Oracle® Web Conferencing

Administrator's Guide

Release 2 (2.0.4)

Part No. B10877-01

June 2003

Use this manual to deploy, configure, and administer Oracle Web Conferencing.

ORACLE®

Oracle Web Conferencing Administrator's Guide, Release 2 (2.0.4)

Part No. B10877-01

Primary Authors: Jeffrey Doering, Dinesh Ajmera, PV Shivkumar, Cynthia Stevens

Contributing Authors: Anindo Roy, Mehul Sanghavi, Kiran Choudary

Contributors: Ramu Sunkara, Vimal Chopra, Michael Tselner, Ravindra Dani

Contents

2 Understanding Oracle Web Conferencing

5 Configuration

6 Sample Deployments

7 Monitoring

8 Reports

9 Web Conferencing Sites

10 imtctl Command Line Utility

Send Us Your Comments

Oracle Web Conferencing Administrator's Guide, Release 2 (2.0.4)

Part No. B10877-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: rtc@oracle.com
- FAX: 650-654-6202 Attn: Oracle Web Conferencing
- Postal service:
  Oracle Corporation
  Oracle Web Conferencing Documentation
  200 Oracle Parkway, Mailstop 2op734
  Redwood Shores, CA 94002
  U.S.A.

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Intended Audience

This manual is intended for administrators who need to deploy and monitor the Web Conferencing option of Oracle Real-Time Collaboration. The Administrator's Guide is not intended for end users.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Structure

This document consists of the following chapters:

Chapter 1 Introduction to Oracle Web Conferencing

This chapter provides an overview of the Oracle Web Conferencing product.

Chapter 2 Understanding Oracle Web Conferencing

This chapter explains Oracle Web Conferencing concepts, including terms and definitions, architecture, processes, conference runtime flow, ports, and clusters.

Chapter 3 Planning for Deployment

This chapter discusses factors to consider when deploying Oracle Web Conferencing. The issues range from sizing guidelines to firewall issues to topology considerations for an enterprise that is geographically distributed.

Chapter 4 Post-Installation

This chapter discusses mandatory post-installation steps required to make Oracle Web Conferencing completely functional in the selected deployment configuration.

Chapter 5 Configuration

This chapter explains how to configure: application pages, usage reports, ports and network connectivity, logging, e-mail, Oracle Real-Time Collaboration clusters, conference runtime characteristics, Document and Voice Conversion Servers, monitoring, user privileges.

Chapter 6 Sample Deployments

This chapter discusses some sample deployment scenarios, ranging from a basic deployment to a sophisticated one spanning multiple geographical regions. It also explains how to configure these deployments.

Chapter 7 Monitoring

This chapter describes how Oracle Web Conferencing components are monitored to provide quality of service for conferences and continuous availability for conference service.

Chapter 8 Reports

This chapter explains the usage, feedback, and uptime reports.

Chapter 9 Web Conferencing Sites

This chapter explains Oracle Web Conferencing sites.

Chapter 10 imtctl Command Line Utility

This chapter explains the imtctl utility, which provides a command-line interface for administering and configuring the Oracle Real-Time Collaboration system.

Chapter 11 Troubleshooting

This chapter provides a URL where you can get the most up-to-date troubleshooting information.

Glossary

This manual contains a glossary. Terms that are explained in the glossary appear in bold in the text.

Appendix A Time Zones

This appendix contains a list of supported time zones.

Related Documents

- *Oracle Collaboration Suite Installation and Configuration Guide for Solaris*
- *Oracle Collaboration Suite Release Notes*
- *Oracle Web Conferencing Sizing Guide*

Conventions

The following conventions are used in this manual:

Convention	Meaning
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted.
boldface text	Boldface type in text indicates a term defined in the text, the glossary, or in both locations.
< >	Angle brackets enclose user-supplied names.
[ ]	Brackets enclose optional clauses from which you can choose one or none.

1

Introduction to Oracle Web Conferencing

The Oracle Real-Time Collaboration system is a state-of-the-art, distributed system that offers real time collaboration services, including Web Conferencing. While this guide focuses on the administration and management of the Web Conferencing service, most of the components and administration tools that are involved in the providing this service are generic and will be used for other services in the future.

This chapter provides an overview of the Oracle Web Conferencing product, including its end-user collaboration features and administration features.

1.1 What is Oracle Web Conferencing?

Oracle Web Conferencing brings real-time online collaboration to any enterprise, enabling customers, employees, teams, and partners to meet online within the context provided by the content, commerce, and comprehensive business flows of e-business.

Oracle Web Conferencing consists of client and server applications that let you create and participate in online conferences. Oracle Web Conferencing features multiple ways to collaborate in conferences, including desktop sharing, whiteboarding, chat, polling, and cobrowsing.

1.2 Oracle Web Conferencing Basics

1.2.1 Features

Oracle Web Conferencing contains the following real-time collaboration features:

Cobrowsing

- Synchronous browsing and window-scrolling of HTML pages with a small or a large number of users.
- Navigation to Web pages by typing a Web address during a conference or by selecting from user-predefined bookmarks.
- Collaborative form-filling, including support for real-time individual character display.

Document Presentation

- Selection and display of documents from a user's personal materials repository.
- Presentation of the following formats is supported: Microsoft Word, Excel, PowerPoint, HTML, text and image (.gif and .jpg).

Whiteboarding

- Conference presenters and attendees have access to a rich set of mark-up and drawing tools for the visual review and annotation of shared content, or as an aid in making presentations.

Desktop Sharing

- Ability for presenters to share anything from their Microsoft Windows-based computer with other attendees in real time. Options include sharing any region or multiple regions, any application, or the entire desktop.

Voice Streaming

- Live, listen-only streaming of voice from any telephone or teleconference system through the Oracle Web Conferencing Console, the user-facing component where real-time conferences take place.

Polling

- Ability to create instant polls or select user-predefined polls. Responses are displayed in real time.
- Poll results can be published to all attendees.

Chat

- Live chat with all attendees, a group of attendees, or with just the conference host.
- Ability for registered users to predefine messages for use during live chat.

Shared Control

- Conference host can share control of the conference with attendees.
- Conference host always has ultimate control.

Recording and Playback—Recording of voice synchronized with the on-screen data collaboration for on-demand playback of the conference.

In addition, Oracle Web Conferencing features the following:

- conference scheduling
- e-mail invitations
- conference archives
- personal materials repository for conference materials storage

1.2.2 End-User Activities

Schedule a conference—Plan a conference in advance, send e-mail invitations, and designate materials to be reviewed before the conference.

Create an instant conference—Create a conference and host it right away.

Join a conference—Easily join a conference through an e-mail invitation (if the host has enabled this option), through the lists of conferences, or through the Join Meeting window.

Manage materials—Store conference materials such as documents, bookmarks, messages, and polls in your own repository. You can access your personal materials repository during conferences.

Participate in a conference—With the control delegation feature, the host can allow attendees to present content. Collaboration modes give you options for presenting Web pages, documents, images, and to draw on the whiteboard and share your desktop. With chat, participants can communicate with each other in real time.

Record a conference—Record a conference so that it can be played back anytime.

Publish archives—Publish the recorded conference and information about the conference, such as the public chat transcript, list of conference attendees, and conference duration.

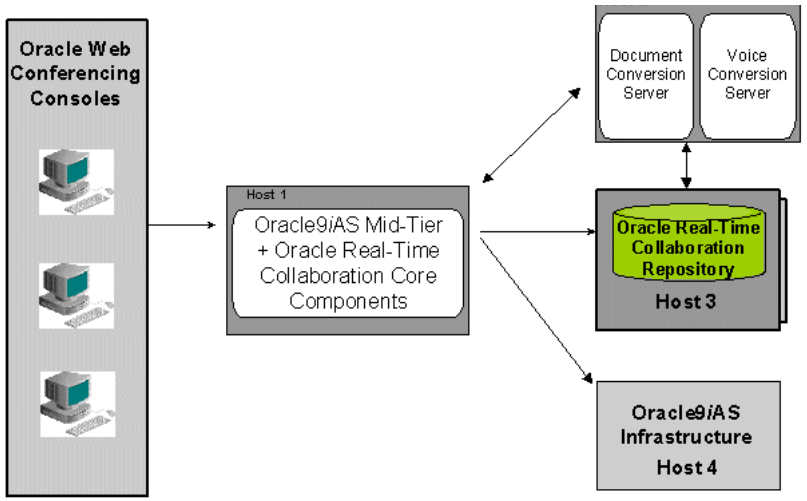
Play back a conference—Play back archived conferences any time.

1.3 Installing Oracle Web Conferencing

Oracle Web Conferencing is an option of Oracle Real-Time Collaboration. A fully functional Web Conferencing system is made up of different components, which are explained in the next section. For details on installing Oracle Real-Time Collaboration, including hardware and software requirements and installation prerequisites, please refer to the *Oracle Collaboration Suite Installation and Configuration Guide for Solaris*.

1.3.1 Basic Oracle Web Conferencing Topology

Figure 1–1 Basic Deployment



1.3.1.1 Web Conferencing Components

A basic Web Conferencing system consists of the following components:

Oracle Web Conferencing Console

When a user joins a conference, the Web Conferencing Console is downloaded if it is not already present on the user's computer or if the installed version is not current. Conferences take place in the Web Conferencing Console. Users access the Web Conferencing Console through Internet Explorer.

Oracle Real-Time Collaboration Core Components

A grouping of Real-Time Collaboration components. Among the components are the Web Conferencing Server (imt-collab), the OC4J application (OC4J_ imeeting), and the Oracle Real-Time Collaboration process monitor (imt-pm) and the Real-Time Collaboration mx. They work together to provide the core real-time collaboration functionality of Oracle Web Conferencing.

The Web Conferencing Console is downloaded from the Oracle Real-Time Collaboration Core Components instance.

As a prerequisite, the Oracle Real-Time Collaboration Core Components need an Oracle9*i*AS mid-tier that is configured to work with Oracle9*i*AS Single Sign-On and Oracle Internet Directory.

Document Conversion Server

A grouping of Oracle Web Conferencing components that converts MS Office documents into HTML for viewing in Document Presentation mode. This grouping includes the Document Conversion Server component. It must be installed on a computer with Microsoft Windows and Microsoft Office.

> **See Also:** *Oracle Collaboration Suite Installation and Configuration Guide for Solaris* for details on prerequisites

Voice Conversion Server

A grouping of Oracle Web Conferencing components that dials into a voice conferencing system, converts the analog voice to digital format, and streams it. This grouping includes the Voice Conversion Server component. It must be installed on a computer with Microsoft Windows and requires specialized telephony hardware and software.

> **See Also:** *Oracle Collaboration Suite Installation and Configuration Guide for Solaris* for details on prerequisites

Real-Time Collaboration Repository

The set of Oracle Real-Time Collaboration database schemas residing in an Oracle9*i* Database .

Table 1–1 Default Database Schemas

Schema	Description
rtc	This account contains all the database tables, views, indexes, triggers, etc. This account is not used by Oracle Real-Time Collaboration components.
rtc_app	This account contains the appropriate synonyms pointing to base Oracle Real-Time Collaboration tables and views that belong in the rtc account. This account has appropriate privileges to modify and select data from these tables. This account is used for connection by Oracle Web Conferencing components.

The schemas contain:

- system map—configuration information of all the instances and components in the system.
- user roles—the roles that are assigned to the users. The user authentication information itself is stored in Oracle Internet Directory.
- conference information—scheduled conferences, conference archives, user documents.
- data for internal use by Oracle Real-Time Collaboration, like application menu definitions.

1.3.1.2 Oracle Web Conferencing Prerequisites

The following prerequisites are necessary for any deployment of Oracle Web Conferencing.

Oracle9*i*AS

Oracle9*i* Application Server is an integrated J2EE application server that provides the Oracle HTTP Server, Oracle9*i*AS Containers for J2EE, and other Oracle Web Conferencing prerequisites.

Oracle9*i*AS Infrastructure

This includes Oracle Internet Directory and Oracle9*i*AS Single Sign-On. Among its services, Oracle9*i*AS Infrastructure provides user provisioning and authentication services.

Oracle9*i* Database

The Oracle9*i* Database, release 2, is a prerequisite for Oracle Web Conferencing. The Oracle Real-Time Collaboration Repository for Oracle Web Conferencing resides in this database.

1.4 Web Conferencing Management Features

Oracle Web Conferencing management consists of system management, business management, and user management.

1.4.1 System Management

1.4.1.1 Features

Support for management of different topologies—Lets you manage Web Conferencing in a geographically-distributed enterprise and provides support for various deployments, taking into account all enterprise considerations like firewall issues and load balancers (LBR).

High availability—Provides process monitoring and automatic restart of processes when component failure is detected.

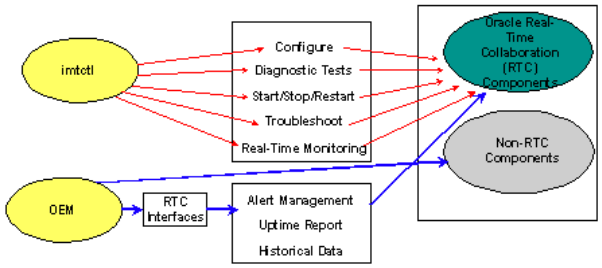
Oracle Enterprise Manager integration—Integrated with Oracle Enterprise Manager.

1.4.1.2 Administration Tools

The Oracle Web Conferencing (Oracle Real-Time Collaboration) system is comprised of both Real-Time Collaboration components and other external components on which the system depends.

The following illustrates Real-Time Collaboration system management:

Figure 1–2 Real-Time Collaboration System Management



1.4.1.2.1 imtctl The imtctl utility provides a command-line interface for administering and configuring the Real-Time Collaboration system. The utility supports a variety of commands:

- Starting and stopping Real-Time Collaboration processes (components)
- Viewing the current state of the system, which includes all the components in the system, the status of the components, and the Real-Time Collaboration processes (components).
- Running diagnostic tests for service availability and configuration check tests
- Real-time conference monitoring
- Configuring Real-Time Collaboration properties
- Assigning administrative roles to users

1.4.1.2.2 Oracle Enterprise Manager Oracle Enterprise Manager is used to manage the external components on which Oracle Real-Time Collaboration depends, like the Oracle9*i*AS mid-tier, Oracle9*i*AS Infrastructure, Oracle9*i* Database. Real-Time Collaboration interfaces are used for monitoring Web Conferencing and generating alerts.

The Oracle Enterprise Manager standalone console provides basic instance management operations, like starting and shutting down an instance and viewing the state of the instance.

Refer to the Oracle Enterprise Manager *Administrator's Guide* for more information.

1.4.1.2.3 Real-Time Collaboration Interfaces Oracle Real-Time Collaboration publishes interfaces that you can plug into any external management framework. With the

interfaces, you can run service availability tests and get real-time monitoring data that can be collected periodically and used for historical analysis to tune the system.

1.4.2 Business Management

Oracle Real-Time Collaboration business management provides a rich set of features for an administrator to manage non-system-related functionality of the Web Conferencing deployment. It provides functionality that lines of business managers and support staff can use to monitor usage and manage business-related activities.

1.4.2.1 Features

Conference Monitoring—With this feature, you can monitor conferences that are currently running on the system. This is useful both in support situations and business administration.

Reporting—Oracle Real-Time Collaboration provides comprehensive usage and feedback reports.

Site Management—With this feature, you can customize a single Oracle Web Conferencing deployment in an enterprise for different lines of business or site usage. Each line of business can have its own look-and-feel and customized integration with the Real-Time Collaboration system.

1.4.2.2 Management Tools

Users with the appropriate Oracle Real-Time Collaboration role can access the administration tabs via the Web-based user interface. Using the tabs, users can access such functionality as monitoring conferences and viewing reports. With the imtctl command-line interface, you can perform various business management tasks. Using the ReportManager infrastructure, you can generate reports that can be automatically e-mailed to recipients.

1.4.3 User Management

The Real-Time Collaboration system uses Oracle Internet Directory for user management. The Oracle Internet Directory host used by Real-Time Collaboration is specified at installation. All users of this Oracle Internet Directory are automatically provisioned to use Real-Time Collaboration with the enduser role.

By default, users are managed using the oiddas interface of Oracle Internet Directory. This is typically available at http://<ldaphostname>:7777/oiddas. A user with

the Oracle Internet Directory administrator account, typically orcladmin, can create, update, and delete users.

> **See Also:** *Oracle Collaboration Suite Installation and Configuration Guide for Solaris* for more information

1.5 Administration Road Map

To successfully deploy Oracle Web Conferencing and manage the system, Oracle Corporation recommends following the steps.

Step 1 Understand Oracle Web Conferencing.

