Getting Started with Cisco Hosted Unified Communication Services

Release 7.1(a)
August, 2010

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

Text Part Number: OL-23314-01

Getting Started with Cisco Hosted Unified Communication Services

CONTENTS

Preface

CHAPTER

This preface includes the following sections:

Overview

This guide explains how to implement Cisco Hosted Unified Communications Services (Hosted UCS) Release 7.1(a). It includes background information about the hardware and software components included in the Hosted UCS 7.1(a) platform and explains how these components fit together. It also provides a high-level overview of the procedures required to configure each component.

This document assumes that the high-level design, the low-level design, and the dial plan are complete.

Audience

This document is written for Cisco Advanced Services (AS), system integrators, Cisco partners, and Cisco customers who are interested in implementing Cisco Hosted UCS 7.1(a).

This document is to be used with the documentation for the individual components of the Hosted UCS 7.1(a) platform after completing the high-level design (HLD) and low-level design (LLD) for a specific customer implementation.

Organization

This document is organized as follows:

Chapter/Appendix	Description
Chapter 1, "Introducing Cisco Hosted Unified Communications Services"	Provides a high-level view of the architecture and overall operation of Cisco Hosted Unified Communications Services (Hosted UCS) 7.1(a).
Chapter 2, "Configuring Hosted Unified Communications Services Components Before Loading Bulk Data"	Describes the high-level tasks required to apply static configuration to Hosted UCS software components.
Chapter 3, "Managing the Hosted Unified Communications Services Platform with VisionOSS USM"	Summarizes the options provided by VisionOSS USM for managing the components of the Hosted UCS platform.
Chapter 4, "Using Bulk Loaders for the Initial Configuration of Hosted Unified Communication Services Components"	Explains how to perform the initial configuration of the Hosted UCS platform components by loading bulk data using VisionOSS USM.
Chapter 5, "Backing Up and Reinitializing Hosted Unified Communications Services Components"	Explains how to clear and reinitialize the components of a Hosted UCS platform and provides general recommendations for upgrading from previous versions.
Appendix A, "Sample Hosted Unified Communications Services Build of Materials"	Provides the standard bill of materials (BOM) for the Hosted UCS 7.1(a) platform.

Related Documentation

The following documentation provides additional information about the Hosted UCS 7.1(a) platform:

- *Release Notes for Cisco Hosted Unified Communications Services (Hosted UCS), Release 7.1(a)*
- *Software Support Matrix for Cisco Hosted Unified Communications Services (Hosted UCS), Release 7.1(a)*
- *Solutions Reference Network Design for Cisco Hosted Unified Communications Services (Hosted UCS), Release 7.1(a)*
- *Provisioning Guide for Cisco Hosted Unified Communications Services, Release 7.1(a)*

Obtaining Documentation and Technical Assistance

Cisco documentation and additional literature are available on Cisco.com. This section explains the product documentation resources that Cisco offers.

Cisco.com

You can access the most current Cisco documentation at this URL:

http://www.cisco.com/techsupport

You can access the Cisco website at this URL:

http://www.cisco.com

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Product Documentation DVD

The Product Documentation DVD is a library of technical product documentation on a portable medium. The DVD enables you to access installation, configuration, and command guides for Cisco hardware and software products. With the DVD, you have access to the HTML documentation and some of the PDF files found on the Cisco website at this URL:

http://www.cisco.com/univercd/home/home.htm

The Product Documentation DVD is created and released regularly. DVDs are available singly or by subscription. Registered Cisco.com users can order a Product Documentation DVD (product number DOC-DOCDVD= or DOC-DOCDVD=SUB) from Cisco Marketplace at the Product Documentation Store at this URL:

http://www.cisco.com/go/marketplace/docstore

Ordering Documentation

You must be a registered Cisco.com user to access Cisco Marketplace. Registered users may order Cisco documentation at the Product Documentation Store at this URL:

http://www.cisco.com/go/marketplace/docstore

If you do not have a user ID or password, you can register at this URL:

http://tools.cisco.com/RPF/register/register.do

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you will find information about how to do the following:

- Report security vulnerabilities in Cisco products

- Obtain assistance with security incidents that involve Cisco products
- Register to receive security information from Cisco

A current list of security advisories, security notices, and security responses for Cisco products is available at this URL:

http://www.cisco.com/go/psirt

To see security advisories, security notices, and security responses as they are updated in real time, you can subscribe to the Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed. Information about how to subscribe to the PSIRT RSS feed is found at this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you have identified a vulnerability in a Cisco product, contact PSIRT:

- For emergencies only — security-alert@cisco.com

  An emergency is either a condition in which a system is under active attack or a condition for which a severe and urgent security vulnerability should be reported. All other conditions are considered nonemergencies.

- For nonemergencies — psirt@cisco.com

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532

Tip We encourage you to use Pretty Good Privacy (PGP) or a compatible product (for example, GnuPG) to encrypt any sensitive information that you send to Cisco. PSIRT can work with information that has been encrypted with PGP versions 2.x through 9.x.

Never use a revoked encryption key or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one linked in the Contact Summary section of the Security Vulnerability Policy page at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

The link on this page has the current PGP key ID in use.

If you do not have or use PGP, contact PSIRT to find other means of encrypting the data before sending any sensitive material.

Document Conventions

This guide uses the following conventions to convey instructions and information:

Convention	Description
boldface font	Commands and keywords.
italic font	Variables for which you supply values.
[ ]	Keywords or arguments that appear within square brackets are optional.
{x | y | z}	A choice of required keywords appears in braces separated by vertical bars. You must select one.
`screen font`	Examples of information displayed on the screen.
`boldface screen font`	Examples of information you must enter.
< >	Nonprinting characters, for example passwords, appear in angle brackets.
[ ]	Default responses to system prompts appear in square brackets.

CHAPTER 1

Introducing Cisco Hosted Unified Communications Services

This chapter provides a high-level overview of the architecture and components of Cisco Hosted UCS, Release 7.1(a), describes applications and features, and defines high-level planning elements for Hosted UCS implementation. It includes the following sections:

Cisco Hosted Unified Communications Services Overview

Service providers can use a single Cisco Hosted UCS platform to offer Cisco Unified Communications Manager (Cisco Unified CM) applications and features to multiple customers (multi-tenant mode), or to one large customer (single-tenant mode).

Hosted UCS allows the sharing of the following centralized resources:

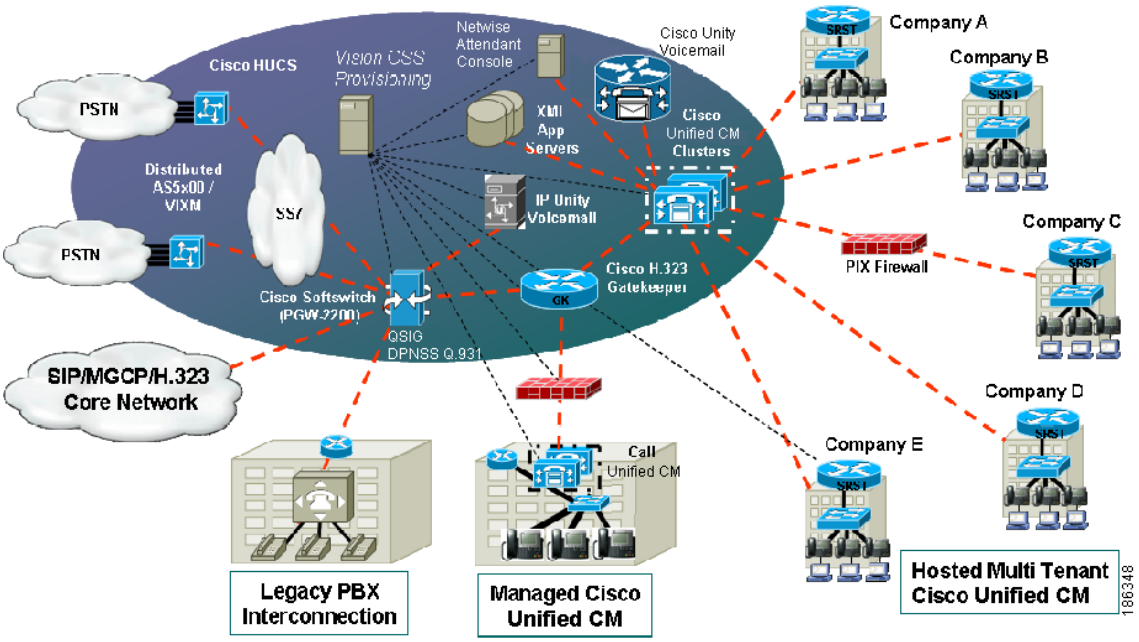
- Cisco PSTN Gateway 2200 Softswitch (PGW)
- Cisco Unified CM
- Trunking gateway
- Media resources

VOSS USM provides provisioning and administration for this multi-tenant solution.

Within a single Hosted UCS platform, a specific set of hardware resources can also be dedicated to a single customer if necessary to support the required level of service. For example, one or more Cisco Unified CM clusters or a Cisco Unity voice mail can be dedicated to a single customer, while other resources can be shared.

The voice network architecture for the Hosted UCS platform integrates the call control capability of a Cisco Unified CM system and the routing and services function of a Cisco PGW. This supports a broad range of Hosted UCS platform deployments. Figure 1-1 illustrates the Hosted UCS platform.

Figure 1-1 Hosted UCS Platform



Voice services for Hosted UCS tenants are provided by Cisco Unified CM and Cisco PGW platforms:

- Cisco Unified CM provides end-user-facing services to individual tenants
- Cisco PGW provides a routing function that mediates among tenants, and connection between the PSTN and each zone in the overall implementation

Virtualized Communications Services

The resources of the Cisco Unified CM system and the Cisco PGW can be shared among several tenants because VoSS USM partitions (*virtualizes*) resources for individual tenants. USM directly supports applications, such as directory services and extension mobility for IP phones, while shielding customers from the complexities of the underlying data structures and routing schema.

Table 1-1 summarizes the applications that are supported by Hosted UCS, Release 7.1(a), in single or multi-tenant mode, and indicates whether the application can be automatically provisioned through USM or if it must be manually provisioned.

Table 1-1 Multi-Tenant and Autoprovisioning Support for Hosted UCS Applications

Component	Multi-tenant?	Auto provision with USM?
Billing Applications (Building Applications Management; billing)	No	Yes[1]
Movius Mereon Voice Messaging (Voice Messaging only)	Yes	Yes
Movius Web Collaboration/Conferencing	Yes	Yes
XML Application Servers	Yes	Yes

1. Sharing customer history files with Mediation system, such as ISI

For more information about VoSS USM, see Chapter 3, "Managing the Hosted Unified Communications Services Platform with VisionOSS USM."

Hosted Unified Communications Services Platform Components

This section identifies and briefly describes the Hosted UCS platform components. It includes the following topics:

VoSS USM

\VoSS USM provides a global view of the Hosted UCS platform, and provides integrated provisioning of most of the major components, including:

- Cisco Unified CM
- Cisco PGW
- Cisco IOS Gatekeepers and local gateways (including SRST)
- VoSS DHCP server
- Movius voice mail and unified messaging system

USM provides the following features and functionality:

- Automates the deployment, provisioning, and management of large-scale, multi-tenant, multi-site hosted VoIP services across regional and international boundaries.
- Provides virtualized services by managing the configuration of the Cisco Unified CM and Cisco PGW dial plans and analog gateways, including multiple versions of code on integrated clusters.
- Supports dial plan management across Cisco Unified CM and the Cisco PGW and allows rapid service activation. Service activation includes configuring dial tone, voice mail, conferencing, corporate directories, and XML applications.
- Provides a comprehensive resource inventory management tool for IP addresses, internal numbers and E.164 external numbers. USM also provides a basic inventory management system for phones, lines, and services.

- Ensures that changes in configuration in one Hosted UCS platform component are mapped across other affected components. USM manages multiple network elements, performing multiple configuration steps for each transaction.
- Supports secure, decentralized administration, which allows customers to perform their own moves, adds, and changes.

For more information about VoSS USM, see the following URL: http://www.voss-solutions.com/

Cisco Unified Communications Manager

In the Hosted UCS platform, Cisco Unified CM provides business IP telephony services to enterprises located within the bounds of a shared voice infrastructure. The Cisco Unified CM is a hosted or managed device that can be partitioned in a multi-tenant manner to provide segregated service to multiple enterprises of all sizes or in a dedicated manner to support a single large-scale enterprise. A Cisco Unified CM cluster or clusters can be deployed within a network provider domain to provide service to IP phones located at an end-user facility.

HUCS 7.1(a) adds the Cisco Unified Mobility application known as Mobile Connect or Single Number Reach (SNR) application that allows an incoming call to an enterprise user to be offered to the user's IP desk phone as well as up to four configurable remote destinations, such as mobile or cellular telephone. Once the call is offered to both the desktop and remote destination phone(s), the user can answer at any of those phones. Upon answering the call on one of the remote destination phones or on the IP desk phone, the user has the option to hand-off or pick-up the call on the other phone.

For more information on Cisco Unified CM, see the following URL:
http://www.cisco.com/en/US/products/sw/voicesw/ps556/index.html

Cisco PGW

The Cisco PGW provides the following key functions within the Hosted UCS platform:

- Time-division multiplexing (TDM) PSTN interconnect—Provides connectivity for all services to the TDM-based PSTN via Signaling System 7 (SS7) or ISDN Primary Rate Interface (PRI), depending on the requirements of the installation. The Cisco PGW also incorporates some capabilities that can assist in meeting local regulatory requirements.
- Business voice access—Provides the business voice access service for TDM PBXes and IP PBXs.
- TDM PBX integration—Provides direct management by the Cisco PGW of PBXes that use PRI-based signaling interfaces, using Media Gateway Control Protocol (MGCP) with backhaul techniques, or can be managed indirectly using the H.323 protocol to the Cisco PGW for both PRI and Basic Rate Interface (BRI). The direct connection model typically offers greater flexibility and functionality.
- Routing and analysis engine—Provides a routing engine for inter-domain routing. All service platforms use the Cisco PGW to route calls that are not local, which ensures that the main dial plan and routing functions for the hosted platform are centrally located. The Cisco PGW also includes A and B number analysis and modification functions, as well as regulatory capabilities that can be applied to satisfy local requirements.

For more information about the Cisco PGW, see the following URL:
http://www.cisco.com/en/US/products/hw/vcallcon/ps2027/index.html

Cisco H.323 Signaling Interface

The Cisco H.323 Signaling Interface (HSI) adds an H.323 interface to the Cisco PGW, which allows calls to be established between the PSTN and an H.323 network. The Cisco H.323 HSI provides these services:

- Translation of signaling protocols for establishing, controlling, and releasing calls
- Administration of network parameters and protocol capabilities
- System and call-related statistics
- Fault reporting
- Overload management
- Event logging
- Simple Network Management Protocol (SNMP) interface

The Cisco HSI operates in a load-sharing configuration, while the Cisco PGW operates in an active/standby configuration. This operation allows the benefits of redundancy (if an HSI fails, the remaining HSIs continue to operate) and simple scaling (you can insert additional HSIs as the network expands). A minimum of two HSIs are required to ensure that the system continues to process calls in case of an equipment failure.

For more information about the Cisco HSI, see the following URL:
http://www.cisco.com/en/US/docs/voice_ip_comm/pgw/hsi/4.3/guide/43ch1.html

Gatekeeper

An H.323 gatekeeper provides basic infrastructure capabilities as well as a registration capability for the Cisco PGW, Cisco Unified CM, H.323 customer premises equipment (CPE), and any H.323 customer devices. The gatekeeper forces all call signaling to use the Cisco PGW.

Trunking Gateway

The trunking gateway is a Cisco AS5x00 platform in the baseline architecture, which is based on Cisco IOS software. A Cisco AS5850 with STM-1 and E1 interfaces and a Cisco AS5350 with E1 interfaces were used in Cisco Hosted UCS tests.

However, because of the requirements of the initial applications, it is relatively easy to incorporate the Cisco MGX, Cisco Voice Internetworking Service Module (VISM), and Cisco Voice Switch Service Module (VXSM) products as needed.

For more information on Cisco AS5x00 platforms, see the following URL:
http://www.cisco.com/en/US/products/hw/iad/index.html

Movius Voice Mail

The Movius servers (formerly IP Unity) Unified Messaging platform provides voice mail services in a multi-tenant environment and the Movius voice mail system supports the multi-tenant Hosted UCS architecture. Only the voice mail functions of the Movius server platform are currently used in the Hosted UCS platform.

The USM platform is integrated with the Movius server to allow provisioning through the USM voice mail graphical user interface (GUI) functionality on a per-tenant basis. The interface between the Cisco PGW and the Movius server (IP Unity) voice mail system is SIP.

The Cisco PGW forwards the incoming calls to the voice mail system. After the caller leaves a message, the voice mail system uses the SIP NOTIFY message to notify the Cisco PGW that a message was left for the user.

The Cisco PGW supports only unsolicited subscription to the voice mail system and does not need to send a SIP SUBSCRIBE message to the voice mail system for every user with voice mail service enabled.

The Cisco PGW inter-works SIP and H.323 between the Movius server and Cisco Unified CM for message deposit, retrieval, and message waiting indicator (MWI). The inter-working of the Movius server through SIP and gateway-fronted Digital Private Network Signaling System (DPNSS)/Q Interface Signaling Protocol (QSIG) PBX is not supported in the Hosted UCS 7.1(a) release, so it is not possible to provide a hosted voice mail service for time division multiplexing (TDM) PBX users.

The USM platform uses the Movius server application programming interface (API), which is Common Object Request Broker Architecture (CORBA)/Extensible Markup Language (XML), to define business groups, provision pilot numbers, add/delete mailboxes assigned against a unique "internal" number and an "extension" number, and to assign class of service.

For further information about Movius server, see the following URL:
http://www.moviuscorp.com/ourofferings/platformsandservers

HUCS 7.1(a) adds the Movius Auto Attendant Integration feature which enables PGW and Cisco Unified Communication Manager to route calls to the Movius Auto Attendant from any PSTN or Hosted UCS phone. After selecting an Auto Attendant menu option, these calls can be transferred to another Hosted UCS or PSTN phone.

Billing

In multi-tenant business voice service, call detail records (CDRs) are generated by Cisco Unified CM system and the Cisco PGW. The Cisco Unified CM generates records for calls between phones in the same tenant. The Cisco Unified CM and the Cisco PGW generate records for calls between different tenants and for calls to the PSTN.

Cisco PGW billing records are produced by a Cisco Billing And Measurement Server (BAMS), which observes carrier-class accuracy and contains all necessary timestamps as well as the called party information delivered to the Cisco PGW and the calling party information sent from the Cisco PGW.

BVSM also provides a source of customer history data accessible through an ODBC link to allow Mediation systems to query BVSM for customer data. For example, this can be used to establish the ownership of a telephone number at a given time and to access customer account data.

For more information on Cisco BAMS, see the following URL:
http://www.cisco.com/en/US/docs/voice_ip_comm/pgw/bams/3.13/guide/3132ch1.html

Application Servers

The Hosted UCS platform includes a number of application services, including voice mail, unified messaging, conferencing, music-on-hold (MOH), auto attendant, and XML applications.

Business CPE

Cisco IOS CPE, such as Cisco 2600, 2800, 3600 and 3800 Series routers, provide ISDN PRI and BRI connections to business tenant telephony equipment. For PRI connections, the Cisco PGW can be used directly via D-channel backhaul and MGCP control. For BRI connections (and also for PRI if necessary), gateways can be configured as H.323 endpoints that register with an infrastructure gatekeeper.

Cisco Unified IP phones and analog telephone adaptors (ATAs) are used to provide devices in the business domain where Cisco Unified CM control is used. Cisco Integrated Access Devices (IADs) can also be used to provide analog telephony service to multi-dwelling/multi-office facilities.

Cisco Emergency Responder

The Cisco Emergency Responder (Cisco ER) can be used to manage emergency calls in the telephony network to respond to these calls effectively, so that the service provider can comply with local ordinances concerning the handling of emergency calls.

In North America, these local ordinances are called "enhanced 911," or E911. Other countries and locales might have similar ordinances. Cisco ER is predominantly deployed in North America.

Planning the Hosted Unified Communications Services Implementation

This section provides high-level guidance for planning a Hosted UCS implementation.

This section includes the following topics:

Note Hosted UCS is a Cisco end-to-end solution architecture. The official Hosted UCS design is fully described in the Hosted UCS platform Solution Reference Network Design (SRND) documentation.

Planning Tasks

Before you begin building a Hosted UCS platform, Cisco recommends that you first complete these related planning tasks:

1. Create a bill of materials (BoM) that covers all equipment, and ensure that the correct software is available.
2. Create an architecture diagram, including a rack diagram.
3. Plan the component naming convention.
4. Plan the IP addressing and create a network design (subnets and VLANs).

5. Plan the classes of service.
6. Plan the dial plan requirements.
7. If SS7 connectivity is required, gather the SS7 PSTN interconnect information; for example, point codes, linksets, links, and CIC information.
8. Develop a set of USM bulk loaders.

Design Workbook

It is recommended that you maintain a design workbook document for each Hosted UCS platform. A sample Excel design workbook is available from VoSS USM and contains the various components described in this chapter.

Note Maintaining a design workbook document is a mandatory requirement for all Hosted UCS reference and production platforms.

Dial Plan

The Hosted UCS platform provides a standard dial plan model for common scenarios, such as service provider and multi-tenant. However, the planning and design of a customized, multi-tenant dial plan configuration is a critical requirement.

Dial plan customization includes the following:

- Inter-site calling prefix
- Outside calling prefix
- Location extension digits
- Site code digits
- Emergency number conventions
- E911 requirements
- DID/DDI number length
- DID/DDI number range allocation
- DDI/internal association format
- Area codes
- PSTN number requirements

Standard Equipment Naming Conventions

During the planning phase, you must define the naming convention to use for your equipment. Because of the integrated nature of the architecture, equipment names must be consistent across the platform.

The Cisco Unified CM server computer name is limited to 11 characters because the MOH name, which includes the Cisco Unified CM server name, must be a maximum of 15 characters. USM cannot statically configure Cisco Unified CM clusters where the server names are more than 11 characters, and this prevents the Hosted UCS platform from being configured further on the affected cluster.

Note The Cisco Unified CM server computer name is restricted to a maximum of 11 characters or less and cannot be altered later without a complete software reload (on Cisco Unified CM 4.x clusters).

Hardware, Network, and Software Requirements

This section summarizes the main hardware and software requirements for implementing the Hosted UCS platform. It includes the following topics:

Hardware Requirements

Before implementing Hosted UCS in a production environment, refer to the high-level and low-level detailed design guide and the build of materials (BOM) for the hardware requirements for a specific deployment. Refer to Appendix A, "Sample Hosted Unified Communications Services Build of Materials" to see a sample build of materials.

Each platform employs a unique rack layout that is dependent on the special requirements of each deployment. For instructions to install specific hardware components, refer to the hardware installation guide for each component.

Network Configuration and NAT

After the devices have been physically rack mounted, the network cabling can be completed by using a suitable Layer 3 switch such as the Cisco Catalyst 3560 Series switch. VLANs and subnets should be defined based on the Hosted UCS platform high-level and low-level design and your network configuration.

When the platform is implemented into a service provider network, it is also important to record the external IP address scheme used by Network Address Translation (NAT), so that remote access to the platform is also possible.

Customers sharing a single Hosted UCS-based service provider-hosted service may have internal IP address space ranges that overlap. NAT can be configured on the firewalls to translate the private (non-unique) addresses that are actually on the IP phones into addresses that are unique in the service provider (shared) domain.

