cannot join the network.

5. Now start creating an HTTPS network by linking two Unity Connection servers together as per hub and spoke topology and top-down approach. See the "Linking Unity Connection Servers with HTTPS Link" section.
6. If any servers in the network require a smart host to transmit and receive SMTP messages from other servers (for example, because a firewall separates the servers, or because the servers are part of a Unity Connection cluster), configure the smart host, and configure the applicable locations to route through the host. See the "Configuring a Smart Host" section.

Note For each Unity Connection cluster that you have added to the network, you must configure all other locations to route to the cluster through a smart host in order for message traffic to reach the cluster subscriber server in the event that the publisher server is down. (You also configure the smart host to resolve the SMTP domain of the cluster to both the publisher and subscriber servers.)

7. For each cluster that you have added to the network, add the IP address of the subscriber server to the IP address access list on every other location on the network. This ensures that the other locations can receive message traffic from the subscriber server if the publisher server is down. See the "Configuring SMTP Access for Cluster Subscriber Servers" section.

8. Verify that replication is complete among locations. See the "Checking Replication Status Within the Network" section.
9. Configure search spaces at each location to allow users that are homed at the location to address the users at other locations. See the "Configuring Search Spaces for HTTPS Network" section.
10. Secure the network so that message transmissions are not misdirected. See the "Securing the HTTPS Network" section.
11. Optionally, set up cross-server features. See the "Cross-Server Sign-In, Transfers, and Live Reply in HTTPS Networking" chapter.
12. Test the network. See the "Testing the HTTPS Network Setup" section.
13. Optionally, set up a network-wide All Users distribution list. See the "Creating a Network-Wide All Voicemail Users Distribution List" section.
14. If you have not already done so, set up VPIM Networking to connect the Unity Connection locations to any other VPIM-compatible voice messaging systems. See the " VPIM Networking " chapter of the *Networking Guide for Cisco Unity Connection Release 12.x*, at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/networking/guide/b_12xcucnetx.html.
15. Optionally, create a mapping of which users are homed on which location. See the "Mapping Users to Home Locations" section.

Procedures for Setting Up an HTTPS Network

See the following sections:

Deciding the Network Topology

Before you start setting up a network, you need to create an HTTPS network map based on the following considerations:

- Number of locations
- Depth of the HTTPS network
- Configuration of the locations
- Number of HTTPS links per location
- Number of homed subscribers on each server

In an HTTPS network, the Unity Connection locations are joined together as per hub and spoke topology. However, the number of hubs and spokes and the depth of the network depend upon the number of locations that we need to connect in the network. It is required to maintain the depth of an HTTPS network map to only second level. Following are the different types of recommended HTTPS network maps based on the number of Unity Connection locations:

- Network Map of 10 Unity Connection Locations
- Network Map of 17 Unity Connection Locations
- Network Map of 25 Unity Connection Locations

Figure 2: Network Map of 10 Unity Connection Locations

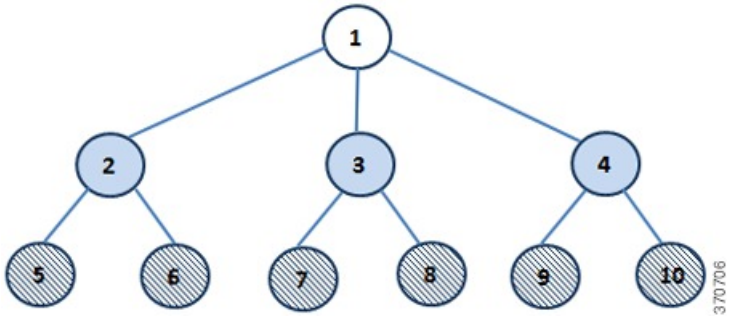


Figure 3: Network Map of 17Unity Connection Locations

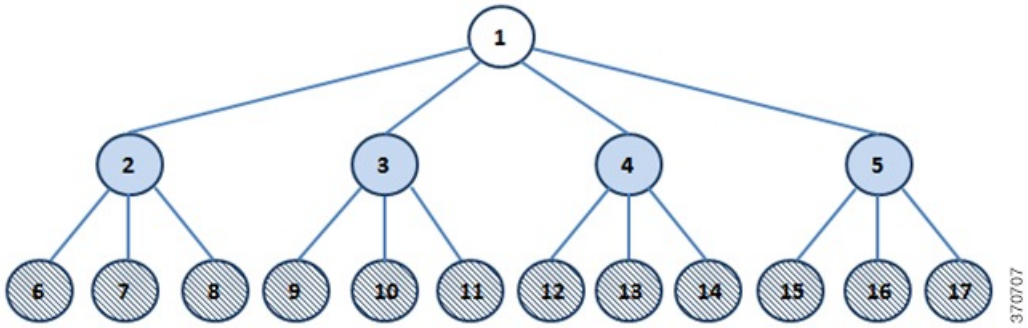
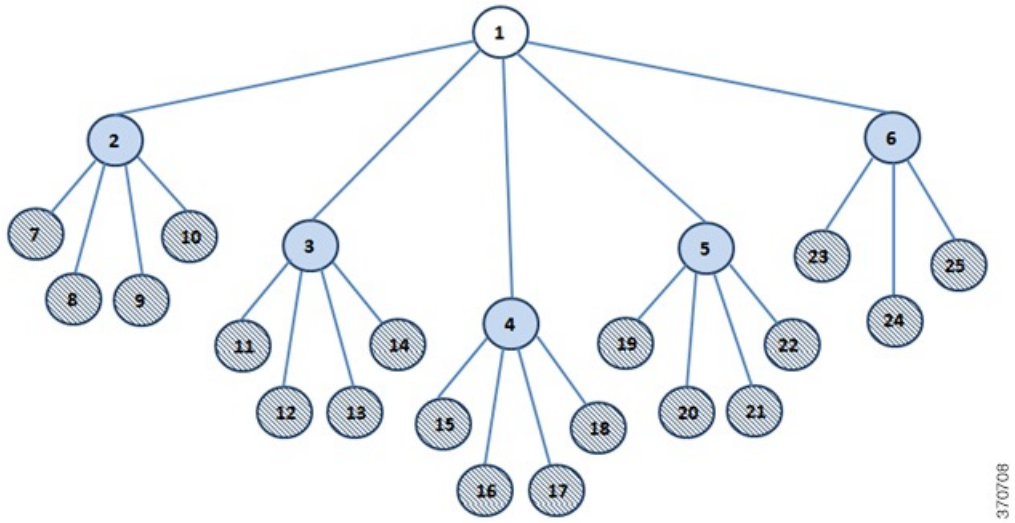
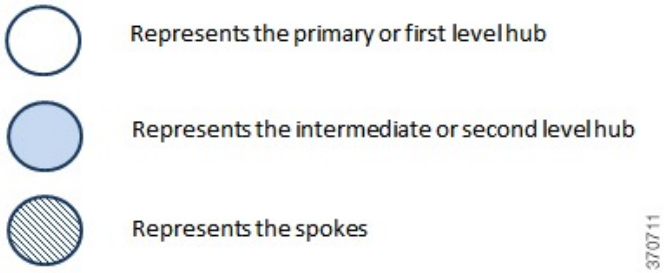


Figure 4: Network Map of 25 Unity Connection Locations



In the above network maps :



While deciding the number of hubs and spokes as per the network maps shown above, try to associate equal number of HTTPS links with every location (except spokes) in the network.

Now after deciding the number of hubs and spokes, you need to determine which location should act as hub and which location should act as spoke based on the configuration of the locations. First of all, arrange all the locations in the network in the descending order of their OVA size. Now the location that has the highest OVA should act as primary or first level hub. However, if the two or more locations have same OVAs then the location that has the least number of homed subscribers should act as primary or first level hub. Similarly, from the rest of the locations, the locations that have the highest OVA and least number of homed subscribers (if the OVA size of two or more locations is same) should act as intermediate or second level hub(s) depending upon the number of second level hubs, The remaining locations should act as spokes connected to the intermediate hubs.