Understand the basic concepts of the Oracle Web Conferencing product, including its architecture, components, end-user connections, load balancing, and network and port considerations for deployment.

Step 2 Plan for deployment.

Follow the sizing guidelines to estimate the hardware requirements for your deployment. Take into account the existing deployment of other Web-based applications, including firewall considerations and load balancers.

Step 3 Install Oracle Web Conferencing.

Install the Oracle Real-Time Collaboration system and follow the post-installation steps. When you are finished, run the post-installation verification tests. Out of the box, some tests might fail. The most common reason is incomplete configuration. Fix the configuration for each of the failures.

Step 4 Monitor Oracle Web Conferencing.

Monitor the health of the Real-Time Collaboration system by running service availability tests periodically. Have the results of the tests tied to an alert management system.

Step 5 Troubleshoot Oracle Web Conferencing.

Troubleshoot the system as required to address user complaints and failures detected by diagnostic tests run manually or through alerts.

Step 6 Tune the Web Conferencing system.

Tune the Web Conferencing (the Real-Time Collaboration) system, as required. You might need to change the existing configuration or add more machines. Follow the guidelines described in the tuning section of this manual.

2

Understanding Oracle Web Conferencing

2.1 Overview

This chapter explains Oracle Web Conferencing concepts and architecture. This chapter describes:

- Terms and definitions
- Real-Time Collaboration processes of Oracle Web Conferencing, including the function of each and their interactions.
- The detailed runtime flow of various kinds of clients connected to the Web Conferencing Server for a conference.
- Ports required for Web Conferencing and all network connectivity issues
- How to create clusters for a distributed deployment

2.2 Terminology

2.2.1 Host

A physical machine.

2.2.2 Real-Time Collaboration Instance

A Real-Time Collaboration instance is a grouping of Real-Time Collaboration components installed within the same Oracle home. These components work together to deliver a well-defined set of services to end-users, other Real-Time Collaboration instances, or external applications.

Three basic installation types—the Core Components, Document Conversion Server, and the Voice Conversion Server—are created when they are installed by the Oracle Collaboration Suite installer. If all are installed on the same machine, there is one instance only. The Real-Time Collaboration Repository is either present in the information store database when Oracle Collaboration Suite information store is installed, or it is created during installation of first Core Components installation.

> **See Also:** *Oracle Collaboration Suite Installation and Configuration Guide for Solaris* for details.

2.2.3 Real-Time Collaboration Component

An Oracle Real-Time Collaboration component is a set of processes within an instance that have identical functionality. Each component has a component type and component name.

The following table contains Real-Time Collaboration component types and names:

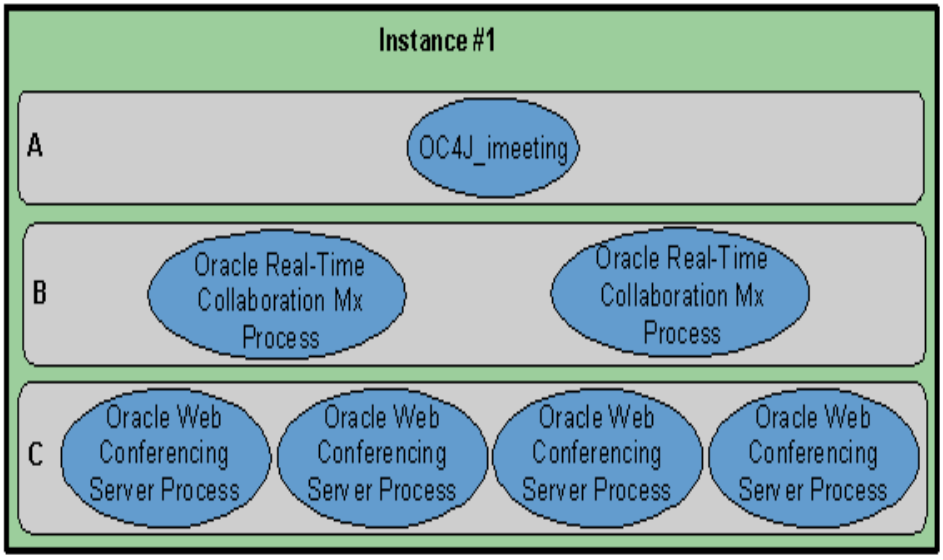
Table 2–1 Real-Time Collaboration Components

Component Type	Component Name	Number of Processes/Component	Full Component Name
clbsvr	imt-collab	1-N	Oracle Web Conferencing Server
oc4j	oc4j_imeeting	1-N	Web Conferencing OC4J
mxcomm	rtc-mx	1-N	Multiplexor
voiceconv	imt-voiceconv	1	Voice Conversion Server
docconv	imt-docconv	1	Document Conversion Server
imt-pm	imt-pm	1	Real-Time Collaboration Process Monitor

Use the `listComponents` imtctl command to see a list of components in an instance. See Chapter 10, "imtctl Command Line Utility" for more information.

The following illustration shows an instance with three components and each of their multiple processes.

Figure 2–1 Component and Processes of a Real-Time Collaboration Instance



A—Component OC4J_imeeting (one process)

B—Component Real-Time Collaboration mx (two processes)

C—Component Web Conferencing (four processes)

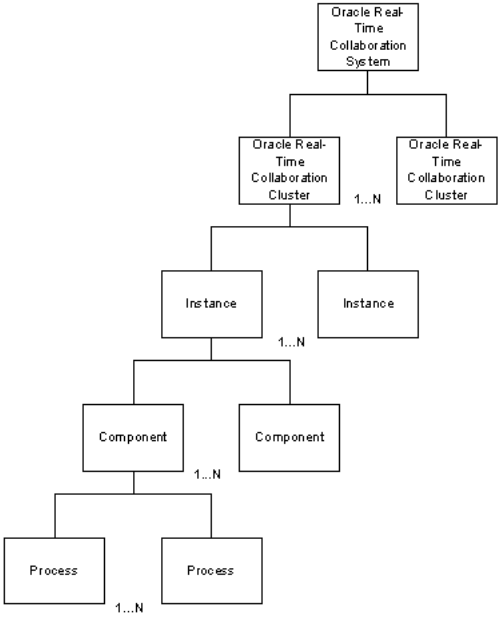
2.2.4 Real-Time Collaboration Cluster

A grouping of Real-Time Collaboration instances that have the same value of "InstanceLocation" property and which use the same Real-Time Collaboration Repository.

2.2.5 Real-Time Collaboration System

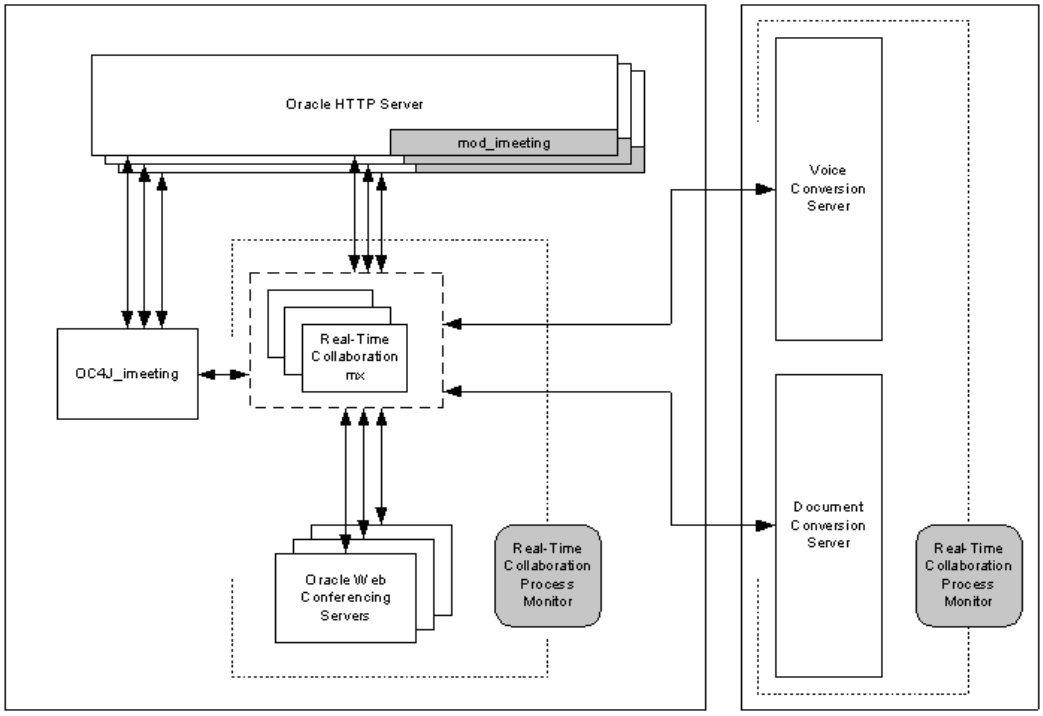
The set of *all* instances that share the same Oracle Real-Time Collaboration Repository. Because instances can be part of clusters, the Oracle Real-Time Collaboration system can be thought of as a set of all clusters.

Figure 2–2 Real-Time Collaboration System



2.3 Real-Time Collaboration Components

Figure 2–3 Real-Time Collaboration Architecture



2.3.1 Oracle Web Conferencing Server

There can be one or more Web Conferencing Server processes in a Real-Time Collaboration instance. The set of Web Conferencing Server processes in an instance constitutes a Web Conferencing Server component. It has the following functions:

- Manages all the conference attendees' states and their permissions within the conference
- Intelligently distributes real-time data for all the collaboration modes that are active during the conference
- Provides services for the recording and archiving of the conference

2.3.2 Real-Time Collaboration mx (Multiplexor)

There can be one or more mx process in an instance. The set of mx processes in an instance constitutes a mx component. The process has the following functions:

- Accepts inbound connections from clients, Web Conferencing Servers, and other Real-Time Collaboration processes.
- Routes data traffic between all clients and all Web Conferencing Servers on a machine.
- Acts as a communication hub for all components.

2.3.3 OC4J_imeeting

The Oracle Web Conferencing J2EE Application running in the Oracle9*i*AS Containers for J2EE. Some of the functions it provides are:

- Provides the Web-based user interface to Oracle Web Conferencing for end-users.
- Provides integration with external applications like Oracle Calendar.
- Interfaces with Oracle Internet Directory for user management.

2.3.4 Real-Time Collaboration Process Monitor (imt-pm)

This process monitors all the other Real-Time Collaboration processes, except OC4J_imeeting processes, and provides high availability by restarting processes if they are down. This process itself is monitored by the Oracle Process Management and Notification system.

2.3.5 Voice Conversion Server

This process dials in to a voice conferencing system, captures the analog voice stream, digitizes it, and streams it to a Web Conferencing Server process through the mx.

2.3.6 Document Conversion Server

This process converts Microsoft Office documents to HTML for document presentation. The Web Conferencing Application provides the front end for a user to convert a document.

2.3.7 HTTPD + mod_imeeting

mod_imeeting is the Real-Time Collaboration plug-in to Oracle HTTP Server. It makes it possible for Oracle Web Conferencing to use Oracle HTTP Server as the only listening point to listen for end-user connection requests. Connections are handed off to the mx using socket hand-off, after the connection has been established using Oracle HTTP Server as the listener.

2.4 Process Interaction

2.4.1 Within an Instance

Within an instance, each Web Conferencing Server process is always connected to each multiplexor process.

Each OC4J process can connect to any mx in the cluster to which the instance belongs. OC4J initially connects to one of the mx processes when required, and from then on caches the connection.

Through the Oracle Web Conferencing Console, each conference participant connects to one of the mx processes, either directly or through a socket hand-off. mod_imeeting connects to all the mx processes in the same instance.

2.4.2 Across Instances

- Across instances, the Voice Conversion Server process in an instance connects to all the mx processes in another instance, which it has been configured to service.
- The Document Conversion Server process in an instance connects to all the mx processes in another instance, which it has been configured to service.

2.4.3 Connections

2.4.3.1 Virtual Channels

Given that all processes connect to the mx, which acts as a communication hub, the following essential virtual channels are created:

- Web Conferencing Console to the Web Conferencing Server process through the mx during a conference.

- The Oracle Web Conferencing OC4Js to the Document Conversion Server process through the mx used for document conversions.
- Each Web Conferencing Server process to the Voice Conversion Server process through the mx for voice streaming during a conference.

2.4.3.2 Database Connections

- Each Web Conferencing Server process maintains a pool of connections to the Real-Time Collaboration Repository.
- Each Oracle Web Conferencing OC4J process maintains a pool of connections to the Real-Time Collaboration Repository.
- Each Document Conversion Server maintains a connection to Real-Time Collaboration Repository.
- Each Real-Time Collaboration Process Monitor maintains a connection to Real-Time Collaboration Repository.

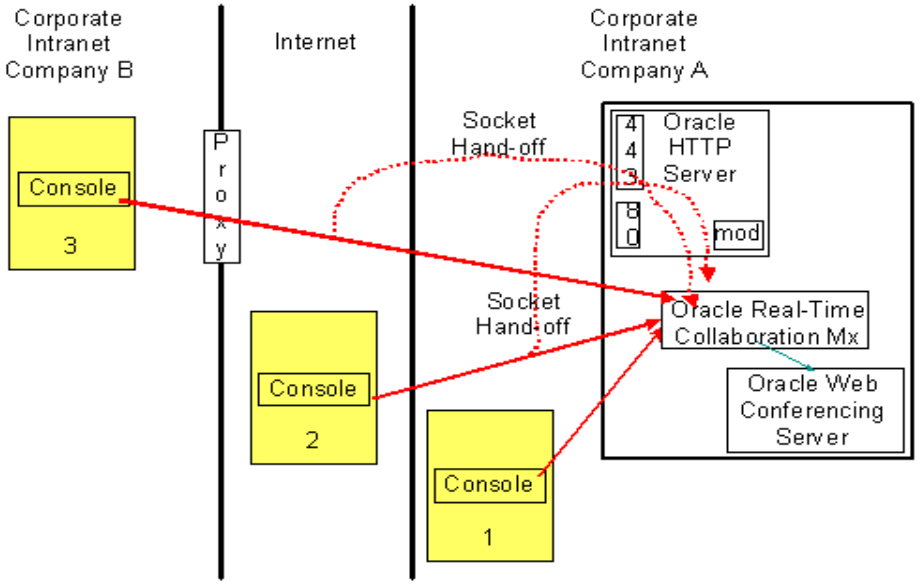
2.5 Runtime Flow

2.5.1 Client Connection Details

All conference attendees fall into one of three categories:

- Attendees connect directly to Oracle Web Conferencing without traversing any firewall. Example: all attendees are in the corporate intranet.
- Attendees connect to Oracle Web Conferencing from the Internet crossing a company firewall. Example: Web Conferencing is deployed in a company demilitarized zone (DMZ), and attendees from the Internet connect to the conference.
- Attendees are coming through the Internet from another company's corporate intranet through their proxy.

Figure 2–4 Client Connection



The Web Conferencing Console attempts to connect to the mx in the following way:

1. Direct TCP/IP—This method is typically successful for clients/computers within a corporate intranet and from the Internet, if the mx port is open to the Internet.
2. HTTPS direct (through Oracle HTTP Server/mod_imeeting)—If direct TCP/IP fails, the Web Conferencing Console tries to connect through HTTPS. This connection is typically successful for client/computers in the open Internet or across transparent proxies. Once a connection is established by Oracle HTTP Server, it is handed off to the mx by mod_imeeting using socket hand-off. Mx and the Web Conferencing Console then communicate directly with each other.
3. HTTPS tunnel (through Oracle HTTP Server/mod_imeeting)—If both direct TCP/IP and HTTPS direct fail, this is the only way to connect for client/computers that are part of a different intranet coming through their own internal proxy. The Web Conferencing Console tries to retrieve proxy information from the browser settings on the client machine and establish a connection to the Oracle HTTP Server using the proxy. Once established, the connection is handed off to the mx by mod_imeeting. Only this time, the Web

Conferencing Console and mx communicate over the HTTPS tunnel through the remote proxy.

The connection information required to connect using the methods in this section is provided to a Web Conferencing Console (transparent to a user) when a user tries to join a conference. The details are explained later in this chapter.

2.5.2 Create Conference Flow

The create conference flow occurs when a host uses the application to create a scheduled or instant conference. The conference is created, and a conference ID is associated with it.

