IP COMMUNICATIONS

Configuring Cisco Unified Communications Manager and Unity

A Step-by-Step Guide

ciscopress.com

David Bateman

Configuring Cisco Unified Communications Manager and Unity Connection:

A Step-by-Step Guide

David Bateman

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

Configuring Cisco Unified Communications Manager and Unity Connection: A Step-by-Step Guide

David Bateman

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing May 2011

Library of Congress Cataloging-in-Publication Number is on file.

ISBN-10: 1-58714-226-0

ISBN-13: 978-1-58714-226-0

Warning and Disclaimer

This book is designed to provide information about configuration and administrative tasks related to Communications Manager and Unity. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The author, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Corporate and Government Sales

Cisco Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales.

For more information please contact: U.S. Corporate and Government Sales
1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside the U.S. please contact: International Sales international@pearsoned.com

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through e-mail at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Executive Editor: Brett Bartow

Managing Editor: Sandra Schroeder

Project Editor: Mandie Frank

Editorial Assistant: Vanessa Evans

Designer: Sandra Schroeder

Indexer: Tim Wright

Cisco Representative: Erik Ullanderson

Cisco Press Program Manager: Anand Sundaram

Development Editor: Marianne Bartow

Technical Editors: David Mallory, Toby Sauer

Copy Editor: John Edwards

Proofreader: Apostrophe Editing Services

Composition: Mark Shirar

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

About the Author

David J. Bateman is a certified Cisco Systems instructor and the director of curriculum development for Skyline-ATS. He has more than 20 years of internetworking experience. For more than 10 years, David was a senior LAN/WAN engineer, working on small, medium, and large networks. Later in his career, he took on the responsibility of running the business operations of a technical services company, while maintaining his existing client base. David has always enjoyed sharing his knowledge, and in 1999, he added to his list of accomplishments by becoming a technical seminar leader. After many successful seminars, he decided to become a full-time Cisco instructor for Skyline Advanced Technology Services. He has been teaching and implementing Cisco voice technologies since 2000. David's years of real-world technical and business knowledge allow him to bring a unique perspective to the classroom, where he not only delivers critical technical knowledge but can also explain how technologies can be used to address various business issues.

About the Technical Reviewers

David L Mallory, CCIE No. 1933, is a technical leader for Learning@Cisco, where he is responsible for content development strategy. For the last seven years, David has been primarily focused on UC certifications and was the technical lead for the Cisco 360 Learning Program for CCIE Voice. Prior to joining Learning@Cisco, David was a systems engineer supporting global accounts. David is a frequent presenter at Cisco Live and has obtained four CCIEs—Routing & Switching, WAN Switching, Security, and Voice.

Toby Sauer is the lead voice instructor and voice curriculum manager for Skyline Advanced Technology Services. He brings 30 years of experience in the traditional voice, data, and VoIP arenas. Toby has been involved in Cisco VoIP since the beginning, working with traditional VoIP, and he was involved in the earliest installations of Cisco Communications Manager. He has installed many different implementations of Communications Manager and was responsible for converting most of the Midwest's Cisco offices from traditional PBX to Communications Manager.

Toby became a Cisco voice instructor in 2000. As the Communications Manager product continued to grow and develop, he was a key instructor to many of the original deployment partners.

Toby currently holds CCNP-Voice, CCNA-Voice, CCNA-RS, CCSI, and various partner-level certifications. He teaches all the Cisco Standard Voice courses and many custom variations of these courses.

Dedications

I'd like to dedicate this book to my parents, who taught me unconditional love; to my wife, Nikki, who is my life, my love, my all; and to Matthew, a young man that I am proud to call my son.

Acknowledgments

There are a number of people that I would like to thank in helping me complete this book. Often the greatest help that can be received is when someone is willing to sacrifice so that you can succeed. With this in mind, I would like to thank my wife, Nikki. She has sacrificed many beautiful summer days that we could have spent out on the motorcycle so that I could work on this book. She sacrificed hours each week reading what I had written in order that I might deliver a more readable copy to the editors. I know it was not always fun for her, but it helped me complete this book. Without her sacrifice, this book would not have been possible.

I would also like to thank the technical editors. Their keen insight and willingness to ask me what the heck I was thinking on some subjects have helped make this a much better book than it was when I first wrote it.

Of course I'd like to thank those at Skyline-ATS, where I work. I would especially like to thank them for the skill they showed in increasing my workload as deadlines for the book drew near. I guess they figured I would do better under pressure. But seriously, I would like to thank Mike Maudlin and Mike Zanatto for their understanding and cooperation during this project. I also need to thank all the others that I worked with at Skyline-ATS. The awesome amount of knowledge that we hold as a team is incredible, and to have such a resource at my disposal has been invaluable.

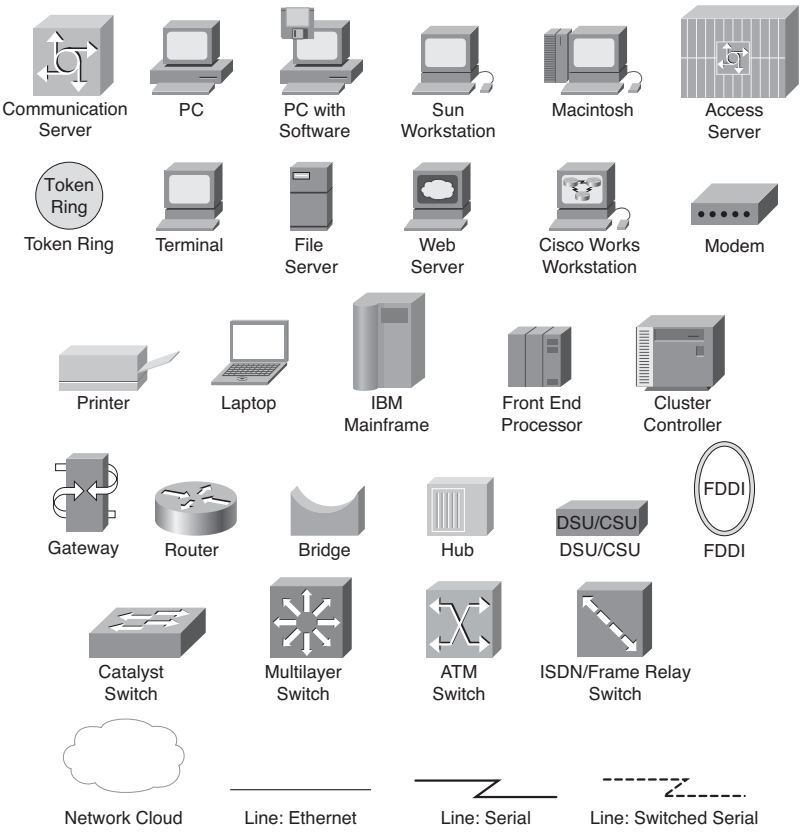
A big thank-you to the folks at Cisco Press: Brett Bartow, who assisted from the beginning of this project and was always there to remind me of upcoming deadlines long enough in advance so that I had time to either meet the deadline or come up with a really good excuse. Also Marianne Bartow, who acted as my development editor and was always helpful and encouraging.

Thanks one and all for all you've done.

Contents at a Glance

Table of Contents

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italics* indicate arguments for which you supply actual values.
- Vertical bars (|) indicate separate alternative, mutually exclusive elements.
- Square brackets [ ] indicate optional elements.
- Braces { } indicate a required choice.
- Braces within brackets [{ }] indicate a required choice within an optional element.

Introduction

On March 10, 1876, Alexander Graham Bell made the first successful telephone call. As with many things, the test was purely accidental. Graham spilled acid on his leg, and Watson, his assistant, heard his call for help through the telephone. So, what has changed over the last 129 years? It would be easier to discuss what hasn't changed. The world of telephony has undergone some significant changes but none as exciting as Voice over IP (VoIP) solutions from Cisco. There are still those who believe we were all a lot better off in an analog world, but you can't stop progress, and the Cisco Unified Communications solutions are starting to grow faster than many had believed.

This new technology brings with it the need for individuals to learn how it works. Although there are many fine Cisco Press books on this technology, I noticed many of my students requesting a task-oriented book. They were looking for a book in which they could look up a specific task and be walked through it. This was the initial goal of the book. Through the writing process, the book evolved from offering only a step-by-step guide into also offering easy-to-understand explanations for many of the Cisco Unified Communications concepts and components.

Goals and Methods

New technologies bring new opportunities and challenges. One of the challenges that we are faced with in the Cisco Unified Communications world is the capability to easily understand the many facets of the configuration and integration process. Because this platform can be deployed in so many different configurations and environments, system administrators and system engineers need a resource that offers quick access to step-by-step solutions. In an environment such as this, it is nearly impossible to keep track of the exact steps for each configuration task. Those tasks that you do on a daily basis are easy to perform, but when you are called upon to perform unfamiliar tasks, you don't always have the time to learn the proper steps. *Configuring Communication Manager and Unity Connection* shows readers how to complete many of the common tasks, and some not-so-common tasks, performed within a Cisco Unified Communications solution.

Who Should Read This Book

The book is aimed at individuals who are required to configure Communications Manager and Unity and Unity Connection solutions as a primary part of their jobs. The book is unique because it covers Communications Manager, Unity, and Unity Connection.

Although this book focuses on the tasks that must be performed, it also offers easy-to-understand explanations for many of the technologies that are commonly found with Cisco Unified Communications environments, which makes it an excellent resource for individuals who are new to this technology.

How This Book Is Organized

Within the book, tasks are organized in the same order in which they would naturally be performed. Some tasks include cross-references to prerequisite tasks. Whenever possible, however, all tasks are presented within the same section.

Different people, depending on their knowledge and background, will use this book in different ways. Many will find it a useful reference tool when completing an unfamiliar task, and those new to this technology will find that reading this book from cover to cover will help them gain a solid understanding of this technology. Although the step-by-step guides were written with the assumption that you have access to a Communications Manager while reading the steps, this is not required. This book includes numerous screen shots, which enable you to see what is happening in the administration interface even if you do not have access to a Communications Manager.

Chapter 1 offers you a high-level overview of most of the concepts and components that are found within Communications Manager and Unity. Basically, the information found in two weeks of classes has been compressed to quickly bring you up to speed. This by no means is a replacement for these classes—just a quick overview.

Chapters 2 through 6 cover Communications Manager configuration, whereas Chapters 7 through 11 discuss Unity and Unity Connection configuration. The last chapter speaks to more advanced features of both technologies and offers a few ways to leverage the strengths of both to create a more feature-rich environment.

The following is a brief description of each chapter.

Chapter 1: CUCM and Unity Connection Overview

This chapter offers a broad overview of the Cisco Unified Communications solutions to ensure that you are comfortable with what follows in the book. The intent of this chapter is to offer you an overview of the various components of a Cisco Voice over IP solution. You are strongly encouraged to refer to suggested reference material for additional information on any topic with which you might be unfamiliar. You can find this material in the appendix.

Chapter 2: Preparing CUCM for Deployment

To ensure a smooth deployment, tasks must be performed in a certain order. In this chapter, you learn what tasks must be completed before adding devices. As with most things, if you fail to create a solid foundation, you will encounter problems in the future. This chapter ensures that the proper foundation is created and future problems are avoided. Topics covered include services configuration, enterprise parameters, and device registration tasks. Additionally, this chapter includes step-by-step instructions for each task.

Chapter 3: Deploying Devices

After the predeployment tasks are completed, you are ready to add devices. This chapter focuses on the tasks required to add various devices to your Communications Manager environment. Devices have been divided into two major categories: clients (IP phones, softphones, and so on) and gateways. The chapter includes step-by-step instructions for adding each device.

Chapter 4: Implementing a Route Plan

Before you can place calls to destinations that are not directly connected to your Communications Manager environment, you must configure a route plan. This chapter discusses all the components of a route plan, such as route patterns, route lists, and route groups and the tasks that are needed to implement an efficient dial plan. The step-by-step tasks show how to create and configure route patterns, route lists, and route groups and more advanced components, such as CTI route points, translation patterns, and route filters.

Chapter 5: Configuring Class of Service and Call Admission Control

After a dial plan is created, you might want to limit what destinations certain devices can reach. This chapter discusses how to do this by configuring Calling Search Spaces and partitions. It is also necessary that some types of Call Admission Control be deployed on WAN links so that the quality of voice is maintained. To this end, Call Admission Control features are covered. Finally, this chapter discusses the importance of special services, such as 911, and describes how to properly configure the dial plan to handle these types of calls.

Chapter 6: Configuring CUCM Features and Services

After basic call-processing functions are configured and working properly, you need to add new features and monitor the health of the system. This chapter explores a number of the features that can be implemented, including IP phone service, media resources, and Extension Mobility. The need for, and the functions of, SRST is also covered in this chapter. Furthermore, this chapter examines some of the monitoring services that are included in Communications Manager. Step-by-step instructions that explain how to add each feature and service are included.

Chapter 7: Unity Predeployment Tasks

The first step to proper configuration is verifying that the integration is correct and that all predeployment tasks are complete. This chapter includes step-by-step instructions for completing predeployment tasks, such as verifying integration, defining system parameters, and creating templates, distribution lists, and CoS.

Chapter 8: User Reference

After a proper integration between Unity/Connection and Communications Manager is achieved and the predeployment tasks discussed in the previous chapter are completed, the user can be added. In this chapter, the different types of users are examined. Then, the process for adding, importing, and managing users is explored. Within the "Managing Users" section, various administrative tasks are discussed, which range from "How to reset a user's password" to "How to properly remove users." Each task includes step-by-step instructions.