Not only must these addresses be unique, they must be reachable from the individual customer networks to allow voice calls to flow between customers through the common address and security domain. To achieve this, routes that represent the address scheme used in the common address and security domain must be injected to each customer domain.

It is therefore important that no customer subscribing to the service is actually using the address space chosen for the IP address and security domain of the common service provider. The NAT pools that are configured on each customer firewall must be large enough to supply addresses to all the IP phones that are deployed in the specific customer.

Firewall Rules

Table 1-2 shows an example of the rules that can be implemented on the firewall to protect both the service provider from the customer and the customers from each other.

Table 1-2 Firewall Rules

Rule	Function
SCCP (TCP port 2000) to the Cisco Unified CMs only	Allows the Cisco Unified CMs to control the phones in the customer domain using the SCCP protocol
TAPI (CTIQBE) to the Cisco Unified CMs running CTI manager only	Used for third-party phone call control or for call control to TAPI-based softphones or software applications
HTTP (TCP port 80) to the Publisher Cisco Unified CMs and USM only	Required for access to phone XML services hosted on the Cisco Unified CM and USM (for example, directory), and also for customer self-provisioning of USM using a web browser
TFTP (UDP port 69) to the TFTP server only	Required to allow phones to download their configuration files and software updates
H.323 (and H.245), to the Cisco Unified CM and maybe the HSI and gatekeeper if customer site applications that use H.323 are required; for example, a customer site-located PSTN gateway using H.323	Required only to support H.323 endpoints in the customer address space; applications for this include site-located PSTN gateways
RTP traffic UDP ports are opened dynamically by the ALG function within the firewall by MGCP, H.323, TAPI, SIP, and SCCP Call Control	Allows voice to flow between customers and to PSTN gateways and conference bridges hosted in the common domain
MGCP (UDP 2427/2428) to the Cisco PGW Various backhaul protocols also need to be allowed to the Cisco PGW depending on the L3 protocol at the gateway; for example, Sigtrans.	Allows the Cisco PGW to control customer site-located PSTN and PBX gateways

For more information about NAT and firewall issues when implementing the Hosted UCS platform, refer to the *Hosted Unified Communications Services, Release 7.1(a) SRND*.

Class of Service

You must define each class of service (CoS) and the naming convention to be used in the dial plan.

Table 1-3 shows an example of CoS and naming conventions.

Table 1-3 CoS and Naming Conventions

Phone Group	Proposed CoS	
	Service Name	**Description**
Unassigned	COS1InternalOnlyNo911	Internal access only (no 911 calls)
Unsecured common area	COS2AllCallsCMCAllButInternal	Internal + 911 + CMC all other calls
Secured common area	COS3AllCallsCMCInternational	All calls allowed (CMC for international)
User (client)	COS4AllCalls(NotIntersite)	All calls allowed (no 9-digit inter-site)
User (service provider)	COS5AllCalls	All calls allowed (+ 9-digit inter-site)

Software Requirements

This section summarizes the software requirements for the core components of the Hosted UCS platform. For information about software compatibility for all the supported platform components, refer to the *Hosted Unified Communications Services, Release 7.1(a)*, *Software Compatibility Matrix*.

Implementation and Configuration Summary

The following summarizes the basic tasks required to implement and configure the Hosted UCS platform, after completing the initial design and planning phase.

Implementation Step	Refer to
1. Install hardware and initialize device software.	Installation and hardware guides for each Hosted UCS platform component. The *Cisco Hosted Unified Communications Services, Release 7.1(a) Software Compatibility Matrix* summarizes the software requirements for each component.
2. Apply static configuration to each device.	Chapter 2, "Configuring Hosted Unified Communications Services Components Before Loading Bulk Data."
3. Load bulk data for each component.	Chapter 3, "Managing the Hosted Unified Communications Services Platform with VisionOSS USM."
4. Customize each component as necessary.	Configuration guides or online help for each Hosted UCS platform component. Chapter 3, "Managing the Hosted Unified Communications Services Platform with VisionOSS USM" summarizes the options provided by USM for configuring the Hosted UCS platform components.

CHAPTER 2

Configuring Hosted Unified Communications Services Components Before Loading Bulk Data

This chapter describes the manual configuration and static configuration scripts that must be applied to each Hosted Unified Communications Services Components (Hosted UCS) platform component before the component can be integrated into the Hosted UCS platform by using USM to load bulk data. For the supported Hosted UCS platform components and software versions, which are required for full support and functionality, refer to the *Cisco Hosted Unified Communications Services, Release 7.1(a) Software Compatibility Matrix*.

For information about configuring the components using USM to load bulk data, refer to Chapter 4, "Using Bulk Loaders for the Initial Configuration of Hosted Unified Communication Services Components."

This chapter includes the following sections:

Manually Configuring Cisco Unified Communications Manager

This section explains how to manually configure and verify the Cisco Unified CM parameters that cannot be provisioned through AVVID XML Layer Simple Object Access Protocol (AXL SOAP).

This section includes the following topics:

Cisco Unified Communications Manager Configuration

Some Cisco Unified CM configuration parameters cannot be provisioned through AXL SOAP. Due to this limitation, some manual configuration is required on the Cisco Unified CM servers before loading bulk data through USM.

This section describes the elements that must be configured using the Cisco Unified CM publisher. It includes the following topics:

Unless specifically mentioned, the configuration steps apply to Hosted UCS platforms based on *Cisco Unified CM 4.2(3), Cisco Unified CM 5.1(3),* Cisco Unified CM 6.1(x) and Cisco Unified CM 7.1(x).

Server Identity

This section describes how to specify identify the Cisco Unified CM server for integrating it into a Hosted UCS Release 7.1(a) platform.

To identify the Cisco Unified CM server, complete the following steps.

Procedure

Step 1 Go to **System > Server.**

Step 2 In the **Host Name/IP Address** field, enter the full IP address of the server.

For example, *10.131.4.2*.

Step 3 In the **Description** field, enter a description of the server.

For example, *e4c1p*

Step 4 Repeat Step 1 through Step 3 for each Cisco Unified CM in each cluster.

Date/Time Groups

Date/Time Groups define the time zones for the various devices that are connected to Cisco Cisco Unified CM. Each device exists as a member of only one device pool, and each device pool has only one assigned Date/Time Group. USM uses the international standard zoneinfo database, also called the tz database. In USM, the timezone names are all in the form *area/location*, where *area* is the name of a continent or ocean, and *location* is the name of a specific location (typically a city or small island) within a large region, such as “America/New_York.”

To configure Date/Time groups, complete the following steps.

Procedure

Step 1 Go to **System > Date/Time Group**.

Step 2 Choose the correct values for each of the following settings:

- **Group Name**—Enter the name that you want to assign to the new date/time group; for example, *Europe-London*

- **Time Zone**—From the drop-down list box, choose the time zone for the group that you are adding, for example, *GMT Standard/Daylight Time*
- **Separator**—Choose the separator character to use between the date fields; for example, **/**
- **Date Format**—Choose the date format for the date that appears on Cisco Unified IP Phones; for example, *D/M/Y*
- **Time Format**—Choose a 12-hour or 24-hour time format; for example, *24-hour*

Step 3 Repeat Step 2 for all required Date/Time Groups.

Note Group name format in USM is "Area/Location," (using a slash as the separator) while in Cisco Unified CM the format is "Area-Location" (using a dash as the separator).

Enterprise Parameters

Enterprise parameters define the default settings that apply to all devices and services in the same cluster.

To define the Enterprise parameters, complete the following steps.

Procedure

Step 1 Go to **System > Enterprise Parameters**.

Note In the **Enterprise Parameters Configuration** section, note that the **Advertise G722 Codec field** is disabled; this applies to Cisco Unified CM 6.1(4) and Cisco Unified CM 7.1(3)

Step 2 Define the correct value for the following setting in the **Phone URL Parameters** section:

URL Directories—This parameter specifies the URL that Cisco Unified IP Phone models use when you click the Directory button. This should point to the virtual IP address of the BVSM server (*not* the Cisco Unified CM server).

- For Cisco Unified CM 7.1(3) and Unified CM 6.1(4):

  http://virtual_IP_address_of_USM_server/bvsmweb/directoryservices.cgi?device=#DEVICENAME

  For example:
  http://10.120.3.66/bvsmweb/directoryservices.cgi?device=#DEVICENAME#

- For Cisco Unified CM 5.1(3):

  http://virtual_IP_address_of_USM_server:8080/bvsmweb/directoryservices.cgi?device=#DEVICENAME#

  For example:

  http://10.120.3.66:8080/bvsmweb/directoryservices.cgi?device=#DEVICENAME#

Step 3 If the Hosted UCS platform network does not use DNS services, replace the host name of the Cisco Unified CM Publisher Server name with its IP address in the following fields:

- In the **URL Help field**, in the General Parameters section:

  For Cisco Unified CM 7.1(3) and Unified CM 6.1(4):

http://*IP_address_of_Publisher_server*/help

For example, *http://10.131.4.2/help*

- In the **CDR UNC Path** field, in the CDR Parameters section:

  For Cisco Unified CM 7.1(3) and Unified CM 6.1(4):

  IP_address_of_Publisher_server\CDR

  For example, *http:\\10.131.4.2\CDR*

- In the **URL Authentication**, **URL Directories**, and **URL Services** fields, in the Phone URL Parameters section:

  For Cisco Unified CM 7.1(3) and Unified CM 6.1(4):

  - http://*IP_address_of_Publisher_server*/CCMCIP/authenticate.asp

    For example, *http://10.131.4.2/CCMCIP/authenticate.asp*

  - http://*IP_address_of_Publisher_server*/CCMCIP/GetTelecasterHelpText.asp

    For example, *http://10.131.4.2/CCMCIP/GetTelecasterHelpText.asp*

  - http://*IP_address_of_Publisher_server*/CCMCIP/getservicesmenu.asp

    For example, *http://10.131.4.2/CCMCIP/getservicesmenu.asp*

- For Cisco Unified CM 5.1(3):

  - http://*IP_address_of_Publisher_server*:8080/ccmcip/authenticate.jsp

    For example, *http://10.132.4.2:8080/ccmcip/authenticate.jsp*

  - http://*IP_address_of_Publisher_server*:8080/ccmcip/GetTelecasterHelpText.jsp

    For example, *http://10.132.4.2:8080/ccmcip/GetTelecasterHelpText.jsp*

  - http://*IP_address_of_Publisher_server*:8080/ccmcip/getservicesmenu.jsp

    For example, *http://10.132.4.2:8080/ccmcip/getservicesmenu.jsp*

Automated Alternate Routing Group

Automated alternate routing (AAR) provides a mechanism to reroute calls through the PSTN or other network by using an alternate number when Cisco Unified CM blocks a call due to insufficient location bandwidth.

To define the AAR parameter, complete the following steps.

Procedure

Note This procedure applies only to Cisco Unified CM 7.1.x, Cisco Unified CM 6.1.x and Cisco Unified CM 5.1(3) because in Cisco Unified CM 4.2(3) this parameter is provisioned through USM.

Step 1 Go to **Call Routing > AAR Group**.

Step 2 Specify the correct values for the following parameters:

AAR Group Name: `defaultaar`

Prefix Digits: Leave this blank

Note The `defaultaar` AAR group is not required by Cisco Unified CM 5.x. However, when the bulk data is loaded for the cluster, BVSM returns an error message and stops the bulk data loading if the parameter is missing.

Conference Bridge

Note This is an optional step, and is only required if Conference Bridges are deployed in the network

Conference bridge configuration is only required if one or more conference bridges are to be integrated into the Hosted UCS platform.

Conference Bridge for Cisco Unified CM can be implemented as a software or hardware application. It allows both ad hoc and meet-me voice conferencing. Each conference bridge can host several simultaneous, multiparty conferences. For details about how to configure the conference bridge for integration into the Hosted UCS 7.1(a) platform, refer to the following websites:

- For Unified CM 6.1(x)
  http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/admin/6_1_1/ccmcfg/b04cnbrg.html
- For Unified CM 7.1(x)
  http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/admin/7_1_2/ccmcfg/b04cnbrg.html

Transcoder

Note This is an optional step, and is only required if Transcoders are deployed in the network.

Transcoder configuration is only required if one or more transcoders are to be integrated into the Hosted UCS platform. A transcoder takes the stream of one codec and transcodes (converts) it from one compression type to another compression type.

The Cisco Unified CM invokes a transcoder on behalf of endpoint devices when the two devices are using different codecs and would normally not be able to communicate. When inserted into a call, the transcoder converts the data streams between the two disparate codecs to enable communications between them.

The Media Resource Manager (MRM) has responsibility for resource registration and resource reservation of transcoders within a Cisco Unified CM cluster. Cisco Unified CM simultaneously supports registration of both the Media Termination Point (MTP) and transcoders and concurrent MTP and transcoder functionality within a single call.

For details about how to configure a transcoder for integration into the Hosted UCS 7.1(a) platform, refer to the following websites:

- For Unified CM 6.1(x)
  http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/admin/6_1_1/ccmcfg/b04trans.html
- For Unified CM 7.1(x)
  http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/admin/7_1_2/ccmcfg/b04trans.html

Cisco Unified IP Phone Services

Users can subscribe to Cisco Unified IP Phone Services at each site, which vary depending on the deployment. This section describes the manual configuration required for the following services:

- Login/Logout Services for Extension Mobility, page 2-6
- Roaming Login/Logout Services for BVSM User Roaming, page 2-6
- Cisco Unified IP Phone XML Services, page 2-7

Note Login/Logout Services for extension mobility are used if only one customer is to be provisioned per cluster. If this is not the case then only Roaming Login/Logout Services for USM user roaming will be used.

Login/Logout Services for Extension Mobility

To configure Login/Logout Services for Extension Mobility, complete the following steps.

Procedure

Step 1 Choose one of the following options:

- For Cisco Unified CM 4.2(3):

  Go to **Feature > Cisco Unified IP Phone Services**
- For Cisco Unified CM 7.1(x) and 6.1(x):

  Go to **Device > Device Settings > Phone Services**

Step 2 Add the Cisco Unified IP Phone Service using the following settings:

- For Cisco Unified CM 4.2(3):

  Service Name: `Login/Logout`

  Service Description: `Extension Mobility Service`
- **Service URL**:
  http://USM_Virtual_IP_Address/bvsmweb/bvsmroaming.cgi?device=#DEVICENAME#
  For example, http://10.120.3.62/bvsmweb/bvsmroaming.cgi?device=#DEVICENAME#
- For Cisco Unified CM 5.1(3) and 6.1(2):

  Service Name: `Login/Logout`

  Service Name (ASCII Format): `Login/Logout`

  Service Description: `Extension Mobility Service`

  Service URL:
  http://USM_Virtual_IP_Address/bvsmweb/bvsmroaming.cgi?device=#DEVICENAME#
  For example, http://10.120.3.62/bvsmweb/bvsmroaming.cgi?device=#DEVICENAME#

Roaming Login/Logout Services for BVSM User Roaming

Roaming Login/Logout Services for USM User Roaming uses Cross Cluster Forwarding, and allows users to login or logout at remote locations.

To configure Roaming Login/Logout Services for USM user roaming, complete the following steps.

Procedure

Step 1 Choose one of the following options:

- For Cisco Unified CM 4.2(3):

  Go to **Feature > Cisco Unified IP Phone Services**
- For Cisco Unified CM 7.1(x) and 6.1(x):

  Go to **Device > Device Settings > Phone Services**

Step 2 Specify the appropriate values for the following settings:

- For Cisco Unified CM 4.2(3):

  Service Name: `Roaming Login/Logout`

  Service Description: `Extension Mobility Service`

  Service URL:
  http://*USM_Virtual_IP_Address*/bvsmweb/bvsmroaming.cgi?device=#DEVICENAME

  For example, *http://10.120.3.62/bvsmweb/bvsmroaming.cgi?device=#DEVICENAME#*
- For Cisco Unified CM 7.1x3) and 6.1(x):

  Service Name: `Roaming Login/Logout`

  Service Name (ASCII Format): `Roaming Login/Logout`

  Service Description: `Extension Mobility Service`

  Service URL:
  http://*USM_Virtual_IP_Address*/bvsmweb/bvsmroaming.cgi?device=#DEVICENAME

  For example, *http://10.120.3.62/bvsmweb/bvsmroaming.cgi?device=#DEVICENAME#*

Step 3 Check the **Enable** Check box.

Cisco Unified IP Phone XML Services

This section describes the configuration required to enable Cisco Unified IP Phone XML Services.

To configure Cisco Unified IP Phone XML Services, complete the following steps.

Procedure

Step 1 Choose one of the following options:

- For Cisco Unified CM 4.2(3):

  Go to **Feature > Cisco Unified IP Phone Services**
- For Cisco Unified CM 7.1(x) and 6.1(x):

  Go to **Device > Device Settings > Phone Services**

Step 2 Specify the appropriate values for the following settings:

- For Cisco Unified CM 4.2(3):

Service Name: `Phone Services`

Service Description: `Phone Services`

Service URL:
http://*USM_Virtual_IP_Address*/bvsmweb/bvsmservices.cgi?device=#DEVICENAME

For example, *http://10.120.3.62/bvsmweb/bvsmservices.cgi?device=#DEVICENAME#*

- For Cisco Unified CM 7.1(x) and 6.1(x):

  Service Name: `Phone Services`

  Service Name (ASCII Format): `Phone Services`

  Service Description: `Phone Services`

  Service URL:
  http://*USM_Virtual_IP_Address*/bvsmweb/bvsmservices.cgi?device=#DEVICENAME

  For example, *http://10.120.3.62/bvsmweb/bvsmservices.cgi?device=#DEVICENAME#*

Template Configuration

This section describes the manual configuration required to enable the following templates that are provided by Cisco Unified CM:

- Phone Button Template, page 2-8
- Softkey Template, page 2-9

Phone Button Template

Cisco Unified CM includes several default phone button templates. When adding phones, you can assign one of these templates to the phones or create a new template. Creating and using templates provides a fast way to assign a common button configuration to a large number of phones. A number of default phone button templates are loaded into USM during initial setup.

If you want to use any non-standard phone button templates, you need to define them in USM, and also need to add them manually into Cisco Unified CM. This section describes how to add non-standard phone button templates.

Ensure that the required number of lines are set up on the template settings and that each phone button template is cloned from the standard phone type for each variant. For example, confirm that the Standard 7960-2line is based on the standard 7960 template, and set the number of lines to 2.

Note The USM InitPBX Load fails if Phone Button Templates exists as a BVSM Service Setting but does not exist in the Cisco Unified CM. If a validation failure occurs, either add the missing phone button template into the Cisco Unified CM, or delete unnecessary phone button templates from USM. In USM 7.x, phone button templates can not be deleted from USM database without disconnecting the dial plans from the harware sets (In USM, Dial Plan Tools > Hardware Sets > Associated Dial plans). Remember to reconnect the required dial plans afterwards.

USM does not distinguish between different Cisco Unified CM versions, and therefore during USM InitPBX Load it verifies that all phone button templates configured in BVSM also exist in Cisco Unified CM. The USM BaseData worksheet for the 7.1(a) model loader lists all the phone button templates that need to be added to Cisco Unified CM clusters.

To configure non-standard phone button templates, complete the following steps.

Procedure

Step 1 Go to **Device > Device Settings > Phone Button Template**.

Step 2 Specify the appropriate values for the following parameters:

- **Phone Button Template**: *required_phone_button_template*

  For example, *Standard 7960*

  Use the following setting:

- **Button Template Name**: *unique_button_template_name*

  For example, *Standard 7960-2lines*

Step 3 Configure the required number of buttons:

- **Feature**: Choose the function of the phone button that you want to specify in the template

  For example, *Line*

- **Label**: Enter a description of the button

  For example, *Line 1*

Softkey Template

Softkey template configuration allows the administrator to manage softkeys that the Cisco Unified IP Phones (such as model 7960) support.

By default two softkey templates are configured in USM: Softkey_Basic and Softkey_Advanced. If these templates are not required, they can be deleted from the USM (in **Setup Tools > Service Types**). The list of Softkey templates in USM has to correspond with the list of Softkey Templates in all Cisco Unified CM clusters.

To configure a softkey template, complete the following steps.

Procedure

Step 1 To add a softkey template to USM, go to **Setup Tools > Service Types**.

Step 2 To add Softkey templates to Cisco Unified CM clusters, go to **Device > Device Settings > Softkey Template**.

Step 3 Create a softkey template based on: *available_softkey_template*.

For example, *Standard User*

Step 4 Configure the following setting:

Softkey Template Name: *unique_softkey_template_name*

For example, *Softkey_Advanced*

Step 5 After creating the new softkey template, add additional application softkeys, and configure softkey positions.

For detailed instructions refer to the appropriate Cisco Unified CM administration guide.

Note If different softkey templates are required on different clusters, load the softkey templates after loading the Cisco Unified CM.

Verifying Cisco Unified Communications Manager Configuration

This section describes how to verify that the Cisco Unified CM cluster configuration, required before loading the base data using USM, is complete.

Note Confirm that the settings are correct.

To verify the Cisco Unified CM configuration, complete the following steps.

Procedure

Step 1 Go to **Publisher – Tools > Service Activation** and turn on the following services:

- Cisco RIS Data Collector
- Cisco Database Layer Monitor
- Cisco Serviceability Reporter
- Cisco Extension Mobility
- Cisco Unified CM (if Publisher used as a back-up subscriber)
- Cisco TFTP (if required on Publisher server)
- Cisco IP Voice Streaming (if required for MoH and announcements)
- Cisco CTIManager (if used as a backup subscriber)

Step 2 Go to **Subscriber/TFTP/MOH Server – Tools > Service Activation** and turn on the following services:

- Cisco RIS Data Collector
- Cisco Database Layer Monitor
- Cisco Serviceability Reporter
- Cisco Telephony Call Dispatcher
- Cisco Unified CM (if used as a subscriber)
- Cisco TFTP (if required)
- Cisco IP Voice Streaming (if required for MoH and announcements)
- Cisco CTIManager (if used as a subscriber)
- Cisco Extended Functions (on multi-tenant clusters, not large enterprise)

Step 3 Verify the following from the Cisco Unified CM Publisher Administration **System** menu:

- Cisco Servers—Must be IP addresses (not hostnames).
- Cisco Unified CM(s)—Should be IP addresses or hostnames.
- Cisco Unified CM Groups—Should show default.
- Date/Time Group—Should include DTG required by USM (set within Cisco Unified CM model loader under global settings). For example, CMLocal, Europe-London, Europe-Copenhagen, America-New_York, or as defined by the project team.
- Region—Should show default.
- Device Pools—Should show default.
- Enterprise Parameters—Set "Enable Dependency Records" to "True" to support dependency record function (on test platforms).
- Confirm that IP addresses rather than hostnames are used in the Phone URL Parameters.
- Also confirm that the Directories URL points to the bvsmweb service at the virtual IP address of the USM cluster (not the Publisher).

Note the following example, where Publisher = 10.10.1.3 and USM-VIP = 10.10.6.16:

- URL Authentication—*http://10.10.1.3/CCMCIP/authenticate.asp*
- URL Directories— *http://10.10.6.16/bvsmweb/directoryservices.cgi?device=#DEVICENAME#*
- URL Information—*http://10.10.1.3/CCMCIP/GetTelecasterHelpText.asp*
- URL Services—*http://10.10.1.3/CCMCIP/getservicesmenu.asp*

Step 4 Verify the following from the Cisco Unified CM Publisher Administration **Route Plan** menu:

a. AAR Group—Should include "defaultaar" (supports alternative routing). Add if necessary.

b. Check that there are no unneeded partitions.

c. Check that there are no unneeded calling search spaces.

d. Check that Route Plan Report is clear.