2.5.3 Join Conference Flow

The join conference flow occurs when an attendee joins a conference through the Oracle Web Conferencing Application, an e-mail invitation, or through an application that is integrated with Oracle Web Conferencing. The sequence of events follows:

1. The application looks up the conference record from the database and retrieves hosting server process information, that is, the list of Web Conferencing Servers that can host the conference. Then, the application authorizes the request based on the person joining and the attributes of the conference the person intends to join.
2. The server load balancer functionality in the Oracle Web Conferencing Application (OC4J_imeeting) chooses a Web Conferencing Server process, initiates the session there, and records the conference ID-Web Conferencing Server process association.
3. The list of client parameters for the Web Conferencing Console to connect is generated:
   - mx load balancer (running in OC4J) functions in the Oracle Web Conferencing Application (OC4J_imeeting). It chooses one mx, which can be used to connect to that server process.
   - HTTPS connection information is taken from the mx description (a description of which Oracle HTTP Server/mod can be used to redirect the connection to the given mx) based on how it has been configured.
   - Encrypted client authentication token is generated.

- URLs to send user feedback information and retrieve Java/JSP components used in the console is also generated.

4. The Oracle Web Conferencing Application response causes a pop-up window on the user's Web browser (Internet Explorer 5.5 or above) to open, which contains a Web Conferencing Console Installer (an ActiveX control) with all these parameters.
5. If the user does not have the Web Conferencing Console Installer yet or has an earlier version, the most recent version is downloaded and installed by Internet Explorer automatically (with a permission alert).
6. The Web Conferencing Console Installer performs compatibility checks. If compatibility checks are successful, it checks the version of the Web Conferencing Console available on the client system.
7. If the Web Conferencing Console is not installed on the client machine, or if the Web Conferencing Console version does not match the current one, the new console package is downloaded and installed.
8. The Web Conferencing Console Installer starts the Web Conferencing Console (as a separate process) with all parameters.
9. The Web Conferencing Console tries to establish connection to mx using the algorithm mentioned earlier in Section 2.5.1, "Client Connection Details".
10. If all attempts fail, the client receives an error message.
11. If a connection is established, then the Web Conferencing Console creates a virtual channel through mx to the conference session.
12. The Web Conferencing Console sends an authorization token identifying the client to the server.
13. The Web Conferencing Server sends all conference state (list of attendees, shared content, chat transcripts, and so on) to the Web Conferencing Console to initialize it.
14. The Web Conferencing Console starts to work.

2.6 Ports and Network Connectivity

The following table contains port and network connectivity information.

Table 2–2 Ports and Network Connectivity

Component	Protocol	Port	IP	Number of Ports	Accessibility (Mandatory)	Accessibility (Recommended)
Oracle HTTP Server/mod_imeeting	HTTP	80	Primary	1	All clients	
Oracle HTTP Server/mod_imeeting	HTTPS	443	Primary	1	All clients	
Oracle HTTP Server/mod_imeeting	HTTPS tunnel	443	Secondary[1]	1	All clients	
mx	mx[2]	2400-2700[3]	Primary	n	Voice Conversion Server and Document Conversion Server *must* be able to access the mx(s) on the instances they serve.	Intranet clients could connect using direct TCP/IP. Given that Real-Time Collaboration Core Components are deployed in a DMZ, accessing these port(s) from the intranet is not an issue.
mx (on NT)	redirect[4]	2400-2700	Primary	n	Local host	
voiced	HTTP	2400-2700	Primary	1	All machines with the Real-Time Collaboration Core Components it is servicing.	For remote status
imt-pm	HTTP	2400-2700	Primary	1	Local host from all Real-Time Collaboration instances.	

[1] Second IP address required only if Oracle9*i*AS Web Cache is present on the machine with the Real-Time Collaboration Core Components.

[2] mx is a Real-Time Collaboration internal proprietary protocol.

[3] Port will be chosen from this range.

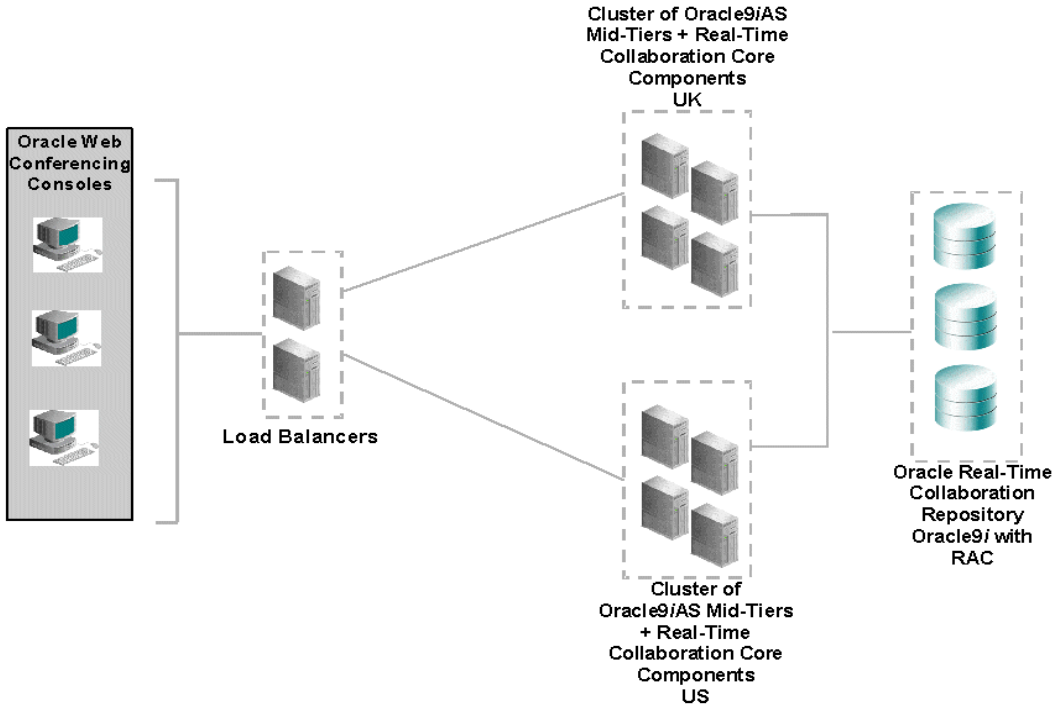
[4] redirect is a Real-Time Collaboration internal proprietary protocol.

For deployments that are accessible from the extra-net, it is enough for Internet- or extranet-facing firewalls of the DMZ to have just the traditional ports (443 and 80) open.

2.7 Real-Time Collaboration Clusters

Without explicit partitioning of the instances in the system, all Web Conferencing Server processes in all instances are considered part of one group. Whenever a new conference is created and a Web Conferencing Server process needs to be chosen for a conference, the conference could be assigned to any one of the Web Conferencing Servers in the system. But, sometimes, it is useful to partition the system into clusters based on geographical distribution.

Figure 2–5 Real-Time Collaboration Clustering



Please note the following points regarding the InstanceLocation property:

- Instances that have the same value for the InstanceLocation property are part of a Real-Time Collaboration cluster. The advantages of creating a Real-Time Collaboration cluster are:
  - Load balancing. For example, all Web Conferencing Server processes in all instances that are part of the cluster become part of one pool of available servers, and the load is balanced intelligently between the different Web Conferencing Servers.
  - High availability. An instance could be down, which means that all Web Conferencing Servers in the instance are down, but other Collaboration Servers from other instances that are part of the same cluster could provide undisrupted service to users.
  - Load separation. With proper us of LBRs, geographically distributed users can be served by separate Real-Time Collaboration clusters that are more locally situated to the users. For geographically distributed companies, you might need to partition the group into smaller groups, with each group servicing a set of users. Use the InstanceLocation property for this purpose. By assigning a different value to the location attribute for the different sets of instances, you can create Real-Time Collaboration clusters in the system. For example, you could set the InstanceLocation property of each instance in a set of Real-Time Collaboration Core Components machines in the United Kingdom to "UK," and set the InstanceLocation property of each instance in the set of Real-Time Collaboration Core Components in the United States to "US."

2.7.1 Real Time Collaboration Clusters for Load Separation

Leveraging the InstanceLocation property requires that geographically separated users are routed to Oracle Web Conferencing Application (OC4J_imeeting) in their respective areas. This can be accomplished automatically with a geographically enabled load balancer or manually by having users in different locations access different Web Conferencing URLs. The Oracle Web Conferencing Application (OC4J_imeeting) in an instance is the component that picks the Web Conferencing Server process for a conference. The Oracle Web Conferencing Application (OC4J_imeeting) knows the Web Conferencing Server process location and picks a Collaboration Server process in an instance which has the same value as its own location.

The instance created by a Document Conversion Server or Voice Conversion Server installation must be assigned a value for the location attribute (like all instances) and it can be assigned to provide service to an Real-Time Collaboration Core

Components machine with a specific value for the location attribute instead of all Core Components machines. For example, each instance of the Document and Voice Conversion Servers in the United Kingdom is assigned to service a set of Real-Time Collaboration Core Components whose location attribute value is "UK." Each instance in the set of Document and Voice Conversion Servers in the United States is assigned to service a set of Real-Time Collaboration Core Components whose location attribute value is "US."

2.8 Load Balancer Configuration

While there are no special requirements for configuring the LBRs themselves, please note the following:

- Oracle Real-Time Collaboration provides the capability for an administrator to partition the system into different Real-Time Collaboration clusters based on geographical distribution. To leverage this feature, set up the LBR to partition user requests based on geographical distribution, as well.
- All machines behind the LBR should have external routable IP addresses and must be directly accessible from the Internet at least on standard HTTP and HTTPS ports (80 and 443). See Section 2.6, "Ports and Network Connectivity" for more information.

See Chapter 6, "Sample Deployments" for more information.

2.9 User Management

2.9.1 User Creation and Provisioning

Oracle Real-Time Collaboration uses the Oracle Internet Directory store, which uses LDAP (Lightweight Directory Access Protocol), to authenticate its users. Any Oracle Internet Directory user can use Real-Time Collaboration. Users are created using the standard mechanisms available through Oracle Internet Directory.

See Also: *Oracle Internet Directory Administrator's Guide* for details

2.9.2 Roles

Real-Time Collaboration users can be assigned different roles. Roles determine the Oracle Web Conferencing functionality to which a user has access. There are three roles in Web Conferencing:

2.9.2.1 End-User Role

The end-user role, *enduser*, is the default role given to any user who logs in to the system for the first time. This role is intended for all regular users of Oracle Web Conferencing.

2.9.2.2 Business Monitor Role

The business monitor role, *businessmonitor*, is intended for those Oracle Web Conferencing users who want to monitor the system and have access to various reports that can be run on the system. Users with this role have access to the Monitor and Reports tabs, in addition to all end-user tabs in the Oracle Web Conferencing Application.

2.9.2.3 Business Administrator Role

The business administrator role, *businessadmin*, is intended for those Web Conferencing users who are in charge of administering the Web Conferencing deployment. This includes users who are responsible for supporting the end-users. Users who have this responsibility have access to the Site Management and the System Configuration tabs in the Web Conferencing Application.

2.9.3 Assigning Roles to Users

Use the imtctl command, `modifyRole`, to assign roles to Web Conferencing users. See Chapter 10, "imtctl Command Line Utility" for more information on imtctl.

2.10 Web Conferencing Sites

As an enterprise-class platform for real-time collaboration, Oracle Web Conferencing allows a single deployment instance to easily be configured to meet the requirements of various lines of business. This objective can be achieved by creating individual sites and customizing system, application, and conference level properties.

> **Note:** Users with the businessadmin role can click the Sites tab in the Web Conferencing Application to access the Sites functionality.

Web Conferencing sites are created by the Web Conferencing business administrator. For every site created, the integrating application or line of business

uses a unique authentication token along with site ID to communicate with the Web Conferencing Application. Creating a site provides the following benefits:

- Customized system, application, and conference properties.
- Support for custom integrated flows which can bypass the Web Conferencing Application.
- Support for custom Web Conferencing Console behavior based on the properties set for the site.
- Site-level reports that are similar to global Oracle Web Conferencing reports, except they use site-specific data.
- Custom user interfaces, including a separate Oracle Web Conferencing prelogin application page with full support for listing public conferences only for the specific site.
- Custom branding to uniquely identify the integrating site/line of business.

Note: The global site is pre-created and cannot be deleted. By default, all users belong to site "*i*Meeting," which is a global site. However, a user can belong to more than one site.

2.11 Reports

Oracle Real-Time Collaboration provides various reporting capabilities, including e-mailed reports and usage trend information available within the Oracle Web Conferencing Application. Some aspects of these features require post-installation configuration, such as including sender's and receiver's e-mail addresses. See Chapter 4, "Post-Installation" for details on the properties that must be configured.

Anyone with the businessadmin or businessmonitor role can access the reports functionality by clicking the Reports tab in the Oracle Web Conferencing Application.

2.11.1 Usage Report

The usage report is designed to give high-level Oracle Web Conferencing usage metrics to business administrators and business managers. This report contains data which is based on one of the following two metrics:

- Time (minutes/hours)
- Usage numbers (count of conferences, users, and so on)

The Key Performance Indicators (KPI) are then summarized with a trending pattern that displays the count and percentage change over the previous time period. The report gives business administrators and business managers an idea of what collaborative modes and features are used, top users, and line-of-business information. The usage report is computed for a weekly period. Any user who has access to the report can view historical data in addition to the current data by keying in the relevant dates.

The following metrics are captured in the usage report:

- Conference minutes summary:
  - Total user minutes
  - Total conference minutes
  - Total voice minutes
  - Total recorded minutes
  - Total playback minutes
  - Host rating
  - Average conference length
  - Longest conference (minutes)
- Conference usage summary:
  - Conferences
  - Voice conferences
  - Total users
  - Total recordings
  - Total playbacks
  - Users in conferences
  - Voice users
  - Average users per conference
  - Largest conference (users)
- Top 10 lists
  - Users
  - Sites

In addition to the weekly KPIs table, the conference minutes and usage details are displayed in a time series of week, month, quarter, and year. The weekly average over the highest period is also computed.

2.11.2 Feedback Reports

The feedback report is designed to give high-level system performance feedback through user ratings to the business administrator and to business managers. The report contains data which is based on performance ratings given by conference hosts at the end of every conference.

The host of the conference can give a comment (raw text feedback) and can rate a conference excellent, good, or poor. The host can also decide to not rate the conference, in which case the conference is classified with no response. The KPIs are then summarized with a trending pattern displaying the count and percentage change over the previous time period.

Below the KPIs table, the top ten conferences with poor, good, excellent, and no response ratings with user comments are listed. These are grouped into four individual tables. Only conferences with host comments are listed in the tables. This convenient format allows administrators to respond proactively to the users who rate the conferences poor.

2.11.3 Uptime Reports

Oracle Web Conferencing provides tests that can be used by monitoring infrastructures to periodically monitor the system for service availability. This data can then be used to produce uptime reports. See Section 7.6, "Real-Time Collaboration Interfaces" for more information on the tests and how to integrate them with your monitoring infrastructure.

3

Planning for Deployment

3.1 Overview

This chapter highlights the factors to consider when deploying Oracle Web Conferencing. The issues range from sizing guidelines to firewall issues to topology considerations for an enterprise that is geographically distributed.

3.2 Distributed Deployment Considerations

3.2.1 Web Conferencing Issues

Given the intrinsic nature of Oracle Web Conferencing functionality where data from one user's desktop is distributed to other end-user desktops, there could be latency issues for end-users, depending on their location. Oracle Web Conferencing, by itself, cannot address the network latency issues. You must resolve these issues through other mechanisms.

In environments in which users are geographically dispersed, a deployment with a single set of Real-Time Collaboration Core Components instances is not optimal for preventing network latency issues. Instead, consider the following scenarios:

Scenario 1—If there are a lot of conferences where most of the attendees are in the same geographical region, then Oracle Corporation recommends deploying a set of Real-Time Collaboration Core Components instances in that geographical region.

Example: A company has multiple divisions all over the world, and each division's employees have a lot of conferences with each other, and, sometimes, they have conferences with employees from other divisions that are in different locations.

In such a scenario, it does not make sense to force users to use a Real-Time Collaboration cluster in a different location, thereby causing network latency

problems for users. Each location could have its own set of Real-Time Collaboration Core Components instances.

Scenario 2—If attendees of most conferences are in different regions, then having a set of Real-Time Collaboration Core Components instances in each geographical region does not help.

Example: A US-based company has outsourced its sales/support organization to a site in India. If a typical conference involves a sales agent from the site in India and a customer in the US, then there is no reason to deploy Real-Time Collaboration Core Components instances in the India site.

3.2.2 Voice Conversion Server Issues

You should deploy Voice Conversion Servers in areas where they are able to successfully dial in to all conference numbers that will be used. In view of phone charges, it might be beneficial to deploy a Voice Conversion Server in the region where most calls terminate. For example, if a significantly large number of Web Conferencing users dial out to a particular region like the United Kingdom, it would be beneficial to have a Voice Conversion Server deployed in the UK instead of using a Voice Conversion Server in the United States to dial the UK numbers.