Chapter 9: Call Management

One of the system's most useful and often underutilized features is call management. This chapter ensures that the reader understands the way that the system processes a call. The most basic object of the call management system is a call handler. A brief review of how

call handlers work is included in the beginning of this chapter. Additionally, a common use of the system's call management feature is to deploy a basic auto-attendant, which is described within this text. The chapter also addresses some of the more advanced call management features, such as call routing rules and audio-text applications. Complete step-by-step instructions are included within this chapter.

Chapter 10: Implementing Unity Networking

Because many organizations are migrating to Unity/Connection from a voicemail system or have other voicemail systems deployed at other locations, Unity/Connection must communicate with them. Unity can be integrated with these systems through a number of industry-standard protocols. This chapter discusses the different types of networking that can be deployed and looks at how to determine the proper one to use.

Chapter 11: Exploring Additional Tools

Although most day-to-day tasks can be accomplished using the system administrative interface, it is often more efficient to use one of the many tools that are included with Unity/Connection. The tools help accomplish tasks that range from making bulk user changes to migrating users to another server. This chapter introduces the reader to these tools and includes step-by-step details on how to use each of them.

Chapter 12: Maximizing CUCM and Unity Connection Capabilities

As Communications Manager and Unity/Connection evolve, more and more advanced features are added. This chapter looks at a few of these more advanced features, such time-of-day routing and call queuing. In addition, the chapter offers a few examples of features that can be created by taking existing features of each application and adding a new twist to them, such as using Unity as a conference manager.

Target Version

This book was written for Communications Manager, Unity, and Unity Connection versions 8.0 and 8.5. This is not to say that you must run any of these versions for this book to be of value to you. It does, however, mean that some of the step-by-step guides might be slightly different. With each new version, the menus are sometimes moved or slightly changed, or there might be an additional field in the new version. However, none of these issues should cause you great concern. If the field isn't there, don't worry about it. If a menu isn't exactly where you expect it, just look above or below, and you are sure to find it. Including the exact steps for every version of these applications would have made the book larger than you would care to lift, let alone read. Remember that the value of this book goes beyond the step-by-step guides, because it also provides easy-to-understand explanations of many Cisco Unified Communications concepts.

Chapter 1

CUCM and Unity Connection Overview

Before embarking on any worthwhile adventure, it is important that you have a good map and a solid understanding of the purpose of your trip. This chapter provides just that—an introduction to some of the many components that make up a Cisco Unified Communications environment.

Technical books can be divided into one of two categories: "why" books and "how-to" books. Why books provide you with a solid understanding of the technology and explain why you would want to deploy it. How-to books tell you how to deploy a given technology. This is a how-to book. The main purpose of this book is that of a configuration reference. However, it is important that you have a solid understanding of the technology. This chapter provides you with a broad overview of this technology and references to further information. If you are new to this technology, you are strongly encouraged to pursue more in-depth information than is presented in this chapter before deploying this technology. If you haven't been involved in this technology for a while, you might be thinking of skipping this chapter and moving on to the meat of the book. This, of course, is your decision, but reading this chapter can give you a better understanding of the specific technologies discussed later in this book.

After reading this chapter, you should have a high-level understanding of the Communications Manager, Unity, and Unity Connection components and how they fit into a Cisco Voice over IP (VoIP) solution. This chapter has been divided into the following sections:

- Reliable foundation
- Communications Manager overview
- Unity overview
- Unity Connection overview
- Security concerns

Because this technology is a mixture of two preexisting technologies, traditional telcom and traditional data, it is likely that you started out solely in one of these disciplines. Often when you start to learn a new technology, you try to compare it to technologies you've learned. This sometimes causes learners to miss an important point because they were preoccupied with trying to make this new information fit in with previous learning. If you are new to this technology, you should take any current knowledge you have and place it aside while reading. After you have read this chapter and feel that you understand it, you should then integrate it with your current knowledge base.

At first, this can be difficult because we all seem to want to fall back on what we already know. So each time you find yourself doing this, just stop reading for a moment and refocus on acquiring new information, knowing that later you can integrate it with what you already know. Also, try not to make judgments while reading. Many times people have made up their minds about a product or technology before they have even seen it. Even if you are learning this technology because "you have to," be as open to it as possible. Regardless of any person's resistance, technology will not stop or even slow down.

Ensuring a Reliable Foundation

Whether you are building a house or a network, a solid foundation is crucial. In a VoIP network, the foundation is even more crucial because both data and voice will use the same network. This means that you need to implement an even higher level of redundancy than you feel is necessary in a traditional data network. The term *five 9s* is used a lot in the traditional telcom world; this stands for 99.999 percent uptime. The expectation is that any network that carries voice should be up 99.999 percent of the time. This calculates to just a little more than 5 minutes a year of downtime, not including planned downtime for upgrades and maintenance. You might be saying, "That's impossible," but actually it is possible. With the proper planning and design, you can expect to see nearly no downtime. Make note that I said, "with the proper planning and design." There have been a number of VoIP deployments that failed solely because a proper infrastructure was not implemented. Typically, a VoIP environment is broken into four layers. Each layer plays a vital role. An example of the devices that are in each layer follows:

- **Clients:** IP phones
- **Applications:** Unity
- **Call processing:** Communications Manager
- **Infrastructure:** Switches and gateways

Note Calculations of five 9s varied by telephony vendor and typically discounted issues that affected a single user or a small group of users. For example, if you lost a desktop switch and 24 users had no phone service for 3 hours, this wouldn't count against your five 9s.

The foundation of the network is at the infrastructure level, where components such as switches, routers, and gateways reside. A solid understanding of these components is needed to design a solution that can withstand common day-to-day problems that arise on most networks. The discussion begins with a look at these components.

Infrastructure Overview

A properly deployed infrastructure is the key to a reliable network. This section begins by examining the foundation of the infrastructure. The cable is one of the most often overlooked components of the network. This is often because it rarely causes problems after it is installed. Cabling problems normally don't appear until some new type of technology is added to the network. I remember one client that was running a 4-megabit network with no trouble. When he upgraded to 16 megabits, the network started failing and he had to rewire the entire network.

Nowadays, twisted-pair Ethernet is installed in most environments. The Cisco VoIP solution is designed with the assumption that twisted-pair Ethernet is installed at each desktop.

One of the common issues that arises with cabling is when the installer takes a few shortcuts. A common shortcut is failing to terminate all the pairs of the cable. The installer assumes that because Ethernet uses only pins 1, 2, 3, and 6, there is no need to terminate the others. In most cases, the network can function when cabled this way. The problem is, however, that such a network is not installed to industry standards, and all Cisco solutions are based on the assumption that the existing infrastructure is installed according to industry standards. In an environment such as this, you cannot use the Cisco power patch panel because it relies on pins 4, 5, 7, and 8 to deliver power to the phone. Ensure that you have all cabling tested and certified before the deployment begins. As the saying goes, "An ounce of prevention is worth a pound of dropped calls" (or something like that).

After you have the cabling under control, you need to look at the equipment to which the cabling connects. On the one end of the cable you have phones, which is quite straightforward. On the other end, you have the phone plugged into a switch.

Note This discussion assumes that the phone is plugged into a switch, not a hub. Plugging phones into hubs is not advised because all devices on a hub share the same bandwidth, and this can lead to poor voice quality. In addition, do not daisy-chain phones (plug one phone into another).

When deciding which switch to use, a few things must be considered. First, it is recommended that all switches you plan to use within the Cisco VoIP solution are Cisco switches. This is not simply because Cisco wants to sell more switches, but because certain Cisco switches include special features that allow greater functionality within your network. These features include inline power, voice virtual LANs (VLAN), and Cisco Discovery Protocol (CDP) support. This does not mean that switches from other manufacturers cannot be used. It simply means that some features discussed in the following sections might not be supported.

Inline Power

This is the ability to provide power to the phones through the Ethernet cable. There are two inline power schemes:

- The Cisco inline power convention, which uses pins 1, 2, 3, and 6 to provide power. These same wires are used for the transfer of data.
- Power over Ethernet (PoE), as defined in IEEE Standard 802.3af. This is an approved industry standard, which uses pins 1, 2, 3, and 6 or 4, 5, 7, and 8 to provide power. The standard differs from the Cisco inline power scheme in a number of ways, the most significant being the way that power requirement is detected. Current Cisco phones support the 802.3af standard, but keep in mind that older models might not.

The net effect of either standard is the same—power is supplied to the phone through the Ethernet cable. The switches that support either of these conventions can detect whether the attached device requires inline power, and if the device does require inline power, the switch provides it. Having two methods of supplying inline power can be confusing, so it is best that the phones and switches you purchase use the same method.

Voice VLANs

This allows the use of a single switch port to simultaneously support both a phone and a PC by allowing a single port to recognize two VLANs. The PC is plugged into the back of the phone, and the phone is plugged into the switch. The switch then advertises both VLANs. The phone can recognize the voice VLAN and use it. PCs cannot recognize voice VLANs and use the native VLAN.

CDP Support

CDP is a Cisco-proprietary protocol that allows Cisco equipment to share certain information with other Cisco equipment. The phones use CDP to determine whether a voice VLAN is present on that port. It also shares other information such as port power information and quality of service (QoS) information with the Cisco Catalyst switch.

Make sure that the switch you choose supports these features. Most currently shipping Catalyst switches are capable of supporting all these features.

Voice Gateways

After you ensure that the cabling and switches are adequate for a VoIP solution, you are ready to deploy the endpoints. Endpoints can be any of the following: phones, soft clients, or gateways. Of these devices, only gateways are considered to reside at the infrastructure level. Phones and Communications Managers are covered later in this chapter.

In its simplest form, a gateway is a device that allows connectivity of dissimilar networks. In the VoIP world, a gateway connects the Communications Manager voice network to

another network. The public switched telephone network (PSTN) is the most popular network with which IP phones must communicate. The job of the gateway is to convert the data traveling through it to a format that the other side understands. Just as a translator is needed when a person speaks German to someone who understands only Spanish, a gateway is needed to convert VoIP to a signal that the PSTN understands.

The hardware that acts as a gateway varies, depending on what type of network you connect to and what features you require. When choosing a gateway, ensure that it supports the following four core gateway requirements:

- **DTMF relay:** Dual-tone multifrequency (DTMF) are the tones that are played when you press the dial pad on a phone. Many people refer to this as touch tones. Because voice is often compressed, the DTMF can become distorted. The DTMF relay feature allows the DTMF to be sent out of band, which resolves the distortion problem.
- **Supplementary services:** Include hold, transfer, and conferencing.
- **Cisco Unified Communications Manager (CUCM) redundancy:** Supports the capability to fail over to a secondary Communications Manager if the primary Communications Manager fails.
- **Call survivability:** Ensures that the call will not drop if the Communications Manager, to which either endpoint is registered, fails.

Later, the various types of gateways are discussed. For now, understand the purpose of a gateway and the required features.

Creating a Reliable VoIP Infrastructure

In the summer of 2003, the northeastern portion of the United States experienced a widespread power outage. The power outage lasted from 6 hours to 3 days depending upon the area. One of the most impressive and yet understated events that occurred during this time is what didn't happen. For the most part, the PSTN didn't fail, and no one even noticed. Because no one actually noticed shows how much people expect the phones to always work. The power was out, and yet most people didn't think for a second that the PSTN might fail. The system didn't fail because of the highly reliable and redundant infrastructure that has been developed over the years. This is the type of reliability that people have come to expect from the phone system. It has been stated that many people view dial tone as a God-given right, or even one of the inalienable rights in the constitution. (I doubt anyone thinks that, but you get the idea.) With this in mind, you must make every effort to ensure that nothing short of a natural disaster prevents your customer from having dial tone.

The most important thing to keep in mind is that individual components of the system will fail. It is not a question of if something will fail, but when. Because components will fail, it is up to you to determine how to prevent the failure from affecting dial tone. This is done during the design phase of the project.

Note The design phase is perhaps the single most important part of any deployment. Countless times I've had panicked customers, whom I inherited from other integrators, calling me with problems that could have been averted if dealt with during the design phase. Often when I ask clients how this problem was dealt with during the design phase, they answer, "What design phase?" Make certain that you cover as many foreseen and unforeseen eventualities as possible during the design phase. Although your customers might never see how good you are at fixing a system when it fails, they will know how good you are because it doesn't fail.

Redundancy is the core component in a reliable infrastructure. The system design should include redundancy at every level. This starts in the wiring closet.

Reducing the cable infrastructure and allowing ease of cable management are two of the motivating factors for migrating to a VoIP solution. Therefore, it does not make sense that redundancy is extended to the cable level. Remember, your goal is to achieve the same level of reliability that people expect from a phone system. People understand that if there is a cabling problem, the phone won't work. This is one of the few acceptable reasons for a phone system to fail. So, as far as the cabling goes, you just need to ensure that the existing network cabling infrastructure is certified as previously mentioned.

Switches are the next piece of the infrastructure that needs to be considered. Redundancy at the switch level is nothing new. Although redundancy has always been encouraged in data networks, it is no longer just a suggestion; it is required to achieve the expected level of reliability. In smaller environments that might have only a single switch, redundancy at the switch level doesn't apply; however, in large networks, make sure that you design a highly available network by building redundancy in at the core and distribution switch level. This means that there will be multiple paths a packet can take to get to its destination. Because of a protocol called Spanning Tree Protocol (STP), only one path is available at any given time. STP ensures that if a link fails, an alternative path will be opened. To find out more about STP, see the additional references listed in the appendix "Additional Reference Resources."