Step 5 Verify the following from the Cisco Unified CM Publisher Administration **Services** menu:

a. Check that the MOH Server name is MOH_*winshostname*.

For example, MOH_2; that is, not MOH_10.10.1.3.

Note Note that there is a maximum of 15 characters in this name, so *winshostname* must be 11 characters or less on Cisco Unified CM 4.x clusters. This restriction is relaxed on Unified 7.x, 6.x and 5.x clusters.

b. Check that at least one conference bridge has been set up on each cluster and that the name(s) of the conference bridge are the same as that entered into BVSM. For example,

- Host Server—10.10.1.3
- Conference Bridge Type—Cisco Conference Bridge Software
- Conference Bridge Name—e2c2p
- Device Pool—Default

Step 6 Verify the following from the Cisco Unified CM Publisher Administration **Features** menu:

In IP Phone Services, verify that the following service has been added:

- For Single Enterprise User Mobility:
  - IP Phone Service—Login/Logout

- Description—Text description
  - Service URL— *http://10.11.226.73/emapp/EMAppServlet?device=#DEVICENAME#* (where the IP address is the IP address of the Cisco Unified CM cluster publisher)
- For Multi-Tenant User Mobility:
  - IP Phone Service—Roaming Login/Logout
  - Description—User Mobility Service provided by USM
  - Service URL— *http://xxxxx/bvsmweb/bvsmroaming.cgi?device=#DEVICENAME#* (where xxxxxx is the virtual IP address, or DNS name if used, of the USM cluster)
- For IP Phone Service:
  - Phone Services (or as defined in the USM Cisco Unified CM model loader "global settings" line)
  - Description—XML Applications provided by USM
  - Service URL— *http://xxxxx/bvsmweb/bvsmservices.cgi?device=#DEVICENAME#* (where xxxxxx is the virtual IP address, or DNS name if used, of the USM cluster providing the corporate directory service providing the corporate directory service).

Step 7 Verify the following Service Parameters from Cisco Unified CM Publisher Administration:

- Check the following Cisco Extension Mobility Settings and adjust as required:
  - Enforce Max Log in Time—False
  - Multiple Login Behavior—Auto Logout
  - Alphanumeric User ID—True
  - Remember last login—True (on reference platforms)

Step 8 Verify the following from the Cisco Unified CM Publisher Administration **Device** menu:

a. Ensure that all the Device Profiles have been deleted.

Note These are added automatically by USM.

b. Ensure that the required phone button templates have been added. For e*xample:*

- *USM 7940*
- *USM 7970*
- *USM 7971*
- *USM 7960-14*
- *USM 7960-28*
- *USM 7961-14*
- *USM 7961-28*
- *USM 7970-14*
- *USM 7970-28*
- *USM 7971-14*
- *USM 7971-28*

Note Additional phone button templates may be required on CCM7.x clusters; for example, Standard 7941, Standard 7961 in addition to Standard 7941 SCCP and Standard 7961 SCCP.

c. Ensure that additional softkey templates have been added as required by USM. The following softkey templates are typically added as examples:

   – *Softkey_Basic*

   – *Softkey_Advanced*

   You can add them by copying the "Standard User" template and then change the settings. In the short term, these can have the same configuration as the "Standard User" softkey template.

Note If the default Softkey_Basic and Softkey_Advanced templates are not required, delete them from the USM "Service Types" within USM Setup Tools (accessible as a USM super user only).

Applying Static Configuration to the Cisco PGW

This section explains how to apply the static configuration to the Cisco PGW. This is required before using USM to load the base data and bulk data that integrates the Cisco PGW into the Hosted UCS platform.

This section contains the following topics:

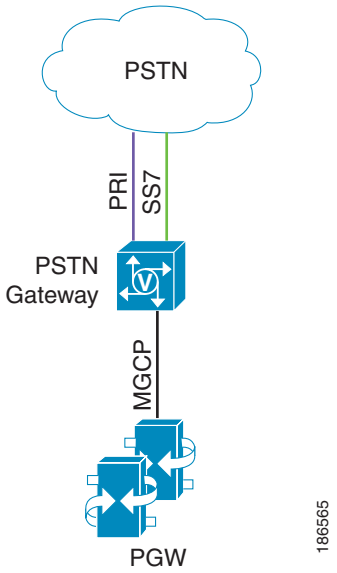
Central Gateway Cisco PGW Breakout

One of the main features of the Cisco PGW in Hosted UCS 7.1(a) is routing calls to and from the PSTN. The Central Gateway Cisco PGW PSTN breakout is achieved using the following signalling:

- SS7
- PRI (MGCP controlled)

This is shown in Figure 2-1:

Figure 2-1 Central Gateway Cisco PGW PSTN



Depending on the deployment, you must provision a number of settings on the Cisco PGW, including the following:

- External Nodes
- Session Sets
- MGCP Paths
- IPFAS Paths
- D-Channels
- IP Links
- DPCs
- OPCs
- APCs
- Linksets
- SS7 Routes
- SS7 Paths
- IP Routes

For detailed information, refer to the Cisco PGW 2200 Softswitch Release 9.8 Provisioning Guide.

The Route List to PSTN for each country must be provisioned on the interface between the Cisco PGW and the PSTN, using the following parameters:

prov-add:rtlist:name="rtlist2pstn*Country_code***",rtname="route2pstn",distrib="OFF"**,

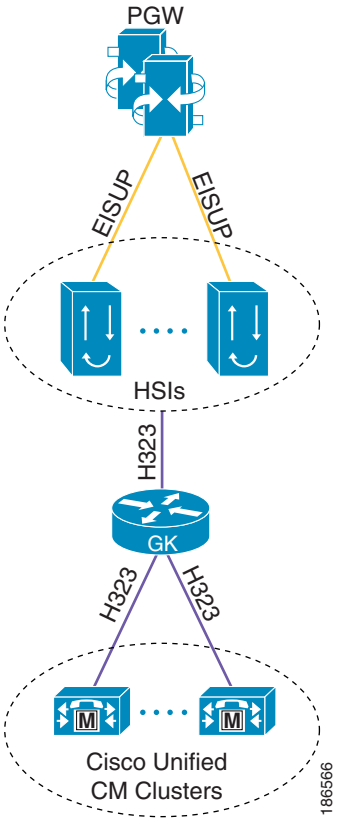
For example: **(for UK)**
prov-add:rtlist:name="rtlist2pstn44",rtname="route2pstn",distrib="OFF"

This configuration is required for each supported country.

Cisco PGW, HSI and Cisco Unified Communications Manager Interface Configuration

In Hosted UCS Release 7.1(a), the interface between the Cisco PGW/HSI and the Cisco Unified CM clusters is an H323 trunk (provisioned as an H.225 gatekeeper controlled trunk on Cisco Unified CM. The interface between the Cisco PGW and HSI is an EISUP trunk, as shown in Figure 2-2.

Figure 2-2 Cisco PGW/HSI/Cisco Unified Communications Manager Interfaces



A number of settings must be provisioned on the Cisco PGW for the interface between the Cisco PGW and HSIs, such as the following:

- External nodes
- EISUP paths
- IP links

For detailed information, refer to the Cisco PGW 2200 Softswitch Release 9.8 Provisioning Guide.

To perform the provisioning required on the Cisco PGW, complete the following steps.

Procedure

Step 1 Configure the ICCM dial plan.

ICCM is the dial plan which needs to be attached to the HSI trunk groups. This dial plan will be accessed when calls are passed from the HSIs to the PGW.

To add the ICCM dial plan, use the following parameter:

```
numan-add:dialplan:custgrpid=”ICCM”, OVERDEC=”YES”
```

Step 2 Configure the Trunk Group for each HSI.

To add the trunk group, enter the following command:

prov-add:trnkgrp:name="*trnkgrp_name***",clli="***clli_name***",svc="***signaling_svc***",type="***type***",qable=n**

For example:

```
prov-add:trnkgrp:name="1001",clli="hsi",svc="eisup-hsi-ent4a",type="IP",qable="n"
```

Step 3 Configure the Routing Trunk Group for each HSI.

To add the routing trunk group, enter the following command:

prov-add:rttrnkgrp:name=”*rttrnkgrp_name***”,type=4,reattempts=0,queuing=0,cutthrough=3,resincperc=0**

For example:

```
prov-add:rttrnkgrp:name=”1001”,type=4,reattempts=0,queuing=0,cutthrough=3,resincperc=0
```

This configuration is required for each HSI.

Step 4 Configure the route to the HSI.

To add the route, enter the following command:

prov-add:rttrnk:weightedTG=”OFF”,name=”route2hsi”,trnkgrpnum=*rttrnkgrp_name*

For example:

```
prov-add:rttrnk:weightedTG=”OFF”,name=”route2hsi”,trnkgrpnum=1001
```

To associate routing trunk groups for the remaining HSIs to the “route2hsi” route, add the following for each remaining HSI:

prov-ed:rttrnk:name=”route2hsi”,trnkgrpnum=*rttrnkgrp_name*

For example:

```
prov-ed:rttrnk:name=”route2hsi”,trnkgrpnum=1002
```

Step 5 Configure the route list to the HSI.

To add the route list, enter the following command:

prov-add:rtlist:name=”rtlist2hsi”,rtname=”route2hsi”,distrib=”OFF”

Step 6 Configure the following HSI Trunk Group Properties:

- CustGrpId
- AllowH323Hairpin
- GatewayRBToneSupport

Other parameters may be necessary depending on the deployment.

From HUCS 7.1(a) onwards, the trunk group properties should be added on a profile and the profile should be attached with trunk group for PGW 9.8(1).

Follow the steps below if the profile is not available on PGW:

```
prov-add: profile:
name="<profile_name>",type="EISUPPROFILE",custgrpid="<custgrpid>",allowh323hairpin =
"1",gatewayrbtonesupport="1"
prov-add: trnkgrpprof:name="<trnkgrp_name>",profile="<profile_name>",
```

For example:

```
prov-add: profile:name="lvl1eisupf-1001",type="EISUPPROFILE",custgrpid="ICCM",
allowh323hairpin="1",gatewayrbtonesupport="1"
prov-add:trnkgrpprof:name="1001",profile=" lvl1eisupf-1001"
```

Step 7 Repeat this step for each HSI.

ILGW Dial Plan

The ILGW Dial Plan is used to route calls from Local Gateways. Because this dial plan is provisioned every time a country is added using USM, it must be manually created. To add the ILGW dial plan, enter the following command:

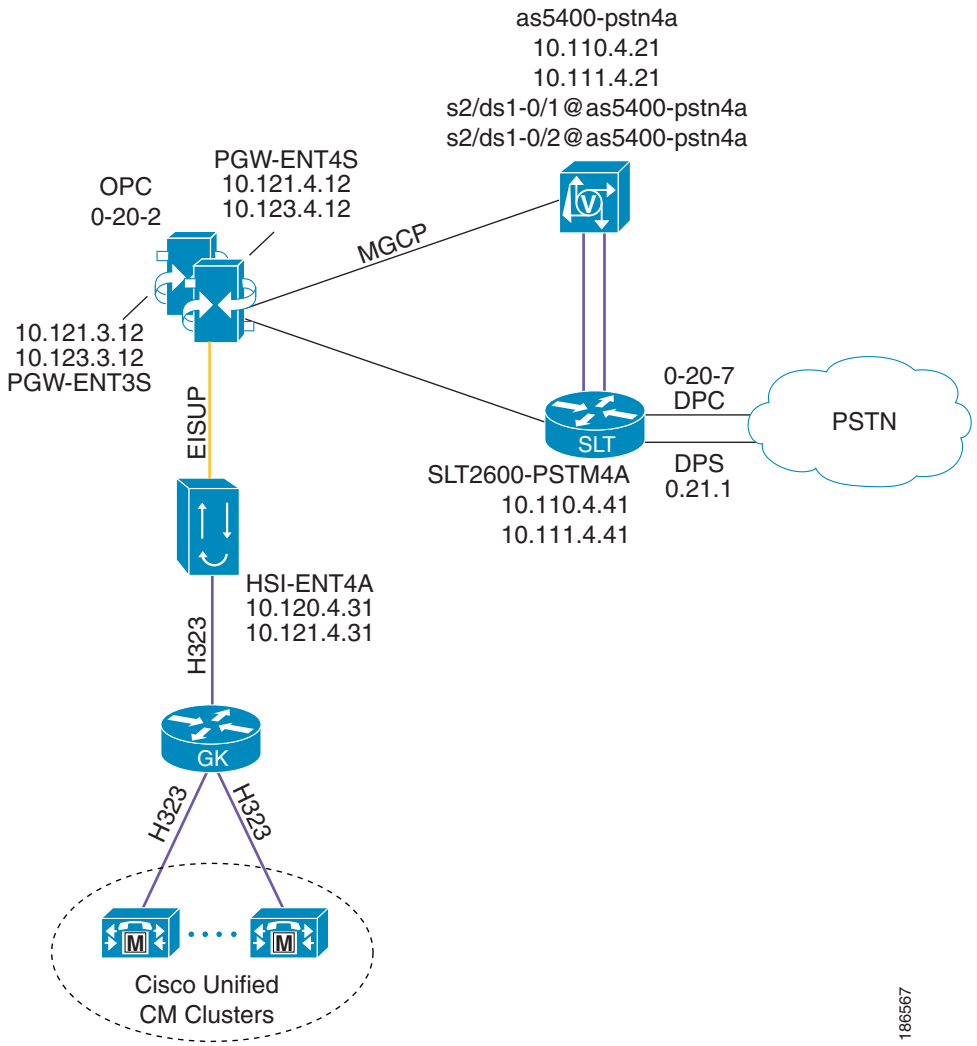
```
numan-add:dialplan:custgrpid="ILGW", OVERDEC="Yes"
```

Example Cisco PGW Static Configuration

This section provides the following sample static configurations for the Cisco PGW within a Hosted UCS 7.1(a) platform:

These example configurations are based on the network illustrated in Figure 2-3.

Figure 2-3 Sample Network



The following examples illustrate the static configuration for each of the following files:

Example 2-1 config.mml

```
prov-add:IPROUTE:NAME="iproute-2",DESC="IPRoute",DEST="10.121.2.0",NETMASK="255.255.255.0"
,NEXTHOP="IP_NextHop2",IPADDR="IP_Addr2",PRI=1
prov-add:IPROUTE:NAME="iproute-1",DESC="IPRoute",DEST="10.120.2.0",NETMASK="255.255.255.0"
,NEXTHOP="IP_NextHop1",IPADDR="IP_Addr1",PRI=1
prov-add:OPC:NAME="opc",DESC="opc",NETADDR="0.20.1",NETIND=2,TYPE="TRUEOPC"
prov-add:DPC:NAME="pstn1",DESC="pstn1 dpc",NETADDR="0.20.7",NETIND=2
prov-add:DPC:NAME="pstn2",DESC="pstn2 dpc",NETADDR="0.21.1",NETIND=2
prov-add:SS7PATH:NAME="ss7p-pstn1",DESC="SS7 path to
pstn1",MDO="ISUPV3_UK",CUSTGRPID="0000",SIDE="network",DPC="pstn1",OPC="opc",M3UAKEY="",O
RIGLABEL="",TERMLABEL=""
prov-add:EXTNODE:NAME="hsi-ent4a",DESC="hsi-ent2a",TYPE="H323",ISDNSIGTYPE="N/A",GROUP=0
prov-add:EXTNODE:NAME="slt2600-ent4a",DESC="slt2600-ent4a",TYPE="SLT",ISDNSIGTYPE="N/A",GR
OUP=0
```

```
prov-add:EXTNODE:NAME="slt2600-ent4b",DESC="slt2600-ent4b",TYPE="SLT",ISDNSIGTYPE="N/A",GR
OUP=0
prov-add:EXTNODE:NAME="as5400-ent4a",DESC="as5400-ent4a",TYPE="AS5400",ISDNSIGTYPE="N/A",G
ROUP=0
prov-add:EXTNODE:NAME="as5400-ent4b",DESC="as5400-ent4b",TYPE="AS5400",ISDNSIGTYPE="N/A",G
ROUP=0
prov-add:SESSIONSET:NAME="sset-slt-ent4a",EXTNODE="slt2600-ent4a",IPADDR1="IP_Addr1",PEERA
DDR1="10.120.4.41",PORT=7001,PEERPORT=7001,TYPE="BSM
V0",IPROUTE1="iproute-1",IPROUTE2="iproute-2",IPADDR2="IP_Addr2",PEERADDR2="10.121.4.41"
prov-add:SESSIONSET:NAME="sset-slt-ent4b",EXTNODE="slt2600-ent4b",IPADDR1="IP_Addr1",PEERA
DDR1="10.120.4.42",PORT=7001,PEERPORT=7001,TYPE="BSM
V0",IPROUTE1="iproute-1",IPROUTE2="iproute-2",IPADDR2="IP_Addr2",PEERADDR2="10.121.4.42"
prov-add:EISUPPATH:NAME="eisup-hsi-ent4a",DESC="eisup-hsi-ent4a",EXTNODE="hsi-ent4a",MDO="
EISUP",CUSTGRPID="ICCM",ORIGLABEL="",TERMLABEL=""
prov-add:MGCPPATH:NAME="mgcp-as5400-ent4a",DESC="MGCP path for
as5400-ent4a",EXTNODE="as5400-ent4a"
prov-add:MGCPPATH:NAME="mgcp-as5400-ent4b",DESC="MGCP path for
as5400-ent4b",EXTNODE="as5400-ent4b"
prov-add:LNKSET:NAME="lnkset-pstn1",DESC="pstn1
lnkset",APC="pstn1",PROTO="SS7-UK",TYPE="IP"
prov-add:IPLNK:NAME="hsi-ent4a-iplnk-1",DESC="hsi-ent4a-iplnk-1",SVC="eisup-hsi-ent4a",IPA
DDR="IP_Addr1",PORT=8003,PEERADDR="10.120.4.31",PEER
PORT=8003,PRI=1,IPROUTE="iproute-1"
prov-add:IPLNK:NAME="hsi-ent4a-iplnk-2",DESC="hsi-ent4a-iplnk-2",SVC="eisup-hsi-ent4a",IPA
DDR="IP_Addr2",PORT=8003,PEERADDR="10.121.4.31",PEER
PORT=8003,PRI=2,IPROUTE="iproute-2"
prov-add:IPLNK:NAME="as5400-ent4a-iplnk1",DESC="IP link 1 to
as5400-ent4a",SVC="mgcp-as5400-ent4a",IPADDR="IP_Addr1",PORT=2427,PEERADDR="10.12
0.4.21",PEERPORT=2427,PRI=1,IPROUTE="iproute-1"
prov-add:IPLNK:NAME="as5400-ent4a-iplnk2",DESC="IP link 2 to
as5400-ent4a",SVC="mgcp-as5400-ent2a",IPADDR="IP_Addr2",PORT=2427,PEERADDR="10.12
1.4.21",PEERPORT=2427,PRI=2,IPROUTE="iproute-2"
prov-add:IPLNK:NAME="as5400-ent4b-iplnk1",DESC="IP link 1 to
as5400-ent4b",SVC="mgcp-as5400-ent4b",IPADDR="IP_Addr1",PORT=2427,PEERADDR="10.12
0.4.22",PEERPORT=2427,PRI=1,IPROUTE="iproute-1"
prov-add:IPLNK:NAME="as5400-ent4b-iplnk2",DESC="IP link 2 to
as5400-ent4b",SVC="mgcp-as5400-ent4b",IPADDR="IP_Addr2",PORT=2427,PEERADDR="10.12
1.4.22",PEERPORT=2427,PRI=2,IPROUTE="iproute-2"
prov-add:SS7ROUTE:NAME="ss7r-pstn1",DESC="SS7 Route to
pstn1",OPC="opc",DPC="pstn1",LNKSET="lnkset-pstn1",PRI=1
prov-add:C7IPLNK:NAME="pstn1-c7lnk-1",DESC="C7 IP link 1 to
pstn1",LNKSET="lnkset-pstn1",SLC=0,PRI=1,TIMESLOT=0,SESSIONSET="sset-slt-ent2a"
prov-add:C7IPLNK:NAME="pstn1-c7lnk-2",DESC="C7 IP link 2 to
pstn1",LNKSET="lnkset-pstn1",SLC=1,PRI=1,TIMESLOT=0,SESSIONSET="sset-slt-ent4b"
prov-add:PROFILE:NAME="lvl2cmpf-1001",TYPE="commonprofile",clli="HSI"
prov-add:PROFILE:NAME="lvl1eisupf-1001",TYPE="eisupprofile",commonprofile="lvl2cmpf-1001",
custgrpid="ICCM"
prov-add:DNSPARAM:CacheSize="500",DnsServer1="0.0.0.0",DnsServer2="0.0.0.0",KeepAlive="30"
,Policy="HIERARCHY",QueryTimeout="1000",TTL="3600"
prov-add:TOS:DSCP = "CS3"
prov-ed:accrespcat:name="default",acl1drcant=50,acl1drskip=20,acl1arcant=50,acl1arskip=20,
acl2drcant=90,acl2drskip=10,acl2arcant=90,acl2arskip
=10,acl3drcant=100,acl3drskip=0,acl3arcant=100,acl3arskip=0
prov-ed:mclcallreject:name="mcl1",callreject=25
prov-ed:mclcallreject:name="mcl2",callreject=50
prov-ed:mclcallreject:name="mcl3",callreject=100
prov-ed:mclthreshold:name="callrate",mcl1onset=0,mcl1abate=0,mcl2onset=0,mcl2abate=0,mcl3o
nset=0,mcl3abate=0
prov-ed:mclthreshold:name="cpu",mcl1onset=82,mcl1abate=75,mcl2onset=90,mcl2abate=77,mcl3on
set=95,mcl3abate=85
prov-ed:mclthreshold:name="memoryaddress",mcl1onset=84,mcl1abate=80,mcl2onset=88,mcl2abate
=82,mcl3onset=93,mcl3abate=85
prov-ed:mclthreshold:name="queuelen",mcl1onset=75,mcl1abate=60,mcl2onset=80,mcl2abate=70,m
cl3onset=85,mcl3abate=75
```

```
prov-ed:mclthreshold:name="virtualmemory",mcl1onset=80,mcl1abate=75,mcl2onset=85,mcl2abate
=80,mcl3onset=90,mcl3abate=80
prov-dlt:inservice:name="ansi-ain-800-npa"
prov-dlt:inservice:name="ansi-ain-800-npa-nxx"
prov-dlt:inservice:name="ansi-ain-800-npanxxx"
prov-dlt:inservice:name="ansi-ain-800-ti"
prov-dlt:inservice:name="ansi-pre-ain-800"
prov-dlt:inservice:name="ansi-pre-ain-800-ssn"
prov-dlt:inservice:name="ansi-pre-ain-800-ti"
prov-dlt:inservice:name="ansi-pre-ain-800-ts"
prov-dlt:inservice:name="c1-lnp"
prov-dlt:inservice:name="cs1-inap-cli-initdp"
prov-dlt:inservice:name="cs1i-nap-cli-srr"
prov-dlt:inservice:name="generic-lnp"
prov-dlt:inservice:name="genesys-800"
prov-dlt:inservice:name="inap-freephon-initdp"
prov-dlt:inservice:name="inap-lnp-initdp"
prov-dlt:inservice:name="inap-lnp-norway"
prov-dlt:inservice:name="inap-lnp-portugal"
prov-dlt:inservice:name="inap-pp-bcsm"
prov-dlt:inservice:name="inap-pp-charge-atexp"
prov-dlt:inservice:name="inap-pp-charge-final"
prov-dlt:inservice:name="inap-pp-charge-texp"
prov-dlt:inservice:name="inap-pp-initdp"
prov-dlt:inservice:name="inap-precarr-initdp"
prov-dlt:inservice:name="inap-cs1-initdp"
prov-dlt:inservice:name="inap-cs1-dummy-25"
prov-dlt:inservice:name="inap-cs1-dummy-26"
prov-dlt:inservice:name="inap-cs1-dummy-27"
prov-dlt:inservice:name="inap-cs1-dummy-28"
prov-dlt:inservice:name="inap-cs2-initdp"
prov-dlt:inservice:name="ansi-pre-ain-cnam"
prov-add:inservice:name="ansi-ain-800-npa",skortcv=4,gtorssn="ROUTEBYGT",gtformat="GTTT",m
sname="ansi-ain-800-npa"
prov-add:inservice:name="ansi-ain-800-npa-nxx",skortcv=5,gtorssn="ROUTEBYGT",gtformat="GTT
T",msname="ansi-ain-800-npa-nxx"
prov-add:inservice:name="ansi-ain-800-npanxxx",skortcv=8,gtorssn="ROUTEBYGT",gtformat="GTT
T",msname="ansi-ain-800-npanxxx"
prov-add:inservice:name="ansi-ain-800-ti",skortcv=0,gtorssn="ROUTEBYGT",gtformat="GTTT",ms
name="ansi-ain-800-ti"
prov-add:inservice:name="ansi-pre-ain-800",skortcv=0,gtorssn="ROUTEBYGT",gtformat="GTTT",m
sname="ansi-pre-ain-800"
prov-add:inservice:name="ansi-pre-ain-800-ssn",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NO
GT",msname="ansi-pre-ain-800-ssn"
prov-add:inservice:name="ansi-pre-ain-800-ti",skortcv=0,gtorssn="ROUTEBYGT",gtformat="GTTT
",msname="ansi-pre-ain-800-ti"
prov-add:inservice:name="ansi-pre-ain-800-ts",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NOG
T",msname="ansi-pre-ain-800-ts"
prov-add:inservice:name="ansi-pre-ain-cnam",skortcv=0,gtorssn="ROUTEBYGT",gtformat="GTTT",
msname="ansi-pre-ain-cnam"
prov-add:inservice:name="c1-lnp",skortcv=0,gtorssn="ROUTEBYGT",gtformat="GTTT",msname="c1-
lnp"
prov-add:inservice:name="cs1-inap-cli-initdp",skortcv=1,gtorssn="ROUTEBYSSN",gtformat="NOG
T",msname="cs1-inap-cli-initdp"
prov-add:inservice:name="cs1i-nap-cli-srr",skortcv=1,gtorssn="ROUTEBYSSN",gtformat="NOGT",
msname="cs1i-nap-cli-srr"
prov-add:inservice:name="generic-lnp",skortcv=37,gtorssn="ROUTEBYGT",gtformat="GTTT",msnam
e="generic-lnp"
prov-add:inservice:name="genesys-800",skortcv=0,gtorssn="ROUTEBYGT",gtformat="GTTT",msname
="genesys-800"
prov-add:inservice:name="inap-cs1-dummy-25",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NOGT"
,msname="inap-cs1-dummy-25"
prov-add:inservice:name="inap-cs1-dummy-26",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NOGT"
,msname="inap-cs1-dummy-26"
```

```
prov-add:inservice:name="inap-cs1-dummy-27",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NOGT"
,msname="inap-cs1-dummy-27"
prov-add:inservice:name="inap-cs1-dummy-28",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NOGT"
,msname="inap-cs1-dummy-28"
prov-add:inservice:name="inap-cs1-initdp",skortcv=90001,gtorssn="ROUTEBYSSN",gtformat="NOG
T",msname="inap-cs1-initdp"
prov-add:inservice:name="inap-cs2-initdp",skortcv=90001,gtorssn="ROUTEBYSSN",gtformat="NOG
T",msname="inap-cs2-initdp"
prov-add:inservice:name="inap-freephon-initdp",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NO
GT",msname="inap-freephon-initdp"
prov-add:inservice:name="inap-lnp-initdp",skortcv=1,gtorssn="ROUTEBYSSN",gtformat="NOGT",m
sname="inap-lnp-initdp"
prov-add:inservice:name="inap-lnp-norway",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NOGT",m
sname="inap-lnp-norway"
prov-add:inservice:name="inap-lnp-portugal",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NOGT"
,msname="inap-lnp-portugal"
prov-add:inservice:name="inap-pp-bcsm",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NOGT",msna
me="inap-pp-bcsm"
prov-add:inservice:name="inap-pp-charge-atexp",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NO
GT",msname="inap-pp-charge-atexp"
prov-add:inservice:name="inap-pp-charge-final",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NO
GT",msname="inap-pp-charge-final"
prov-add:inservice:name="inap-pp-charge-texp",skortcv=0,gtorssn="ROUTEBYSSN",gtformat="NOG
T",msname="inap-pp-charge-texp"
prov-add:inservice:name="inap-pp-initdp",skortcv=1,gtorssn="ROUTEBYSSN",gtformat="NOGT",ms
name="inap-pp-initdp"
prov-add:inservice:name="inap-precarr-initdp",skortcv=2,gtorssn="ROUTEBYSSN",gtformat="NOG
T",msname="inap-precarr-initdp"
prov-add:sigsvcprop:NAME="eisup-hsi-ent4a",H323AdjunctLink="1"
prov-add:sigsvcprop:NAME="mgcp-as5400-ent4a",mgcpDomainNameRemote="s2/ds1-0/1@AS5400-ENT4A
"
prov-add:sigsvcprop:NAME="mgcp-as5400-ent2b",mgcpDomainNameRemote="s2/ds1-0/1@AS5400-ENT4B
"
prov-add:files:name="tkgfile",file="Static_12_05/export_trkgrp.dat",action="IMPORT"
prov-add:TRNKGRPPROF:name="1001",profile="lvl1eisupf-1001"
prov-add:files:name="bcfile",file="Static_12_05/export_trunk.dat",action="IMPORT"
```