For example, if you have 10 locations that need to be connected in an HTTPS network, then the recommended network map is shown in Figure 1, which decides the number of hubs and spokes in the network. It also decides the depth of the network as second level. As per the network map shown in Figure 1, for 10 locations, you need one primary hub, three intermediate hubs and six spokes. Now, you need to decide which location should act as hub and which location should act as spoke. To determine the hubs and spokes, arrange the locations in the descending order of their OVA size and increasing order of their homed subscribers (if the OVA size is same for two or more locations), as shown in Table 3: An Example of 10 Unity Connection Locations Arranged As Per Their Configuration Details:

Table 3: An Example of 10 Unity Connection Locations Arranged As Per Their Configuration Details

Location Name	OVA Size	Number of Homed Subscribers
1	7 vCPU	10 K
2	7 vCPU	15 K
3	4 vCPU	8 K
4	4 vCPU	10 K
5	4 vCPU	10 K
6	2 vCPU	4 K
7	2 vCPU	5 K
8	2 vCPU	5 K
9	2 vCPU	5 K
10	2 vCPU	5 K

As shown in Table 3: An Example of 10 Unity Connection Locations Arranged As Per Their Configuration Details, location 1 and location 2 has the highest and the same OVA size but the number of homed subscribers is less on location 1 as compared to location 2. Therefore, location 1 should act as primary hub. Now, from the rest of the locations, three locations should act as intermediate hub. The locations 2 and 3 have the highest OVA size and least number of the homed subscribers (if the OVA size is same). Therefore, locations 2 and 3 should act as intermediate hubs. For third intermediate location, you can treat either location 4 or location 5 as intermediate hub as both the locations have same configuration. The remaining locations, which are 5, 6, 7, 8, 9, and 10 act as spokes connected to the intermediate hubs.

In an HTTPS network, the directory synchronization between two locations occurs through the connecting hub(s) locations whereas the voice message exchange occurs point to point. For example, in the network topology of 10 locations shown above, if spoke 5 needs to synchronize directory information with spoke 6, it occurs through hub 2. However, if the spoke 5 needs to send voice message to spoke 6, it directly sends the message to spoke 6.

For more information on hub and spoke topology see the Overview of HTTPS Networking chapter.

Making Deployment Decisions and Gathering Needed Information for Setting Up an HTTPS Network

After creating the network map, be sure to plan for the following, and gather the applicable information:

- If your network includes voice messaging servers that do not meet the prerequisites for joining a HTTPS network but support the Voice Profile for Internet Mail (VPIM) protocol (for example, Cisco Business Edition, Unity Connection 2.x servers, Cisco Unity 4.x and 5.x, or other VPIM-compatible systems), use VPIM Networking to connect them.
- We recommend the following approaches:
  - Unless your servers are already configured for VPIM, set up rest of the network first, then set up VPIM Networking.
  - Select the Unity Connection locations in the network to handle the configuration of VPIM locations and contacts. These locations are referred to as the “VPIM bridgeheads.” The VPIM location and contact objects are replicated from the VPIM bridgeheads to all other Unity Connection locations in the HTTPS network so that the locations can address VPIM messages; the networked locations then forward the messages to the VPIM bridgehead for delivery to the remote voice messaging server.

    Note It is recommended that the location that has the minimum number of homed subscribers and highest OVA size should act as VPIM bridgehead.

  - If you are migrating a VPIM location to an HTTPS network (for example, because you used VPIM Networking to connect two or more Cisco Unity Connection 2.x servers and have upgraded the servers to Unity Connection 12.x) set up the HTTPS network first. After the directory is fully replicated and you have tested message exchange between the Unity Connection locations, remove the VPIM locations and VPIM contacts that represent the migrated servers and their users. The task list reminds you when to do this task. For more information on migration from Cisco Unity to Unity Connection, see the “ Migrating from Cisco Unity 4.x and Later to Unity Connection 7.x and Later ” section of the “Maintaining Cisco Unity Connection Server” chapter of the *Install, Upgrade, and Maintenance Guide for Cisco Unity Connection, Release 12.x*, available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/install_upgrade/guide/b_12xcuciumg.html.

- By default, every Unity Connection location (server or cluster) includes several predefined system distribution lists, which you can modify but not delete. If you have not renamed these lists so that the list names are unique on each location, or if you have added additional lists whose names are identical across locations, during initial replication each location automatically adds the remote server name to the display name of any remote lists whose names overlap with local list names. (The default lists are All Voicemail Users, Undeliverable Messages, and All Voicemail-Enabled Contacts.) This can cause confusion when local users try to address to those remote lists.
- To solve this problem, you can use one of the following approaches:
  - If you want to maintain separate lists on each location, you can modify the name of each list on its home location so that it is unique (for example "All Voicemail Users on <Location Name>") and notify your users of the new list names for each server. If you choose this approach, you should also modify the recorded name of each list to indicate its source.
  - Alternatively, after setting up the network, you can create a master list that includes all users on all networked locations. The task list includes instructions on when and how to do this task.
- If you want to synchronize Unity Connection user data with user data in an LDAP directory, we recommend that you configure Unity Connection for integration with the LDAP directory prior to setting up the network, to simplify testing and troubleshooting.
- Make note of the following information about each server that is joining the network:
  - The IP address or fully qualified domain name (FQDN) of the server.
  - The user name and password of a user account that is assigned to the System Administrator role.
  - The dial strings that other servers use to call this server, if cross-server sign-in or transfer are configured on other servers to hand off calls to this server.

Determining SMTP Routing Between Locations

In order for directory synchronization and message exchange to occur between the two locations in an HTTPS network, both the locations must have the following connectivity with each other:

- HTTPS (if you choose to encrypt the connection) or HTTP connectivity, for directory synchronization.
- SMTP connectivity, for voice message exchange.

In each direction, you can either route messages directly or use an SMTP smart host to route messages to the recipient. It is recommended to use an SMTP smart host in the following situations:

- The locations are separated by a firewall that blocks SMTP transmissions.
- Any of the locations is a Cisco Unity Connection cluster.

When a location is a cluster, you must configure the opposite location to route to the cluster through a smart host in order for message traffic to reach the cluster subscriber server in the event that the publisher server is down, and configure the smart host to resolve the SMTP domain of the cluster to the IP addresses of both the publisher and subscriber servers. In this case, we recommend that you route traffic in both directions through the smart host.

Verifying Each Unity Connection Server has a Unique Display Name and SMTP Domain

Each Unity Connection server that you join to an HTTPS network must have a unique display name. The display name must be unique both among Unity Connection locations and among VPIM locations. If the display name is not unique, the server cannot join the network. For new Unity Connection installations, the display name is typically the same as the host name of the server; however, if you changed the display name or upgraded the server from Unity Connection 2.x (which uses "Local VMS" as the default display name), you may need to change the display name so that it does not overlap with other locations on the network.

Tip Choose a display name for each server that is descriptive and that helps you identify the location when it is listed among all locations in the organization in Cisco Unity Connection Administration.

Each Unity Connection server that you join to the network must also have a unique SMTP domain, both among Unity Connection locations and among VPIM locations. By default, the SMTP domain is configured during installation to include the hostname of the server, in order to insure that it is unique. However, if the SMTP domains of multiple servers have been modified to the same value, you must change the domains to unique values before joining the servers in a network.

If you are migrating a server from VPIM Networking to HTTPS networking, it is likely that the display name or SMTP domain of the server overlaps with the VPIM location configured for the server. If the domain name overlaps, you need to disrupt messaging to the VPIM location while doing the migration-either by changing the SMTP domain of the VPIM location, or by removing the VPIM location.

Steps to Verify Each Cisco Unity Connection Server Has a Unique Display Name and SMTP Domain

Step 1 Check the **Display Name** of the first server:

a) In Cisco Unity Connection Administration on the first server, expand **Networking**, then select **Locations**.
b) On the Search Locations page, note the Display Name of the local server. We recommend that you make a list of all Display Names that you can consult later.