A redundant path can ensure that a packet reaches its destination, but it is also important that it gets there in a timely manner. Voice traffic does not handle delay very well. If too much delay is introduced, the quality of the conversation tends to degrade rapidly. You have probably noticed the effect delay can have on a conversation when watching a TV news reporter through a satellite link. It seems to take the reporter a few seconds to respond to a news anchor's question. This is because there is a several-second delay between the time the question is asked and the time when it reaches the reporter's destination.

Many things can affect the delay that is introduced into a conversation. One of the most common is the competition between voice and other traffic for bandwidth. To help alleviate this, QoS must be implemented within the network. QoS gives certain traffic priority over other traffic. The proper configuration of QoS is essential for any network that has

both voice and data on the same wire. A detailed discussion on QoS is beyond the scope of this book. Refer to Appendix A, for suggested references on this subject.

Before leaving the wiring closet, one more thing requires attention: power. Remember that a power failure is not an acceptable reason to lose dial tone. A power outage is not necessarily an acceptable reason for data networks to fail. There was a time when people expected and accepted the loss of data during a power outage. They were never happy about it, but they weren't surprised either. Nowadays with the reasonable price of uninterruptible power supplies (UPS), data networks are no longer as susceptible to power outages as they were in the past. It is nearly unheard of not to have a UPS on file servers and, in many cases, throughout the network. Switches are no exception. As with any equipment, you need to do some research to determine the proper size of the UPS you need. To do this, determine the amount of power that the switch draws and then determine the amount of time you want the switch to run without power. You don't need to worry about redundant power at the phone if you use inline power. Keep in mind that the more phones that draw power from the switch, the larger the UPS you need.

As mentioned previously, gateways are also considered part of the infrastructure. Therefore, whenever possible, redundancy should be included at the gateway level. In some cases, such as an environment that has only a single trunk from the PSTN, redundancy is not feasible. If the environment has other Cisco routers, try to use the same model router for your PSTN gateway. This way, if the PSTN gateway does fail, you might swap equipment for a short-term solution or, at the very least, use the other router for testing purposes after hours. If you do have multiple trunks, it is a good idea to have at least two physical gateways connecting the network to the PSTN. A level of redundancy can be added by using multiple service providers. For example, if you have two trunks, use a different service provider for each. This way, if either of the service providers has a widespread outage, the other trunk will still be functional.

This section dealt with the reliability of the infrastructure. This is only a portion of the solution that must be considered when implementing a reliable system. A system is only as good as its weakest link, so you need to ensure that the entire system is designed with the same goal in mind—"Don't affect dial tone." In the next section, you look at the call-processing layer, more specifically, the Communications Manager.

Communications Manager Overview

In the previous section, the infrastructure was discussed, and you learned what was necessary to create a solid foundation on which to build the rest of the system. As when building a house, you can move to the heart of the project after the foundation is set.

The Communications Manager is considered the heart of the Cisco Unified Communications solution. It is responsible for device registrations and call control. Communications Manager is an application that runs on a media convergence server (MCS). Often the term *Communications Manager* is used to refer to the physical device that the application runs on, but the hardware should be referred to as the MCS. Communications Manager is the software running on the hardware.

Note Cisco has certified certain servers for use as MCSs. Currently only certain HP and IBM servers are certified. Servers can be purchased from Cisco or directly from IBM or HP. The different platforms offer various features, such as redundant hard drives and power supplies. Be sure to take this into consideration when choosing a server. Keep in mind that not all IBM and HP servers are approved, so be certain to check with Cisco to ensure that the server you choose is approved. Also be aware that the HP servers that are supported are older models, and Cisco does not plan on approving newer HP models. Many integrators choose to purchase the MCS from Cisco to have a single-vendor solution.

Every system should have at least two Communications Managers, and the two are referred to as a Communications Manager cluster. The exception to this rule is if you run Communications Manager Business Edition. Later in this chapter, you learn why a minimum of two Communications Managers is strongly recommended. Based on the previous section, you might guess for yourself. Does the word *redundancy* come to mind?

Defining Communications Manager Components

Communications Manager is responsible for all device registration and call control. Much of the configuration is performed through the Communications Manager administrative interface. This section introduces you to the various components of Communications Manager and the devices that it controls.

Most configuration and administration is performed through Communications Manager's web browser interface. Using this interface, you can configure phones, add users to Communications Manager's directory, define the dial plan, and perform various other tasks. The majority of the tasks that you learn how to perform later will be done using this interface. The interface is fairly simple to navigate, and after a short time, most people are quite comfortable using it. It is important to remember that it is a web-based interface and hence might sometimes not be as responsive as you would expect. The delays are more noticeable when you access a remote Communications Manager over a WAN link. Each evolution of this interface improves the end-user experience, and nowadays it is much more enjoyable to use than in years past.

All the information that you enter through the web interface must be stored. Communications Manager uses IBM Informix to store this information. All configuration information is stored in this database.

As mentioned earlier, each Communications Manager cluster should have at least two Communications Managers. The reason for this is redundancy. Remember, the system needs to deliver the same level of reliability that people are used to with a traditional phone system. Having multiple Communications Managers also provides a more scalable system, which will be explained shortly. For now, the focus is on the role that the various Communications Managers play in regard to the database. A Communications Manager is referred to as either a publisher or a subscriber. Each Communications Manager cluster has only one publisher. All other Communications Managers within that cluster are referred to as subscribers.

The job of the publisher is to maintain the most current copy of the database. Whenever anything is added to the database, the information is sent from the publisher to all the subscribers. The data is never written to the subscribers first and then transferred to the publisher.

So far, we have discussed only the roles that the Communications Managers play in the database. The other job of the Communications Manager is device control. All devices register to a Communications Manager. This Communications Manager is known as that device's primary Communications Manager. Each device also has a secondary Communications Manager that it can register to if the primary fails. Devices use a subscriber as their primary Communications Manager. This leaves the publisher alone so that it can take care of its main responsibility, which is to maintain the database. In some cases, a device can have a tertiary server to which it can fail over if both the primary and secondary are not available. In most cases, primary, secondary, and tertiary Communications Managers should be subscribers.

If the primary Communications Manager fails, the device registers to the secondary. The device registers with the secondary Communications Manager only if it is not on a call when the Communications Manager fails. If the device is active when the Communications Manager fails, it registers with the secondary Communications Manager when the call ends. In most cases, a call stays up even if the participating Communications Manager that is controlling devices in the call fails. The reason is that during a call, the communication is point to point, meaning that the Communications Manager is not involved with the actual voice stream. The device has no idea that the Communications Manager has failed because it does not communicate with the device again until either the call is over or a feature that requires Communications Manager is invoked, such as hold or transfer. If a device whose Communications Manager has failed tries to invoke such a feature, the phone display indicates a Communications Manager failure. The feature either fails or is unavailable (grayed out), depending on phone type. The call itself is not affected. A message also appears on the phone stating that the Communications Manager is down and the feature is not available.

Note In certain cases, the failure of a Communications Manager could cause the call to drop. One example is if the call were connected through a Media Gateway Control Protocol (MGCP) PRI.

In small environments where there are only two Communications Managers, it is acceptable to use the publisher as a secondary Communications Manager. If you have more than 1250 users, it is not recommended to use the publisher as a secondary Communications Manager. Figure 1-1 shows a typical Communications Manager environment that can support up to 5000 phones. This figure is an example of what is referred to as one-to-one redundancy. In this configuration, 2500 phones register to Communications Managers B and C. Communications Managers D and E are secondary servers for these phones. Communications Manager A is the publisher, and no phones register to it. This example is based on the assumption that all the servers on which the Communications Manager is

loaded are MCS-7835s or equivalent. Other server models support a different number of phones per server. For example, the MCS 7845 supports up to 7500 phones per server.

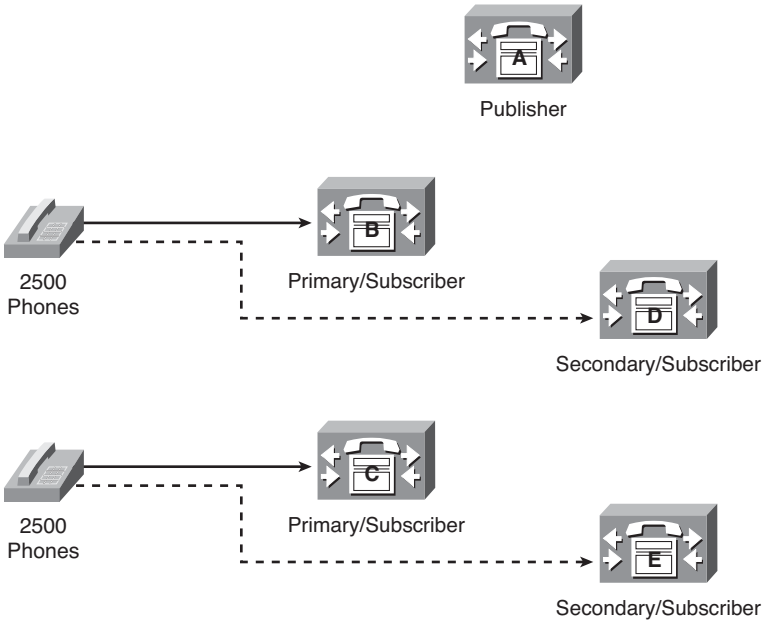


Figure 1-1 *One-to-One Redundancy Communications Manager Cluster*

Communications Manager Business Edition

Cisco offers a version of Communications Manager that is called Communications Manager Business Edition (CMBE). It is just like the standard Communications Manager with a few limitations. CMBE can only support up to 575 phones and does not support redundancy. In other words, there are no subscribers and only one publisher. It comes packaged with Unity Connection, which runs coresident (on the same server). This is an excellent "all in one" solution for small- to medium-size companies. All administration tasks and features configurations are performed the same as they are on the standard Communications Manager. They both use the same administration interface, and the software architecture is identical.

Communications Manager Devices

A large number of devices register with a Communications Manager, but they typically fall into one of the following categories:

- Phones
- Gateways

- Gatekeeper
- Media resources

Each of these devices has its own unique role within the Communications Manager environment, and this section briefly describes each one. For more information on these devices, refer to Appendix A.

Phones

A number of different model phones can be used in a Communications Manager environment. Table 1-1 lists and describes some of the more popular models.

Table 1-1 *Phone Models Used in a Communications Manager Environment*

Phone	Description
Model 7911	The 7911 is a single-line, entry-level phone. It offers a switch port in the back to which you can attach a PC. It only supports audio and text eXtensible Markup Language (XML) support.
Model 7942/7945	These phones support two lines and XML applications. These are considered midrange phones and are typically used in environments where two lines are adequate. They also have a switch port in the back for attaching a PC. The main difference between these phones is that the 7945 offers a color screen and the 7942 is only grayscale.
Model 7962/7965	These phones support four lines and XML applications. They also have a switch port in the back for attaching a PC. The main difference between these phones is that the 7965 offers a color screen while the 7962 is only grayscale.
Model 7975	This is an eight-line phone that offers a touch screen; this allows you to invoke certain features by touching the screen. It also supports XML applications.
Model 7925	The 7925 is a color-screen wireless phone that connects to the network through a wireless access point. The phone's shape and size are similar to a cell phone, but it only works in a Communications Manager environment.
IP Communicator	The Cisco IP Communicator is an application that runs on a PC and allows the PC to be used as a phone. Typically, a headset is attached to the PC, and the user can make and receive calls using the PC.
6900 Series	This is a series of cost-effective phones that work well for companies that are switching from a traditional analog phone system. While not as feature-rich as other phones, they offer a lower-cost entry yet still provide access to many Cisco Communication Manager's features.

Table 1-1 *Phone Models Used in a Communications Manager Environment*

Phone	Description
8900 Series	The 8900 series is a newer series of phone from Cisco that offers high-definition voice in addition to a high-resolution adjustable display. This series also includes USB ports, which support wired headsets.
9900 Series	This series is very similar to the 8900 but also supports a directly attached USB camera for video calls and Bluetooth. The 9971 also offers Wi-Fi.

Note One of the advanced features of many Cisco IP Phones is their capability to parse XML. These phones have an LCD screen on which the user can look up others in the directory, receive messages, log in and out of services, and perform many other functions. Through the use of XML programming, many companies have developed a variety of applications such as time clocks and inventory lookup.

Gateways Overview

As mentioned earlier, gateways are used to connect dissimilar systems together, such as connecting Communications Manager to the public switched telephone network (PSTN). The core requirements were discussed earlier, so this section examines the different types of gateways and how they communicate, meaning the protocol they use. There are three main protocols that are used today for communicating between Communications Manager and gateways. They are Media Gateway Control Protocol (MGCP), H.323, and Session Initiation Protocol (SIP). These are industry-standard protocols and offer similar features.

The type and number of trunks that a customer has also affects the type of gateway you select. Communications Manager is connected to the PSTN using either an analog or a digital trunk. The trunk used also affects the type of equipment you use for the gateway. Gateways differ in interface types and capacities. If analog trunks are used, typically a Foreign Exchange Office (FXO) port is used for each line. With analog lines, each call takes up a port on a gateway. This is not always a practical solution in a large environment. Typically, a T1 or E1 line is used to connect to the PSTN if a company needs more than a few lines. These types of trunks are normally more cost-effective if more than eight simultaneous connections to the PSTN are required.