Example 2-2 routing.mml

```
prov-add:rttrnkgrp:name="1001",type=4,reattempts=0,queuing=0,cutthrough=2,resincperc=0
prov-add:rttrnkgrp:name="2001",type=1,reattempts=1,queuing=0,cutthrough=2,resincperc=0
prov-add:rttrnk:weightedTG="OFF",name="route2hsi",trnkgrpnum=1001
prov-add:rttrnk:weightedTG="OFF",name="route2pstn",trnkgrpnum=2001
prov-add:rtlist:name="rtlist2hsi",rtname="route2hsi",distrib="OFF"
prov-add:rtlist:name="rtlist2pstn1",rtname="route2pstn",distrib="OFF"
```

Example 2-3 iccm.mml

```
numan-add:dialplan:custgrpid="ICCM", OVERDEC="YES"
numan-ed: resulttable: custgrpid="ICCM", name="CSCOADRST1",
resulttype="RETRY_ACTION", dw1="Reattempt", dw2="0",setname="CSCOADRST1"
numan-ed: resulttable: custgrpid="ICCM", name="CSCOADRST2", resulttype="RETRY_ACTION",
dw1="Redirect",dw2="0",setname="CSCOADRST2
numan-ed:cause:custgrpid="ICCM",causevalue=1,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=11,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=26,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=29,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=38,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=41,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=44,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=49,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=50,setname="CSCOADRST1"
```

```
numan-ed:cause:custgrpid="ICCM",causevalue=58,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=69,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=87,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=94,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=107,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=118,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ICCM",causevalue=145,setname="CSCOADRST2"
```

Example 2-4 ilgw.mml

```
numan-add:dialplan:custgrpid="ILGW", OVERDEC="NO"
numan-ed: resulttable: custgrpid="ILGW", name="CSCOADRST1", resulttype="RETRY_ACTION",
dw1="Reattempt",dw2="0",setname="CSCOADRST1"
numan-ed: resulttable: custgrpid="ILGW", name="CSCOADRST2", resulttype="RETRY_ACTION",
dw1="Redirect",dw2="0",setname="CSCOADRST2"
numan-ed:cause:custgrpid="ILGW",causevalue=1,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=11,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=26,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=29,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=38,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=41,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=44,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=49,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=50,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=58,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=69,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=87,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=94,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=107,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=118,setname="CSCOADRST1"
numan-ed:cause:custgrpid="ILGW",causevalue=145,setname="CSCOADRST2"
```

Example 2-5 properties.dat

```
sigmgcp-5400.mgcpHeartbeatInterval = 10
ss7-i-1.chkPtPort = 2001
tg-1001.AllowH323Hairpin = 1
tg-1001.CLLI = HSI
tg-1001.CustGrpId = ICCM
tg-1001.GatewayRBToneSupport = 1
tg-1001.commonProfile = lvl2cmpf-1001
tg-2001.CLLI = PSTN1
tg-2001.FAXsupport = 1
tg-2001.GatewayRBToneSupport = 1
001.GatewayRBToneSupport = 1
```

Note Default properties prefixed by an "*", SS7-<ver>.<property_name> properties, and TALI-IOCC.<property_name> properties, are not shown.

Example 2-6 export_trkgrp.dat

```
<!--#xml - 9.8001-->
<trunk-groups>
<version base="9.8001" revision="0"/>
<trunkgroup name="1001" type="IP" svc="eisup-hsi-ent2a" clli="HSI" selseq="LIDL" qable="N"
origlabel="0" termlabel="0">
<property name="CustGrpId">ICCM</property>
<property name="default">0</property>
</trunkgroup>
```

```
<trunkgroup name="2001" type="TDM_ISUP" svc="ss7p-pstn1" clli="PSTN1" selseq="LIDL"
qable="N" origlabel="0" termlabel="0">
<property name="GatewayRBToneSupport">1</property>
<property name="FAXsupport">1</property>
<property name="default">0</property>
</trunkgroup>
</trunk-groups>
```

Example 2-7 export_trunk.dat

```
#format3 - 0.0
2001 1 ffff 1 as5400-pstn4a s2/ds1-0/1@as5400-pstn4a
2001 2 ffff 2 as5400-pstn4a s2/ds1-0/2@as5400-pstn4a
2001 3 ffff 3 as5400-pstn4a s2/ds1-0/3@as5400-pstn4a
2001 4 ffff 4 as5400-pstn4a s2/ds1-0/4@as5400-pstn4a
2001 6 ffff 6 as5400-pstn4a s2/ds1-0/6@as5400-pstn4a
2001 7 ffff 7 as5400-pstn4a s2/ds1-0/7@as5400-pstn4a
2001 8 ffff 8 as5400-pstn4a s2/ds1-0/8@as5400-pstn4a
2001 9 ffff 9 as5400-pstn4a s2/ds1-0/9@as5400-pstn4a
2001 10 ffff 10 as5400-pstn4a s2/ds1-0/10@as5400-pstn4a
2001 11 ffff 11 as5400-pstn4a s2/ds1-0/11@as5400-pstn4a
2001 12 ffff 12 as5400-pstn4a s2/ds1-0/12@as5400-pstn4a
2001 13 ffff 13 as5400-pstn4a s2/ds1-0/13@as5400-pstn4a
2001 14 ffff 14 as5400-pstn4a s2/ds1-0/14@as5400-pstn4a
2001 15 ffff 15 as5400-pstn4a s2/ds1-0/15@as5400-pstn4a
2001 16 ffff 16 as5400-pstn4a s2/ds1-0/16@as5400-pstn4a
2001 17 ffff 17 as5400-pstn4a s2/ds1-0/17@as5400-pstn4a
2001 18 ffff 18 as5400-pstn4a s2/ds1-0/18@as5400-pstn4a
2001 19 ffff 19 as5400-pstn4a s2/ds1-0/19@as5400-pstn4a
2001 20 ffff 20 as5400-pstn4a s2/ds1-0/20@as5400-pstn4a
2001 21 ffff 21 as5400-pstn4a s2/ds1-0/21@as5400-pstn4a
2001 22 ffff 22 as5400-pstn4a s2/ds1-0/22@as5400-pstn4a
2001 23 ffff 23 as5400-pstn4a s2/ds1-0/23@as5400-pstn4a
2001 24 ffff 24 as5400-pstn4a s2/ds1-0/24@as5400-pstn4a
2001 25 ffff 25 as5400-pstn4a s2/ds1-0/25@as5400-pstn4a
2001 26 ffff 26 as5400-pstn4a s2/ds1-0/26@as5400-pstn4a
2001 27 ffff 27 as5400-pstn4a s2/ds1-0/27@as5400-pstn4a
2001 28 ffff 28 as5400-pstn4a s2/ds1-0/28@as5400-pstn4a
2001 29 ffff 29 as5400-pstn4a s2/ds1-0/29@as5400-pstn4a
2001 30 ffff 30 as5400-pstn4a s2/ds1-0/30@as5400-pstn4a
2001 31 ffff 31 as5400-pstn4a s2/ds1-0/31@as5400-pstn4a
```

Example 2-8 XECfgParm.dat

This update to XECfgParm.dat is required for overlap support of PBX gateways (definitely required for the support of DPNSS):

```
*.analysisCapabilityLevel = 1
```

Number Translation with TimesTen Database

The Full Number Translation feature provides a large-scale number translation function on the Cisco PGW. This feature enhances the current PGW database query mode, which is used for local number portability (LNP) and CLI screening, by handling continuous ranges of numbers with analysis and modification capabilities.

The Full Number Translation feature supports large-scale changes of individual numbers. This feature adds the NUM_TRANS result type that is implemented in analysis where the existing Times Ten database is used to store the dial plan numbers.

The full number replacement mechanism adds a general number replacement result type, NUM_TRANS, available for A-number and B-number analysis. In addition, it includes a Times Ten query and full number translation table.

Hosted UCS 7.1(a) introduces the Full Number Translation with TimesTen Database feature. This means that the association of E.164 numbers to Internal numbers will use this feature instead of configuring via mml.

This section contains the following:

Sparc Based Platform Configuration

You must upload the HUCSprovx10 script on the PGW in order to use this feature.

Note If you have an Active/Standby system, ensure that you perform the steps described here on both platforms.

To configure the Sparc based platform on the Cisco PGW, complete the following steps.

Procedure

Step 1 Upload the HUCS_x10_package.gz package onto an FTP server reachable by the PGW.

Step 2 Login to the PGW as the PGW application user (default is mgcusr).

Step 3 Download HUCS_x10_package.gz from the FTP server into /opt/CiscoMGC/local.

Step 4 Unzip HUCS_x10_package.gz. Enter:

```
gunzip HUCS_x10_package.gz
```

Step 5 Untar HUCS_x10_package. Enter:

```
tar -xvf HUCS_x10_package
```

Step 6 The following output appears:

```
x ./HUCS_x10, 0 bytes, 0 tape blocks
x ./HUCS_x10/java_vm64, 0 bytes, 0 tape blocks
x ./HUCS_x10/java_vm64/jdk64-sparc-1_5_0_06.gz, 9424713 bytes, 18408 tape blocks
x ./HUCS_x10/java_vm64/jdk64-amd64-1_5_0_06.gz, 5439360 bytes, 10624 tape blocks
x ./HUCS_x10/java_appl, 0 bytes, 0 tape blocks
x ./HUCS_x10/java_appl/data, 0 bytes, 0 tape blocks
x ./HUCS_x10/java_appl/data/fnt_sample_data, 180 bytes, 1 tape blocks
x ./HUCS_x10/java_appl/data/lnp_fnt_sample_data, 246 bytes, 1 tape blocks
x ./HUCS_x10/java_appl/data/lnp_sample_data, 67 bytes, 1 tape blocks
x ./HUCS_x10/java_appl/bin, 0 bytes, 0 tape blocks
x ./HUCS_x10/java_appl/bin/HUCSprovx10, 246 bytes, 1 tape blocks
x ./HUCS_x10/java_appl/bin/HUCSprovx10.jar, 8143 bytes, 16 tape blocks
```

Step 7 Go to the **java_vm64** folder. Enter:

```
cd HUCS_x10/java_vm64
```

Step 8 Unzip jdk64-sparc-1_5_0_06.gz. Enter:

```
gunzip jdk64-sparc-1_5_0_06.gz
```

Step 9 Untar jdk64-sparc-1_5_0_06. Enter:

```
tar -xvf jdk64-sparc-1_5_0_06
```

The following output appears:

```
x ./SUNWj5rtx, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/pkgmap, 7335 bytes, 15 tape blocks
x ./SUNWj5rtx/pkginfo, 571 bytes, 2 tape blocks
x ./SUNWj5rtx/install, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/install/copyright, 93 bytes, 1 tape blocks
x ./SUNWj5rtx/install/depend, 1063 bytes, 3 tape blocks
x ./SUNWj5rtx/reloc, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9/java, 81440 bytes, 160 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9/keytool, 74520 bytes, 146 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9/orbd, 74664 bytes, 146 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9/pack200, 74552 bytes, 146 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9/policytool, 74536 bytes, 146 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9/rmid, 74520 bytes, 146 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9/rmiregistry, 74520 bytes, 146
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9/servertool, 74520 bytes, 146 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9/tnameserv, 74696 bytes, 146 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/sparcv9/unpack200, 205960 bytes, 403 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9/java, 81440 bytes, 160 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9/keytool, 74520 bytes, 146
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9/orbd, 74664 bytes, 146 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9/pack200, 74552 bytes, 146
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9/policytool, 74536 bytes, 146
tape blocks
```

```
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9/rmid, 74520 bytes, 146 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9/rmiregistry, 74520 bytes, 146
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9/servertool, 74520 bytes, 146
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9/tnameserv, 74696 bytes, 146
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/sparcv9/unpack200, 205960 bytes, 403
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/awt_robot, 26432 bytes, 52
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/gtkhelper, 7760 bytes, 16
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/headless, 0 bytes, 0 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/headless/libmawt.so, 40400
bytes, 79 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/jvm.cfg, 659 bytes, 2 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libJdbcOdbc.so, 56552 bytes,
111 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libawt.so, 1057000 bytes,
2065 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libcmm.so, 388400 bytes, 759
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libdcpr.so, 187368 bytes, 366
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libdt_socket.so, 19560
bytes, 39 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libfontmanager.so, 479320
bytes, 937 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libhprof.so, 292680 bytes,
572 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libinstrument.so, 86784
bytes, 170 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libioser12.so, 14568 bytes,
29 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libj2pkcs11.so, 66144 bytes,
130 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjaas_unix.so, 7344 bytes,
15 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjava.so, 179264 bytes, 351
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjava_crw_demo.so, 46616
bytes, 92 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjawt.so, 3160 bytes, 7
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjdgaSUNWcg6.so, 11224
bytes, 22 tape blocks
```

```
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjdgaSUNWffb.so, 11632
bytes, 23 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjdgaSUNWm64.so, 7912
bytes, 16 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjdwp.so, 336848 bytes, 658
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjpeg.so, 204264 bytes, 399
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjsig.so, 14264 bytes, 28
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjsound.so, 329360 bytes,
644 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjsoundsolmidi.so, 20872
bytes, 41 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libmanagement.so, 29040
bytes, 57 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libmlib_image.so, 1370616
bytes, 2677 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libmlib_image_v.so, 1870136
bytes, 3653 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libnet.so, 84240 bytes, 165
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libnio.so, 34024 bytes, 67
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/librmi.so, 2840 bytes, 6 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libsaproc.so, 49280 bytes, 97
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libsunwjdga.so, 10304 bytes,
21 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libunpack.so, 95064 bytes,
186 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libverify.so, 82200 bytes,
161 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libxinerama.so, 9832 bytes,
20 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libzip.so, 83568 bytes, 164
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/motif21, 0 bytes, 0 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/motif21/libmawt.so, 607480
bytes, 1187 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/native_threads, 0 bytes, 0
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/native_threads/libhpi.so,
47832 bytes, 94 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/server, 0 bytes, 0 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/server/Xusage.txt, 1423
bytes, 3 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/server/libjvm.so, 12163008
bytes, 23756 tape blocks
```

```
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/server/libjvm_db.so, 46656
bytes, 92 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/xawt, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/sparcv9/xawt/libmawt.so, 257176
bytes, 503 tape blocks
```

Step 10 As a root user, add the SUNWj5rtx package. Enter:

```
pkgadd -d . SUNWj5rtx
```

Step 11 The following output appears:

```
Processing package instance <SUNWj5rtx> from </opt/CiscoMGC/local/HUCS_x10/java_vm64>
JDK 5.0 64-bit Runtime Env. (1.5.0_06)(sparc) 1.5.0,REV=2004.12.06.22.09
Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Using </usr> as the package base directory.
## Processing package information.
## Processing system information.
   7 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.
Installing JDK 5.0 64-bit Runtime Env. (1.5.0_06) as <SUNWj5rtx>
## Installing part 1 of 1.
/usr/jdk/instances/jdk1.5.0/bin/sparcv9/java
/usr/jdk/instances/jdk1.5.0/bin/sparcv9/keytool
/usr/jdk/instances/jdk1.5.0/bin/sparcv9/orbd
/usr/jdk/instances/jdk1.5.0/bin/sparcv9/pack200
/usr/jdk/instances/jdk1.5.0/bin/sparcv9/policytool
/usr/jdk/instances/jdk1.5.0/bin/sparcv9/rmid
/usr/jdk/instances/jdk1.5.0/bin/sparcv9/rmiregistry
/usr/jdk/instances/jdk1.5.0/bin/sparcv9/servertool
/usr/jdk/instances/jdk1.5.0/bin/sparcv9/tnameserv
/usr/jdk/instances/jdk1.5.0/bin/sparcv9/unpack200
/usr/jdk/instances/jdk1.5.0/jre/bin/sparcv9/java
/usr/jdk/instances/jdk1.5.0/jre/bin/sparcv9/keytool
/usr/jdk/instances/jdk1.5.0/jre/bin/sparcv9/orbd
/usr/jdk/instances/jdk1.5.0/jre/bin/sparcv9/pack200
/usr/jdk/instances/jdk1.5.0/jre/bin/sparcv9/policytool
/usr/jdk/instances/jdk1.5.0/jre/bin/sparcv9/rmid
/usr/jdk/instances/jdk1.5.0/jre/bin/sparcv9/rmiregistry
/usr/jdk/instances/jdk1.5.0/jre/bin/sparcv9/servertool
/usr/jdk/instances/jdk1.5.0/jre/bin/sparcv9/tnameserv
/usr/jdk/instances/jdk1.5.0/jre/bin/sparcv9/unpack200
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/awt_robot
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/gtkhelper
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/headless/libmawt.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/jvm.cfg
```

```
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libJdbcOdbc.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libawt.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libcmm.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libdcpr.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libdt_socket.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libfontmanager.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libhprof.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libinstrument.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libioser12.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libj2pkcs11.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjaas_unix.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjava.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjava_crw_demo.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjawt.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjdgaSUNWafb.so <symbolic link>
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjdgaSUNWcg6.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjdgaSUNWffb.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjdgaSUNWm64.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjdwp.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjpeg.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjsig.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjsound.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libjsoundsolmidi.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libmanagement.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libmlib_image.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libmlib_image_v.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libnet.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libnio.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/librmi.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libsaproc.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libsunwjdga.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libunpack.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libverify.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libxinerama.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/libzip.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/motif21/libmawt.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/native_threads/libhpi.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/server/Xusage.txt
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/server/libjsig.so <symbolic link>
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/server/libjvm.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/server/libjvm_db.so
/usr/jdk/instances/jdk1.5.0/jre/lib/sparcv9/xawt/libmawt.so
[ verifying class <none> ]
Installation of <SUNWj5rtx> was successful.
```

Step 12 Go to /opt/CiscoMGC/local/HUCS_x10/java_appl/bin.