3.3 Sizing Guidelines

See the Oracle Web Conferencing Sizing Guide to determine the hardware required to set up Web Conferencing for your specific needs.

3.4 Internet/Intranet Considerations

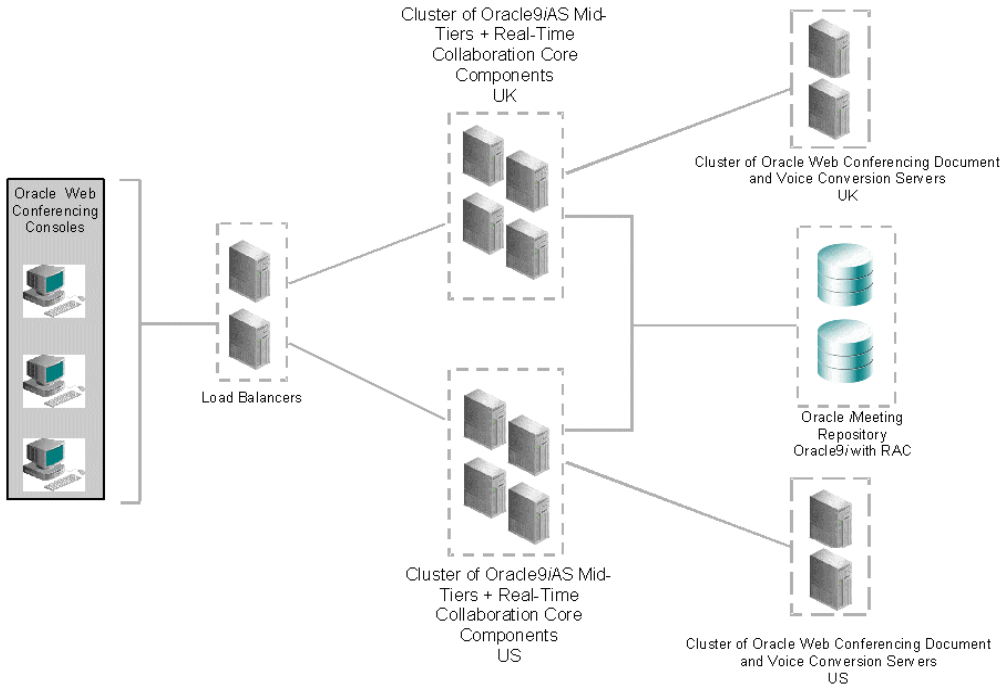
All typical firewall considerations for Web-based applications (Real-Time Collaboration Core Components in DMZ listening on port 80 for HTTP and port 443 for HTTPS, Real-Time Collaboration Repository behind the inner firewall, Oracle9*i*AS Infrastructure behind the inner firewall) apply when deploying Web Conferencing. Additional requirements for Web Conferencing follow:

- Mx listening port(s): As explained in Chapter 2, "Understanding Oracle Web Conferencing", it is mandatory that Document and Voice Conversion Servers be able to connect to the Real-Time Collaboration Core Components they are servicing. If they are deployed in different DMZs, intra-DMZ firewalls need to be opened up on the port(s) to which the mx component on the Real-Time Collaboration Core Components machine is listening.

- Mx listening port(s): All intranet users should be able to make a direct TCP/IP connection to the Real-Time Collaboration Core Components (the ports that the mx is listening on).

3.5 LBR Considerations

Figure 3–1 Geographically Distributed Deployment Using LBRs



For geographically distributed deployment, you can use load balancers (LBR) to leverage the Real-Time Collaboration clustering support.

- Leveraging the InstanceLocation property requires that geographically separated users are routed to the Oracle Web Conferencing Application (OC4J_imeeting) in their respective areas. This can be accomplished automatically with a geographically enabled load balancer or manually by having users in different locations access different Web Conferencing URLs. The Oracle Web Conferencing Application (OC4J_imeeting) in an instance is the component that picks the Web Conferencing Server process for a conference. The Oracle Web

Conferencing Application (OC4J_imeeting) knows the Web Conferencing Server process location and picks a Collaboration Server process in an instance which has the same value as its own location.

- Even when an LBR is used, it is mandatory that all Real-Time Collaboration Core Components behind the LBR have externally routable addresses and are accessible directly from the Internet. The Web Conferencing Console needs to establish a connection to mx. See Chapter 2, "Understanding Oracle Web Conferencing" for more information.
- When using an LBR, follow the special configuration guidelines in the Advanced Configuration document at http://otn.oracle.com/products/webconferencing/ to properly configure Oracle9*i*AS Single Sign-On for use by all the Oracle Real-Time Collaboration Core Components.

4

Post-Installation

This chapter discusses common post-installation steps required to make Oracle Web Conferencing completely functional in the selected deployment configuration.

To configure Oracle Web Conferencing, you must:

- Set Web Conferencing properties
- Configure prerequisite products on which Web Conferencing depends.

4.1 Overview

A standard configuration is a deployment in which corporate intranet users can hold online conferences with voice streaming and document conversion service. For additional functionality like e-mail invitations and conference access to users from outside the corporate firewall, follow the instructions in this chapter.

dcmctl is the command line utility used for most management tasks like start and stop for Oracle9iAS components like Oracle HTTP Server, OC4J_imeeting, and Oracle Process Management and Notification. There are other options, like the 9iAS Enterprise Manager for performing the same tasks. This document uses dcmctl to illustrate the tasks.

> **See Also:** *Oracle9i Application Server Administrator's Guide* for information on dcmctl

4.1.1 Introduction to imtctl

The imtctl utility provides a command-line interface for administering and configuring Web Conferencing and the Real-Time Collaboration system. The utility supports a variety of commands to support actions such as starting and stopping Real-Time Collaboration components, viewing their current state, and configuring

them. You can run imtctl in an interactive shell mode, or you can issue a single command as a command-line argument, in which case imtctl executes the command and exits.

The utility is available on all platforms under <ORACLE_HOME>/imeeting/bin. It is invoked as "imtctl" on all platforms. Invoking imtctl without any arguments opens the interactive shell. See Chapter 10, "imtctl Command Line Utility" for more information.

```
$ORACLE_HOME/imeeting/bin> imtctl
```

To display a list of supported commands, invoke the help command:

```
imtctl> help
```

To display specific instructions for a particular command, invoke the command followed by the `-help` option:

```
imtctl> <COMMAND> -help
```

The exit command ends the shell session:

```
imtctl> exit
```

4.1.2 imtctl Scripts

You can write simple scripts for the imtctl utility. These scripts consist of commands that are executed by imtctl, as well as optional comments. The advantage of these scripts over standard operating system scripts is that imtctl scripts are platform-independent and do not require multiple invocations of the utility to perform multiple commands.

Oracle Web Conferencing comes with a set of standard scripts you can use to perform typical post-installation configuration tasks. These scripts reside in the <IMT_ HOME>/bin/scripts directory and end with the extension ".imt". Each script has been templated to indicate the values that must be set before running the script. Invoke the scripts using standard input redirection as in the following example:

```
imtctl> $ORACLE_HOME/imeeting/bin> imtctl < ../scripts/sample.imt
```

Note: The command line might vary slightly across platforms, but the same basic usage is supported.

You can create new scripts, as needed. The scripts can contain any supported imtctl commands, plus comment lines (any line that begins with #) and the special "echo [on/off]" command for echoing commands executed by scripts.

4.1.3 Starting the Real-Time Collaboration Instance

At installation time, all the processes are automatically started by the installer.

At other times, all Web Conferencing components in an instance can be started using imtctl, except the Web Conferencing Application (OC4J_imeeting). The non-Web Conferencing component products on which Web Conferencing depends are started independently using their own management tools. Only basic startup commands for such products are listed. To start the Real-Time Collaboration Core Components instance:

Start OC4J with the following command:

```
$ORACLE_HOME/dcm/bin/dcmctl start -co OC4J_imeeting -t -v 120
```

Start the Oracle HTTP Server, if it is not already up. The following command shows the status of the Oracle HTTP Server, OC4J_imeeting, etc.:

```
$ORACLE_HOME/dcm/bin/dcmctl getState -v
```

To start the Oracle HTTP Server, invoke:

```
$.../dcmctl start -ct ohs -v -t120
```

To start the Oracle Process Monitor and Notification system, invoke:

```
$.../dcmctl start -ct opmn
```

To use imtctl, invoke the following start command:

```
$ORACLE_HOME/imeeting/bin> imtctl start
```

Alternatively, you can invoke imtctl in shell mode and then issue the start command. The second option makes it easier to verify the state of the Real-Time Collaboration system after the start command is completed:

```
$ORACLE_HOME/imeeting/bin> imtctl
imtctl> start
imtctl> getState
```

4.1.4 Stopping the Real-Time Collaboration Instance

Stop the system by invoking the following stop command:

```
imtctl> stop
```

The Oracle HTTP Server and OC4J_imeeting are stopped using the corresponding dcmctl stop commands.

4.2 Real-Time Collaboration Core Components Installation

When you install the Real-Time Collaboration Core Components, you are installing the Web Conferencing Server and other related components. Every time you perform a new installation that results in the creation of an instance, you must perform the following two types of tasks:

- System tasks—These tasks must be done once for the system. In most cases, there is no need to repeat these steps when a new instance is added to the system. However, you can use the global settings or override them for specific instances.
- Instance tasks—These must be done for every instance. The set of specific tasks that must be completed is determined by the deployment scenario (for example, Internet-facing and SSL-enabled).

Table 4–1 Task Checklist

Task	Reason it is Required	Nature of Task	Refer to:
Integrate with existing e-mail	Oracle Web Conferencing must be integrated with the e-mail infrastructure in order to send e-mail invitations for conferences.	System task	See Step 2 of "System Tasks" in this chapter.
Integrate with existing proxy server	Oracle Web Conferencing must be integrated with the existing proxy settings for use by the Web Conferencing Console. This includes cobrowsing functionality, and cross-instance HTTP-based diagnostics.	System task	See Step 1 of "System Tasks" in this chapter.
Integrate with Oracle HTTP Server	Oracle Web Conferencing must be integrated with the Oracle HTTP Server associated with that instance. This allows access to intranet users behind remote firewalls.	Instance task	See Step 1 of "Instance Tasks" in this chapter.
Configure Oracle Web Conferencing for reports	Oracle Web Conferencing provides automatic generation and e-mailing of reports. Web Conferencing must be configured for this functionality.	System task	See Step 3 of "System Tasks" in this chapter.
Set up for periodic monitoring	The instance can be monitored periodically and the results plugged into an alert management system.	Instance task	See Step 2 of "Instance Tasks" in this chapter.
Assign businessadmin role to an administrator	This allows an administrator to view the administration pages in the Web Conferencing Application.	System task	See Step 4 of "System Tasks" in this chapter.

4.2.1 Post-Installation Tasks

4.2.1.1 System Tasks

Step 1 Integrate with existing proxy servers.

The Web Conferencing Cobrowse mode accesses Web sites using the HTTP/S protocols from the Real-Time Collaboration Core Components instance machine. Depending on network topologies and firewall configurations, it might require an

HTTP/S proxy server to access some sites. The following script provides a template for enabling proxy functionality:

```
$ORACLE_HOME/imeeting/bin/scripts/proxy.imt
```

Before executing the script, edit it to replace templated-values with actual deployment values. See the instructions in the script for details.

Set the following properties:

- UserAgentProxyHost
- UserAgentProxySSLHost
- UserAgentProxyPort (only if it is not 80)
- UserAgentProxySSLPort (only if it is not 443)
- UserAgentProxyExclusions
- UserAgentProxyEnabled

See Chapter 5, "Configuration" for more details on these properties.

Step 2 Integrate with existing e-mail.

The Web Conferencing Application can generate outbound e-mail for various purposes, such as inviting users to a conference. The Real-Time Collaboration system must be configured with certain e-mail-related properties to enable this feature. See the following imtctl script for details:

```
$ORACLE_HOME/imeeting/bin/scripts/mail.imt
```

Before executing the script, edit it to replace templated-values with actual deployment values. See the instructions in the script for details.

Set the following properties:

- SmtpHost
- SmtpPort (only if it is not 25)
- EmailEnabled

See Chapter 5, "Configuration" for more details on these properties.

Step 3 Set up reports.

Oracle Web Conferencing supports various reporting capabilities. Web Conferencing reports can be generated and sent via e-mail using the imtreport

script. The script is located at $ORACLE_HOME/imeeting/bin and is invoked as "imtreport" on all platforms. The actual script file is "imtreport" for UNIX and Linux and "imtreport.cmd" for Microsoft Windows. Mandatory report options such as the recipient list are set by editing variables at the top of the report script. The script contains comments on the options provided. Be sure to edit the proper script for the platform you are using. You may want to send out automated weekly reports by using some operating system provided mechanism for invoking the script on a weekly basis (for example, cron job on UNIX or the Task Scheduler on Windows).

Step 4 Set up the administrator role.

```
imtctl> modifyRole -username "username@company.com" -rolename "businessadmin"
```

4.2.1.2 Instance Tasks

These tasks assume that Oracle9*i*AS Web Cache is turned off on the machine. If Oracle9*i*AS Web Cache is turned on, a second IP address and additional configuration steps are needed. See Chapter 5, "Configuration" for more instructions.

Step 1 Integrate Web Conferencing with Oracle HTTP Server on this Real-Time Collaboration Core Components machine.

Enable the mod_imeeting.

Make the necessary configuration changes to the Oracle HTTP Server in which mod_imeeting is configured. mod_imeeting is included through the `$ORACLE_HOME/imeeting/conf/mod_imeeting.conf file`, which in turn is included in `$ORACLE_HOME/Apache/Apache/conf/oracle_apache.conf` file for Oracle HTTP Server.

To activate the mod:

1. Add the following line for mod_imeeting.conf in oracle_apache.conf line:

   ```
   include "$ORACLE_HOME/imeeting/conf/mod_imeeting.conf"
   ```

2. Update the Oracle HTTP Server configuration using DCM.

   See Also: *Oracle9i Application Server Administrator's Guide* for information on dcmctl

   ```
   $ORACLE_HOME/dcm/bin/dcmctl -v updateConfig -ct ohs
   ```

3. Restart Oracle HTTP Server through DCM.

```
$ORACLE_HOME/dcm/bin/dcmctl -v restart -ct ohs
```

4. Set up mod_imeeting/mx redirect.

   See the following imtctl script for details:

   ```
   $ORACLE_HOME/imeeting/bin/scripts/Redirect.imt
   ```

   Before executing the script, edit it to replace templated-values with actual deployment values. See the instructions in the script for details.

   Set the MxRedirectPort property. See Chapter 5, "Configuration" for more details on this property.

5. Configure Web Conferencing to use basic Oracle HTTP Server settings.

   The Real-Time Collaboration installer assumes that the actual host name configured for the Real-Time Collaboration Core Components machine is used by clients to access Real-Time Collaboration resources. This might not be true in some deployments where DNS aliasing and other advanced network deployment scenarios might require customization of the names used to access Real-Time Collaboration resources from external locations. The following script provides examples on how to support such configurations:

   See the following imtctl script for details:

   ```
   $ORACLE_HOME/imeeting/bin/scripts/hostname.imt
   ```

   Before executing the script, edit it to replace templated-values with actual deployment values. See the instructions in the script for details.

   Set the following properties:

   - ApacheWebHost
   - ApacheWebPort
   - ApacheWebSecurePort
   - ApacheProtocolSecure

   See Chapter 5, "Configuration" for more details on these properties.

Step 2 Set up service availability tests for periodic monitoring.

Set up your monitoring infrastructure to periodically ping the following URLs. For example, assuming the URL for the instance (ApacheWebHost) is imeeting1.company.com:

For conference service availability monitoring:

```
http://imeeting1.company.com/imtapp/servlet/ImtTestServlet?mtgtest=true
```

For Voice Conversion Server availability monitoring:

```
http://imeeting1.company.com/imtapp/servlet/ImtTestServlet?voiceconvtest=true
```

For Document Conversion Server availability monitoring:

```
http://imeeting1.company.com/imtapp/servlet/ImtTestServlet?docconvtest=true
```

See Section 7.6, "Real-Time Collaboration Interfaces" for more information.

4.2.2 Verification

Invoke the following test to verify that the Real-Time Collaboration Core Components have been configured correctly.

```
imtctl> runtests
```

For each test that has failed, verify that the post-installation steps have been done correctly.

> **Note:** The Voice Conversion Server availability test and the Document Conversion Server availability test will fail on the Core Components instance if the Document and Voice Conversion Servers have not been configured to serve the Real-Time Collaboration cluster to which the Core Components instance belongs.