So far, only using gateways as a way to connect to the PSTN has been discussed. Gateways are also needed to connect Communications Manager to traditional phones systems. In many cases, customers choose to integrate the Communications Manager into their existing voice solution and slowly replace the traditional PBX. This is done by connecting the Communications Manager to the traditional PBX through either analog or digital interfaces. The interface used depends on the volume of traffic expected to travel between the two phone systems and the interface available. For environments that expect large volumes of calls to travel between the phone systems, a T1 or E1 line is used. Figure 1-2 shows how this integration might look.

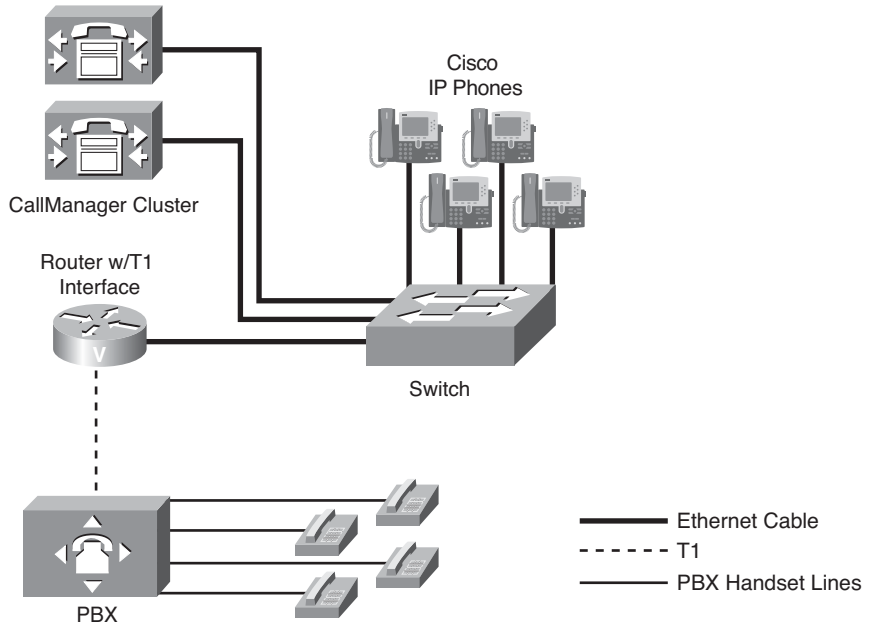


Figure 1-2 *Communications Manager–to–PBX Integration*

Note The physical connection used to connect a traditional PBX with Communications Manager through T1 interfaces is a crossover T1 cable from the T1 interface of the traditional PBX to the T1 interface on the Communications Manager gateway.

Gateways are used not only to connect Communications Manager to traditional PBXs but also to connect multiple Communications Manager environments together. As mentioned earlier, two or more Communications Managers are known as a Communications Manager cluster. All the IP devices within a cluster can communicate with each other without a gateway. However, when two Communications Manager clusters need to be connected, a gateway must be configured. The connection between the two Communications Manager clusters is called an ICT (Intercluster Trunk). In earlier versions of Communications Manager, these were configured under gateways. Now they are referred to as trunks in the configuration menu. Chapter 3, "Deploying Devices," discusses these gateways more fully.

Gateways are also used to provide analog connectivity within a Communications Manager environment. Although the goal of VoIP is to use IP to transport voice whenever possible, there are times when an analog connection is required; modems and fax machines are examples. To connect an analog device such as a fax machine, a Foreign Exchange Station (FXS) port is required. A gateway with FXS ports allows analog devices to operate within a VoIP network.

Gatekeepers

Now that you understand how to connect Communications Manager to the other systems, you need to make sure that the path used to connect to another system does not become congested. It is not possible to allow more calls than a connection can handle when connecting to the PSTN or a traditional PBX using analog lines or voice T1s. However, when connecting devices using an IP connection, oversubscribing is possible. Oversubscribing occurs when more calls connect across a link than the link can adequately handle. When connecting multiple Communications Manager clusters together, you can use gatekeepers to prevent oversubscribing. This is referred to as Call Admission Control (CAC).

A gatekeeper is an H.323 device and typically runs on a router such as a 2800. Hence, Communications Manager communicates with it using H.323. Figure 1-3 shows a typical deployment. This diagram shows two Communications Manager clusters connected through an ICT. The gatekeeper manages the available bandwidth between the sites. The total allowable bandwidth for voice calls is configured in the gatekeeper.

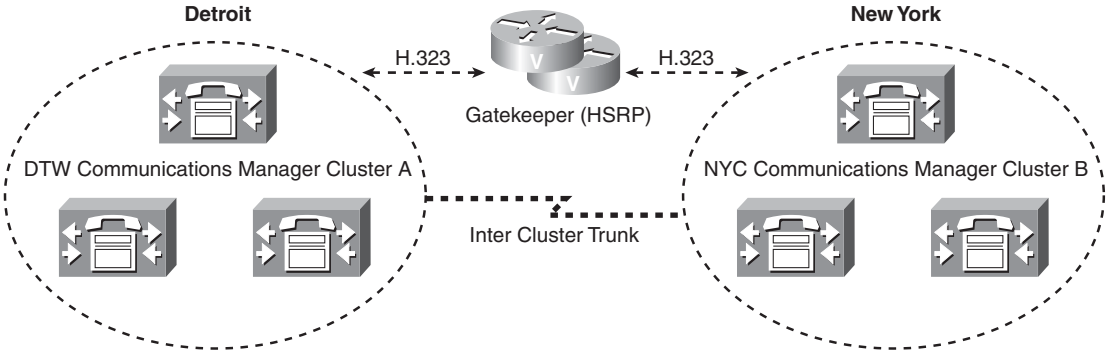


Figure 1-3 *Gatekeeper*

An example call flow would go something like this:

Step 1. Joe, who resides in Detroit, attempts to place a call to Fred, who resides in New York. The Communications Manager in Detroit sends a request to the gatekeeper to see whether there is enough bandwidth to place the call.

Step 2. The gatekeeper replies with either a confirmation (bandwidth is available) or a rejection (bandwidth is not available).

Step 3. If the bandwidth is available, the call setup proceeds and the Communications Manager in New York is informed that a call is being placed to Fred.

Step 4. The Communications Manager in New York then sends a request to the gatekeeper to see whether there is enough bandwidth for its side of the call.

Step 5. If the gatekeeper sends a confirmation, the call setup is complete, and Joe and Fred can talk about how the Lions actually won a game that weekend.

Much more occurs during setup. The previous example was presented to help you understand how a gatekeeper enforces CAC.

Assume for a moment that the gatekeeper in the previous example failed. What happens when the Detroit Communications Manager sends a request to the gatekeeper? Because the gatekeeper isn't working, the Communications Manager does not receive a confirmation and the call does not go through. How do you prevent a failed gatekeeper from negatively affecting call setup? The best solution is to have two gatekeepers. Once again, the theme running throughout this chapter is redundancy. Redundant gatekeepers can be configured by using Hot Standby Router Protocol (HSRP) or gatekeeper clustering. It is recommended that one of these solutions be implemented.

Not only can the gatekeeper be used for CAC purposes, but it can also determine the location of a requested device. This feature is discussed more fully in the CAC sections of Chapter 5, "Configuring Class of Service and Call Admission Control."

There are times when CAC is needed within a cluster. Because the gatekeeper is used when communicating outside a cluster, some other type of CAC must be implemented. For example, some type of CAC is needed within a cluster when a number of offices use the same Communications Manager at a central site.

CAC is accomplished in an IP WAN with centralized call processing deployment by configuring *locations* in Communications Manager. When configuring locations, the amount of bandwidth that is available for voice calls is entered in each location. When calls are placed between locations, bandwidth is deducted from the available bandwidth and calls are allowed or disallowed based on the available bandwidth. Locations are configured within Communications Manager, and no additional hardware is required. Often people ask whether they can use locations instead of a gatekeeper to save money. Remember that locations are designed to be used when the call is placed between two devices within the same cluster. It is best to use a gatekeeper for calls placed between separate networks.

Media Resources

To accomplish certain tasks such as conferencing and Music on Hold (MoH), Communications Manager needs to call upon additional resources. The core Communications Manager application does not have the capability to perform these tasks, so it relies on other resources, which are either hardware or software. Some resources reside on the same server as Communications Manager and others require additional hardware. The sections that follow describe the various resources.

Conference Bridge (CFB)

Conference bridges (CFB) are required for a caller to have a conference call with at least two other callers. CFBs can be either software or hardware; however, hardware is recommended. Software CFBs run as a process in Communications Manager, whereas hardware CFBs require additional equipment. Hardware CFBs require digital signal processors

(DSP). Not all devices that have DSPs can be configured as a CFB. The following devices can be configured as CFBs:

- Catalyst 6000 T1/E1 ports
- Catalyst 4000 Access Gateway Modules
- Supported Cisco routers with DSP farms

Note The Catalyst 600 and 4000 can no longer be purchased from Cisco.

Transcoders

A transcoder allows devices that are using different codecs to communicate. Transcoders can change an incoming codec to another codec that the destination device can understand.

Note A codec (which stands for compression/decompression) is used to express the format used to compress voice. An algorithm is used to compress voice so that it requires less bandwidth. The two most common codecs used in a Communications Manager environment are G.711 and G.729a.

MoH

Music on Hold (MoH) allows an audio source to be streamed to devices that are on hold. It is a process that runs on a Communications Manager. The audio can be streamed either multicast or unicast, and the codec can be configured. Up to 51 audio sources can be configured, including live audio that is plugged into a sound card (USB device) installed in the Communications Manager. It is also possible to configure the audio source to be streamed from a router.

Annunciator

The Annunciator is a software service that runs on the Communications Manager and provides audio announcements to callers. An example of this is if a caller dialed a number that Communications Manager could not process, the caller would hear a message something like, "The number you have dialed cannot be reached; please hang up and try your call again."

Now that you have an overview of the components required in a Communications Manager environment, you should review various deployment models. As this technology matures, the way it is deployed continues to evolve. It is essential that it be deployed only in a supported fashion. The next section discusses the various supported deployment models.

Understanding Communications Manager Deployment Models

There are essentially three main Communications Manager deployment models currently supported by Cisco. Although during the past few years these models have evolved to create what appear to be new deployment models, all support models fit into one of the following three categories of sites as described in the sections that follow:

- Single-site
- Multisite WAN with centralized call processing
- Multisite WAN with distributed call processing

Single-Site

In this model, a Communications Manager is deployed within a single building or perhaps in a campus area. However, some would argue a campus area would typically fall into the centralized model. In the past, the core theme behind this model was exclusive connectivity. There was no VoIP connectivity to any system outside its own. At present, service provides are offering SIP trunks that allow you to send calls outside of your system using VoIP. Any call placed to a destination outside its own is sent to the PSTN or an Internet Telephony Service Provider (ITSP).

Multisite WAN with Centralized Call Processing

A centralized deployment includes remote locations that have IP phones registering to the Communications Manager at the main site. Normally, there is only one Communications Manager cluster in a deployment such as this. Remote phones send all requests across the WAN to the Communications Manager. If the WAN fails and the phones have no local device to which to register, the phones are unusable.

Note Technology called Survivable Remote Site Telephony (SRST) has been developed that allows enables remote offices to have dial tone and make calls across the PSTN if the WAN or remote network fails. SRST runs on a router such as a 3800. The number of phones that can register to an SRST system depends on the hardware on which it is running. SRST is discussed in Chapter 6, "Configuring CUCM Features and Services."

Multisite WAN with Distributed Call Processing

There are multiple sites in this model, each having its own Communications Manager cluster. There is IP connectivity between them, and ICTs are configured to send the voice across. A gatekeeper is highly recommended in this deployment to prevent oversubscribing and assist in call routing.

Often companies that need distributed deployment also have remote sites with just a few phones. A Communications Manager cannot be justified for the remote site, so the phones register to a remote Communications Manager just as they would in a centralized

deployment. When centralized and distributed deployments are merged together like this, it is referred to as a *hybrid deployment*.

Another form of this deployment is referred to as a Clustering over the IP WAN. In this scenario, a single Communications Manager cluster is split among multiple sites. For example, a publisher and a subscriber are in the Detroit office and a subscriber is at the New York office. The phones within the respective offices register with the local Communications Manager.

One of the major requirements in a deployment such as this is that, as of Communications Manager 6.1, the round-trip delay can be no greater than 80 ms (no greater than 40 ms one delay). The standard one-way delay allowed enabled for voice is 150 ms, so the requirements in this type of deployment are much more stringent. Figure 1-4 shows an example of a Clustering over the IP WAN deployment.