Step 2 Check the SMTP domain of the first server:

a) Expand **System Settings** > **SMTP Configuration**, then select **Server**.
b) On the SMTP Server Configuration page, note the SMTP Domain of the local server.

Step 3 Check the **Display Name** and **SMTP Domain Name** of all VPIM locations homed on the local server:

a) Expand **Networking**, then select **VPIM**.
b) On the Search VPIM Locations page, note the **Display Name** of each VPIM location.
c) Select the first VPIM location in the table. On the Edit VPIM Location page, note the SMTP Domain Name of the VPIM location.
d) Select Next and note the SMTP Domain Name of the next VPIM location.
e) Repeat Step 3d. d for each remaining VPIM location.

Step 4 Repeat Step 1 through Step 3 on each location joined to the network.

Step 5 If the Display Name of a location conflicts with that of another location, or you want to modify a name to be more descriptive, change one of the display names:

- To change the Display Name of a Unity Connection location, do Step 6.
- To change the Display Name of a VPIM location, do Step 7.
- If the Display Names are all unique, skip to Step 9.

Step 6 Change the Display Name of the Unity Connection location:

a) On the server for which you want to change the Display Name, expand Networking, then select Locations.
b) Select the Display Name of the local server.
c) On the Edit Location page, modify the Display Name value, and select Save

Step 7 To change the Display Name of a VPIM location:

a) On the server on which the VPIM location is homed, expand Networking, then select VPIM.
b) On the Search VPIM Locations page, select the Display Name of the location that you want to change.
c) On the Edit VPIM Location page, modify the Display Name value, and select Save.

Step 8 If there are any remaining Display Name conflicts, repeat Step 5 as necessary to resolve each conflict.

Step 9 If the SMTP domain of a server conflicts with that of another location, change one of the domain names:

- To change the SMTP Domain of a Unity Connection location, do Step 10.
- To change the SMTP Domain Name of a VPIM location, do Step 11.

Step 10 To change the SMTP Domain of a Unity Connection location:

a) Expand **System Settings** > **SMTP Configuration**, then select **Server**.
b) On the SMTP Server Configuration page, select **Change SMTP Domain**, change the value of the **SMTP Domain** field, and select **Save**.
c) Select OK to confirm the change.

Step 11 To change the SMTP Domain Name of a VPIM location:

a) On the server on which the VPIM location is homed, expand **Networking** and select **VPIM**.
b) Select the Display Name of the VPIM location for which you want to change the **SMTP Domain Name**.
c) On the Edit VPIM Location page, change the value of the **SMTP Domain Name** field, and select Save.

Caution Changing the SMTP Domain Name of a VPIM location may disrupt messaging with the remote voice messaging system.

Step 12 If there are any remaining SMTP domain conflicts, repeat Step 9 as necessary to resolve each conflict.

Linking Unity Connection Servers with HTTPS Link

To create an HTTPS network of Unity Connection servers, you start by linking two servers together via an HTTPS link. At a single point of time, a particular location can be joined with only one location in the network. After joining one location to another, make sure that the directory synchronization is completed between the two locations before joining another location to the network. Each Unity Connection server becomes a location in the network. When a Unity Connection cluster is linked to a location, the cluster is also counted as one location in the network.

If you are setting up a new HTTPS network, you should follow the bottom-up approach for joining two Unity Connection locations in the network. In the bottom-up approach, you start by joining the spokes with their intermediate hubs and then the intermediate hubs with the primary hub. The main advantage of the bottom-up approach is that you can join multiple intermediate hubs with their spokes simultaneously. However, at a single point of time, a particular intermediate hub can make connection with one spoke only. Similarly, at a particular instance of time, the primary hub can join with only one intermediate hub. For example, if you need to create a network as per network map shown in Figure 2.1, you should start creating network by joining location 5 to location 2, location 7 to location 3, and location 9 to location 4 simultaneously.

Note It is not recommended to join two locations to the same location or hub simultaneously. For example, you should not join location 5 and 6 simultaneously to location 2.

In the next step, you can start joining location 6 with location 2, location 8 with location3, and location 10 with location 4 simultaneously. After joining the spokes with their intermediate hubs, start joining the intermediate hubs with the primary hub one by one to complete the network.

Note
- You can join only Unity Connection 10.x and later servers in an HTTPS network.
- If the Unity Connection location that you are joining to the network is a cluster server, it is recommended to join the location through publisher server only.

Joining Two Cisco Unity Connection Servers

Step 1 In Cisco Unity Connection Administration (on either server), expand **Networking** and select **HTTPS Links.**

Note You might see an error on the HTTPS Links page if any Legacy link exists on the Unity Connection location.

Step 2 On the Search HTTPS Links page, select **Add.**

Step 3 On the New HTTPS Link page, select Link to **Cisco Unity Connection Remote Location**.

Step 4 In the **Publisher** field, enter the IP address or fully-qualified domain name (FQDN) or hostname of the Unity Connection server that you want to connect to create the network.

Step 5 In the **Username** field, enter the user name of an administrator at the location specified in the **Publisher** field. The administrator user account must be assigned the System Administrator role.

Step 6 In the **Password** field, enter the password for the administrator specified in the **Username** field.

Step 7 For **Transfer Protocol** settings, enable the SSL to encrypt directory synchronization traffic between the different locations.

Step 8 By default, two tasks that run on their own schedule for data and recorded name directory synchronization from the remote location are enabled immediately after you create the HTTPS link. To disable either type of directory synchronization until you manually edit and enable the applicable synchronization task, uncheck the **Enable** task to synchronize directory data after the join or Enable task to synchronize recorded names after the join check boxes.

Step 9 Select **Link**.

Step 10 Select **OK** to confirm and a success message pops up as **"You have successfully linked to the location**".

Note

- In an HTTPS network, by default the system distribution lists and its members are not synchronized between different locations in the network. To enable the synchronization of distribution lists and its members of a particular Unity Connection location, you need to check the **Include Distribution Lists When Synchronizing Directory Data** check box on the Edit HTTPS Link page of the corresponding location.
- It is recommended that synchronization of distribution list should be initiated after the completion of first directory synchronization cycle.
- When you enable system distribution list synchronization, you cannot disable it after the link is created except by removing and recreating the HTTPS link.
- If you enable the **Include Distribution Lists When Synchronizing Directory Data** check box on one location in the network, it is recommended to check the check box on all the locations in the network.

Configuring a Smart Host

SMTP is used to transmit messages between Unity Connection locations in a network.

If any pair of locations in the network cannot transmit and receive SMTP messages directly (for example, because a firewall separates the servers), you must configure these locations to route these messages through an SMTP smart host.

In addition, for each Unity Connection cluster that you add to the network, you must configure all other network locations to route to the cluster through a smart host in order for message traffic to reach the cluster subscriber server in the event that the publisher server is down, and configure the smart host to resolve the SMTP domain of the cluster to the IP addresses of both the publisher and subscriber servers. For example, a network has a single smart host and the following three locations:

- Server A, which is not a cluster member
- Cluster 1, which is made up of Server B, a publisher, and Server C, a subscriber
- Cluster 2, which is made up of Server D, a publisher, and Server E, a subscriber

In order to create an HTTPS network, you would join Server A, Server B and Server D together to form the network. Note the following:

- On Server A, you would configure the Unity Connection locations for Server B (which represents cluster 1) and Server D (which represents cluster 2) to route through the smart host.
- On Server B (the cluster 1 publisher), you would configure the Unity Connection location for Server D (which represents cluster 2) to route through the smart host.
- On Server D (the cluster 2 publisher), you would configure the Unity Connection location for Server B (which represents cluster 1) to route through the smart host.
- On the smart host, you would configure the SMTP domain name of cluster 1 to resolve to the IP addresses of both Server B and Server C (for example, using DNS MX records). You would also configure the SMTP domain name of cluster 2 to resolve to both Server D and Server E.