4.3 Installing the Document and Voice Conversion Servers

When you install the Document Conversion Server or Voice Conversion Server, a new instance of the newly installed server is created. At the end of installation, it serves the Core Components instances in the default Real-Time Collaboration cluster. If you want it to serve instances in another cluster you must shut it down then reconfigure to use the other cluster and restart.

See Section 2.7, "Real-Time Collaboration Clusters" for more information on Real-Time Collaboration clusters.

4.3.1 Configuring Voice

4.3.1.1 VoiceDialInPrefix Property

Description: This property sets the dial-in prefix for an instance of the Voice Conversion Server. This prefix would be added to all dial-in numbers received by the Voice Conversion Server before dialing out. Typically, corporations have a dial-in prefix ("9," for example) for dialing numbers outside the company's internal phone network.

Default Value: None

Valid Value: Any valid dialing sequence. In most cases, it would be a number example: "9" A valid dialing sequence contains numbers 0-9 and any of the following characters: * # ,

Scope: Instance

To configure the dial-in prefix for a Voice Conversion Server instance to "9" invoke:

```
imtctl> setProperty -i <instance-name> -pname "VoiceDialinPrefix" -pvalue "9"
```

4.3.1.2 System Dial-Ins

System dial-ins are system-wide dial-ins that an administrator creates. They are available to all Oracle Web Conferencing users. Typically, system administrators create system dial-ins for phone conference vendors that are used commonly by the users of the system. System dial-ins serve as templates for users to create their own dial-ins; however, users cannot edit them. In many cases, system dial-ins are incomplete dialing sequences in that they contain information that is common to all users and indicate the information a user must add to make it a usable dial-in. Users should make the recommended addition to the system dial-in and save it as a user-defined dial-in for use in a conference.

For example, phone conference vendors typically provide a PIN or conference ID to its users. A system administrator might create a system wide dial-in for the particular conference vendor that contains the toll-free number for the vendor, followed by a placeholder for the PIN/Conference ID and the required separators like ',', '#' or '*'. Users would then modify the system dial-in by adding their PIN/Conference ID and then save it under a different name.

Table 4–2 Dial-Ins

Dial-in Name	Dial-in Sequence
Audio Conferencing Vendor (US)	18005555555,<ADD CONF ID HERE>#,,,,,,,,,#,,,,,,,,#
Joe's account with Audio Conferencing Vendor	18005555555,8282828#,,,,,,,,, #,,,,,,,,#

System dial-ins can be managed through the imtctl utility.

4.3.1.3 Creating a New System Dial-In

Invoke the following:

```
imtctl> addSysDialin -name <dialin-name> -sequence <dialin-sequence> -default
<true/false>
```

The default indicates whether the system dial-in should be made the default for all users. Users can override the default, from the Web Conferencing Console, if they choose.

4.3.1.4 Deleting a System Dial-In

Invoke the following:

```
imtctl> deleteSysDialin -name <dialin-name> -id <dialin-id>
```

The name or ID is required. If both are specified, name is used. To get the name or ID, invoke `getSysDialins`.

4.3.1.5 Viewing Existing System Dial-Ins

Invoke `getSysDialins`. It lists the ID, name, and sequence for existing system dial-ins.

4.3.1.6 Configuring the Voice Conversion Server to Serve Instances in a Specific Real-Time Collaboration Cluster

Invoke the following:

```
imtctl> setProperty -pname InstanceLocation -pvalue  "[\"InstanceLocation value
for cluster 1, 2, etc.\"]"
```

4.3.1.7 Starting the Voice Conversion Server.

Invoke `imtctl> start` to start the Voice Conversion Server.

4.3.1.8 Verification

To verify that this Voice Conversion Server is set up correctly, shut down the other Voice Conversion Servers, if any, that are assigned to serve the same Real-Time Collaboration Core Components to which the new server is assigned. Then, on a Real-Time Collaboration Core Components instance machine do:

```
imtctl> runtests -testlist voiceconvtest
```

Do the same for the Document Conversion Server:

```
imtctl> runtests -testlist docconvtest
```

Verify that the Core Components machine is able to connect to it.

4.4 Configuring Time Zones

Users can set their time zone by logging in to the Web Conferencing Application and clicking Preferences. A list of available time zones displays.

After users have set their time zone, all date and time references in the Web Conferencing Application and Web Conferencing Console are automatically converted to their time zone.

Example: User A sets his time zone to "(-05:00) US Eastern Time." He then schedules a conference for 30-Mar-2004 5:00 PM. He intends to have the conference at 5 pm Eastern Time. User B, whose time zone is set to "(-08:00) US Pacific Time," is invited to this conference. When User B sees the invitation in her Upcoming Meetings tab, it displays as 30-Mar-2004 2:00 PM. The time has been converted to Pacific Time for User B.

See Appendix A, "Time Zones" for a complete list of supported time zones.

Setting the Default System-Level Time Zone

When a user registers with Oracle Web Conferencing for the first time, the user's default time zone is set based on the system-level property DefaultTimeZoneName. The default value of this property is UTC. An administrator might want to change this value. For example, if the Real-Time Collaboration system is deployed in New York, you can change the property as follows:

```
imtctl> setProperty -system -pname DefaultTimeZoneName -pvalue America/New_York
```

This only affects the default time zones for users who have not set their time zone yet. You should encourage users to set their time zones.

See Appendix A, "Time Zones" for a complete list of valid values for DefaultTimeZoneName.

4.5 Custom Configuration

For custom configuration, including Real-Time Collaboration cluster creation and use of SSL, please refer to Chapter 5, "Configuration" and Chapter 6, "Sample Deployments".

5 Configuration

5.1 Overview

This chapter provides a greater level of detail about configuration than Chapter 4, "Post-Installation". You should use this as the main configuration reference.

To configure Oracle Web Conferencing, you must:

- Set Web Conferencing properties
- Configure the prerequisite products on which Web Conferencing relies

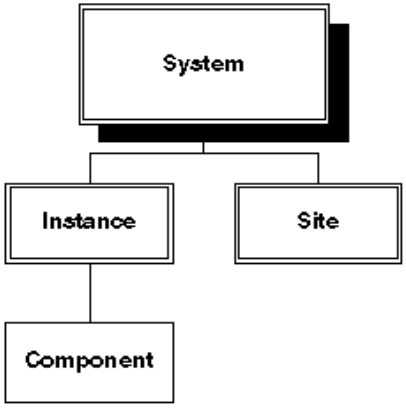
This chapter divides all aspects of Web Conferencing configuration into different tasks. The section on each task discusses the properties applicable for that task and the external product configuration.

This chapter also explains the basics of the property management system and configuration management.

5.2 Property Management

Properties are configuration parameters. The Real-Time Collaboration property manager maintains all properties in the Real-Time Collaboration system.

Figure 5–1 Configuration Properties



Each node in the hierarchy in Figure 5–1, "Configuration Properties" corresponds to a scope, and every property has a scope associated with it. The scopes are:

- system properties—These properties are set at the system level. They are typically set once and do not need to be changed when a new instance (or new set of Real-Time Collaboration Core Components) is added to the system.

  An example of this are load balancer (LBR) settings. If an enterprise has one LBR, it only needs to be set once.

- instance properties—These are properties that are set at an instance level. Any change in the value of these properties affects that instance only. An example of this is location. Each instance in the system could have its location attribute set to a different value.

- component properties—These properties are set at a component level for a specific instance. A change in the properties affects the configuration of that component only. An example of this is the number of processes. Each component in an instance can have a different number of processes. For example, the Mx component could have two mx processes, and the Web Conferencing Server component could have four Collaboration Server processes.

  Also, changes to component properties apply to that component in that instance only. Changes do not affect the same components in other instances. Therefore, if you want to set the number of mx processes to five on all instances, you must do this to each instance.

5.2.1 Site Properties

These are properties that are set at a site level. An example of this is the text of a message that appears below the login link on the login page.

5.2.2 Multiple Scope Properties

These are properties that are meaningful at multiple scopes. Oracle Real-Time Collaboration provides the following:

- Inheritance—The property value at a particular scope can be inherited from higher up in the hierarchy. For example, log level could be set at the system scope, and all instances and components in instances will log at that level.
- Allow Override—A property could be set at a certain scope, but the value can be overridden lower down in the hierarchy and set to a different value. For example, the log level could be set at the system scope, but you can change the log level of a specific instance. All other instances will maintain the log level set at the system scope.

This approach provides an administrator maximum flexibility and ease in configuring the system.

Scope Definitions

system—property can be set at system scope.

instance—property can be set at instance scope.

component—property can be set at component scope.

system and instance—property can be set at system scope but can be overridden for a specific instance.

instance and component—property can be set at instance scope but can be overridden for a specific component in that instance.

system, instance, component—property can be set at a system scope, overridden for a specific instance, and in turn overridden for a specific component in that instance.

site—property can be set at site scope.

system and site—property can be set at system scope but can be overridden for a specific site.

5.2.3 System Map

All configuration information for the Real-Time Collaboration system is stored in the Real-Time Collaboration Repository in the system. This includes configuration information of all the instances, all the components in the instances and how the instances are connected together.

See Chapter 10, "imtctl Command Line Utility" for a view of the system map that you can obtain using imtctl.

5.3 Configuration Tasks Overview

The following options are available for configuring Oracle Web Conferencing:

- Application Pages—See Section 5.4, "Configuring the Application Pages"
- Usage Reports—See Section 5.5, "Configuring Reports"
- Ports and Network Connectivity—See Section 5.6, "Configuring Ports and Network Connectivity"
- Logging—See Section 5.8, "Configuring Logging"
- E-mail—See Section 5.9, "Configuring E-mail"
- Real-Time Collaboration Clusters—See Section 5.10, "Configuring Clusters"
- Conference Runtime Characteristics—See Section 5.11, "Configuring Conference Runtime Characteristics"
- Monitoring—See Section 5.12, "Configuring Monitoring"
- User Privileges—See Section 5.13, "Configuring User Privileges"

When you add a new instance to the system, follow the steps in Chapter 4, "Post-Installation". Use this chapter as a reference for all configuration details.

5.4 Configuring the Application Pages

You can customize various elements of the Oracle Web Conferencing Application. These customizations can have a system scope and also a site scope. When a property has a site scope and the site property value is available, if the Web Conferencing Application is invoked from the default site, and then the site property value supersedes the system-level value.

PreLoginMessage

Description: This property sets the text of a message that appears below the login link on the login page.

Default value: "Login to *i*Meeting is based on Single Sign-On. To log in, enter your global e-mail ID (first.last@oracle.com) and your e-mail password."

Valid values: Any text message.

Scope: System, Site

Example: To set it to "Log in to OracleWeb Conferencing" invoke:

```
imtctl> setProperty -system true -pname PreLoginMessage -pvalue "Log in to
Oracle Web Conferencing"
```

PublicMeetingNumRows

Description: This property sets the number of rows that are displayed in the Public Meetings table.

Default value: 10

Valid values: -1 for all rows, or any positive integer.

Scope: System, Site

Example: To show all the rows in the Public Meetings table for a particular site identified by CompanySupport (siteID 123456), invoke:

```
imtctl> setProperty -siteID 123456 -pname PublicMeetingNumRows -pvalue -1
```

PublicMeetingSortBy

Description: This property sets the "sort by" column of the Public Meeting table.

Default value: The number of the "Start Date" column.

Valid values: Any integer indicating the column position, starting from 0.

Scope: System, Site

Example: To sort the Public Meetings table by the second column in the whole system, invoke:

```
imtctl> setProperty -system true -pname PublicMeetingSortBy -pvalue 1
```

PublicMeetingSortOrder

Description: This property sets the sort order of the column specified in the PreLoginPublicMeetingSortBy property.

Default value: asc

Valid values: asc=ascending, desc=descending

Scope: System, Site

Example: To sort the rows in descending order in the selected column, invoke:

```
imtctl> setProperty -system true -pname PublicSortOrder -pvalue desc
```

5.4.1 Configuring the Quicklinks Bin

Using the Quicklinks bins on the prelogin and the home pages, you can make commonly used URLs easily accessible. You can add four user-defined links to this bin. Use the Quicklinks properties to set the name of these links, as well as the URL associated with them.

Quicklink<n>Name

Description: The Quicklink<n>Name property, where <n> is any number from 1 through 4, sets the name of the Quicklink. This is the name that appears in the bin.

Default value: Quicklink<n>

Valid values: Any string.

Scope: System, Site

Example: To configure the first user-defined link to be called "Real-Time Meetings," invoke:

```
imtctl> setProperty -system true -pname Quicklink1Name -pvalue "Real-Time
Meetings"
```

Quicklink<n>URL

Description: The Quicklink<n>URL property, where <n> is any number from 1 through 4, sets the URL associated with Quicklink<n>. This is the URL that is invoked when a user clicks the link.

Default value: None

Valid values: Any valid URL

Scope: System, Site

Example: To configure the first user-defined link to be associated with http://realtimemeetings.company.com, invoke:

```
imtctl> setProperty -system true -pname Quicklink1URL -pvalue
"http://realtimemeetings.company.com"
```

Note: If the value associated with any of the Quicklinks is null, then that link does not display in the Quicklinks bin.

5.5 Configuring Reports

Oracle Web Conferencing reports can be generated and sent via e-mail using the imtreport script. The script is located at $ORACLE_HOME/imeeting/bin and is invoked as "imtreport" on all platforms. The actual script file is "imtreport" for UNIX and Linux and "imtreport.cmd" for Microsoft Windows. Mandatory report options such as the recipient list are set by editing variables at the top of the report script. The script contains comments on the options provided. Be sure to edit the proper script for the platform you are using. You may want to send out automated weekly reports by using some operating system provided mechanism for invoking the script on a weekly basis (for example, cron job on UNIX or the Task Scheduler on Windows).

In addition, you must configure the following Web Conferencing property.

ReportEnvironmentName

Description: This property sets the name of the environment for which the report is sent. This value appears in the report header.

Default value: None

Valid values: Any text

Scope: System

Example: To set the report environment name to "Company Web Conferencing System," invoke:

```
imtctl> setProperty -system true -pname ReportEnvironmentName -pvalue "Company
Web Conferencing System"
```

5.6 Configuring Ports and Network Connectivity

This section discusses all issues and properties that pertain to configuring Real-Time Collaboration network connectivity. The following are explained:

- Setting up the global Web host when using an (LBR)
- Setting up the multiplexor (mx), the communication hub
- Setting up Oracle HTTP Server integration on the Real-Time Collaboration Core Components machine.
- Setting up Web Conferencing to work with proxy server(s) that are deployed in the corporate intranet.
- Special case of Web Conferencing and Oracle9*i*AS Web Cache on the same machine as the Real-Time Collaboration Core Components.
- LBR/Oracle9*i*AS Single Sign-On interaction

5.6.1 Integrating with LBR (Setting up Global Web Host)

The purpose of an LBR is to provide a single published address to the client browsers, and provide multiple Real-Time Collaboration Core Components mid-tiers which actually service the requests, based on the distribution of the requests done by the LBR. The LBR itself is a very fast network device which can distribute Web requests to a large number of physical servers. The term global Web host is used to refer to such an LBR.

Generally, you must set the Web Conferencing properties described below once for the entire system. However, in some scenarios, you might need to have a set of published addresses based on geographical considerations. Each of these addresses would have multiple Real-Time Collaboration Core Components mid-tier servicing the requests. In such a scenario, the properties need to be set at instance scope.

Use the following properties to set up the global Web host.

GlobalWebHost

Description: This is the name of the global Web host. For example, there could be multiple machines behind an LBR (imeeting1.oracle.com, imeeting2.oracle.com), but the global Web host name is imeeting.oracle.com. This is used to construct, among other things, the URL used to join a conference.

This name cannot be arbitrarily DNS registered and expected to work. There are implications related to the use of Oracle9*i*AS Single Sign-On. Please see later in this chapter for more details on this.

Default value: None

Valid values: host name

Scope: System, Instance.

Example: To set the global Web host to imeeting.company.com, invoke:

```
imtctl> setProperty -system true -pname GlobalWebHost -pvalue
"imeeting.company.com"
```

GlobalWebPort

Description: The HTTP port of the global Web host.

Default value: 80

Valid values: Any legal port value

Scope: System, Instance.

Example: If the global Web host is listening on port 7777 instead of 80 for HTTP requests, invoke:

```
imtctl> setProperty -system true -pname GlobalWebSecurePort -pvalue 7778
```

> **Note:** Setting the port to a value other than 80 may prevent access to users behind remote proxies.

GlobalWebSecurePort

Description: The HTTPS port of the global Web host.

Default value: 443

Valid values: Any legal port value

Scope: System, Instance.