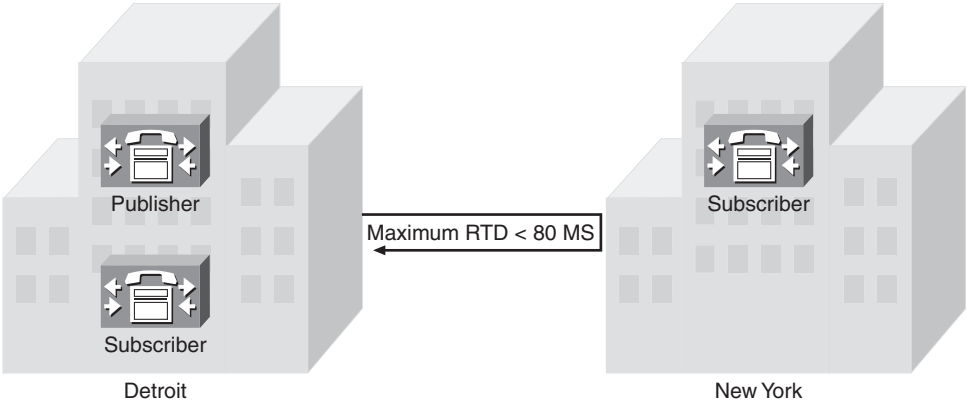


Figure 1-4 *Distributed Single Cluster*

The environment in which you deploy Communications Manager typically dictates which deployment model you choose. Regardless of which model is being deployed, it will most likely connect to an outside system such as the PSTN. When a call is placed outside the local system, the Communications Manager must know where to send the call based on the numbers dialed. The next section discusses how these call-routing decisions are made.

Route Plan Overview

Communications Manager knows about all devices that are registered within the cluster and knows how to route calls to any destination within the cluster. However, if a call is placed to a destination outside the cluster, Communications Manager needs to know where to send the call. This is the purpose of a route plan. A route plan is configured with Communications Manager to determine where to send calls based on the number that is dialed.

A route plan consists of a number of components, which are discussed in this section. Here Following is a brief description of each of the components that make up a dial plan. The list is in the order in which these devices must be configured. Later, the actual flow will be discussed.

Step 1. **Device (gateway):** Gateways were explained earlier as devices that connect dissimilar systems. The gateway is the last component in a route plan because it sends the call to an outside system. As mentioned before, the gateway connects the Communications Manager system to the PSTN and to other destinations.

Step 2. **Route group:** A route group is used to route the call to a gateway. Each route group includes at least one gateway, but because there is often more than one gateway, a call can be routed through a route group that can point to multiple gateways. The order in which the gateways appear in the route group determines which gateway the call is routed to first. If the intended gateway is not available, the route group sends the call to the next gateway in its list.

Step 3. **Route list:** A route list is used to route the call to a route group. Each route list must have at least one route group in it. Just as a route group can point to multiple gateways, a route list can point to multiple route groups. The order in which the route groups appear in the route list determines which route group the call is routed to first. If all the devices in the first route group are unavailable, the route list routes the call to the next route group in the list. If all gateways in the route groups within the list are unavailable, the call fails.

Step 4. **Route pattern:** When a user dials a number, Communications Manager compares the digits dialed against all the patterns it knows. As mentioned earlier, Communications Manager knows about all devices within the cluster. If the number dialed matches the number assigned to a device in the cluster, Communications Manager sends the call to that device. If the call is placed to a device outside the cluster, a matching route pattern must be configured in the Communications Manager. Because it is impossible to enter every possible number that might be dialed, wildcards are used to allow a single route to pattern-match multiple numbers. An example of one of the wildcards used in Communications Manager is an X. In a route pattern, an X matches any single digit 0–9. For example, a route pattern of 5XXX would match all numbers from 5000 to 5999.

Typical Call Flow

Now that you understand the basic components of a dial plan, examining a typical call flow is in order. Communications Manager in this example has the route pattern 9.1248547XXXX:

Step 1. A user dials 912485479000.

Step 2. Communications Manager analyzes the dialed digits and determines that the closest match it knows of is 91248547XXXX.

Step 3. Communications Manager routes the call to the route list that has been configured for the route pattern 91248547XXXX.

Step 4. The call is then routed to the first route group in the route list.

Step 5. If there is more than one gateway in the route group, the call is sent to the first gateway in the list. If the gateway is available, the call is sent out that gateway.

At times, a dialed number matches more than one pattern. When this occurs, Communications Manager selects the closest match. For example, if a user dials 5010 and Communications Manager has 5XXX and 50XX as route patterns, 50XX is selected because there are only 100 possible matches, whereas there are 1000 matches with the route pattern 5XXX.

Wildcards

Up to this point, the only route pattern wildcard that has been discussed is the X. Chapter 4, "Implementing a Route Plan," discusses wildcards further. For now, the following sections present wildcards that can be used in route patterns.

The Wildcard @

This wildcard matches any phone number that is part of a public number plan such as the North American Numbering Plan (NANP). The easiest way to understand what numbers are part of NANP is that any number you can dial from a home phone in North America would be part of the NANP. This includes numbers such as 911 and all international numbers. The NANP is installed by default, but other numbering plans can be downloaded/installed.

The Wildcard !

This wildcard represents any digit and any number of digits. At first, people think that "!" is the same thing as the X wildcard, but remember that the X represents only a single digit. The "!" can represent any number of digits.

The Wildcard [x-y]

This wildcard represents a single digit range. For example, 5[3-5] would match 53, 54, or 55. The numbers inside the brackets always represent a range and match only a single digit.

Note In a pattern such as 5[25-7], the only matches are 52, 55, 56, and 57. When people first look at this pattern, they think it matches 525, 526, and 527, but remember that the brackets can only represent a single digit.

The Wildcard [^x-y]

This wildcard represents an exclusion range, that is, any single digit that is not included in the range matches. For example, the pattern 5[^3-8] matches 50, 51, 52, and 59.

Calling Privileges

While creating a dial plan, you will find that some individuals within the company are allowed to place calls that others are not. For example, some companies do not allow all employees to make long-distance calls. Also, it is highly unlikely that you would want to allow international calls to be placed from a lobby phone. These issues are addressed by assigning calling privileges to devices. Calling privileges are configured by creating what is referred to as a class of control. The term class of service (CoS) was used in previous versions of Communications Manager, but it is now configured within the class of control area.

Note The terms *class of service (CoS)* and *class of control* are often used when talking about calling privileges. These two terms are synonymous with calling privileges. Those of you that come from a data background should not confuse this with the QoS component known as CoS. Although it stands for the same thing, class of service, it does not mean the same thing. Think of it this way: CoS in the data world is about prioritizing; CoS in the IP telephony world involves rights and restrictions.

Class of control is essentially a way to determine what destinations a given device can reach. For example, if you want to call someone in Germany, your CoS would have to allow international calls. In the same regard, if you want to prevent a device from calling Germany, you would assign a CoS that does not allow international dialing.

A class of control comprises two components, a Calling Search Space (CSS) and partitions. A solid understanding of these two components is essential to create an effective class of control. These two components seem to cause some confusion for those who are new to the concept, but at the core, they are simple concepts.

The simplest way to view this concept is to imagine a partition as a lock and a CSS as a key chain. If a number has a lock on it, you need the key on your key chain to reach that number. Simple, right? Well, as with anything, as things grow they can become more complex.

Partitions are assigned to anything that can be called, such as a directory number or a route pattern. Calling search spaces are assigned to anything that can place a call, such as phones and gateways. The configuration of these components is discussed in Chapter 5. Taking a closer look at how the components are configured can help clarify the concept.

Figure 1-5 shows how assigning partitions to a pattern and CSS to phones affect dialing privileges.

The two phones in Figure 1-5, Phone A and Phone B, each have a different CSS that determines where they can call. They both dial 912485479000. Based on the CSS of the phone, a match is determined. The only pattern that the dialed digits match is the 9.1248547XXXX pattern, which has the LocalLD partition. Because Phone A's CSS allows access to the LocalLD partition, its call goes through. However, Phone B's CSS does not allow access to the LocalLD partition, so that call fails.

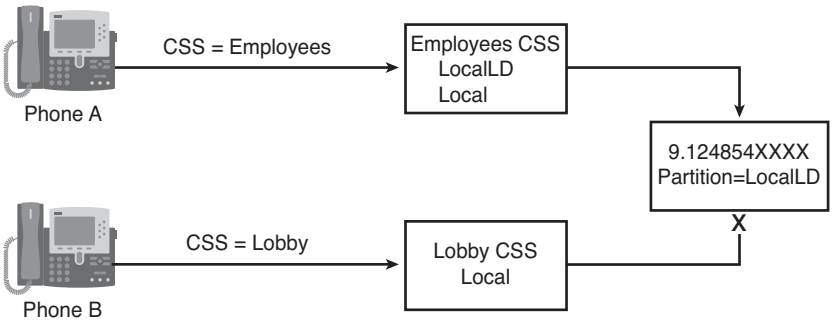


Figure 1-5 *CoS Example*

Note If you feel a little uncertain on this concept, fear not. I have had many students who worked with CSS and partitions for years and still didn't have a handle on it. The best advice is to start small, such as the example in Figure 1-5, and build on that. Although the design can become complex, the way that CSS and partitions work does not change and is simple.

As you can see, the components that are required to implement a Cisco Communications Manager solution are quite extensive. This section has presented only a high-level overview. By now, you should understand the importance of a solid infrastructure and be able to describe the role that redundancy plays in providing reliable service. Furthermore, you should be comfortable with the various devices that can register to Communications Manager and the various deployment models of Communications Manager. Finally, you should understand the basics behind dial plans and see how CoS can affect a device's capability to reach certain destinations.

Unified Messaging Overview

Typically, when you have a phone, you want to have a way for someone to leave a message when you cannot take the call. This means that most Communications Manager environments will have some type of voicemail system. Cisco offers three voicemail solutions: Unity, Unity Connection, and Unity Express. That is right; they all contain the word Unity in their name. This sometimes causes some confusion. This section looks at the two systems that are most often deployed with CUCM. Unity Express is a solution more often found in environments that have CUCM Express deployed and, for that reason, is outside of the scope of this course.

First, there is Unity. Many people view Unity as a voicemail solution, and although it does handle all the functions of a voicemail system, it is much more than that. Unity is a completely unified communication system. That is, it not only handles voicemail functions for a company, but it also can integrate into an existing email and faxing infrastructure. Because the voicemail, email, and faxing systems can be integrated with one another, they can use a single message store. This means that end users can now retrieve all

their voicemail, email, and faxes from one location. In the past, users would have to use the phone to get their voicemail, the PC to get their email, and a fax machine to retrieve their faxes. With unified messaging, all these messages can be retrieved from a single device, be it a PC or a phone. Using a PC, users can read their email and faxes or listen to their voicemail over the PC speakers. The users can also use a phone to retrieve their voicemail and email, because Unity's text-to-speech engine converts the email to a synthesized voice and streams it over the phone.

The other voicemail system is Unity Connection. Recently it has risen in popularity. It offers many of the same features as Unity and a few that Unity does not. The main difference, though, is not in the features it offers but rather in its infrastructure. Although Unity requires an external mail store (Exchange or Dominos), Unity Connection does not. It stores the messages on-box within its own store. However, even though Unity Connection does not require an email server such as Exchange, it can still integrate with an email server so that features such as listening to your email over the phone or checking your voicemail from your email client are still available.

Both products are powerful applications that require a variety of components to accomplish its tasks. The topics covered in this section provide you with a good understanding of the components within these systems and how these components fit into the solution. The areas covered in this section are as follows:

- Software architecture
- Call flow
- Call handlers
- Subscribers
- Networking

Software Architecture

Although Unity and Unity Connection offer similar services and features, their software architecture is not similar at all, as you see in the following sections.

Unity Software Architecture

Unlike some applications, Unity is not a standalone application. That is, in and of itself, it cannot provide voicemail services. Unity depends on a number of other applications to provide its array of services. If Unity were merely a voicemail system, it could have been designed not to require other applications. However, because it is a unified communications solution requires that it integrate with a company's email server. Because both a voicemail system and an email server have to send, receive, and store messages, Unity simply uses the existing email server to handle these functions. This is not to say that Unity does not handle the messages. This is discussed later. The various software components with which Unity interfaces include a mail store; a directory, such as Active Directory (AD); Structured Query Language (SQL); and Internet Information Server (IIS).

A closer look at the software architecture can help you understand the need for each component.

The discussion begins at the lowest level of the software architecture. The first layer of any software architecture is the operating system (OS). The current version of Unity requires Windows 2003 Server OS. Unity integrates tightly with Microsoft's AD when using Exchange 2003. Previous versions of Unity supported Windows 2000, so don't worry if the version you run isn't on Windows 2003.

Note Much like Communications Manager, Unity is supported only on Cisco-approved servers. Currently there are a number of models approved by Cisco. These servers can be purchased from Cisco or directly from HP or IBM. You can find the most current list of approved servers by searching for "Cisco Unity supported platforms list" at Cisco.com.

After the OS is on the server, other supporting applications have to be available before Unity can be configured. Unity depends on an outside message store to store messages. Currently you can use Microsoft Exchange or Lotus Notes as the message store. If Exchange is used, it can run on the same server as Unity, but this should be done only if you use Unity as a voicemail-only solution. If Unity is used as a unified communications solution, Exchange should be loaded on a separate server. If Notes is the message store, it must run on a separate server.

Note As of Unity 8.0, Domino is no longer supported as a mail store.

Note If Notes is the message store, an additional piece of software is required. It is called Domino Unified Communication Services (DUCS). This can be purchased from IBM and is used to allow the transfer of information between Unity and the Domino environment.

All the information entered into Unity must be stored. Unity uses SQL as its database, and most of the information you enter, such as usernames and phone passwords, are stored in SQL. Hence, SQL must run on the Unity server.

As mentioned earlier, Unity integrates with AD when using Exchange 2003, so the server must be part of an AD. If the Unity server is a standalone computer, that is, it is not part of an existing AD, it must be configured with AD and act as its own domain controller. Some of the information—for example, name, telephone number, and alias stored in SQL—is replicated to the AD.