Do the following tasks for each server that requires routing to other locations through a smart host:

1. Configure the SMTP smart host to accept messages from the Unity Connection server. If your network includes Unity Connection clusters, also configure the smart host to resolve the SMTP domain of the cluster to the IP addresses of both the publisher and subscriber servers. See the documentation for the SMTP server application that you are using.
2. Configure the Unity Connection server to relay messages to the smart host. See the Configuring the Cisco Unity Connection Server to Relay Messages to a Smart Host procedure .
3. Configure the Unity Connection server to route messages to the other Unity Connection locations through the smart host. See the Configuring the Cisco Unity Connection Server to Route Inter-Location Messages through the Smart Host procedure.

Configuring the Cisco Unity Connection Server to Relay Messages to a Smart Host

Step 1 In Cisco Unity Connection Administration, expand **System Settings > SMTP Configuration**, then select Smart Host.

Step 2 In the **Smart Host** field, enter the IP address or fully qualified domain name of the SMTP smart host server. (Enter the fully qualified domain name of the server only if DNS is configured.)

Step 3 Select **Save**.

Configuring the Cisco Unity Connection Server to Route Inter-Location Messages through the Smart Host

Step 1 In Cisco Unity Connection Administration, expand **Networking**, then select **Locations**.

Step 2 Select the name of a location that requires routing through a smart host.

Step 3 Check the **Route to This Remote Location Through SMTP Smart Host** check box.

Step 4 Select Save.

Step 5 Repeat Step 1 through Step 4 for each additional location that requires routing through the smart host.

Configuring SMTP Access for Cluster Subscriber Servers

When you create an HTTPS network that includes a Unity Connection cluster server pair, you join only the publisher server of the pair to the network. In order for all locations in the network to communicate directly with the cluster subscriber server when the subscriber status is Primary, you must configure all network locations (except for the publisher server that is clustered with the subscriber server) to allow SMTP connections from the subscriber server.

Direct SMTP connectivity is needed so that locations can continue to receive user message traffic from the cluster while the publisher server does not have Primary status and the routing from the cluster to other locations is not done via a smart host. Direct SMTP connectivity with the subscriber server does not impact directory updates, because directory updates are only replicated from the publisher server.

For example, a network has the following three locations:

- Server A, which is not a cluster member
- Cluster 1, which is made up of Server B, a publisher, and Server C, a subscriber
- Cluster 2, which is made up of Server D, a publisher, and Server E, a subscriber

In order to create an HTTPS network, you would join Server A, Server B and Server D together. For direct SMTP access, the following steps are required:

- On Server A, you would need to add the IP addresses of both Server C and Server E (the two subscriber servers) to the IP address access list so that Server A can communicate with either subscriber server if it has Primary status.
- On Server B (the cluster 1 publisher), you would add the IP address of Server E (the cluster 2 subscriber) to the IP address access list; and on Server D (the cluster 2 publisher), you would add the IP address of Server C (the cluster 1 subscriber) to the IP address access list.

Alternatively, you can configure each cluster location to route messages to every other location through a smart host; when you do this, the other Unity Connection locations do not need to accept SMTP connections directly from the cluster subscriber when it has Primary status, because the cluster subscriber establishes the SMTP connection with the smart host rather than directly with every other location. In the example above, the alternate configuration would entail the following:

- On Server B (the cluster1 publisher), you would configure a smart host, and configure the Unity Connection locations for Server A and Server D (the cluster 2 publisher) to route through the smart host.
- On Server D (the cluster 2 publisher), you would configure a smart host, and configure the Unity Connection locations for Server A and Server B (the cluster 1 publisher) to route through the smart host.

For instructions on configuring routing through a smart host, see the Configuring a Smart Host section. Note that when more than one cluster is joined to a network, you should have already configured each cluster to route messages to other clusters through the smart host; in this case, all you need do in addition is to configure the cluster to route through the smart host to any servers that are not configured as clusters.

Configuring Direct SMTP Access for Cluster Subscriber Servers

Step 1 On a network location, in Cisco Unity Connection Administration, expand **System Settings > SMTP Configuration**, then select **Server**.

Step 2 On the **Edit** menu, select **Search IP Address Access List**.

Step 3 Select **Add New**.

Step 4 On the New IP Address page, enter the IP address of a cluster subscriber server at another location on the network.

Note Do not enter the IP address of the subscriber server on the publisher server that it is paired with.

Step 5 Select **Save**.

Step 6 On the IP Address page, make sure that the **Allow Connection** check box is checked.

Step 7 Select **Save**.

Step 8 Repeat Step 2 through Step 7 for each additional subscriber server on the network (other than the subscriber server that is paired with the server you are configuring).

Step 9 Repeat Step 1 through Step 8 on each network location.

Checking Replication Status Within the Network

When initial replication begins among locations, it can take a few minutes to a few hours for data to be fully replicated between all locations, depending on the size of your directory.

On each location in the network, there are two tasks which control the schedule on which the Reader polls the remote Feeder for directory data and the schedule on which it polls for recorded names. By default, the tasks are enabled and run every 15 minutes. If you have unchecked the **Enable Task to Synchronize Directory Data After the Join** or **Enable Task to Synchronize Recorded Names After the Join** check boxes while linking the Unity Connection locations, you must configure the schedule and enable the tasks before synchronization can begin.

You can use the Edit HTTPS Link page and Task Schedule page in Cisco Unity Connection Administration interface to determine whether synchronization is progressing successfully or has completed. Do the following procedure to check synchronization status between locations and to configure schedules for the two synchronization tasks.

Tip On Unity Connection 12.x locations, you can also use the Voice Network Map tool in Cisco Unity Connection Serviceability to check replication status. With the tool, you can quickly locate replication problems in a network, and get information about the status of replication between any two locations in the network. For more details, select **Help > This Page** from within the tool, or see the "Using the Voice Network Map Tool in HTTPS Networking" chapter of the *Cisco Unified Serviceability Administration Guide Release 12.x* at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/serv_administration/b_12xcucservag.html.

Checking the Status of Synchronization Between Cisco Unity Connection Locations And Configure Task Schedules

Step 1 In Cisco Unity Connection Administration on a location, select **Networking** and **HTTPS Links**.

Step 2 On the Search HTTPS Links page, select the **Display Name** of the HTTPS link.

Step 3 On the Edit HTTPS Link page, check the values of the following fields:

- **Time of Last Synchronization**-Indicates the timestamp of the last time the local reader service attempted to poll the remote location feeder service for directory changes on the remote locations, regardless of whether a response was received.
- **Time of Last Failure**-Indicates the timestamp of the last time the local reader service encountered an error while attempting to poll the remote location feeder service. If the value of this field is 0, or if the **Time of Last Synchronization** value is later than the **Time of Last Error** value, replication is likely to be progressing without problems.
- **Object Count**-Indicates the number of objects (users, contacts, system distribution lists and its memberships if applicable, partitions, search spaces and Unity Connection locations) that the local Unity Connection location has synchronized from the remote location.

Step 4 View the **Synchronize Directory With Local Network** task, and enable it or change the schedule, if necessary:

a) From the Edit HTTPS Link page, in the **Related Links** box in the upper right corner of the page, select **Location Directory Synchronization** and select **Go**.
b) On the Task Schedule page, enable the task if it has not yet been enabled, and modify the schedule so that the task runs at the desired interval or time.
c) Select **Save**.
d) To view the task execution history, select Edit > Task Definition Basics. On this page you can determine whether the task has not started, is in progress, or has completed. If the task has completed, you can select either the **Time Started** or **Time Completed** to view the detailed task results.

Step 5 From the Task Definition Basics page, select **Task Definition** > **Task Definitions** to go to the list of all tasks.

Step 6 View the **Synchronize Voice Names With Local Network** task, and enable it or change the schedule, if necessary:

a) On the Task Definitions page, select **Synchronize Voice Names With Local Network**.
b) Select **Edit** > **Task Schedules**.
c) On the Task Schedule page, enable the task if it has not yet been enabled, and modify the schedule so that the task runs at the desired interval or time.
d) Select **Save**.
e) To view the task execution history, select Edit > Task Definition Basics. On this page you can determine whether the task has not started, is in progress, or has completed. If the task has completed, you can select either the **Time Started** or **Time Completed** to view the detailed task results.