Step 13 Move HUCSprovx10 and HUCSprovx10.jar to /opt/CiscoMGC/local/. For example, *mv HUCS* /opt/CiscoMGC/local*

Step 14 Repeat these steps on the Standby platform (if applicable).

This completes the requred steps for uploading the HUCSprovx10 script on the PGW(s).

Opteron Based Platform Configuration

Note: If you have an Active/Standby system ensure that you perform the steps described below on both platforms.

To use this feature, you must upload the HUCSprovx10 script on the PGW. To do this, complete the following steps.

Procedure

Step 1 Upload the HUCS_x10_package.gz package onto an FTP server reachable by the PGW.

Step 2 Login to the PGW as the PGW application user (default is mgcusr).

Step 3 Download HUCS_x10_package.gz from the FTP server into /opt/CiscoMGC/local.

Step 4 Unzip HUCS_x10_package.gz. Enter:

```
gunzip HUCS_x10_package.gz
```

Step 5 Untar HUCS_x10_package. Enter:

```
tar -xvf HUCS_x10_package
```

Step 6 The following output appears:

```
x ./HUCS_x10, 0 bytes, 0 tape blocks
x ./HUCS_x10/java_vm64, 0 bytes, 0 tape blocks
x ./HUCS_x10/java_vm64/jdk64-sparc-1_5_0_06.gz, 9424713 bytes, 18408 tape blocks
x ./HUCS_x10/java_vm64/jdk64-amd-1_5_0_06.gz, 5439360 bytes, 10624 tape blocks
x ./HUCS_x10/java_appl, 0 bytes, 0 tape blocks
x ./HUCS_x10/java_appl/data, 0 bytes, 0 tape blocks
x ./HUCS_x10/java_appl/data/fnt_sample_data, 180 bytes, 1 tape blocks
x ./HUCS_x10/java_appl/data/lnp_fnt_sample_data, 246 bytes, 1 tape blocks
x ./HUCS_x10/java_appl/data/lnp_sample_data, 67 bytes, 1 tape blocks
x ./HUCS_x10/java_appl/bin, 0 bytes, 0 tape blocks
x ./HUCS_x10/java_appl/bin/HUCSprovx10, 246 bytes, 1 tape blocks
x ./HUCS_x10/java_appl/bin/HUCSprovx10.jar, 8118 bytes, 16 tape blocks
```

Step 7 Go to the java_vm64 folder. Enter:

```
cd HUCS_x10/java_vm64
```

Step 8 Unzip jdk64-amd-1_5_0_06.gz. Enter:

```
gunzip jdk64-amd-1_5_0_06.gz
```

Step 9 Untar jdk64-amd-1_5_0_06. Enter:

```
tar -xvf jdk64-amd-1_5_0_06
```

Step 10 The following output appears:

```
x ./SUNWj5rtx, 0 bytes, 0 tape blocks
```

```
x ./SUNWj5rtx/pkgmap, 6599 bytes, 13 tape blocks
x ./SUNWj5rtx/pkginfo, 573 bytes, 2 tape blocks
x ./SUNWj5rtx/install, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/install/copyright, 93 bytes, 1 tape blocks
x ./SUNWj5rtx/install/depend, 1063 bytes, 3 tape blocks
x ./SUNWj5rtx/reloc, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64/java, 68016 bytes, 133 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64/keytool, 71424 bytes, 140 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64/orbd, 71568 bytes, 140 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64/pack200, 71456 bytes, 140 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64/policytool, 71456 bytes, 140 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64/rmid, 71424 bytes, 140 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64/rmiregistry, 71424 bytes, 140 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64/servertool, 71424 bytes, 140 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64/tnameserv, 71600 bytes, 140 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/bin/amd64/unpack200, 200368 bytes, 392 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64/java, 68016 bytes, 133 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64/keytool, 71424 bytes, 140 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64/orbd, 71568 bytes, 140 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64/pack200, 71456 bytes, 140 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64/policytool, 71456 bytes, 140
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64/rmid, 71424 bytes, 140 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64/rmiregistry, 71424 bytes, 140
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64/servertool, 71424 bytes, 140
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64/tnameserv, 71600 bytes, 140
tape blocks
```

```
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/bin/amd64/unpack200, 200368 bytes, 392
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/awt_robot, 24768 bytes, 49 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/gtkhelper, 7120 bytes, 14 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/headless, 0 bytes, 0 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/headless/libmawt.so, 33024
bytes, 65 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/j2pkcs11.dll, 65666 bytes, 129
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/j2pkcs11_g.dll, 82054 bytes,
161 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/jvm.cfg, 652 bytes, 2 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libJdbcOdbc.so, 64768 bytes,
127 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libawt.so, 481776 bytes, 941
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libcmm.so, 383216 bytes, 749
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libdcpr.so, 190656 bytes, 373
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libdt_socket.so, 18072 bytes,
36 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libfontmanager.so, 457896
bytes, 895 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libhprof.so, 179616 bytes, 351
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libinstrument.so, 74152 bytes,
145 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libioser12.so, 16824 bytes, 33
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libj2pkcs11.so, 61192 bytes,
120 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libjaas_unix.so, 6232 bytes, 13
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libjava.so, 163928 bytes, 321
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libjava_crw_demo.so, 26160
bytes, 52 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libjawt.so, 3432 bytes, 7 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libjdwp.so, 278624 bytes, 545
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libjpeg.so, 187080 bytes, 366
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libjsig.so, 14824 bytes, 29
tape blocks
```

```
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libjsound.so, 294688 bytes, 576
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libmanagement.so, 27448 bytes,
54 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libmlib_image.so, 807296
bytes, 1577 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libnet.so, 71744 bytes, 141
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libnio.so, 30816 bytes, 61 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/librmi.so, 3056 bytes, 6 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libsaproc.so, 62024 bytes, 122
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libunpack.so, 95712 bytes, 187
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libverify.so, 63232 bytes, 124
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/libzip.so, 75200 bytes, 147
tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/motif21, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/motif21/libmawt.so, 528728
bytes, 1033 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/native_threads, 0 bytes, 0 tape
blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/native_threads/libhpi.so,
41312 bytes, 81 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/server, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/server/Xusage.txt, 1423 bytes,
3 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/server/libjvm.so, 12230144
bytes, 23887 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/server/libjvm_db.so, 54776
bytes, 107 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/xawt, 0 bytes, 0 tape blocks
x ./SUNWj5rtx/reloc/jdk/instances/jdk1.5.0/jre/lib/amd64/xawt/libmawt.so, 226704 bytes,
443 tape blocks
```

Step 11 As a root user, add the SUNWj5rtx package. Enter:

```
pkgadd -d . SUNWj5rtx
```

Step 12 The following output appears:

```
Processing package instance <SUNWj5rtx> from </opt/CiscoMGC/local/HUCS_x10/java_vm64>
JDK 5.0 64-bit Runtime Env. (1.5.0_06)(i386) 1.5.0,REV=2005.03.04.02.15
Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Using </usr> as the package base directory.
## Processing package information.
## Processing system information.
   7 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
```

```
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.
Installing JDK 5.0 64-bit Runtime Env. (1.5.0_06) as <SUNWj5rtx>
## Installing part 1 of 1.
/usr/jdk/instances/jdk1.5.0/bin/amd64/java
/usr/jdk/instances/jdk1.5.0/bin/amd64/keytool
/usr/jdk/instances/jdk1.5.0/bin/amd64/orbd
/usr/jdk/instances/jdk1.5.0/bin/amd64/pack200
/usr/jdk/instances/jdk1.5.0/bin/amd64/policytool
/usr/jdk/instances/jdk1.5.0/bin/amd64/rmid
/usr/jdk/instances/jdk1.5.0/bin/amd64/rmiregistry
/usr/jdk/instances/jdk1.5.0/bin/amd64/servertool
/usr/jdk/instances/jdk1.5.0/bin/amd64/tnameserv
/usr/jdk/instances/jdk1.5.0/bin/amd64/unpack200
/usr/jdk/instances/jdk1.5.0/jre/bin/amd64/java
/usr/jdk/instances/jdk1.5.0/jre/bin/amd64/keytool
/usr/jdk/instances/jdk1.5.0/jre/bin/amd64/orbd
/usr/jdk/instances/jdk1.5.0/jre/bin/amd64/pack200
/usr/jdk/instances/jdk1.5.0/jre/bin/amd64/policytool
/usr/jdk/instances/jdk1.5.0/jre/bin/amd64/rmid
/usr/jdk/instances/jdk1.5.0/jre/bin/amd64/rmiregistry
/usr/jdk/instances/jdk1.5.0/jre/bin/amd64/servertool
/usr/jdk/instances/jdk1.5.0/jre/bin/amd64/tnameserv
/usr/jdk/instances/jdk1.5.0/jre/bin/amd64/unpack200
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/awt_robot
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/gtkhelper
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/headless/libmawt.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/j2pkcs11.dll
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/j2pkcs11_g.dll
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/jvm.cfg
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libJdbcOdbc.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libawt.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libcmm.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libdcpr.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libdt_socket.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libfontmanager.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libhprof.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libinstrument.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libioser12.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libj2pkcs11.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libjaas_unix.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libjava.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libjava_crw_demo.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libjawt.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libjdwp.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libjpeg.so
```

```
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libjsig.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libjsound.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libmanagement.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libmlib_image.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libnet.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libnio.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/librmi.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libsaproc.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libunpack.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libverify.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/libzip.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/motif21/libmawt.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/native_threads/libhpi.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/server/Xusage.txt
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/server/libjsig.so <symbolic link>
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/server/libjvm.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/server/libjvm_db.so
/usr/jdk/instances/jdk1.5.0/jre/lib/amd64/xawt/libmawt.so
[ verifying class <none> ]
Installation of <SUNWj5rtx> was successful.
```

Step 13 Go to /opt/CiscoMGC/local/HUCS_x10/java_appl/bin.

Step 14 Move HUCSprovx10 and HUCSprovx10.jar to /opt/CiscoMGC/local/

For example,

mv HUCS /opt/CiscoMGC/local*

Step 15 Repeat these steps on the Standby platform (if applicable).

This completes the requred steps for uploading the HUCSprovx10 script on the PGW.

Applying Static Configuration to the Cisco HSI

This section explains how to apply the static configuration to the Cisco HSI. This is required before using USM to load the base data and bulk data that integrates the Cisco HSI into the Hosted UCS platform.

Cisco HSI enables the Cisco PGW to talk to the Cisco Unified CM using H.323 over the H.323 gatekeeper. The HSI is an adjunct to the Cisco PGW and simply provides an H.323 interface.

Apart from the configuration described in this section, Cisco HSI has mandatory parameters that must be provisioned; for example:

- IP addresses of the HSI
- IP addresses the Cisco PGWs
- Ports used to communicate with the PGW

For further information, refer to the *Cisco H.323 Signaling Interface User Guide, Release 4.2*.

Example 2-9 illustrates the Hosted UCS- specific static configuration settings required on all HSIs.

Example 2-9 RAS Parameters

```
prov-add:name=ras,gatekeeperId=HUCS_ZONE
prov-add:name=ras,gateway.prefix[1]=999#
prov-add:name=ras,manualDiscovery.ipAddress=<gatekeeper_ip_address>, for example:
prov-add:name=ras,manualDiscovery.ipAddress=10.120.4.51
prov-add:name=ras,manualDiscovery.port=1719
prov-add:name=ras,terminalAlias[1].h323ID=<hsi_name>, for example:
prov-add:name=ras,terminalAlias[1].h323ID=hsi-ent4a@cisco.com
```

Example 2-10 illustrates the Hosted UCS- specific static configuration settings required to support T.38 fax:

Example 2-10 T.38 fax support

Example 2-9 illustrates the Hosted UCS- specific static configuration settings required on all HSIs.

```
prov-add:name=sys_config_static,t38maxval=”MaxBit 0x90, FxMaxBuf 0xc8, FxMaxData 0x48"
prov-add:name=sys_config_static,t38options="FxFillBit 0, FxTransMMR 0, FxTransJBIG 0,
FxRate Trans, FxUdpEC Red"
```

Example 2-11 illustrates the Hosted UCS- specific static configuration settings required to support DTMF.

Example 2-11 DTMF support

```
prov-add:name=sys_config_static, dtmfsupporteddirection=both
prov-add:name=sys_config_static, dtmfsupportedtype=dtmf
```

Example 2-12 illustrates the static configuration settings required to support transit of the redirecting number parameter.

Example 2-12 Transit of the redirecting number parameter

This is contained in Cisco Unified CM H.225 setup messages—nonStandardControl field)

```
pov-add:name=sys_config_static, h225pavosupported=enabled
```

Example 2-13 illustrates the static configuration settings required to support CLIP and CLIR.

Example 2-13 CLIP/CLIR support

```
prov-add:name=SYS_CONFIG_STATIC,ClipClirSupported=enabled
prov-add:name=CCPackage,A_CC_AnumDataSI=1
prov-add:name=CCPackage,A_CC_Clir=1
```

Applying Static Configuration to the Cisco H.323 Gatekeeper

This section explains how to apply the static configuration to the Cisco H.323 gatekeeper. This is required before using USM to load the bulk data that integrates the Cisco H.323 Gatekeeper into the Hosted UCS platform.

An H.323 gatekeeper is included in the Hosted UCS platform to provide basic infrastructure capabilities. It provides registration capability for the Cisco PGW (through the Cisco HSI), Cisco Unified CM, and any H.323 customer devices. The gatekeeper forces all routing to use the Cisco PGW rather than to operate between Cisco Unified CM clusters.

Configure the following static configuration settings on the gatekeepers in global configuration mode:

Example 2-14 Gatekeeper Static Configuration

```
gatekeeper
zone local HUCS_ZONE cisco.com
gw-type-prefix 999#* default-technology
no shutdown
```

CHAPTER 3

Managing the Hosted Unified Communications Services Platform with VisionOSS USM

This chapter explains how to use VisionOSS BVSM to view, configure, and provision the resources and components of the Hosted Unified Communications Services (UCS) platform. It includes the following topics:

USM GUI Overview

USM provides integrated, hierarchical, role-based administration of Hosted UCS platform components. This is required to support scalable, decentralized administration of a multi-tenant hosted communications service. The administrative hierarchy is as follows, from top to bottom:

```
Providers

   Channels (resellers)

      Customers

         Division administrators

            Location administrators

               Users
```

Each administrator or user has access to lower levels in the hierarchy, but not at higher levels. This allows the secure delegation of authority from provisioning at the provider or reseller level, down to self-care at the level of customer end users.

The following list shows configuration capabilities at various levels:

- Self-care (manual configuration of a specific phone)—Users
- Auto-provisioning and device configuration for each site—Location/division administrators, channel/reseller administrators
- Bulk data loading—Location/division administrators, channel/reseller administrators

Caution To maintain platform-wide data integrity, use USM for configuring or provisioning all Hosted UCS components whenever possible. If it is necessary to configure Hosted UCS platform components directly, be certain that the changes will not affect system integrity.

Loading Bulk Data for Initial Configuration

Before performing any other configuration, you will typically use the USM Deployment (Bulk Data) Tool to perform the initial configuration of the Hosted UCS platform components.

For information about using bulk data loading for the initial configuration of the Hosted UCS platform components, refer to "Chapter 4, "Using Bulk Loaders for the Initial Configuration of Hosted Unified Communication Services Components."

USM provides a platform-wide view that includes Cisco Unified CM and the Cisco PGW. When administrators enter or edit data using USM, the necessary configuration for Cisco Unified CM and Cisco PGW is performed automatically.

Setup Tools

The **Setup Tools** option lets you set up and configure the USM management platform when establishing the platform for a new deployment. It sets up all the internal capabilities for USM in terms of the services, phones, and features that are presented on the USM GUI.

When you select the **Setup Tools** option on the VisionOSS menu, the system displays the screen shown in Figure 3-1.

Figure 3-1 Setup Tools Options

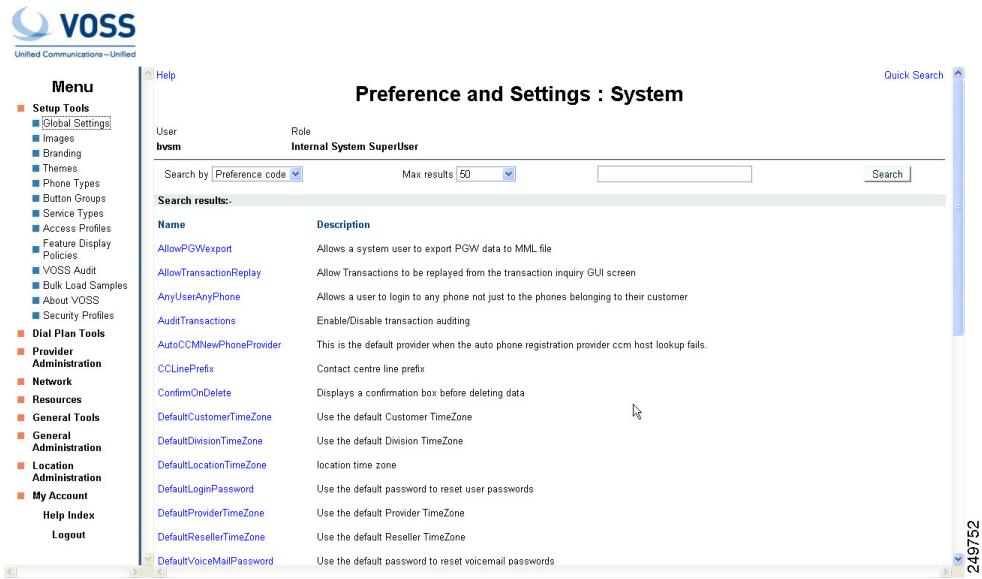


Table 3-1 Summarizes the function of each option provided on the Setup Tools menu.

Table 3-1 Setup Tools Menu Options

Option	Description
Global Settings	Sets preferences at the global level for the whole platform. These settings are similar to preferences at the provider, customer, and location levels.
Images	Loads images, such as logos, used for branding.
Branding	Lets service providers customize the USM GUI with specific colors, labels, icons, and logo, on a platform-wide basis, or for each customer.
Theme	The system supports skinning of the web interface via the use of themes. This includes customization of display elements, including Cascading Style Sheets, images and page layout. The themes are managed via a themes archive, a ZIP file containing the styles, images and template files. A system-wide default theme is provided. The default theme can be used as a base for customization but the default them cannot be changed or deleted using the Theme management screens.
Phone Types	Creates phone types within USM and adds them to the USM menus, which helps deliver services consistently. After the phone type is added, it appears to users in the USM phone menus.
Button Groups	Creates service types within USM and adds them to the USM menus, which helps deliver services consistently. After the service type is added, it appears to users in the USM services and feature group menus.
Service Types	Creates phone types within BVSM and adds them to the BVSM menus, which helps deliver services consistently. After the phone type is added, it appears to users in the BVSM phone menus.

Table 3-1 Setup Tools Menu Options (continued)

Option	Description
Access Profiles	Access profiles are used to manage a users access to various features and functions within the system. Access Profiles enable administrators to control, to a much better granularity, access to features within the predefined security roles.
Feature Display Policies	Customer administrators must be able to enforce access rules for any phone line and mobility feature setting exposed in Self Care. These access rules are grouped together as policies.
VOSS Audit	Transaction Auditing is a powerful tool that enables administrators to closely monitor all or specific transaction types processed within the system.
Bulk Load Samples	Sample bulk loader and configuration model spreadsheets are provided with the default installation.
About VOSS	The About VOSS page contains two important pieces of information, the VOSS software (system) version and the VOSS Server platform.
Security Profiles	Security profiles are sets of rules that govern password strength and other security related preferences. Security Profiles can be applied at System level and at various hierarchical levels which are Provider, Reseller, Customer, Division, Location and Users.

Dialplan Tools

The **Dialplan Tools** option lets you create the dial plan configuration for the Hosted UCS platform during initial deployment.

The dial plan tells the Cisco Hosted UCS platform how to route calls against a number plan. USM is responsible for configuring and loading the dial plan into the various components within Hosted UCS, such as the Cisco PGW and Cisco Unified CM. Super users can create and manage the dial plan when first deploying the platform as well as whenever the dial plan is revised.

The dial plan used for Hosted UCS is designed using Excel spreadsheets, which are loaded using the **Deployment** (**Bulk Load Tools**) option from the General Tools menu.

Dial plan templates can be customized for each provider and after loading, can be further customized for specific customers and locations. For example, each location may require a different extension number length.

When you select the **Dialplan Tools** option on the VisionOSS menu, the system displays the screen shown in Figure 3-2.

Figure 3-2 Dialplan Tools Menu Options

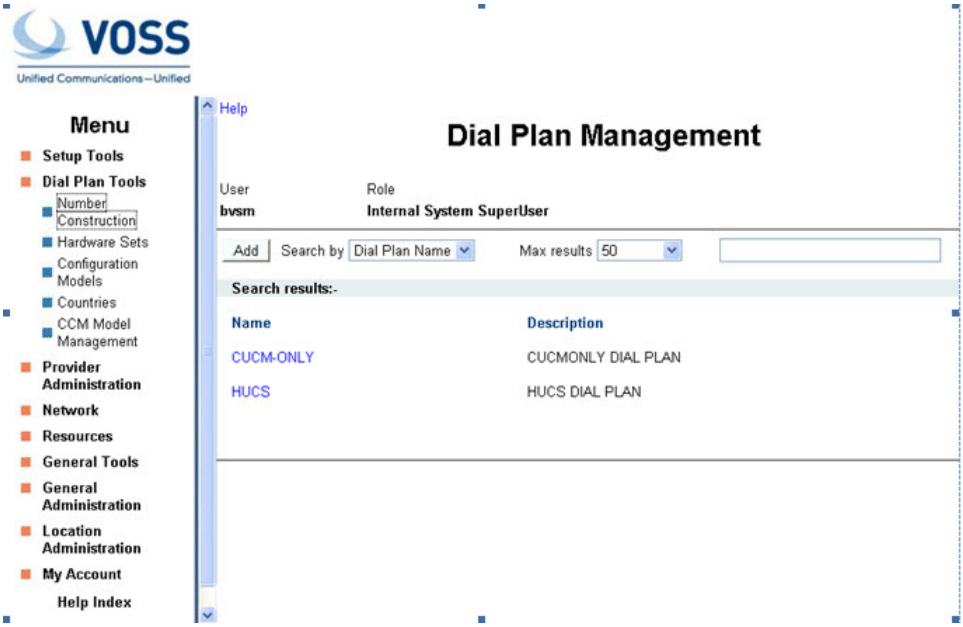


Table 3-2 summarizes the function of each option provided on the Dialplan Tools menu.

Table 3-2 Dialplan Tools Options

Option	Description
Number Construction	Configures dial plan variables such as the following: • Codec settings (compression and decompression standards) • Site number formats • Site display formats • Multi-tenant capabilities • Dial prefixes • E.164 number formats
Hardware Sets	Defines the templates for USM to use for hardware configurations and defines the dial plan associated with each hardware set.
Configuration Models	Provides model loader programs (similar to bulk data loaders) as well as various sample templates. Each model loader is designed for a specific Hosted UCS platform component, including the following: • Cisco PGW • Cisco Unified CM • Gateways • Application services such as voice mail

Table 3-2 Dialplan Tools Options (continued)

Option	Description
Countries	Providers are able to operate in a multi-country environment. Each country however, has unique dial plan elements and number configurations, so the system needs to apply these different configurations to each location based on the country they are allocated.
CMM Model Management	The CCM Model Management page enables admistrators to manage their dial plan configuration models, primarily the models route patterns and translation patterns.