Unity is similar to Communications Manager in that most of the configuration of Unity is done through a web browser interface. This requires that IIS run on the Unity server. During the installation process, IIS is installed. Additional tools perform many of the same tasks that are accomplished using the web browser interface. These tools are discussed in Chapter 11, "Exploring Unity/Connection Tools." The main advantage to using

the browser interface is that no additional software has to be loaded on a PC to administer Unity. This is a useful feature when you find yourself away from your desk needing to change something in Unity.

Unity Connection Architecture

In contrast to Unity, Unity Connection is a standalone application. That is, it can be a self-contained voicemail solution without any dependency on any separately installed third-party software. Installing Unity Connection is as simple as inserting the disk into a new server, powering it up, and answering a few questions. This install process installs the OS and all required applications. The install for Unity Connection is much easier and faster than that of Unity.

The OS that is used is often referred to as the VTG OS. VTG is an acronym for Voice Technology Group. This is the same Linux-based OS that is used for Communications Manager. Unity Connection stores all its information in an Informix database, just as Communications Manager does. Unity Connection and Communications Manager share a lot of the same software infrastructure. This is one reason why Unity Connection and Communications Manager can run on the same server if you deploy Communications Manager Business Edition.

Just as with Communications Manager and Unity, most of the configuration for Unity Connection is done through a web browser.

Following the Call Flow

Before discussing the actual components that determine the path a call takes, now take a high-level look at the various ways that a call can enter either of these systems and see how each type is dealt with. Because Unity and Unity Connection handle calls in much the same way, this section applies to both products. To keep things simple, both Unity and Unity Connection are referred to as "the system" for the rest of this section.

Typically, calls are forwarded to voicemail systems because the phone called is either busy or unanswered. When the system receives a call, it examines the reason that it is receiving the call. The phone system that forwarded the call includes a call-forwarded reason to the system. In this case, it was forwarded because the called party was busy or did not answer. If the call is forwarded because the caller did not answer, the system plays the called party's standard greeting. If the call were forwarded because the called party was on the phone, the system plays the called party's busy greeting.

In Unity, each user is known as a subscriber, but in Unity Connection, each user is known as a user. To keep consistent in this section, they are referred to as users. Each user can have five different greetings, which are as follows:

- **Standard:** Played during open hours when the user does not answer the phone. This is the default greeting and is always enabled.
- **Busy:** Played when the user is on another call.

- **Closed:** Played after hours.
- **Internal:** Played when another user reaches voicemail.
- **Alternate/holiday:** If enabled, always plays regardless of the forwarded reason or time of day. This greeting can be used as a vacation greeting. Note: Unity refers to this greeting as the Alternate; UC refers to it as the Holiday.

The system is often used as an auto-attendant, which allows all incoming calls to be answered by the system and then forwarded to the desired destination. In this instance, the call is not forwarded and is considered a direct call. The system handles direct calls differently than forwarded calls. The first thing the system does is attempt to determine whether the caller is a user. It does this by identifying whether the caller ID matches any phone number that is associated with a user. If it finds a match, it assumes that the user is calling and asks for the user's password.

Note All users have at least one phone number associated with them. This is their phone extension. Additional numbers can be assigned to subscribers. They are referred to as alternate extensions. A common use for alternate extensions is to associate a subscriber's cell phone or home number so that when they call in to check messages they are taken directly to the login prompt.

If the system determines that the caller ID is not associated with a user, the call is sent to the system's opening greeting. An opening greeting is the main greeting that outside callers hear when they reach the auto-attendant. It might say something like "Thank you for calling Bailey, Inc. If you know your party's extension, you can dial it at any time during this greeting." The options offered to the caller are determined by the system administrator based on the company's needs.

Exploring Call Handlers

Often companies that use the system as an auto-attendant create a menu that can lead outside callers to the proper department. You have probably experienced memorably frustrating menus. To create menus in the system, objects titled *call handlers* are created. One call handler was mentioned in the previous section. The opening greeting is a call handler. The easiest way to think of a call handler is as an object that can be used to help route calls. You can also think of call handlers as the building blocks that make up the menu system.

There are primarily three types of call handlers that can be used within the system:

- System call handlers
- Interview handlers
- Directory handlers

Table 1-2 *Call Handler Parameters*

Configuration Page	Parameter
Profile (Handler Basics in Unity Connection)	Name, Creation date, Owner (Unity only), Recorded voice, Schedule, Extension, Language, Switch (PBX)
Transfer	Status, Transfer incoming call, Transfer type, Rings to wait, If busy, Announce, Introduce, Confirm, Ask caller's name
Greeting	Greeting, Status, Source, Allow caller input, After greeting action, Reprompt, Number of reprompts
Caller Input	Allow extension dialing during greeting, Milliseconds to wait, Lock key, Action
Message	Message recipient, Max message length, After message action, Caller edit, Urgent marking
Call Handler Owners (Unity Connection only)	Owner of the call handler

The system call handler is used to build the menu system and typically contains a recorded prompt and asks the caller for input, such as "Press 1 for sales, press 2 for technical support, or hold on the line for further assistance." Within Unity, these types of call handlers have no special classification and can be referred to as general call handlers. They are simply known as call handlers. Within Unity Connection, they are called system call handlers. Table 1-2 lists five configurable pages for a call handler and some of the parameters found in each. These parameters vary slightly between Unity and Unity Connection, but this table gives you a good idea of the number of parameters that need to be configured for each call handler. More detail is given to these parameters in the "Configuring Call Handlers" section found in Chapter 9, "Call Management."

The next type of call handler is called an interview handler. It is used to extract information from the caller that requires asking more than one question. An interview handler can contain up to 20 questions. It allows the user time to answer between the questions. Unlike the call handler described previously, there are only two sets of parameters that need to be configured for the interview handler. Table 1-3 shows these parameters in the "Creating Advanced Call Routing Systems" section found in Chapter 9.

Note An example used for an interview handler would be a class survey hotline. After students take a class, they can call a number and answer five questions. The answers are later transcribed and entered into the school's database.

Table 1-3 *Interview Handler Parameters*

Parameter	Description
Profile (Handler Basics in Unity Connection)	Name, Creation date, Owner (Unity only), Recorded voice, Extension, Language, Deliver response to, Response urgency, After interview action
Question	Questions number, Question text, Maximum length in seconds (for response), Recorded question

The third type of call handler is known as a directory handler. The directory handler allows callers to dial by name. As the caller spells the person's name using the dial pad, the system searches to find all matching names. The system then offers the caller name choices and allows the caller to choose the party to which they would like to be forwarded. The system allows multiple directory handlers so that each department could have a separate one if wanted. Directory handlers have four sets of configurable parameters, as shown in Table 1-4. More detail is given to these parameters in the "Configuring Directory Handlers" section found in Chapter 9.

Figure 1-6 shows a typical system menu system flow chart. The various types of call handlers that have been discussed are shown in this figure. Each of the boxes represents a call handler.

Table 1-4 *Directory Handler Parameters*

Parameter	Description
Profile (Handler Basics in Unity Connection)	Name, Creation date, Owner, Recorded voice, Extension, Language, Play all names, Search scope (Unity Connection only), Search result behavior (Unity Connection only)
Search Options (Unity only)	Search in, Search by
Match List Options (Unity only)	On unique, Announce matched using, Announce extension
Caller Input	No input timeout, Last input timeout, Repeat prompt, Send on to exit
Greeting (Unity Connection only)	Greeting

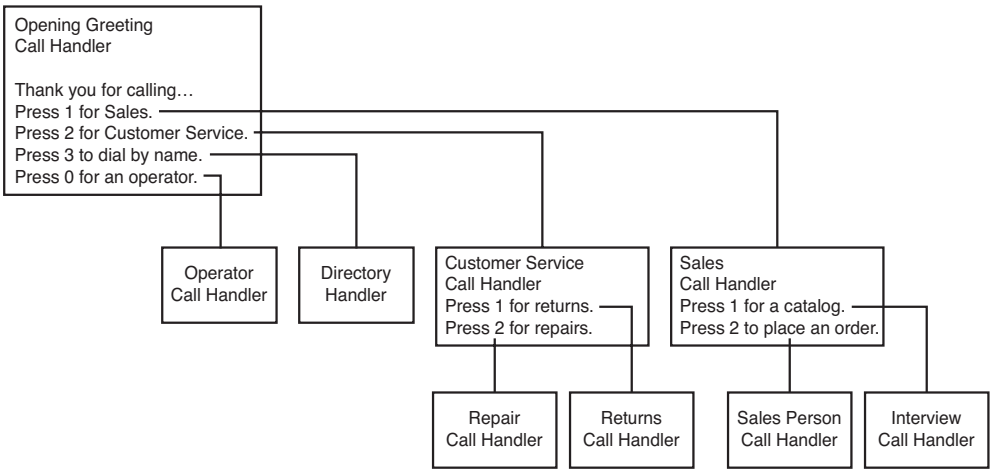


Figure 1-6 *Example Menu System Flow Chart*

Tip It is a good idea to draw your menu system in a flow chart form before creating it on the system. This helps ensure that all the call handlers are linked and that the flow makes sense.

Typically call handlers are used to route a call through the system. However, they can be configured to take messages. If a call handler is configured to take a message, the message must be sent to a user. Call handlers are simply database objects and do not have mailboxes.

As you can see, call handlers are a fundamental part of the system. Even the most basic system contains a number of call handlers. So a solid understanding of how they are configured and work is essential. This section has only touched the surface of the power and flexibility of call handlers. Once again, you are encouraged to review the listed references in Appendix A for further study of this technology.

Defining Various Types of Users

Now that you have a good understanding of how calls are routed through a system, we take a look at the users. As mentioned earlier, in Unity, users are referred to as subscribers, while in Unity Connection, they are simply referred to as users. The following sections look at each type separately to help you better understand the differences.

Unity Connection Users

Within Unity Connection there are three types of users: those with voice mailboxes, those without voice mailboxes, and contacts. Users with voice mailboxes are the type of users that you will be creating the majority of the time. Users without voice mailboxes

are users for people that need administrative access to the system but do not need voicemail on the system. The main purpose of contacts is to add someone to the directory who does not have an account on the system. It is common to list users who have a voicemail account on a different system. This allows voicemails to be forwarded to them.

Unity Subscribers

When it comes to Unity, there are a number of different types of subscribers. This is because Unity relies on another application such as Exchange or Dominos to store the messages. The following sections explore the various types of subscribers that can be created within Unity.

A subscriber is anyone who has an account on the Unity system. Typically, Unity subscribers' messages are stored on the message store that is associated with Unity. However, subscribers are sometimes created where the mailbox actually resides on a different voicemail system. The following sections explore all the types of users who can be configured in Unity and indicate the appropriate uses for each. Six different types of subscribers can be configured as follows: Exchange, Domino, Internet, AMIS-a, VPIM, and bridge.

Exchange Subscriber

An Exchange subscriber is the most common type of subscriber. This subscriber, as the name implies, has its messages stored on an Exchange server. Exchange subscribers have access to voicemail using the phone, also called the telephone user interface (TUI). If licensed, they might also have access to voicemail using a PC. This is referred to as the graphical user interface (GUI).

Note Exchange subscribers can be divided into two subtypes of subscribers. One type is referred to as voicemail-only. These subscribers use the Exchange server for voicemail messaging purposes only. Furthermore, they cannot retrieve their email through Unity or their voicemail through a PC. The other type is known as unified messaging. These subscribers have both their email and voicemail stored on the Exchange server and can access both from either a phone or a PC.

Domino Subscriber

A Domino subscriber is similar to an Exchange subscriber except that it uses Domino as the message store, not Exchange. Unlike Exchange subscribers, a Domino subscriber must be a unified messaging user. Domino is not supported as a voicemail-only solution.

Note As of Unity 8.0 Domino is no longer supported.

Internet Subscriber

An Internet subscriber does not have a mailbox on the message store associated with Unity. The mailbox is an Internet email account. This type of account allows outside workers to be listed in the directory and receive voicemail without any type of access to Unity. This type of subscriber is generally used to provide voicemail for contract employees who work from their homes. When messages are left for Internet subscribers, they are delivered to the Internet email account as WAV file attachments, and the subscribers retrieve them when they download their mail.

AMIS-a (Audio Messaging Interchange Specification–Analog) Subscriber

This type of subscriber is used when Unity is connected to a non-Unity voicemail system using the AMIS protocol. The subscriber's mailbox is located on the non-Unity voicemail system. This type of subscriber has no direct access to Unity.

Voice Profile for Internet Mail (VPIM) Subscriber

This type of subscriber is used when Unity is connected to a non-Unity voicemail system using the VPIM protocol. The subscriber's mailbox is located on the non-Unity voicemail system. This type of subscriber has no direct access to Unity.

Bridge Subscriber

This type of subscriber is used when Unity is connected to an Octel voicemail system using bridge networking. The subscriber's mailbox is located on the Octel voicemail system. This type of subscriber has no direct access to Unity.

User Parameters

When creating and configuring a user, you notice that a number of the parameters are similar to those found in a call handler. This is because each user has an associated call handler. When you create a user, a call handler is also created. So, when you are configuring certain user parameters, you are actually configuring the call handler associated with that user. To help illustrate this, Table 1-5 shows all the Unity parameter sets. Those marked with an asterisk (*) are the parameters used to create the call handler. While the names of the parameters in Unity Connection vary slightly, the concept is the same.