Configuring Search Spaces for HTTPS Network

When you initially set up a network between locations, users that are homed on one location are not able to address messages to users at other locations, because the users on each location are in separate partitions and use search spaces that do not contain the partitions of users on the other locations. After initial replication completes between the locations, you can reconfigure your search spaces to include partitions that are homed on other servers, and you can change the search scope of users, routing rules, call handlers, directory handlers, and VPIM locations to use a search space that is homed on a remote location. (Note that while both partitions and search spaces are replicated between locations, you cannot assign users or other objects to a partition that is homed on another location.)

Note When limit search users to search scope containing users of another location, no users are displayed.

If you have not made any changes to the default partitions and search spaces on any server, at each location, you can add the default partition of each remote Cisco Unity Connection location to the search space that local users are using. For example, in a network of three servers named Server A, Server B, and Server C with no changes to the system defaults, in Cisco Unity Connection Administration on Server A you would add the "Server B Partition" and "Server C Partition" default partitions as members of the "Server A Search Space" default search space; in Unity Connection Administration on Server B you would add "Server A Partition" and "Server C Partition" to "Server B Search Space," and so on.

For instructions on adding partitions to search spaces, see the "Dial Plan" section of the "Call Management" chapter in the *System Administration Guide for Cisco Unity Connection Release 12.x*, available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/administration/guide/b_12xcucsag.html.

Securing the HTTPS Network

No user credentials are transmitted as part of HTTPS communications. However, in order to protect the security of SMTP addresses that are contained in the messages, make sure that any smart hosts that are involved in SMTP message transmission between Unity Connection locations are configured to route messages properly, as it may be possible to extract SMTP addresses from the messages. See the documentation for the SMTP server application that you are using for instructions.

Testing the HTTPS Network Setup

To test the HTTPS network configuration, create test user accounts or use existing user accounts on each Unity Connection location. When setting up user accounts in Cisco Unity Connection Administration to be used in the tests, be sure to do the following for each account:

- Record a voice name.
- Record and enable an internal greeting.
- On the User Basics page, for Search Scope, select a search space that includes the partitions of remote users.
- On the User Basics page, check the **List in Directory** check box.
- On the Playback Message Settings page, check the **Before Playing Each Message, Play the Sender's Information** check box.
- Optionally, if you plan to enable and test cross-server live reply, ensure that the account belongs to a class of service for which the **Users Can Reply to Messages from Other Users by Calling Them** check box is checked on the Edit Class of Service > Message Options page. (The check box is not checked by default.)

Verifying Messaging Between Users on Different Unity Connection Locations

Step 1 Sign in to a Unity Connection location as a user.

Step 2 Follow the prompts to record and send messages to users who are associated with other Unity Connection locations.

Step 3 Sign in to the applicable Unity Connection location as the recipient user to verify that the message was received.

Step 4 Repeat Step 1 through Step 3 on other Unity Connection locations.

Verifying Call Transfers from the Automated Attendant to Users on Other Unity Connection Locations

Step 1 From a non-user phone, call a Unity Connection location that has been configured to handle outside callers, and enter the extension of a user who is associated with another Unity Connection location.

Step 2 Verify that you reach the correct user phone.

Verifying Call Transfers from a Directory Handler to Users on Other Cisco Unity Connection Locations

Step 1 From a non-user phone, call a Unity Connection location that has been configured to handle outside callers, and transfer to a directory handler.

Step 2 Verify that you can find a user who is associated with another Unity Connection location in the phone directory, and that the directory handler transfers the call to the correct user phone.

Verifying Identified User Messaging Between Networked Users (When Identified User Messaging Is Enabled)

Step 1 Verify that Unity Connection plays an internal greeting for users who leave messages, by doing the following sub-steps:

a) From a user phone, call a user who is associated with another Unity Connection location, and allow the call to be forwarded to Unity Connection.

b) Verify that the internal greeting plays.

c) Leave a test message.

Step 2 Verify that users are identified when the recipient listens to a message, by doing the following sub-steps:

a) Sign in to the applicable Unity Connection location as the recipient user and listen to the test message that you recorded in Step 1.

b) Verify that the user conversation announces who the message is from by playing the recorded voice name of the sending user.

c) After listening to the message, verify that the user conversation allows you to reply to the message.

Verifying Live Reply Between Users on Different Cisco Unity Connection Locations

Step 1 From a user phone, call a user who is associated with another Unity Connection location, and allow the call to be forwarded to voicemail.

Step 2 Leave a message.

Step 3 Sign in to the applicable Unity Connection location as the recipient user and listen to the test message that you recorded in Step 2.

Step 4 After listening to the message, verify that the user conversation allows you to live reply to the message by saying "Call sender" or using the applicable key presses for the user conversation type. (To find the key presses for a particular conversation, see the "Cisco Unity Connection Phone Menus and Voice Commands" chapter of the *User Guide for the Cisco Unity Connection Phone Interface, Release 12.x*, available at http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/user/guide/phone/b_12xcucugphone.html).

Step 5 Verify that the live reply call is correctly transferred to the phone of the user who left the message.

Creating a Network-Wide All Voicemail Users Distribution List

If you would like to create a master distribution list that includes all users on all servers in the network, do the following tasks:

1. On each location in the network, rename the All Voicemail Users list with a unique name (for example All Voicemail Users on <Location Name>). For instructions, see the "Configuring System Distribution Lists" section in the "System Distribution Lists" chapter in the *System Administration Guide for Cisco Unity Connection Release 12.x*, available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/administration/guide/b_12xcucsag.html.
2. Create a new All Voicemail Users system distribution list on one location to use as the master list.
3. Add the lists from all locations as members of the master list.
4. Put all lists except the master list in partitions that do not belong to a search space that users use, so that they cannot address to any list except the master. For example, on each location, create a new partition called Hidden DLs on <Location Name> and put the list homed at that location in that partition. (By default, new partitions are not a member of any search space.)

Tip To avoid having users generate large amounts of voice message traffic using reply-all to reply to messages sent to the master list, you should use search spaces to restrict access to the master list to a small subset of users. These users can use a search space that is essentially identical to the search space that other users use, except for the addition of the partition containing the master list.

Mapping Users to Home Locations

Each server or cluster handles a distinct group of users. In large organizations, it is possible that more than one server or cluster is in use at the same physical location. In this case, you need to determine which user accounts to create on each of the servers (the "home" server or location for each user), and keep a record of the mapping. This record is needed for the following reasons:

- User phones must forward calls to the system on which the users are homed.
- If user phones have a “Messages” or a speed-dial button that dials the number to access voicemail, the buttons must be configured to call the system on which the users are homed.
- If you do not configure cross-server sign-in, users must dial the pilot number of the server or cluster that they are associated with to check their messages; in this case, you need to tell the correct number to the users to dial when calling their home server.

To create a record of the mapping, run the Users report on each Unity Connection location. The information in this report includes the user name and primary location. For more information, see the “Reports” section of the “Advanced System Settings” chapter in the *System Administration Guide for Cisco Unity Connection Release 12.x*, available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/administration/guide/b_12xcucsag.html.

Notable Behavior in Networked Unity Connection Servers

See the following sections:

Networked Broadcast Messages are Not Supported

Broadcast messages cannot be sent to multiple locations within a network.

Networked Dispatch Messages are Not Supported

Dispatch messaging is not supported across locations. Dispatch messages addressed to recipients at other locations within a network are delivered to remote users as regular messages. You should configure dispatch messaging only when the message recipient is a system distribution list that does not include users on other networked locations.