Provider Administration

The **Provider Administration** option lets super users create or change provider settings for the entire Hosted UCS platform, including adding or changing provider administrators.

Caution The default super user account is *BVSM*, and the default password is *password*. After accessing USM, change the password for the BVSM super user account to a strong password. For greater security, create a super user account with a less obvious account name and a strong password, and delete the default super user account.

When you select the **Provider Administration** option on the VisionOSS menu, the system displays the screen shown in Figure 3-3.

Figure 3-3 Provider Administration Menu

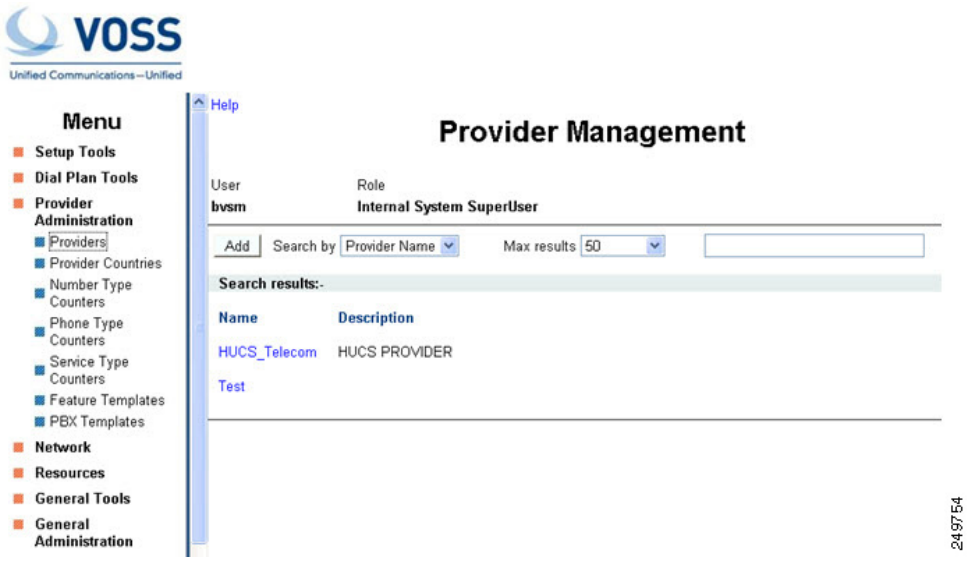


Table 3-3 summarizes the function of each option provided on the Provider Administration menu.

Table 3-3 Provider Administration Options

Option	Description
Providers	Opens the Provider Management page, where you can access and edit provider preferences.
Provider Countries	Add or delete countries under a provider.
Number Type Counters	Manage the reserved inventory for lines by type of line.
Phone Type Counters	Manage the inventory for phones by type of phone.
Service Type Counters	Manage the reserved inventory for phones, lines, and services.
Feature Templates	Manage feature groups, which are a combination of features required for all users or phones.
PBX Templates	All PBX features and functionality managed by the system are defined as PBX Features. Any single version of the system will support a defined set of PBX features. These features can be grouped into PBX templates, a PBX template being a pre-configured set of PBX features.

Network

The **Network** option allows you to set up the network infrastructure so that USM can then perform its management role.

When you select the **Network** option on the VisionOSS menu, the system displays the screen shown in Figure 3-4.

Figure 3-4 Network Menu Options

Table 3-4 summarizes the function of each option provided on the Network menu.

Table 3-4 Network Options

Option	Description
IOS Devices	Add, delete, or modify IOS devices.
Gatekeepers	Add, delete, or modify gatekeepers.
Transit Switches	Add, delete, or modify Cisco PGW switches.
PBX Devices	Add, delete, or modify Cisco Unified CM servers.
DHCP Servers	Add, delete, or modify DHCP servers.
TFTP Servers	Add, delete, or modify TFTP servers.
VoiceMail Gateways	Add, delete, or modify voice mail gateways.
IP Edge Devices	Add, delete, or modify IP edge devices.
Console Servers	Add, delete, or modify switchboard servers.
Music Servers	Add, delete, or modify music on hold (MOH) servers.
Conference Servers	Add, delete, or modify conference servers.
Transcoder Servers	Add, delete, or modify transcoder servers, used to translate between codecs.
Annunciator Servers	Add, delete, or modify Annunciator servers.
Media Termination Point	Add, delete, or modify the media termination points.
VoiceMail Servers	Add, delete, or modify voicemail servers.
Directory Servers	Add, delete, or modify directory servers.

Table 3-4 Network Options (continued)

Option	Description
Emergency Responder	Add, delete, or modify emergency responders.
IVR	Add, delete, or modify IVR.
Hardware Groups	Add, delete, or modify hardware.
Session Border Controllers	Add, delete, or modify Session Border Controllers.
Contact Centre	Add, delete, or modify Contact Centre.

Resources

The **Resources** option allows you to create the necessary resources and make them available to the relevant location.

When you select the **Resources** option on the VisionOSS menu, the system displays the screen shown in Figure 3-5.

Figure 3-5 Resources Menu

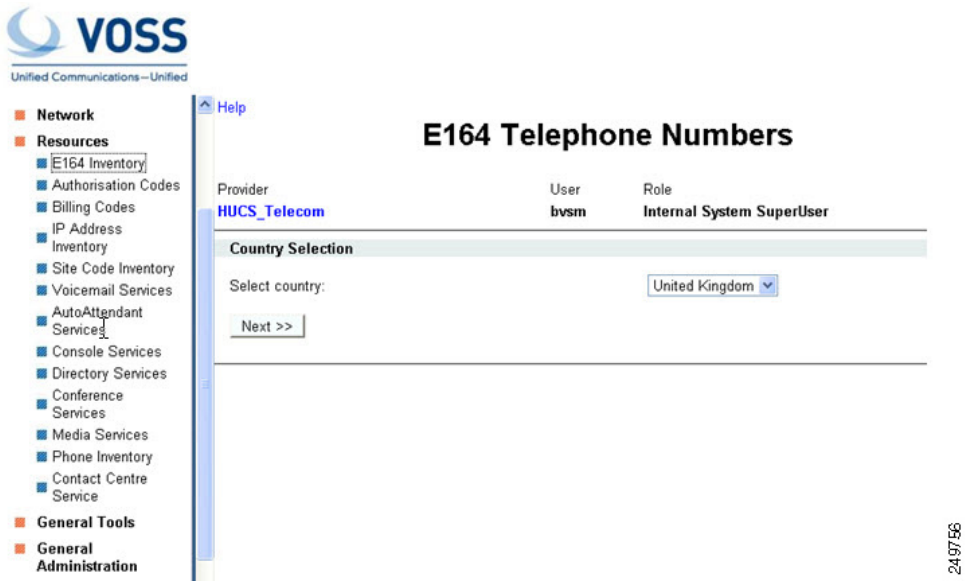


Figure 3-5 summarizes the function of each option provided on the Resources menu.

Table 3-5 Resources Options

Option	Description
E164 Inventory	E164 numbers, also called PSTN or DDI numbers, identify the phone to the external PSTN. BVSM ensures that each E164 number is only allocated once.
Authorisation Codes	Authorization Codes or Forced Authorization Codes (FACs) enable you to manage call access and accounting. The codes regulate the type of calls that specific users can make by forcing the user to enter a valid authorization code before they can make a call.
Billing Codes	Provides access to billing codes.
IP Address Inventory	Provides access to the IP address inventory.
Site Code Inventory	Used as the short-code dial prefix before internal direct dial numbers for a location, allowing internal calls between sites to be routed directly over the internal network.
VoiceMail Services	Creates voice-mail services for each customer, which can then be managed by the customer administrator within each location.
AutoAttendant Services	Creates auto attendant services for each customer, which can then be managed by the customer administrator within each location.
Console Services	Creates console services for each customer, which can then be managed by the customer administrator within each location.
Directory Services	Creates directory services for each customer, which can then be managed by the customer administrator within each location.
Conference Services	Creates conference services for each customer, which can then be managed by the customer administrator within each location.
Media Services	Creates media services for each customer, which can then be managed by the customer administrator within each location.
Phone Inventory	Creates, moves, and deletes phones within or between customer locations.
Contact Centre Service	Creates contact centre services for each customer.

General Tools

The **General Tools** option provides access to the following menu choices:

- **Operations Tools**
- **Bulk Load**

- **Transactions**

For detailed information about using the **Bulk Load** option for initial configuration of the Hosted UCS platform components, refer to Chapter 5, "Configuring Hosted UCS Components."

When you select the **General Tools** option on the VisionOSS menu, the system displays the screen shown in Figure 3-6.

Figure 3-6 General Tools Menu

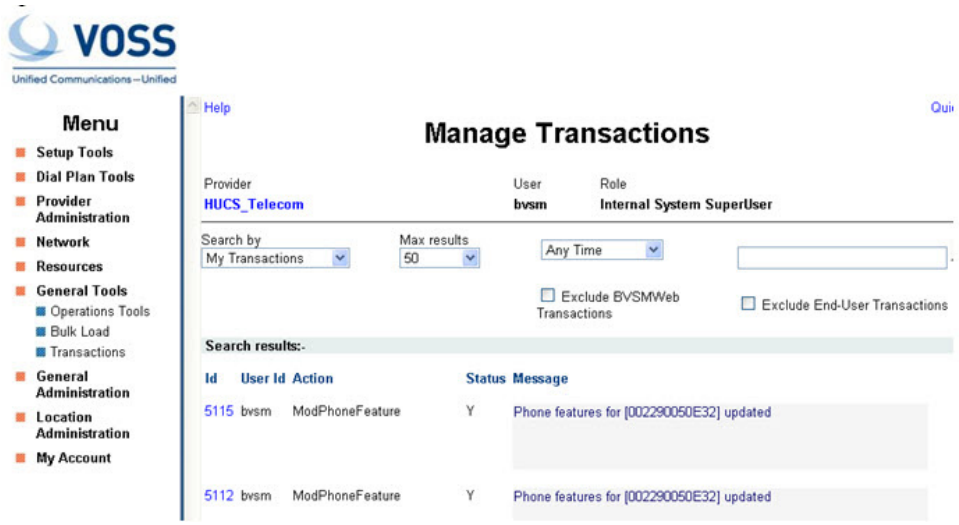


Figure 3-6 summarizes the function of each option provided on the General Tools menu.

Table 3-6 General Tools Options

Option	Description
Operations Tools	Automates multi-step processes. The operations tools are also used for testing purposes when a 360-degree test needs to be performed, such as adding a location, deleting a location, and then adding the same location again. These options are also useful for refreshing a location when adding a new dial plan to legacy locations.

Table 3-6 General Tools Options (continued)

Option	Description
Bulk Load	Loads bulk data into USM using Excel spreadsheets. Providers must load bulk data, including network elements, channels, customers, users, and CPE resources, before services can be delivered. Loading bulk data speeds up platform configuration, especially during the initial phases or with a large amount of data.
Transactions	Provides a chronological record of failed and successful activities associated with each user. This feature is useful for troubleshooting USM and for providing an audit trail for administration moves, adds, and changes.

General Administration

The **General Administration** option lets you navigate between locations, divisions, customers, and resellers. However, the Status menu provides a faster means of changing levels when you are working in the Location Administration menu.

You cannot jump to a lower level because USM does not know which branch of the customer tree you will follow. The best way to jump straight to a location is by selecting the Location key under the **General Administration** option and step down the levels from reseller, customer, and division. This is faster than selecting one level at a time.

When you select the **General Administration** option on the VisionOSS menu, the system displays the screen shown in Figure 3-7.

Figure 3-7 General Administration Menu Options

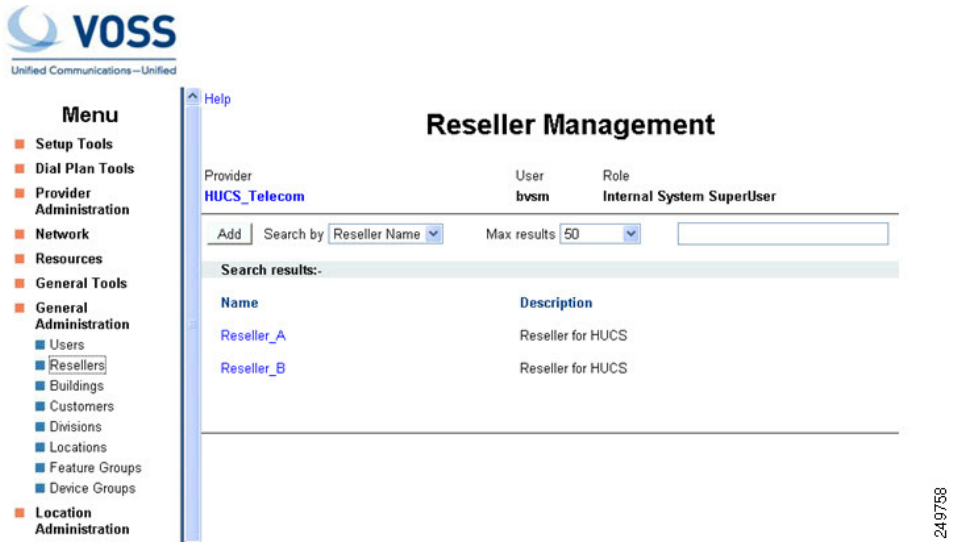


Table 3-7 summarizes the function of each option provided on the General Administration menu.

Table 3-7 General Administration Options

Option	Description
Users	Manage users (repeated in Location Administration).
Resellers	Manage channels (provider administrator only).
Buildings	Manage building (building administrator only).
Customers	Manage customers (reseller administrator only).
Divisions	Manage divisions (customer administrator only).
Locations	Manage locations (division administrator only).
Feature Groups	Manage feature groups (managed at customer level).
Device Groups	Manage device Groups. Device Groups are an optional administrative sub-division of Locations. They are used for defining a set of resources within a location

Location Administration

The **Location Administration** option contains links for the main administrative processes. When you select the **Location Administration** option on the VisionOSS menu, the system displays the screen shown in Figure 3-8.

Figure 3-8 Location Administration Menu Options

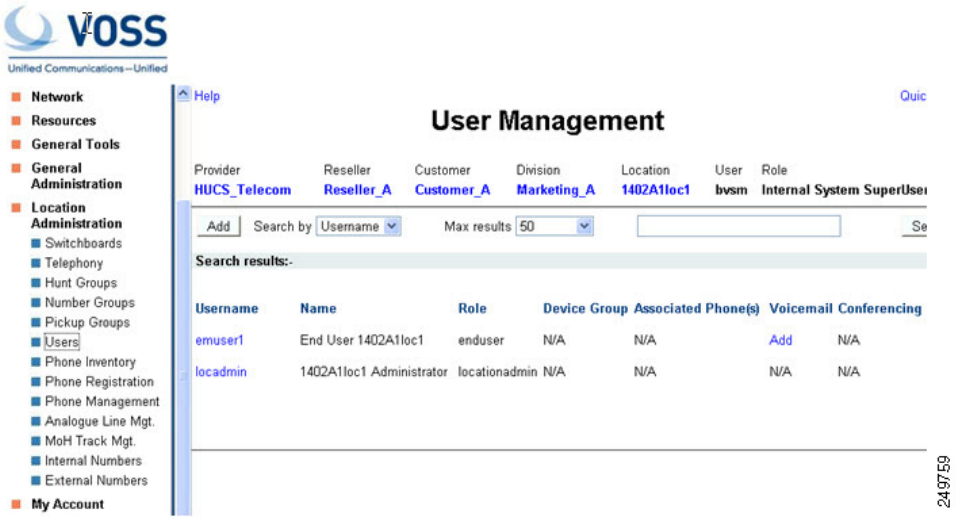


Table 3-8 summarizes the function of each option provided on the Location Administration menu.

Table 3-8 Location Administration Options

Option	Description
Switchboards	Add and manage switchboards.
Telephony	Manage telephony services.

Table 3-8 Location Administration Options (continued)

Option	Description
Hunt Groups	Add and manage hunt groups.
Number Groups	Add and manage number groups.
Pickup Groups	Add and manage pickup groups.
Users	Add, delete, and modify users.
Phone Inventory	Add, move, register, associate, and delete phones.
Phone Registration	Register and un-register phones.
Phone Management	Manage phones.
Analogue Line Mgt	Manage analog lines.
MOH Track Mgt.	Add and manage Music on Hold tracks.
Internal Numbers	Manage internal numbers.
External Numbers	Manage external (DDI) numbers.

My Account

The Account Settings page is used to manage administrator details, passwords and preferences.

Note The My Account section of VOSS can only be used to view and modify the details of the administrator that is currently logged in.

When the user selects the **My Account** option on the VisionOSS menu, the system displays the screen shown in Figure 3-9.

Figure 3-9 Self Care Menu Options

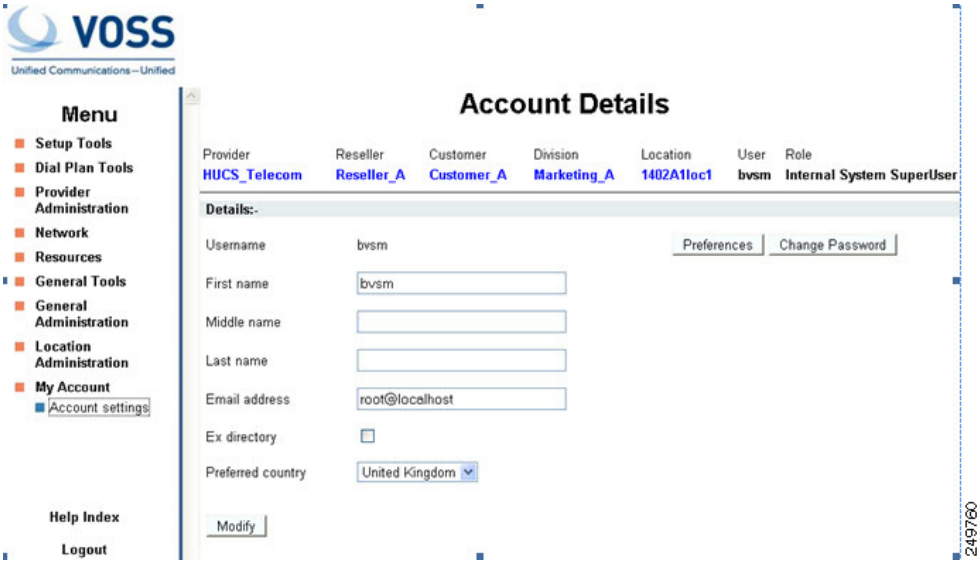


Table 3-9 summarizes the function of each option provided on the My Account menu.

Table 3-9 My Account ***Options***

Option	Description
Account Settings	The Account Settings page is used to manage administrator details, passwords and preferences.
Directory	Displays the customer user accounts and associated phone numbers, but does not allow these details to be modified. When a user account is added to the system, BVSM automatically adds it to the directory.

CHAPTER 4

Using Bulk Loaders for the Initial Configuration of Hosted Unified Communication Services Components

This chapter describes the process for building and configuring a Cisco Hosted Unified Communications Services (Hosted UCS) platform using USM Bulk Loaders.

Before loading bulk data, complete the prerequisites in Chapter 1, "Introducing Cisco Hosted Unified Communications Services," and apply the static configuration described in Chapter 2, "Configuring Hosted Unified Communications Services Components Before Loading Bulk Data."

This chapter contains the following sections:

Using USM Bulk Loaders

This section provides an overview of the USM bulk-data loaders provided with the Hosted UCS platform for getting started with provisioning of the platform.

This section contains the following topics:

Overview

Bulk loaders are specially formatted Excel spreadsheets that let you perform multiple configuration changes in the database of VOSS USM. They can be used to load bulk data into the USM database both faster and easier than manual configuration using Graphical User Interface (GUI) of USM. The bulk loaders are a key part of the platform build process and follow a strict format. However, you should be able to convert an existing set of loaders to your needs rather than starting from the beginning. The bulk

loaders need not be used for all the bulk loading tasks. For most tasks, the data can be loaded using the GUI of the USM itself. These loader sheets are built using Microsoft Excel, and are loaded into USM by importing them directly.

Format

The bulk loaders are in the form of Excel workbooks with several worksheets in each of them. The worksheets have the configuration data grouped according to the function, and worksheets can also be grouped according to the high-level function. Several workbooks are provided with the Hosted UCS platform.

The two sets of bulk loader workbooks provided with the Hosted UCS platform are as follows:

- Dial Plan models
- Bulk Configuration-data loaders.

Dial Plan Models

The workbooks in this set are typically grouped as follows, but this grouping is not mandatory.

1. CCM-HUCS71a Model
2. CCM-Model-CUCM-ONLY
3. IOS-Device-HUCS71a Model
4. IOS-Device-Model-CUCM-ONLY
5. Netwise-HUCS71a Model
6. PGW-MML-HUCS71a Model
7. PGW-TimesTen-HUCS71a Model
8. Movius-HUCS71a Model

Bulk configuration-data loaders

The workbooks in this set are typically grouped as follows, but this grouping is not mandatory.

1. BaseData.xls
2. ProviResel.xls
3. Network.xls
4. Customers.xls
5. Resources.xls
6. Buildings.xls
7. Divisions.xls
8. BuildingLocations.xls
9. Locations.xls
10. LocAdmin.xls

The Dial Plan models and the Configuration-Data loaders are to be loaded separately, and the Dial Plan models are to be loaded first. This is because the dial plan models comprise the intelligence that instructs USM about what it has to provision at the component-level while loading the configuration data.

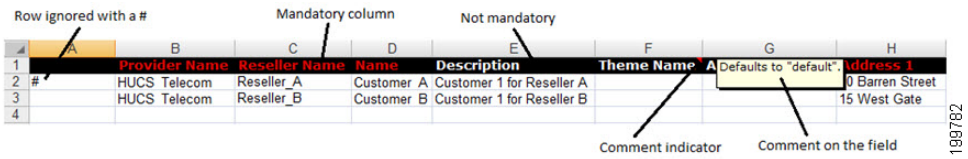
All the sheets in either of the two sets i.e. Dial Plan models and Configuration - data loaders, could be combined to one single workbook also, but it is not advisable because in that case, if a single sheet fails to load, all the others after the failed sheet would also fail to load if they are dependent on the failed sheet's data. It is not required to have the sheets of a workbook in any specific order. Also it is recommended that you use the sample models provided in USM as a reference base for building your customized bulk loader sheets.

Note Currently, bulk loaders can be used by all system administrators by default, but other administrators might not be able to perform all bulk-loading operations. For other administrators to be able to do some operations (add/edit/delete a type of data in USM) they should be associated to access profiles with relevant access rights configured for those operations. These access profiles can be accessed by navigating to **Setup Tools > Access Profiles** page of USM.