Note Not all Unity subscribers have all the parameters listed in Table 1-5. The parameters available depend on the type of subscriber that you are creating. In some cases, there are some parameters that are not listed in this table. Only the most common parameters are listed.

A user is more than just a call handler, as you can see by the additional parameters that need to be configured. A user can be seen as a collection of objects, one being the call handler. It is the call handler's job to offer the caller any available choices and play the greeting. Another piece of a user is the mailbox where the voice messages are stored. For

a Unity subscriber, the mailbox typically resides on an Exchange server. There are also additional user settings. These settings are not part of the mailbox or the call handler. These attributes deal with things such as phone passwords, message notification, and CoS. These settings, much like the call handlers, are stored in the database.

Table 1-5 *Subscriber Parameters*

Parameter	Description
Profile*	Name, Display name, CoS, Extension, FAX ID, Recorded voice, Schedule, Time zone, Self-enrollment setting, Directory, Language, Switch (PBX)
Account	Account status, GUI access status, Creation date, Last phone contact, Billing ID, Call handlers owned, Windows NT account status
Phone Password	User cannot change password, Must change on next login, Never expires, Password, Date of last change
Private Lists	Private list number, Name of list, Recorded name, Current members
Conversation	Menu style, Volume, Language, Time format, Send to, Identify by, Play recorded name, Message counts, Message type menus, Sort by, Message number, Time Message was sent
Call Transfer*	Transfer to, Transfer type, Rings to wait, If busy, Announce, Introduce, Confirm, Ask caller's name
Greetings*	Greeting, Status, Source, Allow caller input, After greeting action, Reprompt, Number of reprompts
Caller Input*	Allow extension dialing during greeting, Milliseconds to wait, Lock key, Action
Messages*	Max message length, After message action, Caller edit, Urgent marking, Language, MWI
Message Notification	Device, Phone number, Extra digits, Dialing options, Status, Schedule
Alternate Extensions	Alternate extensions
Alternate Names	Alternate name
Features	Broadcast messages, Message locator, Message security

*Parameters used to create the call handler

Note The term *class of service (CoS)* is sometimes confusing because it means different things depending on the technology it is referring to. When the term is used with Unity/Connection, it is referring to rights and access, such as what features a user has access to.

As you can see, many parameters need to be configured for each user. If you had to configure each parameter for every user, it would take far too long. To simplify this task, objects known as *templates* are used when creating users. A template defines how the parameters should be set by default. Typically there are large groups of users that need similar settings. A template is created for each group, and after creation, a user can be edited as needed.

Note It is important to understand that templates are significant only at the point of user creation. Changing a template after a user is created has no effect on that user.

As previously stated, not all subscribers have all the parameters listed in Table 1-5. Parameters depend on the type of subscriber that is being created.

Networking Overview

Both Unity and Unity Connection have the capability to network with each other and other voicemail systems. The following sections explore the types of networking that are supported by each system.

Unity Networking

Unity has the capability to network with other types of voicemail systems. This feature allows employees of large companies that have multiple voicemail systems to efficiently exchange voicemails regardless of the system on which they are homed. Cisco understands that companies have a substantial investment in current solutions and might not be ready to do a total cut-over. Often Unity is deployed for a portion of the users as a pilot. It is easier to convince the client to start with a pilot because Unity can talk with other systems and allow the end user to use the most common features.

Following are four types of Unity networking:

- Digital
- VPIM
- AMIS
- Bridge

Typically, people think of networking as plugging two PCs into an Ethernet jack that allows enables them to communicate. Unity networking is not just plugging an Ethernet

cable into the Unity server. Unity networking refers to connecting Unity to another voicemail system. This can be done in a variety of ways, which are described in the following sections. The overall goal is to have Unity interface with another voicemail system as seamlessly as possible.

Digital Networking

Digital networking allows multiple Unity systems within the same directory, such as AD, to seamlessly interact with one another. When using digital networking, the Unity systems have the ability to share the directories of each other. This allows outside callers to search for any subscriber, regardless of which Unity server they are associated with. This type of networking offers many other features, which is discussed in Chapter 10, "Implementing Unity Networking."

VPIM Networking

VPIM networking allows Unity to interface with non-Unity voicemail servers through a TCP/IP link. This link can be an Internet connection or a private network. The non-Unity voicemail system must support VPIM, which is often an add-on that requires an additional investment.

AMIS Networking

AMIS networking allows Unity to interface with a non-Unity voicemail system across analog lines. This is typically done across the PSTN, but it might also be used for in-house migration. AMIS is not as efficient as VPIM, because it uses analog lines, so transmissions can be more time-consuming. For example, a 5-minute message sent to five people through AMIS takes 25 minutes, plus setup and teardown time. With VPIM, the transmission time is much shorter. (The actual transmission time depends on the speed of the connection.) The non-Unity voicemail system must support AMIS, which is often an add-on that requires an additional investment.

Bridge Networking

Bridge networking is used to connect a Unity system with an Octel voicemail system. Bridge networking is unique in that it requires an additional server. This is called the Unity bridge server. The bridge server communicates with the Unity server through a TCP/IP connection and with the Octel server across analog lines. The bridge server acts as a translator between the Octel analog protocol and Unity's digital networking protocol. The overall message delivery can experience the same type of delay that is common with AMIS because they both use analog connectivity. Octel networking adds features that are not found in VPIM or AMIS and creates a more feature-rich environment. Bridge networking is recommended if a company is going to gradually migrate from Octel to Unity.

So which one should you use? The voicemail solution that you choose to integrate with will often dictate the answer. If you have a choice between AMIS and VPIM, most often VPIM is recommended. As mentioned previously, when integrating with Octel, bridge networking should be used. When connecting multiple Unity systems, digital networking is the solution if the Unity systems share the same Active Directory (AD). If they do not share the same AD, the solution is VPIM networking.

Unity Connection Networking

Unity Connection offers two types of networking. The first, digital networking, is used when networking UC systems within a company. The other, VPIM networking, is used to network UC to non-UC voicemail systems and UC systems not within the same company.

Unity Connection digital networking is similar to Unity's digital networking; the big difference is that UC does not use AD. This means that each UC needs to synchronize its directory with the other UC systems on the network. The net effect is the same as it is with Unity, but a little more configuration might be required.

Unity Connection also supports VPIM networking. This is used when networking with a Unity server or other voicemail system. The remote voicemail system must support VPIM. Keep in mind that the remote system will most likely require a license for this feature. VPIM is also used when networking UC systems when digital networking is not an option.

Securing the Environment

With the proliferation of viruses and malicious attacks on computer systems today, it is not just wise, but mandatory, to protect your systems. Many network administrators remember Nimda and Blaster all too well. Even some admins that thought they were fairly well protected got hit. Sometimes it seems that just when one type of attack can be defended against, another begins. To make matters worse, in an IP telephony environment, you need to protect the system not only from computer-related attacks but also from the types of attacks common to voice systems, such as toll fraud. You might as well face it now: There is no silver bullet that completely and forever protects you. But there are steps you can take to make sure that you are not an easy mark. The following sections examine some of the current security concerns that you should be aware of. When possible, solutions are offered. Remember, this is an ongoing battle. The goal of these sections is to make you aware of a few of the security issues and encourage you to be vigilant in protecting your system.

Securing the Operating System

Because Unity uses Windows as its operating system, it is vulnerable to the security issues that exist for Windows. Because Microsoft is undeniably the most commonly installed OS, it is a large target. For various reasons, some people hold grudges against Microsoft and find it fun to attack systems running on a Microsoft OS. Whether they are valid grudges, it is important to protect your systems. What is important is that attacks do occur, and you need to protect your system.

Typically an attack is virus driven. The damage viruses cause can range from something benign to data corruption and destruction. All viruses should be considered dangerous.

For this reason, virus protection software should be installed on all systems. Cisco has qualified the following list of antivirus software for use with Unity 8.0:

- CA Anti-Virus for the Enterprise version 8.0 and later (formerly called eTrust Antivirus)
- Computer Associates InoculateIT for Microsoft Windows
- McAfee
- ePolicy Orchestrator
- GroupShield Domino (Unity 7.x and earlier only)
- NetShield for Microsoft Windows
- Symantec
- AntiVirus Corporate Edition
- Norton AntiVirus for IBM Lotus Notes/Domino (Unity 7.x and earlier only)
- Norton AntiVirus for Microsoft Exchange
- Norton AntiVirus for Microsoft Windows
- Trend Micro
- ScanMail for Lotus Notes (Unity 7.x and earlier only)
- ScanMail for Microsoft Exchange
- ServerProtect for Microsoft Windows

Because Unity uses SQL, keep in mind that in the past, SQL has been the target of attacks. So, attention should be given to this. The best way to secure any SQL installation is to ensure that the latest approved security patches are applied.

Note *Remember to check that a patch has been tested and approved by Cisco before you apply it.* In some cases, there are special instructions on how to load certain patches. It could even be required that you load only patches available directly from Cisco. Don't assume that all Microsoft patches can be loaded on Unity.

Communications Manager Security Issues

Protecting the system from outside threats is only half the battle. There are also internal threats that exist. Ever since a telephone system has existed, people have tried to exploit it. These exploitations range from illegal wiretapping to toll fraud.

Earlier in this chapter, it was mentioned that a PC can be connected to the back of the phone, allowing both devices to share a single Ethernet port. Although this is an efficient use of ports, if not properly configured, both devices could be using the same VLAN. When both devices are on the same VLAN, it is possible for a PC to capture voice

traffic. A tool called Voice Over Misconfigured Internet Telephones (VOMIT) can then reassemble the captured data into a WAV file. The conversation could be played back on a PC. By ensuring that the voice traffic is on a separate VLAN, you help prevent this from occurring, but that alone is not enough. You also need to prevent data and voice networks from communicating with each other as much as possible. It is understood that there are times when the networks might need to send packets to each other, but this traffic should be managed by a firewall to ensure that only the desired traffic is allowed through. Earlier, it was recommended that voice and data traffic be separated on different VLANs so that the voice could be prioritized. Now you see that it can protect not only the quality of the voice but also protect the voice from prying ears.

PC-based phones, such as Cisco IP Communicator, introduce the same problem that having voice and data on the same VLAN produces. Because a PC is being used as phone, the voice traffic is sent on the data VLAN. This allows prying ears on the data network to capture the conversation. For this reason, a properly designed network has voice and data on separate VLANs.

Another often-overlooked area that can be exploited is allowing rogue phones to autoregister. This occurs when autoregistration is used during deployment and not turned off or restricted later. Autoregistration is a useful tool during some deployments. It allows a phone to be plugged into the system and register without having to configure it in Communications Manager first. It is most often used in greenfield deployments. However if autoregistration is not disabled after the initial deployment, rogue phones could register to Communications Manager. This seems like a minor issue until you factor in that if the dial plan has not been secured, a user on a rogue phone could place a call anywhere in the world without detection until the bill arrives.

The administration of Communications Manager should also be considered when securing the voice system. If people get access to Communications Manager administration, they can do anything from changing users' class of control to shutting down gateways, or even the entire system. Security is often a double-edged sword: Too little and anyone can do anything, and too much and no one can do anything. Often, you find that you want to allow limited access to the administration interface. An example might be to allow someone to add users to the Communications Manager directory. You do not want this person to change the dial plan. A Communications Manager add-on, called Multi-Level Access (MLA), enables you to do this by granting limited administration access to individuals. An even higher level of security can be implemented by restricting which physical system can access the Communications Manager through the use of an access list.

The preceding few paragraphs discussed only a few security issues that you need to address. The task of completely securing a Communications Manager environment can seem daunting at times, but all possible efforts must be taken to ensure that the system is protected from both external and internal threats.

Unity Security Issues

Unity offers a new way of looking at message management. It is now possible to store all voicemail, email, and faxes in a single location. In addition, it is possible to retrieve them from nearly anywhere in the world. Of course to do this, there must be a way to access the system. Although this comes as no surprise, remember, at every entry point, there are people trying to exploit that point of entry. Because of the many features that Unity offers, there are many points of entry. By default, Unity enables access through the phone, web browser, and email client. All of these entry points add to the task of securing the system. Securing Unity is not a trivial task, and sufficient attention must be given to this task. The following are just a few examples of areas that must be secured.

Unity uses a third-party message store, and currently this is either Microsoft Exchange or Lotus Notes. Both of these systems are subject to the same types of attacks as any email systems. There are attacks aimed specifically at these systems. Securing these systems requires that you stay current on all security-related patches.

Note *Remember to check that a patch has been tested and approved by Cisco before you apply it.* In some cases, there are special instructions on how to load certain patches. It could even be required that you load only patches available directly from Cisco. When a patch is related to a high-level security issue, Cisco is quick to test and report these patches on Cisco.com. You can find available patches in the software download section of Cisco.com. You can also set up to have alerts emailed to you automatically. Search "Product Alert Tool" at Cisco.com to set this up.

Just as with Communications Manager, administration access has to be secured. By default, the administrator selected when configuring Unity has full administrative access. Often you need to allow others access for limited administrative tasks. You can do this by creating a CoS that enables them only the access that you want them to have.

Unity has the capability to track all administrative access and create reports showing this access. This can prove useful when troubleshooting to determine whether changes were made and who made them. For this reason, it is necessary for every person accessing the administration interface to use individual login credentials. Don't just create one administrative login and allow everyone to use the same one because there is no way to determine who made what changes.