Manual Synchronization and Resynchronization Runs Both Directory and Voice Name Synchronization Tasks

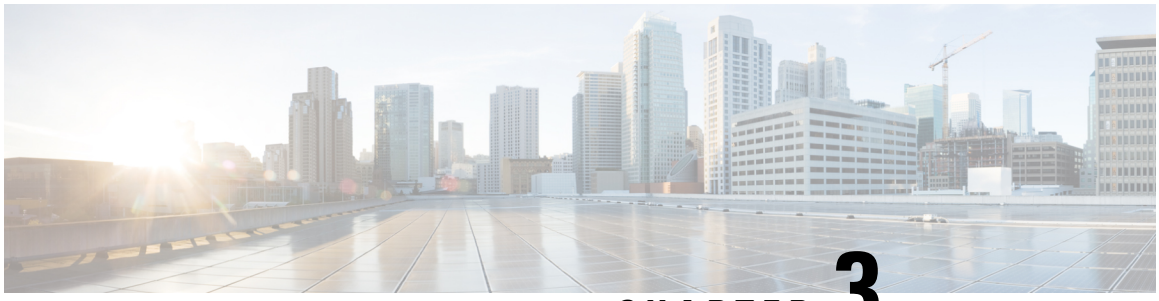
The **Sync and Resync All** button on the Search HTTPS Links page starts the **Synchronize Directory With Local Network** task. When that task completes, it automatically starts the **Synchronize Voice Names With Local Network** task. These tasks normally run independently on separate schedules.

Adding Remote Users as Private Distribution List Members

When creating private lists, users can add members from other locations if allowed by their search scope, in which case the same set of users who are reachable when addressing a message or placing a call can also be added as members of a private list. Private lists are not replicated to other locations; when a user addresses a message to a private list, the home location of the user expands the distribution list and addresses messages to each individual recipient on the list.

Consider notifying users in the event that the following members are inadvertently removed from their lists:

- When you delete a Unity Connection location, remote users at that location are removed from all private lists.
- When a VPIM contact becomes a Unity Connection user, the contact is removed from all private lists.



CHAPTER 3

Migration from Legacy Network to HTTPS Network

• Migration from Legacy Network to HTTPS Network, on page 33

Migration from Legacy Network to HTTPS Network

Migration from Legacy Network to HTTPS Network

In Cisco Unity Connection two or more Unity Connection servers or clusters can be connected through HTTPS links in a hub-spoke topology to form a single site network, referred to as an HTTPS Unity Connection network.

Migration from legacy network to HTTPS network can be done using a flash-cut approach which includes unjoining the existing network and then joining the locations in a pre-decided network topology.

A downtime window is required for the migration activity, which varies depending on the number of locations and the directory size; hence it should be planned during off-hours or over the weekend. Since legacy links are not supported in HTTPS networking, the entire network should collectively be migrated to the HTTPS network.

Prerequisites for Migration from Legacy to HTTPS Network

Before proceeding with the migration activity, ensure the following prerequisites have been met on each server that joins in the HTTPS network (for clusters, verify these prerequisites for the publisher server):

- For a Unity Connection site, the directory replication is working fine and none of the locations have stalled replication. Unity Connection SMTP server and Connection Digital Networking Replication Agent services are up and running. Visit https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/serv_administration/b_12xcucservag/b_12xcucservag_chapter_0100.html for more details on description and management of Cisco Unity Connection Services.
- All locations in the network are in synchronization with each other.
- For Intrasite links, this can be determined as mentioned in section " Checking Replication Status Within a Site " section of the “Setting Up Networking between Cisco Unity Connection Servers” chapter of the *Networking Guide for Cisco Unity Connection, Release 12.x*, available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/networking/guide/b_12xcucnetx.html.

- For Intersite links, see section "Checking the Status of Synchronization Between Unity Connection Sites and Configuring Task Schedules " of the “Setting Up Networking between Cisco Unity Connection Servers” chapter of the *Networking Guide for Cisco Unity Connection, Release 12.x*, available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/networking/guide/b_12xcucnetx.html. Verify that the latest synchronization task has completed without any errors.
- Upgrade all the locations to connection release 12.x. The HTTPS networking feature is available only from release 12.x onwards. See https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/install_upgrade/guide/b_12xcuciumg/b_12xcuciumg_chapter_00.html. for upgrading a network location to 12.x version.
- After upgrade, take a data backup for all locations in the legacy network. See *Install, Upgrade, and Maintenance Guide for Cisco Unity Connection, Release 12.x*,available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/install_upgrade/guide/b_12xcuciumg.html. which provides an overview of the Disaster Recovery System, describes how to use the Disaster Recovery System, and provides procedures for completing various backup-related tasks and restore-related tasks.
- The network administrator needs to decide the HTTPS network topology and arrangement of Unity Connection servers prior to the migration of the network. See section "Deciding the Network Topology" at Setting Up an HTTPS Network

Task List to Migrate from Legacy Network to HTTPS Network using Flash Cut Approach

After ensuring that the conditions specified in Prerequisites for Migration from Legacy to HTTPS Network section are met, proceed with the following steps to migrate the network:

Step 1 On each Unity Connection location in the network, take a backup of distribution lists, both public and private, and distribution list members that are homed at that location. The Distribution List Syncher tool can be used for this purpose. For more information on the tool visit: http://www.ciscounitytools.com/Applications/CxN/DistributionListBuilderCsv/DistributionListBuilderCsv.html.

Step 2 Schedule the task "Update Database Statistics" to run every 2 hours. Visit **Tools** > **Task Management**> **help** >**This Page** for details regarding task definitions, task definition basics and task schedules.

Step 3 Unjoin the locations in the network. It is recommended to unjoin only two locations at a time from the network, since unjoining more locations increases the SMTP traffic during replication and can halt replication.

See section Removing a Location From a Unity Connection Site of https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/networking/guide/b_12xcucnetx/b_12xcucnetx_chapter_0101.html for steps to unjoin an Intrasite network. It is also recommended that the Intrasite links be removed using Remove Self from Site option.For more information visit **Networking**>**Legacy Links**>**Intrasite Links**>**Help**>**This Page**.

Also see section Removing an Intersite Link Between Two Cisco Unity Connection 12.x Sites at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/networking/guide/b_12xcucnetx/b_12xcucnetx_chapter_0101.html for steps to unjoin an Intersite network.

Step 4 Schedule the "Update Database Statistics" task to run at its default time. Visit **Tools**>**Task Management**>**help**>**This Page** for details regarding task definitions, task definition basics and task schedules.

Step 5 Take a DRS backup of all the locations. See section “About Disaster Recovery System” of Backing Up and Restoring Cisco Unity Connection Components chapter in *Install, Upgrade, and Maintenance Guide for Cisco Unity Connection, Release 12.x,* available at

https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/install_upgrade/guide/b_12xcuciumg.html.which provides an overview of the Disaster Recovery System, describes how to use the Disaster Recovery System, and provides procedures for completing various backup-related tasks and restore-related tasks.

Step 6 Join the locations in HTTPS network as per the topology decided by the network administrator and allow the directory replication to complete. See Setting Up an HTTPS Network for details regarding joining locations in HTTPS network and verifying the synchronization status.

Step 7 In an HTTPS network, by default the system distribution lists and its members are not synchronized between the locations in the network. To enable the synchronization of distribution lists and its members, of a particular Unity Connection location, you need to check the Include distribution lists and membership when synchronizing directory data check box on the Edit HTTPS Link page of the corresponding location.

It is recommended that synchronization of distribution list should be initiated after the completion of first directory synchronization cycle.