Example: If the global Web host is listening on port 7778 instead of 443 for HTTPS requests, invoke:

```
imtctl> setProperty -system true -pname GlobalWebSecurePort -pvalue 7778
```

Note: Setting the port to a value other than 443 may prevent access to users behind remote proxies.

GlobalProtocolSecure

Description: Whether the GlobalWebHost is using SSL.

Default value: false

Valid values: true/false

Scope: System, Instance.

Example: If the global web host is using SSL, this property needs to be set to true and so, do the following

```
imtctl> setProperty -system true -pname GlobalProtocolSecure -pvalue true
```

5.6.2 Setting Up the Mx

Use the following properties to set up the mx.

SrvNumProcs

Description: This property sets the processes for each component. Setting this parameter for component type "clbsrvr" does not affect any other parameter. However, if you change this property for the mx component, you must make sure there are enough ports to match.

Default value: 1 for component type "mxcomm," 4 for component type "clbsrv"

Valid values: Depends on the amount of memory on the system.

Scope: Component.

See also: MxListenPort

Examples:

- To set the number of mx processes to 2 on the current instance, invoke:

  ```
  imtctl> setProperty -ct mxcomm -pname SrvNumProcs -pvalue 2
  ```

- To set the number of Web Conferencing Server processes to 8 on the current instance, invoke:

```
imtctl> setProperty -ct clbsvr -pname SrvNumProcs -pvalue 8
```

MxListenPort

Description: This property specifies a list of ports on which the processes in the component of type "mxcomm" listen. The mx is the communication hub to which all component processes on an instance connect, including the Web Conferencing Server processes. Through the Web Conferencing Console, users connect to the mx processes for conferences hosted by a Web Conferencing Server on an instance. The Voice Conversion Server and the Document Conversion Server processes servicing this instance then connect to the mx.

Default value: None (set during installation)

Valid values: 2400-2700

Scope: Component.

See also: SrvNumProcs

Examples:

- To set one mx communication process that listens on 2400 direct port, invoke:

  ```
  imtctl> setProperty -ct mxcomm -pname MxListenPort -pvalue "[\"2400\"]"
  ```

- To set two mx processes, one of which listens on 2400 direct port and the other one on 2401 on an instance, invoke:

  ```
  imtctl> setProperty -ct mxcomm -pname MxListenPort -pvalue
  "[\"2400\",\"2401\"]"
  ```

5.6.3 Setting Up Oracle HTTP Server Integration

5.6.3.1 Enabling the mod_imeeting

After installation, the Oracle Real-Time Collaboration communication subsystem is configured for direct socket connections. This configuration is typically appropriate for intranet users. Supporting Internet users requires a more advanced configuration that supports clients connecting from behind external proxy servers. Oracle Real-Time Collaboration supports integration with the Oracle HTTP Server to address these requirements. As a first step, mod_imeeting needs to be enabled.

1. To enable mod_imeeting:

Make the necessary configuration changes to the Oracle HTTP Server in which mod_imeeting is configured. mod_imeeting is included through the $ORACLE_HOME/imeeting/conf/mod_imeeting.conf file, which in turn gets included in $ORACLE_HOME/Apache/Apache/conf/oracle_apache.conf file for Oracle HTTP Server. Add the following include line for mod_imeeting.conf in oracle_apache.conf:

```
Include "$ORACLE_HOME/imeeting/conf/mod_imeeting.conf"
```

Replace $ORACLE_HOME with the actual Oracle home.

2. Update the Oracle HTTP Server configuration using DCM.

```
$ORACLE_HOME/dcm/bin/dcmctl -v updateConfig -ct ohs
```

3. Restart Oracle HTTP Server through DCM.

```
$ORACLE_HOME/dcm/bin/dcmctl -v restart -ct ohs
```

5.6.3.2 Setting Up the mod_imeeting/Mx Redirect

Set the following property.

MxRedirectPort

Description: Internet end-users connecting through the Web Conferencing Console typically connect to the Apache/mod_imeeting. Ultimately, the socket is handed off to the mx by mod_imeeting, and this property is used to facilitate the communication between mod_imeeting and the mx process on the instance. This parameter has platform-specific connotations:

UNIX: It is not a port, but a name. The socket is handed off using Domain Name Sockets.

NT: It is a port.

Linux: It is not a port, but a name.

Default value: None

Valid Values: Integer within the range 2400-2700

Scope: Component.

Examples:

- For one mx process (installer default), invoke:

```
imtctl> setProperty -ct mxcomm -pname MxRedirectPort -pvalue "[\"2420\"]"
```

- For two mx processes, invoke:

```
imtctl> setProperty -ct mxcomm -pname MxRedirectPort -pvalue
"[\"2420\",\"2421\"]"
```

Typically, these two examples are enough to support all modes of connections if Oracle9*i*AS Web Cache is not present on the Real-Time Collaboration Core Components machine, and Oracle HTTP Server is the listener. More details on configuration when Oracle9*i*AS Web Cache is present is described later in this chapter.

5.6.3.3 Synchronize Web Conferencing Properties with Oracle HTTP Server Settings

Change the following properties to synchronize with the Oracle HTTP Server settings.

ApacheWebHost

Description: The Real-Time Collaboration installer assumes that the actual host name configured for the Real-Time Collaboration Core Components machine is used by clients to access Oracle Web Conferencing resources. This might not be true in some deployments where DNS aliasing and other advanced network deployment scenarios might require customization of the names used to access Web Conferencing resources from external locations.

Default value: Actual host name derived at installation by the installer.

Valid values: Any host name

Scope: Instance

Example: If the actual host name is web10.us.oracle.com, but if the name to be used is "imeeting.company.com", invoke:

```
imtctl> setProperty -pname ApacheWebHost -pvalue "imeeting.company.com"
```

ApacheWebPort

Description: The port on which Oracle HTTP Server is listening for HTTP requests

Default value: 80

Valid values: Any legal port value

Scope: Instance

Example: If the Oracle HTTP Server on the instance is listening on port 7777 instead of 80 for HTTP requests, invoke:

```
imtctl> setProperty -pname ApacheWebPort -pvalue 7777
```

Note: Setting the port to a value other than 80 may prevent access to users coming behind remote proxies.

ApacheWebSecurePort

Description: The port on which Oracle HTTP Server is listening for HTTPS requests.

Default value: 443

Valid values: Any legal port value

Scope: Instance

Example: If the Oracle HTTP Server on the instance is listening on port 4443 instead of 443 for HTTPS requests, invoke:

```
imtctl> setProperty -pname ApacheWebSecurePort -pvalue 4443
```

Note: Setting this to something other than 443 may prevent access to users behind remote proxies.

ApacheProtocolSecure

Description: Whether the ApacheWebHost is using SSL.

Default value: false

Valid values: true/false

Scope: System

Example: If the Oracle HTTP Server Web host is using SSL, this property needs to be set to true and so, do the following

```
imtctl> setProperty -pname ApacheProtocolSecure -pvalue true
```

5.6.4 Setting Up a Proxy

The following properties configure proxy settings used by all Oracle Web Conferencing HTTP client code. This includes cobrowsing functionality and cross-instance HTTP-based diagnostics. This is important for firewall considerations.

UserAgentProxyHost

Description: This property sets the host name of the HTTP proxy.

Default value: None

Valid values: host name

Scope: Component

Examples:

- To set the proxy to www-proxy.company.com for the entire system, invoke:

```
imtctl> setProperty -system true -pname UserAgentProxyHost -pvalue
"www-proxy.company.com"
```

- To set the proxy to www-proxy.company.com for the current instance, invoke:

```
imtctl> setProperty -pname UserAgentProxyHost -pvalue
"www-proxy.company.com"
```

UserAgentProxySSLHost

Description: This property sets the host name of the HTTPS proxy

Default value: None

Valid values: valid host name

Scope: Component

Examples:

- To set the proxy to www-proxy.company.com for the entire system, invoke:

```
imtctl> setProperty -system true -pname UserAgentProxySSLHost -pvalue
"www-proxy.company.com"
```

- To set the proxy to www-proxy.company.com for the current instance, invoke:

```
imtctl> setProperty -pname UserAgentProxySSLHost -pvalue
"www-proxy.company.com"
```

UserAgentProxyPort

Description: This property sets the proxy port.

Default value: 80

Valid values: Any port

Scope: Component

Examples:

- To set the proxy port to 8080 for the entire system, invoke:

```
imtctl> setProperty -system true -pname UserAgentProxyPort -pvalue 8080
```

- To set the proxy port for the current instance, invoke:

```
imtctl> setProperty -pname UserAgentProxyPort -pvalue 8080
```

UserAgentProxySSLPort

Description: Set this to the SSL proxy port

Default value: 443

Valid values: Any port

Scope: Component

Examples:

- To set the SSL proxy port to 4443 for the entire system, invoke:

```
imtctl> setProperty -system true -pname UserAgentProxySSLPort -pvalue 4443
```

- To set the SSL proxy port for the current instance, invoke:

```
imtctl> setProperty -pname UserAgentProxySSLPort -pvalue 4443
```

UserAgentProxyExclusions

Description: This property sets a list of URLs for domains/hosts that should *not* be proxied.

Default value: None

Valid values: (simple tail-matching algorithm). The format of the value must be "[\"domain1\", \"domain2\"]"

Scope: System, Instance. The property can be set at system scope but can be overridden for a specific component.

Examples:

- To set the exclusions for the whole system for domains company.com and companycorp.com, invoke:

```
imtctl> setProperty -system true -pname UserAgentProxyExclusions -pvalue
"[\".company.com\", \".companycorp.com\"]"
```

- To set the exclusions for the current instance for domains company.com and companycorp.com, invoke:

```
imtctl> setProperty -pname UserAgentProxyExclusions -pvalue
"[\".company.com\", \".companycorp.com\"]"
```

UserAgentProxyEnabled

Description: This property enables the proxy settings. None of the other proxy settings take effect unless this is set to true.

Default value: true

Valid values: true/false

Scope: Component

Example: To enable the proxy setting, invoke:

```
imtctl> setProperty -system true -pname UserAgentProxyEnabled -pvalue true
```

5.6.5 Web Conferencing and Oracle9*i*AS Web Cache

5.6.5.1 Second IP Address

If Oracle9*i*AS Web Cache is present on the Real-Time Collaboration Core Components machine and if the listener is on the port 443, a second IP address is required for HTTPS tunneling. In addition, Oracle HTTP Server (the virtual host) must be listening on this address on port 443. To configure Oracle Web Conferencing for a second IP address:

1. Create a virtual host in $ORACLE_HOME/Apache/Apache/conf/httpd.conf to listen on port 443 to serve incoming requests for the second IP address of the machine.

```
<VirtualHost second-ip-address-for-the-machine:443>
</VirtualHost>
```

2. Update the Oracle HTTP Server configuration using DCM.

```
$ORACLE_HOME/dcm/bin/dcmctl -v updateConfig -ct ohs
```

3. Restart Oracle HTTP Server through DCM.

```
$ORACLE_HOME/dcm/bin/dcmctl -v restart -ct ohs
```

5.6.5.2 Configure Web Conferencing to Use a Second IP Address

If Oracle9*i*AS Web Cache is present, you will need a second IP address. If Oracle9*i*AS Web Cache is disabled, a second IP address is not needed. Configure Oracle Web Conferencing to use a second IP address by setting the properties in this section.

ApacheTunnelHost

Description: The second IP address for which a virtual host has been configured in httpd.conf

Default value: None

Valid values: An IP address

Scope: Instance

Example: If the second IP address is 145.35.28.4, invoke:

```
imtctl> setProperty -pname ApacheTunnelHost -pvalue 145.35.28.4
```

ApacheTunnelPort

Description: This is the tunnel port. It should be 443. Having a value other than 443 creates problems, such as preventing clients from other intranets that attempt to connect through their proxy across the Internet.

Default value: 443

Valid values: A port number

Scope: Instance

See also: ApacheTunnelHost

Example: This is the default setting that sets the tunnel port to 443.

```
imtctl> setProperty -pname ApacheTunnelPort -pvalue 443
```

5.6.5.3 Reconfigure Oracle9*i*AS Web Cache

Configure Oracle9*i*AS Web Cache to only bind to the primary IP address on port 443 if it is bound to all network interfaces.

> **See Also:** *Oracle9i Application Server Administrator's Guide* for more information

5.6.6 Load Balancer/Oracle9*i*AS Single Sign-On Configuration

For advanced configuration information such as this, please go to http://otn.oracle.com/products/webconferencing/.

5.7 Configuring SSL

Oracle Real-Time Collaboration provides complete conference runtime security, as well as application security by providing complete support for HTTPS. Before learning how to configure the Real-Time Collaboration system for HTTPS use, it is useful to understand the user and administrator experiences.

5.7.1 User's Perspective

In the Oracle Web Conferencing Application, users can choose HTTP or HTTPS, if both options are allowed by the administrator. Using controls provided in Oracle9*i*AS, an administrator can mandate use of SSL for some or all URLs in the application. Therefore, even if a user goes to a non-secure URL, the user could be redirected to a secure URL.

When logging in to Oracle9*i*AS Single Sign-On, most users expect to use HTTPS.

In addition, all the Web Conferencing properties that need to be in sync with Oracle HTTP Server settings and the LBR setting to be set up, as well. The following properties need to be set and please look at the relevant sections for more details on those properties:

- GlobalWebSecurePort

- GlobalProtocolSecure
- ApacheWebSecurePort
- ApacheProtocolSecure

In addition, update the Oracle HTTP Server as follows:

1. Append imt_mod_osso.conf in imeeting/conf to mod_osso.conf and then do the following
2. Update the Oracle HTTP Server configuration using DCM.

```
$ORACLE_HOME/dcm/bin/dcmctl -v updateConfig -ct ohs
```

3. Restart Oracle HTTP Server through DCM.

```
$ORACLE_HOME/dcm/bin/dcmctl -v restart -ct ohs
```

During a Conference

In Preferences, a user can mandate use of the secure mode for all attendees in the conferences that the user creates. Thus, attendees are forced to use the secure mode, based on the host's preferences.

5.7.2 Administrator's Perspective

5.7.2.1 Application Setup

An administrator can mandate that some or all pages use HTTPS, based on Oracle9*i*AS controls.

Oracle Corporation strongly recommends that an administrator make sure that the Oracle9*i*AS Single Sign-On access during a user login is done using HTTPS.

5.7.2.2 Conference Runtime

By default, SSL is disabled for the entire system. However, an administrator can enable the use of SSL for the entire system. Enabling SSL does *not* mean that it is mandated; it means that the conference runtime SSL is available.

In addition, an administrator can mandate use of SSL for all conferences in the entire system or for a specific site only.

GlobalMeetingSSLSupportEnabled

Description: Set this property to enable the conference runtime SSL for the entire Real-Time Collaboration system.

Default value: false

Valid values: false/true

Scope: System

See also: MxWalletLocation

Example:

```
imtctl> setProperty -system true -pname GlobalMeetingSSLSupportEnabled pvalue
"true"
```

MxWalletLocation

Description: The mx uses the same wallet that the Oracle HTTP Server is using on the Real-Time Collaboration Core Components machine. The only requirement is that the wallet must be in "Auto Login mode." Use the Oracle Wallet Manager to change the wallet to "Auto Login mode." This property will then need to set to the location of the wallet.

> **See Also:** *Oracle9i Application Server Administrator's Guide* for more information on how to use Oracle Wallet Manager

Default value: Default value is unset, in which case the location "$ORACLE_HOME/Apache/Apache/conf/ssl.wlt/default" is used.

Valid values: Full path of the wallet

Scope: Instance

See also: GlobalMeetingSSLSupportEnabled

Example:

```
imtctl> setProperty -pname MxWalletLocation -pvalue
"/902ias/Apache/Apache/conf/ssl.wlt/default"
```

SecureReqForMeetings

Description: This parameter allows an administrator to mandate or choose HTTPS as the default option for all conferences.

Default value: false

Valid values: false/true

Scope: System/Site

Examples:

- An administrator can require that all conferences use HTTPS by default, but allow sites to override this setting by using the following command:

```
imtctl> setProperty -system true -pname SecureReqForMeetings -pvalue true
-force true
```

- An administrator can require that all conferences use HTTPS by default, but allow sites to override this setting by using the following command

```
imtctl> setProperty -system true -pname SecureReqForMeetings -pvalue true
```

- An administrator can mandate use of HTTPS for all the conferences in a particular Oracle Web Conferencing site with the following command:

```
imtctl> setProperty -siteId <site-id> -pname SecureReqForMeetings -pvalue
true -force true
```

5.7.3 Cobrowsing SSL Web Sites

You can use Oracle Web Conferencing to cobrowse SSL Web sites. No additional Web Conferencing configuration is necessary if the Web sites have Web server certificates from the following certificate authorities: RSA, GTE CyberTrust, Baltimore Technologies, and Entrust.