When you create subscribers, you determine which features they can access. A few of these features can affect the security of your system. One of these features is the user's capability to define one-key transfer options that can be performed during his greeting. This enables the outside caller to be transferred to another extension or number by pressing a single key. However, this also opens up the possibility of toll fraud. A subscriber could set up a one-key transfer that transfers the call to a long-distance number, perhaps a friend who lives out of state. This option, of course, would not be announced during the greeting, but the subscriber could dial into his voicemail from home at night and be transferred to his friend's number all on the company's dime. The configuration of such

features should be restricted to administrative personnel only. Even then, you might want to take further action to ensure that individuals with the ability to configure this feature do not exploit the privilege. To prevent possible exploitation, you can restrict the numbers to which Unity can transfer these users. Chapter 7, "Unity Predeployment Tasks," discusses how to perform these tasks.

Each subscriber in Unity has two passwords. The first is the phone password. This is the password that is used when accessing Unity through the phone. This password is only numeric because it is entered using the digits on a phone. Unity should be configured to require minimum-length passwords for each subscriber. The longer the minimum length, the harder it is for someone to figure it out. It is important that you don't take Unity phone passwords lightly. Often people don't think that their voicemail password is as important as their email password, but with unified message, if you can access one, you can access the other. The second password subscribers use to access Unity is their AD or NT account password. This password is used when Unity is accessed using a PC, such as when checking messages over the Internet. This password should contain both letters and numbers, and a minimum length of five digits should be used.

Note All accounts have default passwords. Make sure that these are changed the first time an account is accessed. There are a few default accounts that are created when Unity is installed. Make certain that the default password on these accounts is changed immediately following the installation. On more than one occasion, I have logged in to systems at clients' sites and retrieved messages, unknowingly sent to these accounts, simply by using the default extension and password. This, of course, was done with the full knowledge of the customer and only to show him that his system was not secured by the installer.

This section has touched on only a few security concerns to bring attention to the fact that these are valid concerns that must be addressed. Additional reference material is listed in Appendix A.

Summary

A Cisco IP telephony deployment comprises a number of technologies and platforms. Before beginning to deploy such a solution, much thought must be given to the proper design. It all begins with a reliable, solid infrastructure. A proper infrastructure begins with a redundant physical topology and proper backup power. To allow access to outside systems, gateways are used and methods of Call Admission Control must be implemented.

At the call-processing layer, redundant Communications Managers should be deployed. These systems must run on Cisco-approved hardware called MCSs. Devices such as phones, gateways, and conference bridges register to Communications Manager. When a call is placed or a resource is requested, the request is sent to the Communications Manager to which the device is registered. The Communications Manager uses the configured dial plan to determine how to route the call to the desired location. After the call is complete, Communications Manager ensures that it is properly torn down.

Unity is a unified communications server that enables individuals to have all their voicemails, emails, and faxes stored in a single location. This enables more efficient access to all messages. Unity has the capability to function as an auto-attendant and have outside callers routed through the system when they make selections from the available menu. In the future, unified communications will enable calls to be routed based on predefined rules that individual users can configure.

Unity Connection is a voicemail system that offers many of the same features as Unity but does not use a third-party message store. This prevents it from offering true unified messaging, but it can still offer comparable feature and, in some cases, features that are not offered by Unity.

The advances being made in IP telephony are exciting. As with most new technologies, there are those who find ways to exploit them. It is critical that sufficient attention be given to securing all IP telephony deployments.

Chapter 2

Preparing CUCM for Deployment

To ensure a smooth deployment, certain tasks must be performed in a certain order. In this chapter, you learn what tasks need to be completed before adding devices. As with most things, if you fail to create a solid foundation, you will encounter problems. The topics covered in this chapter give you that firm foundation.

Before adding any devices to the system, ensure that a number of predeployment tasks are accomplished. Because this book assumes that Communications Manager is already installed, most settings discussed in this chapter should already be properly configured.

The goal of this chapter is to help you understand these settings and see how changing them can affect your system. This chapter covers services configuration, enterprise parameters, and device registration tasks. In addition, the chapter includes step-by-step instructions for many of these tasks. It is worth the time to review these settings and make sure that they are configured properly for your system. Although the system might seem to function fine even if some of these tasks are overlooked, it is recommended that all tasks are verified before adding devices.

Configuring Communications Manager for Maximum Performance

Communications Manager is the heart of the call-processing system, and it is important to ensure that it runs at peak performance. A number of processes can run on the Communications Manager, but not all of them are always necessary. The following sections explore the various processes that might run on a Communications Manager and discuss which can be safely disabled to preserve more processing power for other Communications Manager functions.

The following sections look at the services that can be enabled to run on the Communications Manager. There are two types of services: feature and network. For the most part, the average administrator will not need to deal with the network services, so these sections focus on the feature services.

Activating Communications Manager Services

Since the release of Communications Manager 3.3, all Communications Manager services are deactivated by default. Before Communications Manager 3.3, the user determined which services would run by choosing to install them during the installation process. In the current versions of Communications Manager, all services are loaded but none are activated. This section discusses each of these services and explores the proper way to activate them.

Before you can have devices register, you must activate services using the Service Activation screen within the Communications Manager Serviceability interface. You can access this interface by entering **https://[***CM_IP Address***]/ccmservice**. If you are already logged in to the Communications Manger administration interface, you can access the Serviceability interface by selecting Cisco Unified Serviceability from the Navigation drop-down menu in the upper-right corner. After you are in the Serviceability interface, navigate to **Tools > Service Activation.**

Figure 2-1 shows the Service Activation screen as it appears the first time it is accessed.

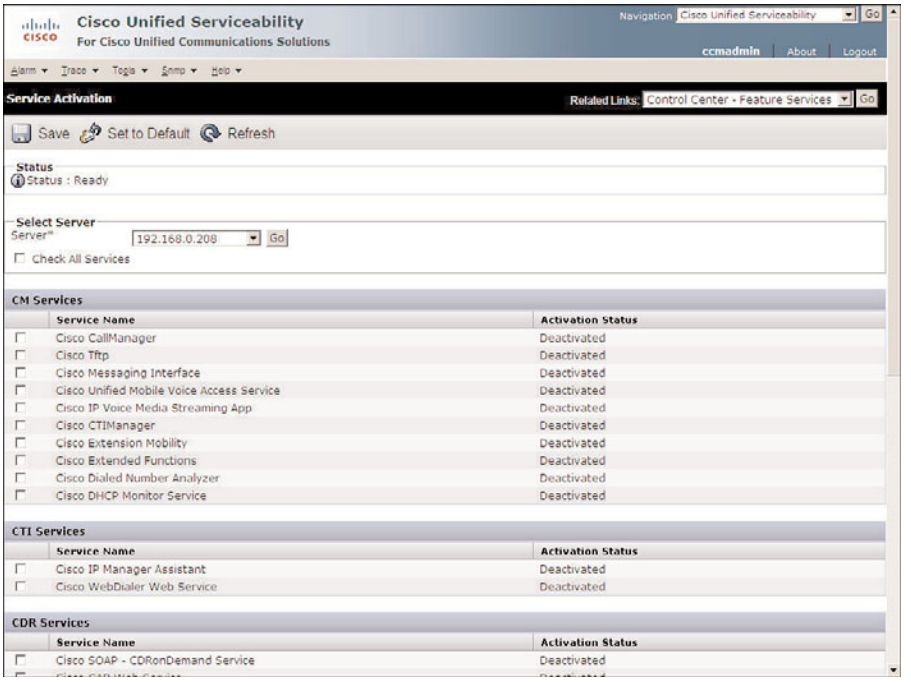


Figure 2-1 *Service Activation Screen*

To activate the Services screen, follow these steps:

Step 1. Enter **https://[***CM_IP Address***]/ccmservice** in the address bar of your browser and press **Enter.**

Step 2. Enter the administrative username and password and click **Login**.

Step 3. Navigate to **Tools > Service Activation**.

Step 4. Select the desired server from the Server drop-down list.

Step 5. Select the check box next to each service that you want to activate and click **Save**.

Note The Set to Default button is next to the Refresh button. If the Set to Default button is clicked, all the services that are required for the functioning of Communications Manager will be activated. In most cases, you do not activate all the same services on all the servers. Only use this button in a single-server environment. You should not select all services on all the Communications Manager servers because doing so would activate services that might not be needed, thereby consuming CPU cycles that could be used for other functions.

Table 2-1 lists all Communications Manager services, with a brief explanation of their functions and activation recommendations.

After these services are activated, they can be deactivated if desired. To deactivate any of these services, follow these steps.

Step 1. Enter **https://[***CM_IP Address***]/ccmservice** in the address bar of your browser and press **Enter**.

Step 2. Enter the administrative username and password and click **Login**.

Step 3. Navigate to **Tools > Services Activation**.

Step 4. From the Server drop-down list, select the server on which you want to deactivate service and click **Go**.

Step 5. Deselect the check box next to each service that you want to deactivate and click **Save**.

After all the services are configured properly on all the servers in the cluster, you can move on to defining Communications Manager Enterprise settings. The next sections explore the various parameters that should be defined before devices are deployed.

Configuring Communications Manager's Enterprise Settings

After services are active, Communications Manager is functioning. However, before devices are added, a few more tasks must be accomplished. The following sections examine the importance of removing Domain Name System (DNS) reliance and the system's enterprise parameters.

Table 2-1 *Communications Manager Services*

Service	Function	Recommendations and Dependencies
Cisco Communications Manager	Provides call-processing, signaling, and call control functions.	In the control center, network services ensure that the Cisco Real-Time Information Serve (RIS) Data Collector service and Database Layer Monitor service are running on the node.
Cisco Trivial File Transfer Protocol (TFTP)	Provides TFTP services for device configuration files. It is also responsible for building the configuration files.	This service should be activated on at least one Communications Manager. This server is responsible for servicing TFTP requests.
Cisco Messaging Interface	Provides Simplified Message Desk Interface (SMDI) connectivity for traditional voicemail systems.	This does not need to be activated if Unity is the voicemail solution.
Cisco Unified Mobile Voice Access Service	Allows Cisco Unified Mobility users to perform the following tasks: • Make calls from a cellular phone as if the call originated from the desk phone. Depending on the device that is used, a user might have to call a local PSTN number owned by the company, enter a PIN, and then establish an outbound toll call through the gateway. Some devices such as the iPhone do not require this. • Turn Cisco Unified Mobility on. • Turn Cisco Unified Mobility off.	For mobile voice access to work, you must activate this service on the first node in the cluster and configure the H.323 gateway to point to the first Voice Extensible Markup Language (VXML) page. In addition, make sure that the Cisco CallManager and the Cisco TFTP services run on one server in the cluster, not necessarily the same server where the Cisco Unified Mobile Voice Access Service runs.

Table 2-1 *Communications Manager Services*

Service	Function	Recommendations and Dependencies
Cisco IP Voice Media Streaming App	Provides service such as Music on Hold (MoH), conferencing, and Media Termination Points (MTP).	You do not need to activate this service on all Communications Managers. Activate it only on Communications Managers that you want to provide these services. This should not be activated in the Publisher.
Cisco Extension Mobility	Used to define time limits for Extension Mobility. Activates the XML service for login function.	This service should be activated on all Communication Mangers in environments for which Extension Mobility is planned.
Cisco Extended Functions	Provides support for Cisco Unified Communications Manager voice-quality features, including Quality Report Tool (QRT). Also enables the callback feature.	The Cisco RIS Data Collector must be running on servers that are running this service. The CTI manager service must be loaded on at least one server.
Cisco Dialed Number Analyzer	Supports Cisco Unified Communications Manager Dialed Number Analyzer.	If you are planning to use Cisco Unified Communications Manager Dialed Number Analyzer, activate this service. This service can consume many resources, so only activate this service on the node with the least amount of call-processing activity or during off-peak hours. It is further recommended that this service only be activated while troubleshooting.
Cisco DHCP Monitor Service	Provides Dynamic Host Configuration Protocol (DHCP) service for phones and monitors IP address changes for IP phones in the database tables.	Activate this service on the Communication Managers that have DHCP server enabled.

Table 2-1 *Communications Manager Services*

Service	Function	Recommendations and Dependencies
Cisco IP Manager Assistant (IPMA)	This is required to enable the IPMA feature. This feature allows managers additional features such as divert and DND.	This service should be active on servers that service these features.
Cisco WebDialer Web Service	Allows users to dial from a web page or desktop application.	Typically, this service is loaded on one server in the cluster.
Cisco SOAP - CDRonDemand Service	Receives Simple Object Access Protocol (SOAP) requests for Call Detail Records (CDR) filename and returns a list of filenames that fit the time duration that is specified in the request.	You can activate the Cisco SOAP - CDRonDemand Service only on the first server.
Cisco CAR Web Service	Loads the user interface for CDR Analysis and Reporting (CAR), a web-based reporting application that generates either CSV or PDF reports by using CDR data.	You can activate the Cisco CAR Web Service only on the first server.
Cisco AXL Web Service	This service allows you to modify database entries and execute stored procedures from client-based applications that use Administrative XML (AXL).	Activate on the first node only. Failing to activate this service causes the inability to update Cisco Unified Communications Manager from client-based applications that use AXL.
Cisco UXL Web Service	This service performs authentication and user authorization checks. The TabSync client in Cisco IP Phone Address Book Synchronizer uses the Cisco UXL Web Service for queries to the Cisco Unified Communications Manager database.	Activate if the Cisco IP Phone Address Book Synchronizer is being used.