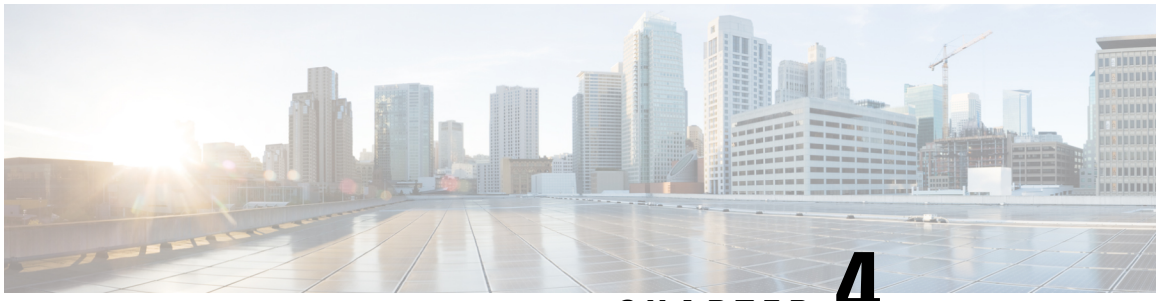
If you enable the Include distribution lists and membership when synchronizing directory data check box on one location in the network, it is recommended to check the check box on all the locations in the network.

Caution When you enable system distribution list synchronization, you cannot disable it afterwards, except by removing and recreating the HTTPS link.

Allow the directory synchronization to complete after enabling the checkbox on the locations.

Step 8 Once the directory replication is complete, start restoring the public and private distribution lists from the backup taken in step Step 1. The Distribution List Syncher tool can be used for this purpose. For more information on the tool see: http://www.ciscounitytools.com/Applications/CxN/DistributionListBuilderCsv/DistributionListBuilderCsv.html .

Step 9 After restoring distribution lists and distribution list members from the backup, allow the directory replication to complete between the locations.



CHAPTER 4

Making Changes to HTTPS Networking Configuration

• Making Changes to HTTPS Networking Configuration, on page 37

Making Changes to HTTPS Networking Configuration

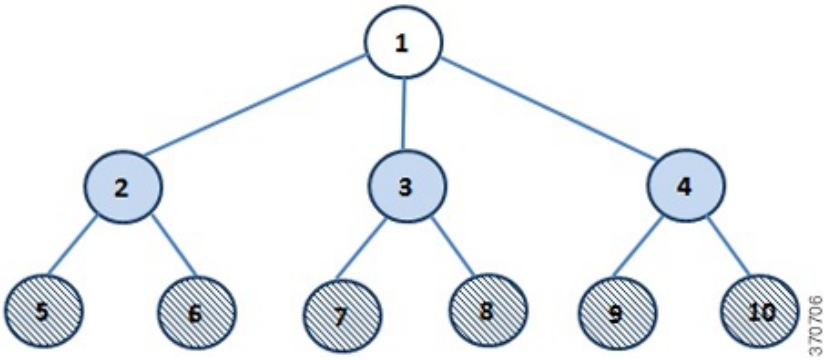
Migration from Legacy Network to HTTPS Network

When you remove an HTTPS link from an HTTPS network, the network gets divided into two parts. However, if you remove an HTTPS location from an HTTPS network, the network gets divided into the equal number of parts as the number of HTTPS links associated with the removed location.Then the each part of the network can be termed as a subtree. A subtree can be of the following two types with respect to a particular location:

- **Local subtree**: The subtree to which a Unity Connection location belongs is termed as local subtree.
- **Remote subtree**: The other subtree(s) to which a Unity Connection does not belong is termed as remote subtree.

For example, in figure 4-1 if you remove HTTPS link between location 1 and 2, it gets divided into two subtrees. One subtree comprising of locations 2, 5, and 6 whereas another comprising of locations 1, 3, 4, 7, 8, 9, and 10. Now with respect to location 2, the network of locations 5 and 6 becomes local subtree. However, the network of locations 1, 3, 4, 7, 8, 9, and 10 becomes the remote subtree for location 2.

Figure 5: HTTPS Network of 10 Unity Connection locations



Removing an HTTPS Link Between Two Unity Connection Locations

When you remove an HTTPS link between two Unity Connection locations, each location stops synchronizing directory information with the locations on the other side of the HTTPS link (which is referred as subtree), removes all objects that are homed on the remote subtree, and removes all configuration information about the mediator location. The mediator location is the location through which a location communicates with the rest of the network. We recommend that you carefully consider the impacts of removing an HTTPS link prior to doing so, particularly if you plan to relink the subtrees later. Consider the following impacts:

- Users, contacts, and system distribution lists on the local subtree are removed from distribution lists that are homed on the remote subtree and vice-versa. If you later relink the subtrees, you need to update distribution list membership to include any users, contacts, and distribution lists that belong to the remote subtree.
- System call handlers and interview handlers that are configured to send messages to a remote subtree user, contact, or distribution list are reconfigured to send messages to the undeliverable messages list of the location on which the handler is configured. If you later relink the subtrees, you need to update the recipients for these handlers to use the correct remote object. (Even if you do not plan to relink the subtrees, you should make sure that someone is checking messages that are sent to the Connection undeliverable messages list, or reassign handlers that use it as a recipient.)
- Partitions that were created for remote subtree locations are removed from search spaces in each Connection location in local subtree. If you later relink the subtrees, you need to review the partition membership of the search spaces that are owned on each location in each subtree. You need to manually re-add remote partitions to each search space or you may need to reorder the partitions within the search space to match the configuration that you had prior to removing the HTTPS link.
- On each location in the network, there are cross-server and SMTP routing configuration settings specific to other locations. When you remove an HTTPS link, these settings are removed and need to be configured again if you later re-add the HTTPS link.
- All HTTPS messaging, addressing between locations, and auto-attendant features unavailable after the HTTPS link is removed.

Note If the location that you are removing from the network is a Unity Connection cluster, it is recommended to remove through the publisher server only.

Do the following procedure to remove an HTTPS link between two Unity Connection 11.x locations. If either of the location is a Unity Connection cluster, do the steps for that location only on the publisher server.

Removing an HTTPS Link

Step 1 On either of the locations, remove the HTTPS link. This stops synchronization with the remote subtree and removes all remote subtree objects from the local subtree directory.

a) In Cisco Unity Connection Administration, expand **Networking** and select **HTTPS Links**.
b) Check the check box next to the HTTPS link that you want to remove.
c) Select **Remove Selected**.
d) At the warning about deleting associated objects, select **OK** to confirm the removal.

Step 2 Navigate to the **Search HTTPS link** page of the other location and select Sync button next to the HTTPS link that you want to remove. Make sure that the status for the HTTPS link is **"Link Removal Pending"**.

Step 3 Optionally, review the schedule for the **Remove Objects Associated With Deleted Network Location** task. By default, to avoid affecting system performance during business hours, this task runs at once a day at 10:00 p.m., and the HTTPS link is not fully removed until the task has run.

To review the schedule, either select the link to the task that is displayed in the Status message on the Search HTTPS Links page after you have removed the selected HTTPS link, or expand **Tools** and select **Task Management**; on the Task Definitions page, select **Remove Objects Associated With Deleted Network Location**.

Caution Because server performance can be impacted by large deletion activities, we strongly recommend that you allow the **Remove Objects Associated With Deleted Network Location** task to run on the default schedule (or at another time during non-business hours).

Step 4 To verify that the link has been removed, expand **Networking**, and select **HTTPS Links**.

On the **Search HTTPS Links** page, if the link has not yet been removed, it is displayed in the **HTTPS Links** table with (Link Removal Pending) listed after the **Display Name**. If the **Remove Objects Associated With Deleted Network Location** task has run and the link has been removed, the entry for the deleted link is not displayed in the HTTPS Links table.

Step 5 Repeat Step 3 through Step 4 on the other location also.

Removing a Location from an HTTPS Network

When you remove a location from an HTTPS network, it stops replicating directory information with other locations in the network and all the objects that are homed on the server are removed from other locations. Similarly, all the objects that are homed on other locations of the network are removed from the server you are removing.