The key points about the USM bulk-data loaders to be noted before creating/using them are as follows:

- USM supports only Microsoft Excel spreadsheets with file extension .XLS (Office 2003 default), and not the XML based spreadsheets with .XLSX extension (Office 2007/2010 default).
- The name of the workbooks could be anything, and in case of a failure in loading a part of the workbook the user can always correct that part, comment out the other parts (by setting # to the first column of the rows) or sheets (by prefixing the sheet names with a #), and re-upload the workbook with the same name as before or a different name.
- The first row of every sheet is the header row, and it contains the column names. The sheet names and the column names should not be changed as these names are referenced by the USM during data loading. The sheet names used for the bulk loaders can be obtained from the sample loaders provided by VOSS in the USM, **Setup Tools > Bulk Load Samples.**
- To comment out any row so that USM does not process it while importing, insert an additional empty column to be the first column (Column A), and then insert # or ## in that column for the rows that are not to be loaded.
- Some headers in the sample loader sheets provided with VOSS are highlighted in Red indicating that they are mandatory fields, and those columns require row values for each row with valid data.
- Some fields require a value only from a valid list of values. To assist the user with this, and also to provide information on the default values, or any other helpful info, comments are added to the corresponding header cells which could be read by placing the mouse pointer over the cells.

Figure 4-1 An example of the Configuration-data Bulk Loader sheets with some features marked

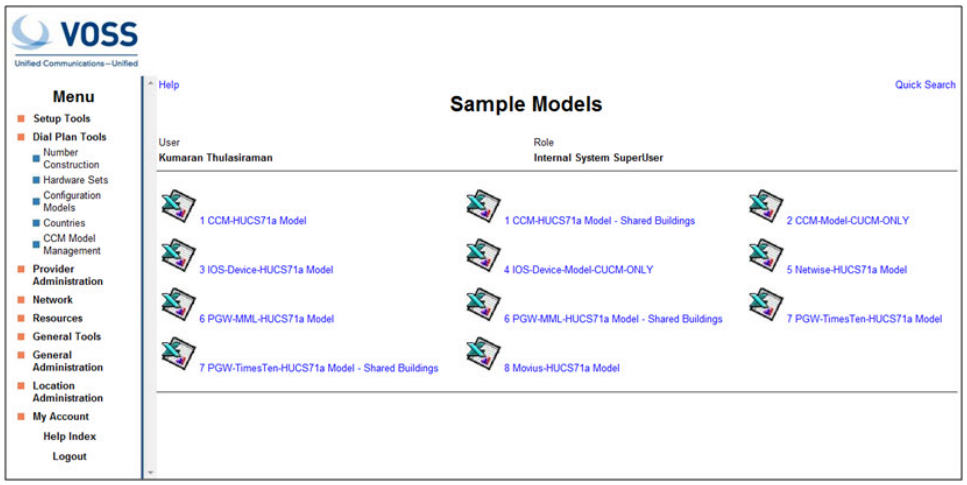


The sample Dial Plan models could be obtained from USM by navigating to the Dial Plan Tools and selecting the Configuration Models tab. The sample Configuration-data bulk loader sheets could be obtained from the Setup Tools > Bulk Load Samples page.

Dial Plan Model Loaders

Dial plan models should be loaded first. To display the sample dial plan model loaders, click the **Configuration Model** option on the Dialplan Tools menu, and select the Sample Models link (see Figure 4-2)

Figure 4-2 Dial Plan Tools-Configuration Models Option-Sample Models



Note Do not make any changes to the dial plan models without specific instructions from your system integrator, VisionOSS, or the Cisco Hosted UCS technical design team.

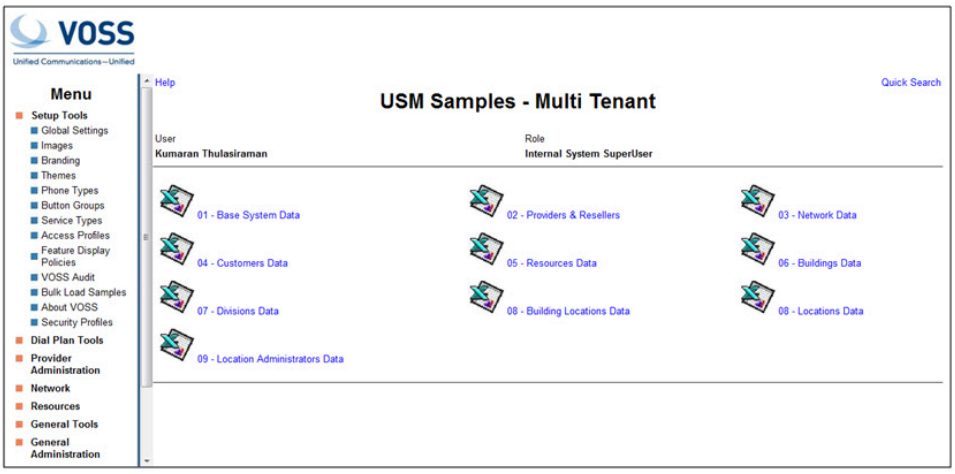
These models have the intelligence that drives the provisioning of the Hosted UCS platform by USM. If loading a particular model loader workbook failed, you can correct the model with the required fix, and re-upload it again even without commenting out the sheets that went through successfully last time. This is because the relevant configurations that are provided by the workbook are always cleared from the system before each model workbook is loaded. The first sheet of the model workbook contains the information on which configurations are cleared, before it starts loading from the second sheet onwards.

Configuration - data Loaders

The configuration loaders contain the network configuration data as well as the configuration data for providers and customers. The Network Configuration data defines the Hosted UCS network components, individual component configuration, and the various platform-wide associations, sets, and groups. Configuration data for Providers and Customers include the associations of the network components to individual customers, Provider and Customer details, and the platform deployment configuration that goes all the way from defining locations under a customer to the configuration for individual phones in those locations.

To display the sample configuration data loaders, click the Bulk Load Samples option under the Setup Tools menu. (see Figure 4-3).

Figure 4-3 Setup Tools-Bulk Load Samples



To save the spreadsheets provided by the Samples option, right-click on the Excel icon and select **Save As**. You can freely explore and experiment with these bulk loaders, which are straightforward in design.

Before loading a production environment with a large amount of data, experiment by loading a small amount of data and use the USM GUI options to explore the effects on the system configuration.

Note The sample bulk data is provided only as a general reference. It is recommended that you contact VisionOSS or Cisco Advanced Services for the most current sample bulk data as your starting point.

Performing the Initial Configuration

This section describes the steps to use the bulk loaders to set up and load the Hosted UCS platform components. This section includes the following topics:

Overview

Before loading bulk data, complete the prerequisites described in Chapter 1, “Introducing Cisco Hosted Unified Communications Services,” and apply the static configuration described in Chapter 2, “Configuring Hosted Unified Communications Services Components Before Loading Bulk Data.”

The procedures in this section assume that a set of USM loaders have been created for the target Hosted UCS platform. These should be pre-tested in simulation (manual) mode.

The following is the order in which the bulk data must be loaded to USM:

1. Dial Plan Models
2. Bulk Configuration-data loaders

The order of loading the individual workbooks under each of the above two groups is the same order in which the workbooks are listed in the section **Format** under **Using USM Bulk Loaders**.

The specific pages in USM to be used for bulk loading are,

- Bulk Configuration-data Loading: General Tools > Bulk Load Tools
- Dial Plan Model Loading: Dial Plan Tools > Configuration Models > Model Loader

Loading the Dial Plan Model Workbooks

To load the dial plan model workbook, follow the following steps.

Procedure

Step 1 Choose Dial Plan Tools > Configuration Models > Model Loader.

Step 2 Select **Schedule New Job**.

Step 3 Click **Browse** and select the Spreadsheet that you would like to upload.

Step 4 Specify whether there is a specific date and time when you would like the spreadsheet to be uploaded in the field **yyyy-mm-dd** and **hh:mm:ss**. If you would like the upload to occur as soon as possible, check the **Execute as soon as possible** check-box.

Note You can schedule multiple sheets one after another and they will queue up and be processed in the order scheduled.

Step 5 Select the relevant file encoding type from the drop-down list then click **Submit** to schedule the upload

Note Select File Encoding - This option is used to set the file encoding. The default is set to UTF8 and will be able to load any sheet containing normal characters. If you are using special characters (àûüý*Úëá ...) in your sheets, UTF8 will display these characters as □□□□□□□□. To load them properly, please select the correct file encoding.

Loading the Configuration - data Loader Workbooks

To load the Configuration-data workbooks in the USM, follow the following steps.

Procedure

Step 1 Choose **General Tools > Bulk Load.**

Select **Schedule New Job.**

Click **Browse** and select the Spreadsheet that you would like to upload.

Specify whether there is a specific date and time when you would like the spreadsheet to be uploaded in the field **yyyy-mm-dd** and **hh:mm:ss**. If you would like the upload to occur as soon as possible, check the **Execute as soon as possible** check-box.

Select the relevant file encoding type from the drop-down list and click **Submit** to schedule the upload. The status and history of all the loading is displayed. The status of a loader can be any of the following:

- Scheduled: This status indicates that the scheduled time of the load has not occurred yet. The load will be started at the time scheduled.

- Validating: The load process is validating the data in the workbook. This occurs before any loading occurs. The current sheet being validated can be seen on the details page for that load. The whole workbook will be validated and if there are any errors the load will not progress to the next phase. The details of the validation failures can be seen in the log.
- Loading: This status indicates the process has passed validation and is starting transactions. The current sheet and row being worked on can be seen on the details page for the load. To check whether the transactions has started, click **Show Transactions**.
- Completed: This status indicates the load process has completed processing the data in the sheets. The errors count indicates the number of transactions that are not completed successfully. The details of the transactions can be seen in the transaction log or by clicking Show Transactions.
- Failed: Generally indicates the internal issues in the load process. The log file for the load can be seen for more details.

The job number or the filename both provide a link to see the details of the load. This provides more detailed information about links to the log files and the transactions scheduled by the load.

Hosted UCS 7.1 Platform configuration

For the Hosted UCS 7.1 platform, the following steps are recommended to be followed in loading the bulk loaders

Procedure

Step 1 Load **1-BaseData.xls** workbook using the procedure explained in the previous section.

Step 2 Load **2-ProviResel.xls** workbook.

Note Changes done to the Base Data configuration will not reflect in the configuration for the already existing providers. So, verify the Base Data configuration in the 1-BaseData.xls workbook to be correct before loading the 2-ProviResel.xls workbook.

Step 3 The functions Add CUCM Groups, Add CUCM Media Resource Groups and Add CUCM MRG Lists cannot be implemented using bulk loaders due to the limitation described in <>. Therefore, remove the sheets with these names from the **3-Network.xls** workbook, and save them separately. Also, remove the **Import CCM Items** sheet. Load the 3-Network.xls workbook after removing the sheets. Check for errors before proceeding to the next step.

Step 4 Do the following steps manually after loading the 3-Network.xls workbook for all CUCM clusters in the platform.

a. Navigate to **Network > PBX Devices**.

b. Select **cluster** by clicking the cluster name.

c. Inside the cluster configuration page, click Groups to add the CUCM groups.

d. Click **Add**.

e. In the form that appears after clicking Add, enter the relevant data exactly as it is in the removed sheet 'Add CUCM Groups'. Click **Submit**.

f. For the media resource groups configuration and the MRG lists configuration, click **Media Resources** and **add** the data in the corresponding sheets.

Step 5 Navigate to **Network > PBX Devices**, select the CUCM cluster and click **Load Static Config** in the bottom of the page. Check for errors before proceeding. Do this for all the CUCM clusters available in the network.

Step 6 Navigate to **Network > Transit Switches**, select the PGW and click **Load** in the bottom of the page. Check for errors before proceeding. Do this for all the transit switches/PGW available in the network.

Step 7 Update media resource group list in trunk config for all the IPPBX as per following steps.

a. Go to **Network > PBX Devices**.

b. Select the PBX.

c. Click **Trunk Config**.

d. Select **External** Trunk.

e. Select the correct **Media resource Group List** from the drop-down menu. Verify that DHCP helper addresses are loaded on the edge devices.

Step 8 Load the **Import CCM Items** sheet as a separate workbook now.

Step 9 After adding the Provider Countries, it is required to manually edit the incoming PGW trunk. Follow these steps to do it. . If you are adding any other country besides GBR, do the following static configuration on PGW before loading the Provider Countries sheet.

For adding e.g. US, prov-add:rtlist:name="rtlist2pstn1",rtname="route2pstn",distrib="OFF"

a. SSH to the PGW.

b. Type mml

c. Start Provisioning (e.g. m**ml command prov-sta::srcver="active",dstver="P044dp",confirm**)

d. Edit trunk group properties (e.g. **prov-ed:trnkgrpprop:name=<incoming PGW trunk name>,custgrpid="P044"**)

e. Issue **Prov-dply** or **Prov-cpy** depending on whether the PGW has a standby server or not.

Step 10 Add CUCM sets by following the steps mentioned below.

a. Navigate to **Network > PBX Devices.**

b. Select **Associated Devices** for a CUCM cluster, and click **Add**.

c. Enter a name and provide a description to the set.

d. Select the **Transaction Type** as **Add Customer**.

e. From the available devices shown, select all the CUCM clusters that will be used by any customer using the selected cluster. This step is mandatory, and should be done before adding the customers to the USM database.

Note Understanding of the CUCM Sets is required before making this configuration.

Load the following workbooks in the same order as they are listed below.

1. 4-Customers.xls
2. 5-Resources.xls
3. 6-Buildings.xls
4. 7-Divisions.xls
5. 8-BuildingLocations.xls
6. 8-Locations.xls

7. 9-LocAdmin.xls

Note At this point of time, the Hosted UCS platform should have been provisioned completely with all the phones registered. Make sure that the transactions in USM are completed successfully with no errors.

Testing and Verifying Initial Configuration

This section describes the various methods to test the configuration added to the USM database using Bulk Loaders, and provisioned on the platform. This section includes the following topics:

Testing the Platform in the Data Center

To test the configuration in the data center, complete the following steps.

Procedure

Step 1 Preconfigure a Cisco line-powered switch in the data center lab and connect test phones.

Ensure that DHCP helper addresses are correctly set.

Step 2 Load a test customer and location using the designated edge device in the data center.

Step 3 Load, provision, and register the test phones.

Step 4 Verify that the phones physically register with the Cisco Unified Communications Manager.

Step 5 Test the platform by making calls on the test phone.

Testing the Platform in the Network

To test the configuration in the live network, complete the following steps.

Procedure

Step 1 Preconfigure a Cisco line-powered switch on the physical provider network and connect test phones.

Step 2 Load a test customer and location using the designated edge device on the network.

Step 3 Load, provision, and register the phones.

Step 4 Check that the phones physically register with the Cisco Unified CM subscriber. If the phones work in the lab but not on the network, troubleshoot the network, including the firewall.

Step 5 Test the platform by making calls on the test phone.

Verifying Bulk Loading

This section describes the procedures for verifying the successful loading of the Bulk configuration data in the USM using Bulk Loaders:

- Verify that USM indicates successful loading with no failure messages.

  Load failure is normally due to incorrect static configuration settings.
- Look for: Request Succeeded—InitIPPBX and InitTransit.

Note After loading the Cisco Unified CM publisher from USM, Cisco recommends restarting all the Cisco Unified CM cluster servers (Windows Restart).

Verifying Cisco Unified Communications Manager Publisher from the System Menu

To verify each instance of Cisco Unified CM Publisher, complete the following steps from the System menu of each Cisco Unified CM Publisher:

Procedure

Step 1 For Cisco Unified CM, verify that only subscriber servers are configured as Cisco Unified CMs.

Note In large clusters, USM configures *all* the cluster servers as Cisco Unified CM (subscribers), and it is necessary to delete out Publisher, TFTP, and MOH servers.

The maximum permitted number of Cisco Unified CM servers is eight.

Step 2 For Cisco Unified CM Group, verify each cluster PhonesGroup and TrunkGroup.

Step 3 For Region, verify Default and Trunk configuration.

Step 4 For Device Pool, verify each trunk; for example, MC.

Step 5 For Location, verify each location.

Verifying Cisco Unified Communications Manager Publisher from the Services Menu

To verify each instance of Cisco Unified CM Publisher, complete the following steps from the Services menu of each Cisco Unified CM Publisher:

Procedure

Step 1 For Media Resource–Media Resource Group, verify that the correct MRGs have been loaded and that these contain the correct resources.

For example, VS-R2-C1-Phones-MRG and VS-R2-C1-Trunks-MRG, containing VS-R2-CONF-1 and MOH_VS-R2-C1-P.

Note It is important to use fully-defined media resource group (MRG) and MRG list (MRGL) naming conventions when supporting multiple clusters.

Step 2 For Media Resource–Media Resource Group List, verify that the correct MRGLs have been loaded and that these contain the correct MRGs.

For example, VS-R2-C1-Phones-MRGL and VS-R2-C1-Trunks-MRGL.

Verifying Cisco Unified Communications Manager Publisher From the Devices Menu

To verify each instance of Cisco Unified CM Publisher, complete the following steps from the Devices menu of each Cisco Unified CM Publisher.

Procedure

Step 1 For gatekeepers, verify that a loaded gatekeeper exists.

Step 2 For trunks, verify that an external trunk exists.

You need to reboot the Cisco Unified CM servers supporting the trunk to ensure that the Cisco Unified CM cluster registers with the gatekeeper.

Step 3 Logon to the gatekeeper (for example, using Telnet), and enter:

```
show gatekeeper endpoints
```

Enter the following command to verify that you can see trunks from all subscribers.

```
show gatekeeper status cluster
```

Step 4 For Device Settings–Device Profile Default, verify that the phone button template on the 7940 default profile has been set to USM 7940.

Step 5 For Device Settings–Device Profile Default, verify that 15 Device Profiles have been set: Cisco 7902 profile through to Cisco ATA 186 profile.

Verifying Cisco PGW After Loading Bulk Data

To check for USM-TO-PGW transaction errors, complete the following steps.

Procedure

Step 1 Use SSH to access PGW (for example, using PuTTY) and log in.

For example user ID *mgcusr* and password, *cisco*.

Step 2 Enter the following commands to login, for example, to the GL-D-PGW host:

```
GL-D-PGW% cd ../etc
GL-D-PGW% cd cust_specific
GL-D-PGW% ls
```

Step 3 To display a list of PGW log files, enter the following command:

```
GL-D-PGW% grep Error  *.output
```

Step 4 To view a particular error file and look for Warnings and Errors, enter the following command.

```
GL-D-PGW% more [filename*]   for example, more 15119aaaaad*
```

Step 5 To confirm that the dial plans exist on the active and standby Cisco PGW, enter the following **mml** command:

```
prov-exp:all:dirname="midrange”
```

Replace *mydirname* with any name you want to use.

Step 6 Enter the following command:

```
cd/opt/CiscoMGC/etc/cust_specific/mydirname
```

Step 7 Look for files such as ICCM.mml, which are the dial plans.

Step 8 Ensure that the same files are present on both active and standby.

Step 9 If not, choose **prov-sync** on the active, or restart the standby.

Step 10 Ensure that ICCM has been populated.

Step 11 View the mml log for error messages.

```
cd /opt/CiscoMGC/var/log
```

Step 12 To list error messages, enter the following command:

```
grep -I DENY mml*
```

This will list any errors.

Step 13 View these files, looking for DENY to identify what failed.

CHAPTER 5

Backing Up and Reinitializing Hosted Unified Communications Services Components

This chapter explains how to backup and reinitialize the Hosted Unified Communications Services (Hosted UCS) platform components. It also provides some recommendations for upgrading to a newer version. This chapter includes the following sections:

Backing Up Cisco Unified Communications Manager and Cisco PGW

This section outlines the process for storing a known, reliable configuration before a platform upgrade and includes the following topics:

After backup, the stored configuration can be restored onto a Hosted UCS platform component, if required. For example, restoring the initial static configuration for the Cisco Unified CM or Cisco PGW eliminates the time-consuming reconfiguration process.

Backing-up Cisco Unified Communications Manager

To use Disaster Recovery System (DRS) for backing up and restoring configuration, complete the following steps.

Procedure

Step 1 Go to **Disaster Recovery System**, using the Cisco Unified CM OS admin credentials. **(CCM7.x)**.

Step 2 Create a backup device on the Cisco Unified CM, to add a network based backup device with access credentials.

Step 3 Choose **Manual Backup > Select the Device Name.**

Step 4 Check the **CCM** check box.

Step 5 Click **Start Backup**.

After approximately five to ten minutes, you should see the TAR file in Backup Device folder.

Backing Up the Cisco PGW

Note If your Cisco PGW 2200 Softswitch is a continuous service system, (active-standby) ensure that you perform backup procedures on both Cisco PGW 2200 Softswitches.

Note You can run the various backup operations that are described in the following sections only when you are logged in to your system as mgcusr. You cannot perform any backup operation while you are logged in as root.

To perform a manual backup operation, enter the following UNIX command on the Cisco MGC:

mgcbackup -d *path* [**-r** *retries* **-t** *retry_time*]

Where:

- *path*—The full path of the directory in which to store the backup file; for example, a directory on a remote server that you have mounted on your system, or the local tape drive.

Note Cisco recommends that you do not store backup files on your local Cisco MGC host, because storage of backup files on the local host reduces the amount of disk space available to process call data and does not ensure that the data is safe in the event of a natural disaster.

- *retries*—The number of times to check for an active provisioning session on the Cisco MGC before aborting the backup operation. The default value is 0 and the maximum value is 100.

Note A backup operation cannot start while there is an active provisioning session on the Cisco MGC.

- *retry_time*—The number of seconds to wait between checks for an active provisioning session on the Cisco MGC. The default value is 30 seconds and the maximum value is 3600 seconds.

For example, to perform a manual backup operation where the backup file is saved to a directory path called /dev/rmt/h0, with a maximum of three attempts, each 60 seconds apart, you would enter the following UNIX command:

```
mgcbackup -d /dev/rmt/h0 -r 3 -t 60
```

The backup file is stored in the specified directory path in the following format:

`mgc_hostname_yyyymmdd_hhmmss_backup`

Where:

- *hostname*—The name of the Cisco MGC host, such as MGC-01.
- *yyyymmdd*—The date the backup file is created, in a year-month-day format, such as 20011130.
- *hhmmss*—The time the backup file is created, in an hour-minute-second format, such as 115923.

For more information on backup operations, see the "Backing Up System Software" in Chapter 3 of the *Cisco PGW 2200 Softswitch Release 9.8 Operations, Maintenance, and Troubleshooting Guide* at the following URL:

http://www.ciscosystems.cg/en/US/docs/voice_ip_comm/pgw/9.8/Maintenance/Guide/r9chap3.html#wp1229244

Restoring Cisco Unified Communications Manager and Cisco PGW Configuration

This section explains how to restore the configuration for Cisco Unified CM and the Cisco PGW. It includes the following topics:

Restoring Cisco Unified Communications Manager Configuration

To restore Cisco Unified CM, follow the Disaster Recovery System (DRS) restore process and then restart the Publisher and the Subscribers.

Restoring the Cisco PGW Configuration

This restoration method uses a script to restore the configuration data for the Cisco MGC software, UNIX administrative files, and the Main Memory Database (MMDB).

Note These procedures assume that you have backed up your system configuration data regularly. The procedures for system configuration backup can be found in Backing Up the Cisco PGW, page 5-2.

Listing the Cisco PGW Backup Files

To list the backup files in a particular directory path, enter the following UNIX command on the Cisco MGC:

mgcrestore -d *path* **-l**

Where *path* is the directory path in which you have stored backup files, such as a directory on a remote server or a local tape drive.

The system returns a response similar to the following:

```
Backup files in /var/cisco
-------------------------------------------------
mgc_venus_20011010_153003_backup
mgc_venus_20011011_153003_backup
mgc_venus_20011012_153003_backup
```

Restoring the Cisco PGW Backup File

CiscoMGC service on the PGW must be stopped before restoring a backup file.