We recommend that you carefully consider the impacts of removing a location from the network prior to doing so, particularly if you plan to add the location back to the network later. Consider the following impacts:

- Users on the server are removed from distribution lists that are homed on other locations in the network, and users on other locations are removed from distribution lists that are homed on the server you remove. Later on, if you add the server back into the network, you need to update the distribution list membership on the re-added server to include any remote users, and update distribution list membership on all other locations in the network to include users on the re-added server.
- System call handlers and interview handlers on other locations that are configured to send messages to a user, contact, or distribution list that is homed on the server you remove are reconfigured to send messages to the undeliverable messages list of the location. Likewise, system call handlers and interview handlers on the server you remove that are configured to send messages to a user, contact, or distribution list that is homed on another location are reconfigured to send messages to the local undeliverable messages list. If you later add the server back into the network, you need to update the recipients for these handlers to use the correct remote object. (Even if you do not plan to add the server back into the network, you should make sure that someone is checking messages that are sent to the undeliverable messages list, or reassign handlers that use it as a recipient.)
- Partitions that are homed on the server are removed from search spaces that are homed on other locations in the network, and partitions that are homed on other locations are removed from search spaces that are homed on the server you remove. Later on, if you add the server back into the network, you need to review the partition membership of search spaces on the re-added server and on all other locations in the

network. You need to manually re-add remote partitions to each search space or you may need to reorder the partitions within the search space to match the configuration that you had prior to removing the location.

- One location in the network retains a copy of search spaces that are homed on the server being removed, and the copy continues to be replicated amongst remaining locations in the network. Likewise, the server being removed makes a copy of remote search spaces that are homed on other locations. The copies replace the original search spaces on any objects that reference them. Later on, if you add the server back into the network, Unity Connection automatically attempts to resolve the ownership of the search space copies to their original location and then delete the copies. If you do not plan to add the server back into the network, you can reassign object references that use the search space copies to use other search spaces, and then delete the copies.
- On each location in the network, there are configuration settings specific to other locations (for example, the fields related to cross-server transfers and SMTP routing). When you remove a server from the network, the settings for all locations in the network are deleted from the server that you remove, and the settings for the server that you remove are deleted from all other locations in the network. Later on, if you add the server back into the network, you need to update the settings for the re-added server on all other locations in the network, and configure the settings for all other locations on the re-added server.

Do the following procedure to remove a location from the HTTPS network:

- Using “Remove Self from Site” Option

It is recommended that you remove only one Unity Connection location from a network at a time.

Depending on the size of the directory, removing a Unity Connection location can take a few minutes to a few hours. Even though the operation may have completed on the local location, it may continue to be in progress on remote locations. We recommend that you wait for the removal operation to complete on all locations in the network before making additional changes to the network.

Using “Remove Self from Site” Option

You can use this option to remove a Unity Connection location from the network using its own Cisco Unity Connection Administration interface.

Step 1 Remove the self-location from the network. This stops synchronization with the other locations in the network and removes all remote objects from the directory of the deleted location.

a) In Cisco Unity Connection Administration, expand Networking, and then select **HTTPS Links**.
b) Select **Remove Self from Site**.
c) At the warning about deleting associated objects, select **OK** to confirm the removal. The status of all the locations on the **Search HTTPS** page becomes **“Link Removal Pending**”.

Step 2 Synchronize other associated HTTPS link with the deleted location.

a) Navigate to the Cisco Unity Connection Administration page of each associated HTTPS link.
b) In Cisco Unity Connection Administration, expand **Networking**, and then select **HTTPS Links**.
c) On the **Search HTTPS link** page, select **Sync** button next to the HTTPS link that you want to remove. Make sure that the status for the HTTPS link is **“Link Removal Pending”**.

Step 3 Optionally, review the schedule for the **Remove Objects Associated With Deleted Network Location** task. By default, to avoid affecting system performance during business hours, this task runs at once a day at 10:00 p.m., and the HTTPS link is not fully removed until the task has run.

To review the schedule, either select the link to the task that is displayed in the Status message on the Search HTTPS Links page after you have removed the selected HTTPS link, or expand **Tools** and select **Task Management**; on the Task Definitions page, select **Remove Objects Associated With Deleted Network Location.**

Caution Because server performance can be impacted by large deletion activities, we strongly recommend that you allow the **Remove Objects Associated With Deleted Network Location** task to run on the default schedule (or at another time during non-business hours).

Step 4 To verify that the link has been removed, expand Networking, and select HTTPS Links.

On the **Search HTTPS Links** page, if the link has not yet been removed, it is displayed in the HTTPS Links table with (Link Removal Pending) listed after the **Display Name**. If the **Remove Objects Associated With Deleted Network Location** task has run and the link has been removed, the entry for the deleted link is not displayed in the HTTPS Links table.

Step 5 Repeat Step 3 and Step 4 on the other associated HTTPS links also.

Making Changes to a Unity Connection Location

The following changes are not supported on a Unity Connection location unless you unlink the location from the network. To make the changes in a Unity Connection location, you need to remove the location from the network, make the change, and add the location back to the network:

- Replacing the Unity Connection location or replacing the hard disk of the selected location.
- Changing the IP address of the Unity Connection location.
- Renaming the Unity Connection location.

To make any of these changes, do the following tasks:

1. Remove the Unity Connection location. See the "Removing a Location from an HTTPS Network" section.
2. Make the change on the Unity Connection location according to the instructions in the "Maintaining Cisco Unity Connection Server " chapter of the *Install, Upgrade, and Maintenance Guide for Cisco Unity Connection, Release 12.x*, available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/install_upgrade/guide/b_12xcuciumg.html.
3. Relink the Unity Connection location to the network. See the " Linking Unity Connection Servers with HTTPS Link " section.

Updating Directly Connected Nodes in HTTPS Networking for IP or Hostname Change

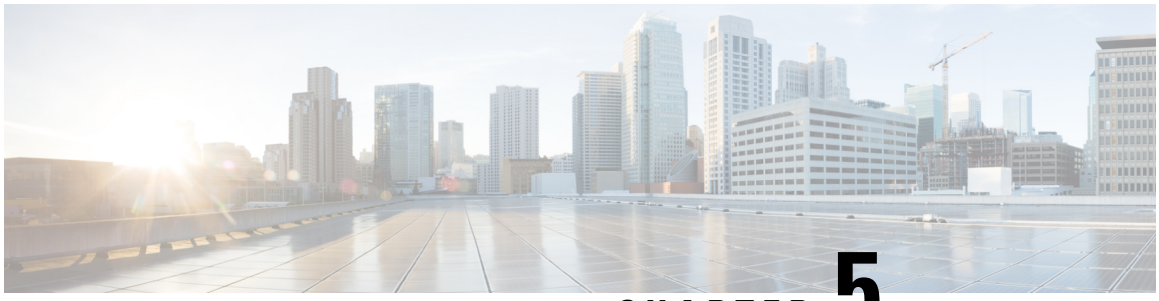
In case of IP or Hostname Change for a location, the IP or Hostname needs to be updated on all the directly connected nodes for the respective location in a HTTPS network.

Do the following procedures to update directly connected nodes in HTTPS networking for IP or Hostname change

Step 1 Find all directly connected nodes in the HTTPS network. Login to Cisco Unity Connection Administration and navigate to **Networking** and then select **HTTPS Links**.

Step 2 Login to Cisco Unity Connection Administration of all the nodes found in previous step.

Step 3 Navigate to **Netwoking** and select **HTTPS Links** and then modify the IP or Hostname.



CHAPTER 5

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

• Cross-Server Sign-In, Transfers, and Live Reply in HTTPS Networking, on page 43

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 4: Cross-Server Features.

Table 4: 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:

a) To enable cross-server sign-in to the remote location, check the **Allow Cross-Server Sign-In to this Remote Location** check box.

b) 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 12.0(1) 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 12.5(1) 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:

a) To enable cross-server transfer and live reply to the remote location, check the **Allow Cross-Server Transfer to this Remote Location** check box.

b) 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:

a) To enable cross-server transfer and live reply to the remote location, check the **Allow Cross-Server Transfer to this Remote Location** check box.

b) 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.

INDEX

C

cross-server features 43, 44, 55
  notable behavior 55
  overview 43
  search space considerations 44
cross-server live reply 52
  overview 52
cross-server sign-in 45
  overview 45
cross-server transfer 48, 50
  overview 48
  procedures 50