Log on PGW as root user and run the following command to stop mgc service.

```
PGW-ENT2M% /etc/init.d/CiscoMGC stop
```

To restore the configuration data stored in a particular backup file, enter the following UNIX command on the affected Cisco MGC to run the restore script:

mgcrestore -d *path* **-f** *filename*
Where:

- *path*—The directory path to the location where your backup files are stored.
- *filename*—The file name of the backup file you want to restore.

For example, to restore a backup file called mgc_venus_20011012_153003_backup stored in a directory path called /var/cisco, you would enter the following command:

```
mgcrestore -d /var/cisco -f mgc_venus_20011012_153003_backup
```
After restoring the backup file, start the mgc service on PGW as root user.

```
PGW-ENT2M% /etc/init.d/CiscoMGC start
```
For more information on backup operations, see “Restoring Procedures for Cisco MGC Software Release 9.1(5) and up” in Chapter 5 of the Cisco PGW 2200 Softswitch Release 9.8 Operations, Maintenance, and Troubleshooting Guide at the following URL:
http://www.ciscosystems.cg/en/US/docs/voice_ip_comm/pgw/9.8/Maintenance/Guide/r9chap6.html#wp1722661

Restoring the Cisco PGW to Clean Status

To clear the Cisco PGW and restore it to its original static configuration, complete the following steps.

Procedure

Step 1 Logon to the Cisco PGW using Reflexion Host – Unix.

On test systems: username=*mgcusr.* password=*cisco*.

Step 2 Enter the text shown in boldface:

```
GL-D-PGW% mml
GL-D-PGW mml> prov-sta::srcver="pure-static",dstver="iBVSconfig",confirm
  MGC-01 - Media Gateway Controller 2005-06-30 14:09:55.352 BST
M  COMPLD
 "PROV-STA"
;
```

Note pure-static in the MML command above is an example name of the static configuration MML session on the PGW. If you have static MML configuration session saved on your PGW, use that session name to restore PGW static configurations

```
GL-D-PGW mml> prov-cpy
   MGC-01 - Media Gateway Controller 2005-06-30 14:10:02.164 BST
M  COMPLD
"PROV-CPY"
;
GL-D-PGW mml> quit
```

In this example, the entry iBVSconfig is a temporary name. The exact name is not important.

Backing up and Restoring USM

The USM automatically backups the database within the cluster and between active and standby USM servers. USM always maintains four copies of the database, two in each headend. If a copy of the database needs to be saved offsite, you can set up an export copy of the database.

In most cases to date, offsite backup has occurred once every 24 hours. This should occur at a time of low provisioning traffic, such as in the early hours of the morning.

For the backup to be useful as part of a disaster recovery plan, the USM backup needs to be in a consistent state with those taken for the Cisco PGW and Cisco Unified CM, along with IP Unity if they are included in the Hosted UCS platform. To ensure a consistent state, there should be a USM transaction freeze while the platform is being backed up.

If all the backups are taken at the same time, it becomes possible to time-shift the entire platform back to the latest backup without any misalignment between USM and the servers that it controls.

Note To use the CLI, a client capable of SSH is required. The most common SSH clients include Putty (Windows platform) or the SSH client in a terminal (UNIX / Linux or Mac OS X platform).

Note The default login username is usmcli and the default password is voss. If you are unable to login using the default login details, please contact your dedicated support person so that they can reset the default password and verify that this functionality has been enabled for your system.

It is important to ensure that the commands below are carried out on the active VoSS server. These commands do not, and should not, be run on the inactive VoSS director.

To use the CLI, do the following:

Step 1 SSH to the cluster/standalone system and login as the user usmcli. You will be presented with a prompt as follows:

```
Password:
Welcome to the VOSS-USM Standalone Console
=>
```

Step 2 To enter enable mode (similar to other networking hardware) type enable as follows:

```
=>> enable
=>#
```

The prompt should changes from "=>>" to "=>#" to reflect this change.

Step 3 Type "dbbackup" command on the CLI as follows:

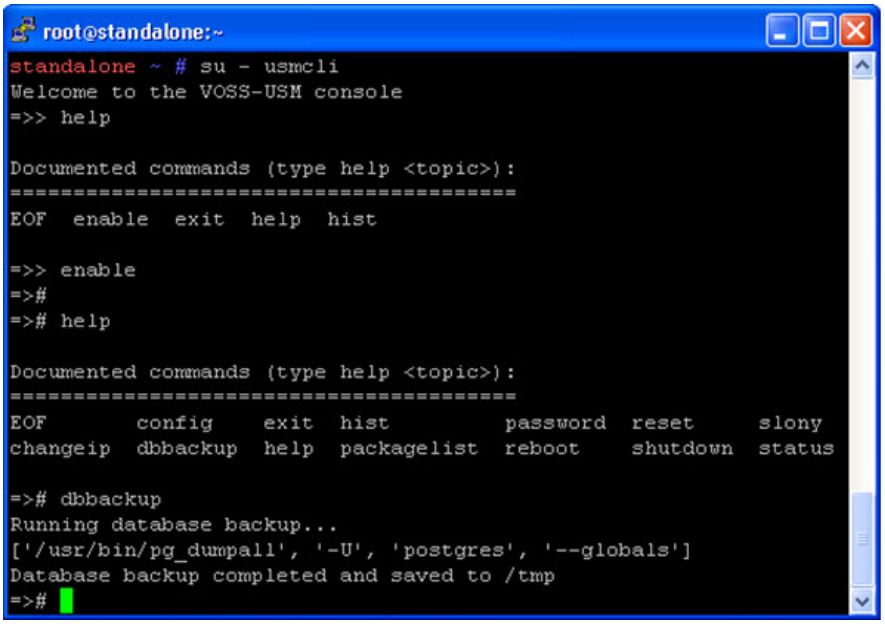
```
=># dbbackup
```

Note Only the database is backed up. Any other configuration information currently such as DNS Configuration, syslogs, Any Manual Changes (i.e. changes to file content), bulk loader sheets etc., must be backed up manually by copying those files.

Note If you are rebuilding a system however, these files would be recreated during the installation of the platform and software.

The USM database backup is performed as shown in Figure 5-1.

Figure 5-1 Backup USM database using CLI



Restoring USM

Before restoring USM batabase backup, you must stop the relevant services on the USM system.

Procedure:

Step 1 Log on USM as root user.

Step 2 Stop all services and restart Postgress: (Standalone, on cluster environment you just need to restart postgress). In a Standalong USM system, run the following command to stop all the services:

```
for svc in usm-autoregister loader_scheduler extdhcp apache2 usm_batch selfcare memcached
estraier iptparent iptqueue iptdevmn voss-monit voss-imq postgresql-standalone postgresql;
do /etc/init.d/${svc} stop; done
```

If USM is deployed as a clutster, use the following command to stop the services:

```
/etc/init.d/heartbeat-wrapper stop
Restart postgresql services:
/etc/init.d/postgresql-stanalone restart
/etc/init.d/postgresql restart
```

Step 3 Perform the restore, see example below:

```
/opt/VOSS/usm/scripts/deployment/db-backup-restore.py --restore
/root/db-backup-ipt-2010-02-11.sql
```

Step 4 After restoring backup file start all USM services. In a Standalone USM system, run the following command to start all the services:

```
for svc in voss-monit iptdevmn iptqueue iptparent estraier memcached usm_batch selfcare
apache2 extdhcp loader_scheduler usm-autoregister; do /etc/init.d/${svc} start; done
```

If USM is deployed as a clutster, use the following command to start the services:

```
/etc/init.d/heartbeat-wrapper start
```

Clearing a Cisco Unified Communications Manager Cluster

This section describes the process for clearing a Cisco Unified CM cluster, in preparation to re-build the Hosted UCS platform.

The order of the clearing steps is not important and further clearing steps may be required on some Hosted UCS platforms. For example, you may need to delete organizations within the Movius server, using the Movius server Sysconfig GUI.

When you start the rebuild process, you must complete all stages. It is not possible to go back after you have cleared one component in the architecture.

You must clear the Cisco Unified CM before a rebuild to ensure that there will be no data duplication or mismatch between USM and the Cisco Unified CM.

You can quickly restore the Cisco Unified CM publisher to its initial state by restoring a CUCM backup file.

When no DRS backup file is available, to clear the Cisco Unified CM and avoid any interdependency issues, complete the following steps.

Procedure

Step 1 Delete phone devices.
Choose 50 phones for the search list to allow deletion of 50 phones at a time.

Step 2 From the **Call Routing > Translation Patterns** menu, delete Translation Patterns, 50 at a time.

Step 3 From the **Device > CTI Route Points** menu, delete any CTI route points used.

Step 4 From the **Call Routing > Route/Hunt > Route Pattern** menu, delete all route patterns.

Step 5 From the **Call Routing > Route/Hunt > Route List** menu, delete all route lists.

Step 6 From the **RoutePlan > Route/Hunt > Route Group** menu, delete all route groups.

Step 7 From the **Device > Trunks** menu, delete all trunks.

Step 8 From the **Device > Gatekeepers** menu, delete all gatekeepers.

Step 9 From the **Device > Gateways** menu, delete all gateways.

Step 10 From the **Media Resources > MediaResourceGroupList** menu, delete all media resource group lists.

Step 11 From the **Media Resources > MediaResourceGroup** menu, delete all media resource groups.

Step 12 From the **System > Locations** menu, delete all locations (show 50 at a time).

Note The Cisco Unified CM does not allow to delete the default locations, device pools and regions. Do not attempt to delete default configurations.

Step 13 From the **Media Resource > Conference Bridge** menu, delete any conference bridges that are not required.

Keep the conference bridges that are required by USM.

Step 14 From the **Application > CiscoCM Attendant Console > Pilot Points** menu, delete pilot points used if any.

Step 15 From the **System > Device Pool** menu, delete all device pools, except Default.

Step 16 From the **System > Region** menu, delete all regions except Default.

Step 17 From the **Call Routing > Route/Hunt > Hunt Pilot** menu, delete all hunt pilots used.

Step 18 From the **Call Routing > Route/Hunt > Hunt List** menu, delete any hunt lists used.

Step 19 From the **Call Routing > Route/Hunt > Line Group** menu, delete any line groups used.

Step 20 Delete all users, either one-by-one via the CCMAdmin group or in bulk using the BAT Tool facilities.

Step 21 From the **Call Routing > Directory Numbers** menu, delete any directory numbers used.

Step 22 From the **Call Routing > Call Pickup** Group, delete any call pickup numbers used.

Step 23 From the **Call Routing > Call Park** menu, delete an call park numbers used.

Step 24 From the **System > Cisco Unified CM Groups** menu, delete all Cisco Unified CM groups except Default.

Step 25 From the **Device > Device Settings > Device Profile** menu, delete all profiles, including “Logout” service.

Step 26 From the **Call Routing > Route Plan Report** menu, search for unassigned DNs and select **Delete All Found Items** at the bottom of the search page.

This allows deletion of 150 unassigned DNs at a time.

Step 27 For voice-mail profiles, voice-mail pilot numbers, and MWI numbers, unless these need to be maintained.

Step 28 From the **Call Routing > Class of Control > Calling Search Space** menu, delete all CSSs except **IncomingToCluster**.

Step 29 From the **Call Routing > Class of Control > Partitions** menu, delete all partitions.

Step 30 From the **Call Routing > Route Plan Report** menu, search for assigned DNs and delete DNs one at a time.

Note If issues occur, use the dependency record feature to search for components that might be preventing deletion of records.

Initializing the Cisco PGW

This section describes the clearing process for the Cisco PGW by deleting the USM-created file before rebuilding a Hosted UCS platform.

You must clear the Cisco PGW before reloading a Hosted UCS platform. Clearing the Cisco PGW means clearing out USM data but not other configuration information that may have been set up on the Cisco PGW servers independently of USM.

To initialize the Cisco PGW, complete the following steps.

Procedure

Step 1 Logon to the active Cisco PGW.

Log in over Telnet or SSH, using a terminal console program, such as PuTTY.

On test systems, the user account/password is mgcusr/cisco.

To configure the up arrow operate to add back previous lines, use the following:

```
PGW % setenv TERM vt100
```

Step 2 To verify that you are logged into the active Cisco PGW, enter the following commands:

```
PGW % mml
mml > rtrv-ne
```

Step 3 To create a binary backup to allow rollback if required, choose your own filename.

For example, *270710-01bin*.

```
mml> prov-sta::srcver="active",dstver="270710-01bin"
mml> prov-stp
```

Step 4 To create a text backup for diagnostics if required, choose your own filename.

For example,***270710-01text*.**

```
mml>prov-exp:all:dirname="270710-01text"
```

Step 5 Restore the process, if rollback is required:

```
mml> prov-sta::srcver="270710-01bin ",dstver="270710-03bin "
mml> prov-dply (Dual server PGW platform)
or
mml> prov-cpy (Single server PGW platform)
```

Step 6 For the Cisco PGW reset process (dial plans only), enter the following commands:

```
mml> quit
% cd /opt/CiscoMGC/etc/cust_specific
% ls -la
```

This gets a list of files stored including Text file.

```
% cd /opt/CiscoMGC/etc/cust_specific/270710-01text
% ls
```

Step 7 Make a note for all four-character mml files loaded by USM.

For example, copy ICCM.mml, P974.mml, XXXX.mml, XXXX.mml into Notepad.

```
% mml
mml> prov-sta::srcver="active",dstver="270710-02bin"
mml> numan-dlt:dialplan:custgrpid="XXXX"
```

where *XXXX* is the name of each four-character mml file.

Step 8 Repeat this process until all XXXX.mml files have been deleted. If you hit a dependency, go to the next file and cycle through until all files are deleted.

Note All the dial plans are not deletable manually due to interdependencies between dial plan. Cisco recommends to back up PGW static configuration and restore the PGW static configuration when PGW needs to be cleaned up.

Step 9 Reload the ICCM dial plan as an empty file:

```
mml> numan-add:dialplan:custgrpid="ICCM",overdec="YES"
mml> prov-dply (Dual server PGW platform)
or
mml> prov-cpy (Single server PGW platform)
```

Step 10 On completion, take a further backup of the Cisco PGW.

This will be the static configuration of the Cisco PGW if, for example, the Cisco PGW needs to cleared by deleting static settings.

Step 11 To create a binary back-up to allow rollback if required, choose your own filename.

For example, *VSR2-151007Static-HB-01bin*.

```
mml> prov-sta::srcver="active",dstver="2710001-01bin"
mml> prov-stp
```

Initializing USM

This section explains how to clear an existing USM platform that has already been loaded with dial plans and data.

Clear the USM database when you are planning to rebuild the Hosted UCS platform. The clearing process is much faster than deleting all the data manually through the USM GUI and even faster than the Delete Bulk Loader tool or Operations tools. This is especially the case if USM has many customers and locations already loaded.

To clear a USM cluster, complete the following steps.

Caution There is no way to undo a destroy procedure, so check whether you are on the correct server.

Note The consequences of running this command are great. Please ensure that you fully understand the impact of running this command prior to using it

Procedure

Step 1 SSH to the cluster/standalone system and login as the user usmcli.
You will be presented with a prompt as follows:

```
Password:
Welcome to the VOSS-USM Standalone Console
=>>
```

Step 2 To enter enable mode (similar to other networking hardware) type enable as follows:

```
=>> enable
=>#
```

The prompt should changes from "=>>" to "=>#" to reflect this change.

Step 3 Type "reset app" command on the CLI as follows:

```
=># reset app
    Really reset the application? ? [y/N] : y
    clearing status...
    Application reset complete.
=>#
```

Step 4 Reboot the USM system.

```
=># reboot
```

Note You only need to destroy Standalone VOSS system or VOSSDir1 because USM automatically replicates to the other servers.

APPENDIX A

Sample Hosted Unified Communications Services Build of Materials

This section covers the standard bill of materials (BOM) for Hosted Unified Communications Services (Hosted UCS) platforms. Each platform BOM differs based on the special requirements of each deployment.

This appendix contains the following sections:

BOM Tool

Cisco and VisionOSS have developed a special BOM tool that calculates the equipment requirements for a set of unique requirements. Users enter their requirements into a spreadsheet and the BOM Tool calculates the equipment requirements. This tool is available from the Cisco Hosted UCS product manager.

BOM Examples

This section contains the following topics:

Reference Platform BOM Example

Table A-1 contains a high-level BOM example for reference platforms.

Table A-1 High-Level BOM Example for Reference Platforms

Hardware Item	Hardware Platform	Vendor	Quantity
Cisco Unified CM	MCS 7845	Cisco	2
PGW2200	Dual Netra X4200 (Dual AC PSU)	Sun	1
HSI	Dual Netra X4200 (Dual AC PSU)	Sun	1

Table A-1 High-Level BOM Example for Reference Platforms

Gatekeeper	2851 VSEC Bundle	Cisco	1
PSTN gateway	AS5400XM	Cisco	1
Distribution switch	Catalyst 3750-48 Port	Cisco	1
Line-powered switch	Catalyst 3750 POE	Cisco	1
IP Phones (testing)	7900, 6900, 8900, 9900 series	Cisco	16
USM provisioning	m2010-FT series servers	VisionOSS	2
Flat-panel display, keyboard, and mouse drawer unit	Tbc	tbc	1
KVM switch	Tbc	tbc	1
Power distribution	Tbc	tbc	1
24 U mobile rack unit	Tbc	tbc	1
Mobile shipping container for mobile rack	Tbc	tbc	1
Rack storage unit	Tbc	tbc	1

Production BOM Example

This section provides an example of a BOM for a 5000-seat Hosted UCS production platform (with licenses to 1000 ports), software licenses for UC applications, call aggregation and Voss USM.

Note Refer to Cisco and VisionOSS for the latest BOM Excel worksheets.

Table A-2 shows a sample BOM for the Cisco PGW 2200.

Table A-2 PGW 2200

Product	Description	Qty
PGW2200	Cisco PGW 2200 Softswitch Server Pair	1
SWP22-CCX-RTU	PGW 2200 (Call Control) Voice or Dial RTU License, 1k Ports	1
SFB22-CC-9.7.3	PGW 2200 Application Software Ver 9.6(1) on CD	1
MGC-X42K-AC216-2	Dual Netra X4200 Server (Dual AC PSU), 2 CPU, 16 GB RAM	1
SWP22-H323-RTU	PGW 2200 H.323 Per Call leg Run Time License, 2k Call Legs	1
SWP22-SIP-RTU	PGW 2200 SIP Per Call leg Run Time License, 2k Call Legs	1
SFB22-PBX	PGW 2200 PBX Interface (DPNSS/QSIG) Right-to-Use License	1

Table A-3 shows a sample BOM for the Cisco Unified CM 7.1.5.

Table A-3 Cisco Unified Communications Manager 7.1.5

Product	Description	Qty
CUCM-USR-LIC	Unified Connect Licensing - Top Level part Number For Ordering	1

Table A-3 Cisco Unified Communications Manager 7.1.5

MCS7845H2-K9-CMB2	Unified CM 7.1 7845-H2	3
LIC-CUCM-USR	UCL - 1 Enhanced User	5000
CUCM-7845-71	Cisco Unified CM 7.1 7845	3

Table A-4 shows a sample BOM for the HSI.

Table A-4 HSI

Product	Description	Qty
SFBHSI-4.3	HSI Application Software v4.3	2
MGC-X42K-AC216-1	Dual Netra X4200 Server (Dual AC PSU), 2 CPU, 16 GB RAM	2

Table A-5 shows a sample BOM for the media gateway.

Table A-5 Media Gateway

Product	Description	Qty
AS54XM-16T1-V-HC	AS5400XM High-Density Voice; 16T1, 16 AS5X-PVDM2-64,IP+ IOS	2

Table A-6 shows a sample BOM for the gatekeeper.

Table A-6 Gatekeeper

Product	Description	Qty
C2851-VSEC-CUBE/K9	2851 VSEC Bundle w/PVDM2-48,FL-CUBE-125,AVS,64F/256D	2

Table A-7 shows a sample BOM for the Gateway.

Table A-7 Gateway

Product	Description	Qty
C2911-VSEC-SRE/K9	Cisco 2911 SRE Bundle, SRE 300, PVDM3-16, UC and SEC License PAK	2

Table A-8 shows a sample BOM for USM provisioning.

Table A-8 USM Provisioning

Product	Description	Qty
VOSS-Server HW	VOSS-Server m2010-FT	4
VUSM-1KUSR-EE	VOSS-USM 1000 User Essential Edition Feature License	5
VUSM-MS-1Y-1K-EE=	VOSS-USM 1000 User License - Essential Ed. 1 Year 24 by 7 Maint. Support	5

Table A-9 shows a sample BOM for the core LAN switch and firewall.

Table A-9 Core LAN Switch and Firewall

Product	Description	Qty
WS-C3750G-48TS-S	Catalyst 3750 48 10/100/1000T + 4 SFP + IPB Image	2

Table A-10 shows a sample BOM for various options.

Table A-10 Options

Product	Description	Qty
BAMS	Billing and Measurement Server	tbc
SLT	SS7 Signaling Gateway	tbc
Voice mail	Movius Mereon	tbc
Auto Attendant	Movius Auto Attendant	
Ancillary equipment	KMV switch, monitors, keyboards, cabling	tbc
Phones	Cisco phones	tbc
Catalyst 3560	Edge Devices—Ethernet ports (line-powered)	tbc

GLOSSARY

A

A number Calling number.

API Application programming interface.

AXL AVVID XML layer.

B

B number Called number.

BVSM Business Voice Services Manager.

C

CC Country code.

CF Call forward.

CLI Command-line interface *and* calling line identification.

CLIP Calling Line Identification Presentation.

CLIR Calling line identification restriction.

CoS Class of service.

CPID Call processing identifier (unique system-wide).

CSS Calling search space.

CT Call type.

CTI Computer telephony integration.

D

DDI Direct Dial Inward.

DID Direct Inward Dialing.

DP Dial plan.

DPNSS Digital Private Network Signaling System.

E

E.164 ITU-T recommendation defining PSTN numbering plan.

EISUP Extended ISDN user part.

EOL End of line (variable used by BVSM to determine the end of line in each model).

EXT Extension *and* external prefix.

F

FINT Full Internal Number = CPID+RID+SLC+EXTN = Cisco Unified CM DN.

FNN Full national number—E.164 telephone number without area code.

G

GK Gatekeeper.

GUI Graphical user interface.

GW Gateway.

H

H.323 ITU-T umbrella recommendation defining audio-visual protocols on a packet network.

H/M-UCS Hosted/Managed-Unified Communications Solution.

HSI H.323 Signaling Interface.

Hosted UCS Short form of Hosted Unified Communications Services.

I

ICPID Call processing identifier, IPPBX-based.

IOS (Cisco) Internetwork Operating System.

IP Internet Protocol.

ISP Inter-site prefix.

ISUP ISDN user part.

L

LRID Routing identifier, location-based.

M

MGCP Media Gateway Control Protocol.

MML Man-machine language.

MT Multi-tenant.

MWI Message waiting indicator.

N

NOA Nature of address.

P

PBX Private branch exchange.

PCC Padded country code

PGW PSTN gateway.

POTS Plain old telephone service.

PRI Primary Rate Interface.

PSTN Public switched telephone network.

Q

QSIG Q Signaling (ISDN-based protocol for signaling between PBXs).

R

RID Routing Identifier (unique per CPID).

S

SIP Session Initiation Protocol.

SOAP Simple Object Access Protocol.

SLC Site location code (unique within a customer).

SRST Survivable Remote Site Telephony.

SS7 Signaling System 7.

T

TOD Time of day.

V

VM Voice mail.

X

XML Extensible Markup Language.

INDEX

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

X