This set is not comprehensive. If you cobrowse a Web site that uses a certificate from a certificate authority not included in the default set, the Web Conferencing Console displays an error message such as:

```
IMT_SRV_CM_SSL_HANDSHAKE_FAILED.
The certificate submitted by the site has either expired or is not trusted
by the iMeeting server.Please request your iMeeting server administrator to
update the Trusted Signer Certificates Database with the certificate for the
site you are visiting.
[https://www.secureWeb site.com/uri/]
...
```

If you get this error, add Trusted Root Certificate Authority certificates to the Real-Time Collaboration Core Components machine as described in Section 5.7.3.1, "Adding Trusted Root Certificate Authority Certificates".

5.7.3.1 Adding Trusted Root Certificate Authority Certificates

SSL communications involve, among other things, digital certificates. Certificates are issued by a third party, called a Certificate Authority (CA). When you use your browser to visit an SSL-secured Web site, your browser validates that the Web site is who it claims to be by verifying the Web site's certificate. Your browser comes bundled with a set of Trusted Root Certificate Authority Certificates, which it uses to validate any certificate a Web site presents. Occasionally, your browser will prompt you to determine whether you want to proceed, and also optionally installs a certificate when a Web site presents a new certificate signed by a CA.

The Web Conferencing Server that browses Web sites on behalf of conference attendees, comes with a default set of Trusted Root Certificate Authority certificates. When an attendee cobrowses a Web site that deploys a certificate from a CA vendor not included in the default set, you need to add the Trusted Root Certificate to the Real-Time Collaboration Core Components configuration. Follow the steps in this section to do this.

You will need to use Internet Explorer 5.5 or later to install the CA root certificate into the browser and export it as a Base64 encoded X.509 (.CER) file. You cannot use Netscape, because it does not allow the export of the root certificates to a file.

1. From an Internet Explorer browser that is not using Oracle Web Conferencing, go to the SSL-secured Web site that gave an error through Web Conferencing.
2. Using your browser, access secured areas of the Web site until you see the yellow lock in your browser status bar at the bottom.
3. Click the yellow lock, and then click the Certification Path tab in the pop-up window.

4. Select all the certificates one by one from the certificate chain. Click the General tab to read the description of the selected certificate. It contains the name of the issuer, its validity period, and so on.
5. If you are comfortable trusting the issuer (CA), then click the Details tab.
6. Click Copy To File and then click Next on the following screen.
7. In the subsequent screen, choose the format you want to export in as Base64 encoded X.509 (.CER) and click Next.
8. Enter a file name in which you want to store the certificate, for example, cavendor.cer, and save the CA certificate. Repeat steps 4 to 8 for all the certificates in the chain.
9. Append the one or more files saved in step 8 to the certdb.txt file specified by the Real-Time Collaboration Core Components property TRUSTDBFILE. This file is typically located in $ORACLE_HOME/imeeting/conf. Each of the files saved in step 8 contains a certificate in the format:

```
-----BEGIN CERTIFICATE-----
MIICYzCCAdACEAuZ4ibKgW0O66lustIp2TowDQYJKoZIhvcNAQEEBQAwXzELMAkG
A1UEBhMCVVMxIDAeBgNVBAoTF1JTQSBEYXRhIFNlY3VyaXR5LCBJbmMuMS4wLAYD
VQQLEyVTZWN1cmUgU2VydmVyIENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTAw
...
-----END CERTIFICATE-----
```

5.8 Configuring Logging

When you configure logging, you can configure log levels, as well as the location of the logs. See the properties in this section for more information.

LogLevel

Description: Log level

Default value: CONFIG

Valid values: NONE, SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST, ALL

Scope: System, Instance and Component. The property can be set at a system scope, can be overridden for a specific instance and in turn can be overridden for a specific component in that instance.

Examples:

An administrator wants to set the system-wide log level to SEVERE and then for a specific instance, the administrator wants to change it to WARNING and then for the Web Conferencing Servers in that instance, wants to change the log level to FINE.

- To set the system-wide log-level to SEVERE, invoke:

```
imtctl> setProperty -system true -pname LogLevel -pvalue SEVERE
```

- To set the instance-wide log-level for the current instance to WARNING, invoke:

```
imtctl> setProperty -pname LogLevel -pvalue WARNING
```

- To set the log level for Web Conferencing Servers only to FINE in this instance, invoke:

```
imtctl> setProperty -ct clbsvr -pname LogLevel -pvalue FINE
```

LogPath

Description: The log path relative to IMT_HOME. The filename must be specified using forward slashes '/', even on the Microsoft Windows platform.

Default value: logs (Because this is relative to IMT_HOME, it becomes IMT_HOME/logs)

Valid values: any directory. The directory name must be specified using forward slashes (/), even on the Microsoft Windows platform.

Scope: Component level property and can be set at System, Instance, or Component scope.

Example: To set the log path to directory IMT_HOME/imtlogfiles/, invoke:

```
imtctl> setProperty -system true -pname LogPath -pvalue "imtlogfiles'
```

LogFlushInterval

Description: System-wide log flush interval in the system-wide log flush interval, duration types must be specified in some combination of days, hours, minutes, seconds, and milliseconds. Valid abbreviations include "d", "hr", "min", "sec", "millis". Values must be in quotation marks due to the spaces.

Default value: 10 sec

Scope: Component level property and can be set at System, Instance, or Component scope.

Example: To set the log flush interval to one second, invoke:

```
imtctl> setProperty -pname LogFlushInterval -pvalue "1 sec"
```

5.9 Configuring E-mail

The Oracle Web Conferencing Application can generate outbound e-mail for various purposes, such as inviting users to a conference. To enable this feature, you must configure the Real-Time Collaboration system with certain e-mail-related properties.

SmtpHost

Description: This is the host name of the SMTP server Oracle Web Conferencing should use.

Default value: None

Valid values: a host name

Scope: System and Instance. The property can be set at system scope but can be overridden for a specific instance.

Examples:

- To set SmtpHost for the whole system to mail-net.company.com, invoke:

  ```
  imtctl> setProperty -system true -pname SmtpHost -pvalue
  "mail-net.company.com"
  ```

- To set SmtpHost for the current instance to mail-net.company.com, invoke:

  ```
  imtctl> setProperty -pname SmtpHost -pvalue "mail-net.company.com"
  ```

SmtpPort

Description: This is the listen port of the SMTP server.

Default value: 25

Scope: System and Instance. The property can be set at system scope but can be overridden for a specific instance.

Examples:

- To set SmtpPort for the whole system to 3000, invoke:

```
imtctl> setProperty -system true -pname SmtpPort -pvalue 3000
```

- To set SmtpPort for the current instance to 3000, invoke:

```
imtctl> setProperty -pname SmtpPort -pvalue 3000
```

5.10 Configuring Clusters

The clusters are created by assigning the same value to the location attribute of multiple instances. Thus, if two instances A and B have the same value for the location, they form a cluster.

InstanceLocation

Description: This property sets the location of an instance. By default, all instances are in the location "default." All core services (Web Conferencing Servers) in a particular location are considered for load balancing.

Default value: Default

Valid values: Any string

Scope: Instance

Example: To set the location for an instance to "US," invoke:

```
imtctl> setProperty -pname InstanceLocation -pvalue "US"
```

Instances containing the Voice Conversion Server and Document Conversion Server can also be clustered by making them available as "services" to one set (typical case) of Real-Time Collaboration Core Components and not necessarily to all sets in a geographically distributed enterprise by having their InstanceLocation property set to the same value as instances with the Real-Time Collaboration Core Components.

Although not recommended, if or any reason, an instance containing voice or doc conversion server needs to provide service to multiple locations, use the AdditionalLocationsServed property in this chapter.

AdditionalLocationsServed

Description: This property sets a list of locations to which these Voice and Document Conversion Server instances will provide service.

Default value: default

Valid values: The value should be in the format: "[\"location1\", \"location2\"]"

Scope: Instance. The property can be set at instance scope.

Example: To have the Document and Voice Conversion Server instance provide service to instances with locations imeeting-hq and imeeting-uk, invoke:

```
imtctl> setProperty -pname AdditionalLocationsServed -pvalue "[\"imeeting-hq\",
\"imeeting-uk\"]"
```

5.11 Configuring Conference Runtime Characteristics

You can configure some characteristics of a conference session using the Meeting Properties. These properties are used when a conference is created from an integrating site using the Oracle Web Conferencing Integration Service. Similar properties can also be set from the Preferences global button available on each Application page.

MeetingStartupMode

Description: Set the default startup mode of the Web Conferencing Console when it loads.

Default value: W

Valid values: W (Cobrowse), D (Document Sharing), B (Whiteboard), C (Desktop Sharing)

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingStartupMode -pvalue D
```

MeetingAutoHide

Description: Whether the console should be automatically hidden at start-up.

Default value: false

Valid values: false, true

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingAutoHide -pvalue true
```

MeetingExpandHelpText

Description: Whether to expand the help text in console.

Default value: false

Valid values: false, true

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingExpandHelpText -pvalue true
```

MeetingExpandSecondRow

Description: Whether to expand the second row in console.

Default value: false

Valid values: false, true

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingExpandSecondRow -pvalue true
```

MeetingAllUseTools

Description: All attendees can use annotation tools.

Default value: true

Valid values: false, true

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingAllUseTools -pvalue true
```

MeetingAllControlLayout

Description: All attendees control console layout.

Default value: true

Valid values: false, true

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingAllControlLayout -pvalue true
```

MeetingAllViewAttendeeList

Description: All attendees view attendee list.

Default value: true

Valid values: false, true

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingAllViewAttendeeList -pvalue true
```

MeetingAllViewStatusMsg

Description: Can all attendees view status messages.

Default value: true

Valid values: false, true

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingAllViewStatusMsg -pvalue true
```

MeetingChatType

Description: Chat type

Default value: PUBLIC_HOST (public chat)

Valid values: ALL, NONE, HOST, PUBLIC_HOST

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingChatType -pvalue PUBLIC_HOST
```

MeetingChatInterface

Description: Select docked chat interface or pop-up interface.

Default value: P (pop-up)

Valid values: P, D (docked)

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingChatInterface -pvalue P
```

MeetingSharedControlUseTools

Description: Attendees with shared control can use annotation tools.

Default value: true

Valid values: true, false

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingSharedControlUseTools -pvalue true
```

MeetingSharedControlSendPoll

Description: Attendees with shared control use can send polls.

Default value: false

Valid values: true, false

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingSharedControlSendPoll -pvalue true
```

MeetingStartUrl

Description: Start URL when the conference starts.

Default value: none

Valid values: Any valid URL

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingStartUrl -pvalue
http://www.company.com
```

MeetingEndUrl

Description: End URL when the conference finishes.

Default value: none

Valid values: Any valid URL

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingEndUrl -pvalue
http://www.company.com
```

MeetingIsInstant

Description: If the conference is an instant conference.

Default value: false

Valid values: true, false

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingIsInstant -pvalue false
```

MeetingDurationMinutes

Description: Suggested conference duration in minutes, as shown in the scheduling user interface.

Default value: 60

Valid values: any valid integer

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingDurationMinutes -pvalue 30
```

MeetingTypeOfMtg

Description: Conference type option in the scheduling user interface.

Default value: REGULAR

Valid values: REGULAR, PUBLIC, RESTRICTED

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingTypeOfMtg -pvalue PUBLIC
```

MeetingDialInAndConfId

Description: Dial-in information for the conference.

Default value: none

Valid values: Any text

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingDialInAndConfId -pvalue "dial in
info"
```

MeetingInformation

Description: Other miscellaneous information related to the conference.

Default value: none

Valid values: Any text

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingInformation -pvalue "misc
information"
```

MeetingPublishOption

Description: Allows the host to publish the attendee list.

Default value: true

Valid values: true, false

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingPublishOption -pvalue false
```

MeetingEmailJoin

Description: Allows attendees to join the conference directly from the e-mail link.

Default value: true

Valid values: true, false

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingEmailJoin -pvalue false
```

MeetingShouldSendEmail

Description: Allows the host to notify attendees by e-mail.

Default value: true

Valid values: true, false

Scope: Site

Example:

```
imtctl> setProperty -siteId 101 -pname MeetingShouldSendEmail -pvalue false
```

EarlyJoinMinutes

Description: This property sets the number of minutes before a conference's scheduled start time during which users can join.

Default value: 30

Scope: System

Example: To configure this value to 15 minutes, invoke:

```
imtctl> setProperty -system true -pname EarlyJoinMinutes -pvalue 15
```

5.12 Configuring Monitoring

By default, Oracle Web Conferencing provides high availability of its own components using process monitoring of all its processes, including automatic restart of processes when they are down. In addition, Web Conferencing provides interfaces for HTTP-based Web site monitoring for monitoring service availability. This can be easily integrated into any monitoring infrastructure. See Chapter 7, "Monitoring" for details.

At a minimum, you should monitor the following on Real-Time Collaboration Core Components:

- Conference service availability test—checks if the machine with Real-Time Collaboration Core Components is capable of hosting a conference.
- Voice Conversion Server availability—checks if the machine with the Real-Time Collaboration Core Components can provide live voice streaming during a conference.
- Document Conversion Server availability—checks if this Core Components machine can successfully handle document conversion requests.

See Section 7.6, "Real-Time Collaboration Interfaces" for more information.

For example, if the instance monitoring URL is imeeting1.company.com, do the following:

- For conference service availability monitoring:

  http://imeeting1.company.com/imtapp/servlet/ImtTestServlet?mtgtest=true

- For Voice Conversion Server availability monitoring:

  http://imeeting1.company.com/imtapp/servlet/ImtTestServlet?voiceconvtest=true

- For Document Conversion Server availability monitoring:

  http://imeeting1.company.com/imtapp/servlet/ImtTestServlet?docconvtest=true

See Section 7.6, "Real-Time Collaboration Interfaces" for more information.

5.13 Configuring User Privileges

Use the imtctl command line interface to assign roles to Web Conferencing users. Example:

```
imtctl> modifyRole -username "anyusername" -rolename "businessadmin"
```

See Section 2.9, "User Management" for details.

5.14 Configuring Voice Streaming

Set the VoiceDialInPrefix property. Then, use the imtctl command line utility to manage system dial-ins.

VoiceDialInPrefix

Description: This property sets the dial-in prefix for an instance of the Voice Conversion Server. This prefix would be added to all dial-in numbers received by the Voice Conversion Server before dialing out. Typically, corporations have a dial-in prefix ("9," for example) for dialing numbers outside the company's internal phone network.

Default Value: None

Valid Value: Any valid dialing sequence. In most cases, it would be a number example: "9" A valid dialing sequence contains numbers 0-9 and any of the following characters: * # ,

Scope: Instance

To configure the dial-in prefix for a Voice Conversion Server instance to "9" invoke:

```
imtctl> setProperty -i <instance-name> -pname "VoiceDialinPrefix" -pvalue "9"
```

5.14.1 System Dial-Ins

System dial-ins are system-wide dial-ins created by an administrator, which are available to all Oracle Web Conferencing users. Typically, system administrators create system dial-ins for phone conference vendors that are used commonly by the users of the system. System dial-ins serve as templates for users to create their own dial-ins; however, users cannot edit them. In many cases, system dial-ins are incomplete dialing sequences in that they contain information that is common to all users and indicate the information a user must add to make it a usable dial-in. Users

should make the recommended addition to the system dial-in and save it as a user-defined dial-in for use in a conference.

For example, phone conference vendors typically provide a PIN or conference ID to its users. A system administrator would create a system wide dial-in for the particular conference vendor that would contain the toll-free number for the vendor, followed by a placeholder for the PIN/Conference ID and the required separators like ',', '#' or '*'. Users would then modify the system dial-in by adding their PIN/Conference ID and then save it under a different name.

Table 5–1 Dial-Ins

Dial-in Name	Dial-in Sequence
Audio Conferencing Vendor (US)	18005555555,<ADD CONF ID HERE>#,,,,,,,,,#,,,,,,,,#
Joe's account with Audio Conferencing Vendor	18005555555,8282828#,,,,,,,,, #,,,,,,,,#

System dial-ins can be managed through the Oracle Real-Time Collaboration command line utility, imtctl.

5.14.1.1 Creating a New System Dial-In

Invoke the following:

```
imtctl> addSysDialin -name <dialin-name> -sequence <dialin-sequence> -default
<true/false>
```

The default indicates whether the system dial-in should be made the default for all users. Users can override the default, from the Web Conferencing Console, if they choose.

5.14.1.2 Deleting a System Dial-In

Invoke the following:

```
imtctl> deleteSysDialin -name <dialin-name> -id <dialin-id>
```

The name or ID is required. If both are specified, name is used. The name or ID can be obtained by using `getSysDialins`.

6

Sample Deployments

This chapter discusses some sample deployment scenarios, ranging from a basic deployment to a sophisticated one spanning multiple geographical regions. It also explains what an administrator should do to set them up. The best way to review this chapter is to go through each example, because each example builds on the previous.

> **Note:** In Chapter 4, "Post-Installation" Oracle Web Conferencing properties are set using .imt files. The examples use the setProperty command in imtctl (the command used by the .imt files) to set Web Conferencing properties. It is important to remember that when the option "-system true" is used in setProperty, the setting of the property has a global scope and affects not just the instance on which imtctl is being invoked. If the `-system true` option is not used in the command, the command applies to that instance.

See Chapter 4, "Post-Installation" for basic information about the following:

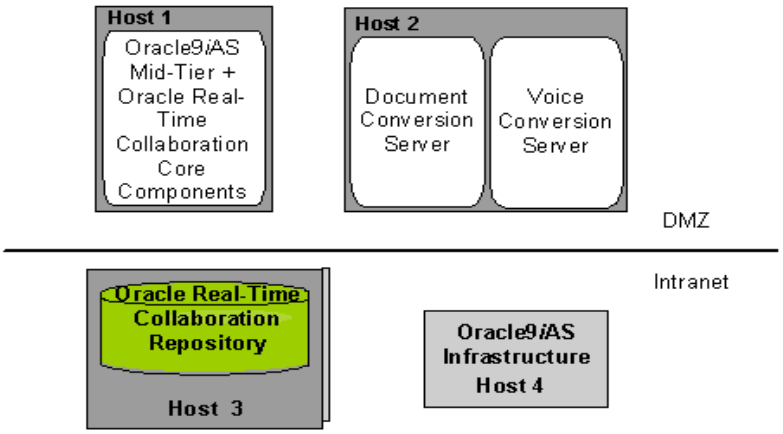
- Checklist of post-installation tasks for a Web Conferencing Server installation.
- Checklist of post-installation tasks for installations of the Document and Voice Conversion Servers.

For all examples, one Oracle9*i*AS instance is on the machine with the Real-Time Collaboration Core Components and only one Real-Time Collaboration instance is on this machine. For these examples, it is assumed that Oracle9*i*AS Web Cache is disabled on the mid-tier.

6.1 Basic Deployment

This is the simplest deployment and is generally used to create a pilot deployment. In this deployment, users can access Oracle Web Conferencing from the Internet, through their proxy across the Internet, and through the intranet.

Figure 6–1 Web Conferencing Basic Deployment



In Figure 6–1, "Web Conferencing Basic Deployment", Host 1 and Host 2 are deployed in the same DMZ, while Host 3 and Host 4 are deployed in the corporate intranet.

These are the mandatory associations that must be made for the various parts of Oracle Web Conferencing to work:

- Real-Time Collaboration Core Components to Real-Time Collaboration Repository. This association is made during installation.
- Real-Time Collaboration Core Components to Oracle9*i*AS Infrastructure. This association is made during installation.
- Document and Voice Conversion Servers to Real-Time Collaboration Repository. This association is made during installation.

Note: For these examples, the Voice Conversion Server and the Document Conversion Server are on the same machine.

- Document and Voice Conversion Servers to Real-Time Collaboration Core Components. This is a mandatory post-installation step.

Additional configuration issues are discussed in the rest of the chapter.

6.1.1 Sequence of Steps

1. Install all prerequisites on all the computers.
2. Install the Real-Time Collaboration Core Components on Host 1.
3. Perform post-installation steps on Host 1.
4. Perform verification tests on Host 1.
5. Install Document and Voice Conversion Servers on Host 2.
6. Follow the post-installation steps on Host 2.
7. Perform verification tests on Host 2.
8. Perform verification tests on Host 1 to confirm proper connection to Host 2.

6.1.2 Prerequisites

Host 1: Oracle9*i*AS plus Real-Time Collaboration Core Components set up and configured to use the Oracle9*i*AS Infrastructure on Host 4.

Host 4: Oracle9*i*AS Infrastructure needs to be set up on this machine.

Host 2: A Windows NT computer with specialized telephony hardware and software.

Host 3: Oracle9*i* Database (9.2.0.1) for the Real-Time Collaboration Repository.

See Also: *Oracle Collaboration Suite Installation and Configuration Guide for Solaris* for additional information on the prerequisites

6.1.3 Settings

Configure Oracle Web Conferencing to leverage existing enterprise e-mail and proxy servers for its functionality. Assume that the settings are as follows:

Table 6–1 System Settings

Setting Name	Value	Comments	Web Conferencing Property
Existing e-mail server host name (one for the enterprise)	"mail-net.company.com"	Post-installation step	SmtpHost
E-mail server listening port	25	Default	SmtpPort
Existing proxy server host name for HTTP (one for the enterprise)	"www-proxy.company.com"	Post-installation step	UserAgentProxyHost
Proxy server host name for HTTPS (one for the enterprise)	"www-proxy.company.com"	Post-installation step	UserAgentProxySSLHost
Proxy port for HTTP	80	Default	UserAgentProxyPort
Proxy port for HTTPS	443	Default	UserAgentProxySSLPort
Proxy domains to exclude	".company.com"	Post-installation step	UserAgentProxyExclusions

Table 6–2 Instance on Host 1

Setting Name	Value	Comments	Web Conferencing Property
Number of mx	1	Default	SrvNumProcs
Listening port for the single mx	2400	Set during installation	MxListenPort
Actual host name of Host 1	"host1.company.com"		N/A
Alias for Host 1	"imeeting1.company.com"	Post-installation step	ApacheWebHost
Instance name	"instance1.company.com"	Set automatically during installation	N/A
Oracle HTTP Server listening port for HTTP on Host 1	7777	Default. Set during installation.	ApacheWebPort
Oracle HTTP Server listening port for HTTPS on Host 1	443	Default	ApacheWebSecurePort

You must configure the settings in Table 6–1 and Table 6–2 every time you install Web Conferencing.

In addition, you must configure Host 2 as follows:

Table 6–3 Host 2 Settings

Setting Name	Value	Comments	Web Conferencing Property
HTTP listening port for the Voice Conversion Server	2460	Set during installation	
Dial-in prefix for voice conversion server instance	9	Post-installation step	VoiceDialinPrefix

6.1.4 Installation

On Host 1, install the Real-Time Collaboration Core Components, and provide the required port numbers. On Host 2, install the Document and Voice Conversion Servers, and provide the required port numbers.

> **See Also:** Oracle Collaboration Suite Installation and Configuration Guide for Solaris for details

6.1.5 Post-Installation

Invoke the imtctl commands in this section, while setting Web Conferencing properties. The commands in this section are based on the data in the Settings tables in Section 6.1.3, "Settings".

Step 1 Integrate Web Conferencing with existing e-mail.

```
imtctl> setProperty -system true -pname SmtpHost -pvalue "mail-net.company.com"
```

Because the port is 25 (which is default), there is no need to set the port.

Step 2 Integrate Web Conferencing to use existing proxy servers.

```
imtctl> setProperty -system true -pname UserAgentProxyHost -pvalue
"www-proxy.company.com"
imtctl> setProperty -system true -pname UserAgentProxySSLHost -pvalue
"www-proxy.company.com"
imtctl> setProperty -system true -pname UserAgentProxyExclusions -pvalue
"[\".company.com\"]"
```

> **Note:** Because the default proxy port for HTTP is 80 and HTTPS is 443, there is no need to set these ports.

Step 3 Integrate Web Conferencing with Oracle HTTP Server on this Real-Time Collaboration Core Components machine.

1. Enable the mod_imeeting.

   Make the necessary configuration changes to the Oracle HTTP Server in which mod_imeeting is configured. mod_imeeting is included through the $ORACLE_HOME/imeeting/conf/mod_imeeting.conf file, which in turn gets included in $ORACLE_HOME/Apache/Apache/conf/oracle_apache.conf file for Oracle HTTP Server.

   Add the following include line for mod_imeeting.conf in oracle_apache.conf:

   ```
   Include "$ORACLE_HOME/imeeting/conf/mod_imeeting.conf"
   ```

   Replace $ORACLE_HOME with the actual Oracle home.

2. Update the Oracle HTTP Server configuration using DCM.

   ```
   $ORACLE_HOME/dcm/bin/dcmctl -v updateConfig -ct ohs
   ```

3. Restart Oracle HTTP Server through DCM.

   ```
   $ORACLE_HOME/dcm/bin/dcmctl -v restart -ct ohs
   ```

4. Set up mod_imeeting/mx redirect.

   For one mx process (installer default), invoke:

   ```
   imtctl> setProperty -ct mxcomm -pname MxRedirectPort -pvalue "[\"2420\"]"
   ```

5. Make Web Conferencing aware of basic Oracle HTTP Server settings by invoking:

   ```
   imtctl> setProperty -pname ApacheWebHost -pvalue "imeeting1.company.com"
   ```

   Because 80 and 443 are defaults, no additional steps are required for setting the ports.

   Note: imeeting1.company.com is the URL users will use to access Web conferences.

Step 4 Set up reports.

Web Conferencing reports can be generated and sent via e-mail using the imtreport script. The script is located at $ORACLE_HOME/imeeting/bin and is invoked as "imtreport" on all platforms. The actual script file is "imtreport" for UNIX and Linux

and "imtreport.cmd" for Microsoft Windows. Mandatory report options such as the recipient list are set by editing variables at the top of the report script. The script contains comments on the options provided. Be sure to edit the proper script for the platform you are using. You may want to send out automated weekly reports by using some operating system provided mechanism for invoking the script on a weekly basis (for example, cron job on UNIX or the Task Scheduler on Windows).

Step 5 Set up for periodic monitoring.

Set up your monitoring infrastructure to periodically ping the following URLs:

- For conference service availability monitoring:

  http://imeeting1.company.com/imtapp/servlet/ImtTestServlet?mtgtest=true

- For Voice Conversion Server availability monitoring:

  http://imeeting1.company.com/imtapp/servlet/ImtTestServlet?voiceconvtest=true

- For Document Conversion Server availability monitoring:

  http://imeeting1.company.com/imtapp/servlet/ImtTestServlet?docconvtest=true

See Section 7.6, "Real-Time Collaboration Interfaces" for more information.

Host 2

No post-installation configuration is required.

6.1.6 Verification

Host 1

Invoke `imtctl> runTests`. This runs the full verification test.

Typically, all tests succeed. However, if this command is run before Host 2 is set up, the Voice Conversion Server test and the Document Conversion Server test will fail.

Run this test as described in Section 6.1.1, "Sequence of Steps".

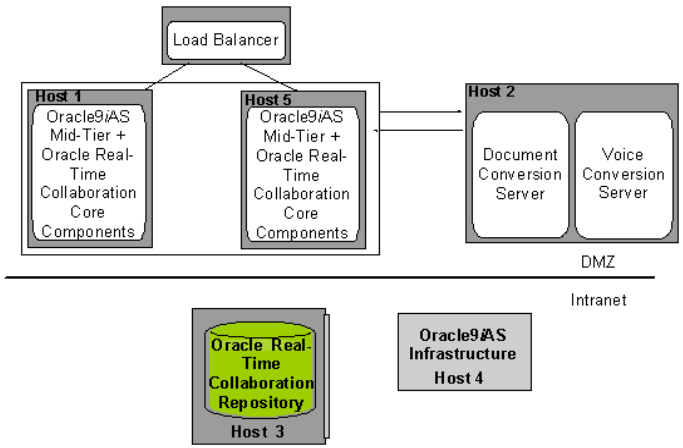
Host 2

Invoke `imtctl> getState` to verify that the processes are up, but the real test needs to be invoked on Host 1 (`imtctl> runTests`) to make sure that the document and voice conversion services provided by Host 2 are available.

6.2 Multiple Real-Time Collaboration Core Components and Load Balancer

In this deployment, Host 1, Host 2, and Host 5 are deployed in the DMZ, while Host 3 and Host 4 are deployed in the corporate intranet.

Figure 6–2 Multiple Real-Time Collaboration Core Components + LBR



6.2.1 Sequence of Steps

1. Install all prerequisites on all the computers, including the load balancer (LBR).
2. Install Real-Time Collaboration Core Components on Host 1.
3. Perform post-installation steps on Host 1.
4. Perform verification tests on Host 1.
5. Install Document Conversion Server and Voice Conversion Server on Host 2.
6. Perform post-installation steps on Host 2.
7. Perform verification tests on Host 2.
8. Perform verification tests on Host 1 to confirm proper connection to Host 2.
9. Install Real-Time Collaboration Core Components on Host 5.
10. Perform post-installation steps on Host 5.

11. Perform verification tests on Host 5.

6.2.2 Prerequisites

Host 1: Oracle9*i*AS mid-tier set up and configured to use the Oracle9*i*AS Infrastructure on Host 4

Host 5: Oracle9*i*AS mid-tier set up and configured to use the Oracle9*i*AS Infrastructure on Host 4

Host 4: Oracle9*i*AS Infrastructure needs to be set up on this machine.

Host 2: A Windows NT computer with specialized hardware and software

Host 3: Oracle9*i* Database for the Real-Time Collaboration Repository

Using the instructions in the Advanced Configuration document at http://otn.oracle.com/products/webconferencing/, configure the LBR/Oracle9*i*AS Single Sign-On for multiple sets of Real-Time Collaboration Core Components. The URL used to access Oracle Web Conferencing will now be "imeeting.company.com," and "imeeting1.company.com" and "imeeting2.company.com" will be behind the LBR. Users will access "imeeting.company.com" and the request will automatically be routed to either imeeting1.company.com or imeeting2.company.com.

See Also: *Oracle Collaboration Suite Installation and Configuration Guide for Solaris* for additional information on the prerequisites

6.2.3 Settings

Configure Web Conferencing to leverage existing enterprise e-mail and proxy servers for its functionality. Assume that the settings are as follows:

Table 6–4 System Settings

Setting Name	Value	Comments	Web Conferencing Property
Existing e-mail server host name (one for the enterprise)	"mail-net.company.com"	Post-installation step	SmtpHost
E-mail server listening port	25	Default	SmtpPort
Existing proxy server host name for HTTP (one for the enterprise)	"www-proxy.company.com"	Post-installation step	UserAgentProxyHost
Proxy server host name for HTTPS (one for the enterprise)	"www-proxy.company.com"	Post-installation step	UserAgentProxySSLHost

Table 6–4 System Settings

Setting Name	Value	Comments	Web Conferencing Property
Proxy port for HTTP	80	Default	UserAgentProxyPort
Proxy port for HTTPS	443	Default	UserAgentProxySSLPort
Proxy domains to exclude	".company.com"	Post-installation step	UserAgentProxyExclusions
Web Conferencing access URL	"imeeting.company.com"	Post-installation step. Requests to imeeting.company.com will be redirected to imeeting1.company.com and imeeting2.company.com.	GlobalWebHost

Host 1 settings are the same as described as in Table 6–2, "Instance on Host 1".

Configure Host 5 as follows:

Table 6–5 Host 5 Settings

Name of Setting	Value	Comments	Web Conferencing Property
Number of mx	1	Default	SrvNumProcs
Listening port for the single mx	2400	Set during installation. Keep it the same as Host 1.	MxListenPort
Actual host name of Host 5	"host5.company.com"		N/A
Alias for Host 5	"imeeting2.company.com"	Post-installation step	ApacheWebHost
Apache listening port for HTTP on Host 5	7777	Default. Set during installation.	ApacheWebPort
Apache listening port for HTTPS on Host 5	443	Default	ApacheWebSecurePort

6.2.4 Installation

On Host 1 and Host 5, install the Real-Time Collaboration Core Components, and provide the required port numbers.

6.2.5 Post-Installation

Invoke the imtctl commands in this section, while setting Web Conferencing properties. The commands in this section are based on the data in the Settings tables in Section 6.2.3, "Settings".

Host 1

Step 1 Set up GlobalWebHost.

```
imtctl> setProperty -system true -pname GlobalWebHost -pvalue
"imeeting.company.com"
```

Follow steps 2-7 as defined in Section 6.1.5, "Post-Installation".

Step 2 Set up e-mail.

Step 3 Set up proxy.

Step 4 Integrate Web Conferencing with Oracle HTTP Server on this Real-Time Collaboration Core Components machine.

Step 5 Set up for monitoring.

Step 6 Set up reports.

Host 5

It is not necessary to set up e-mail, proxy, or global Web host, because they are global settings and will apply to this instance, as well.

Step 1 Integrate Web Conferencing with Oracle HTTP Server on this Real-Time Collaboration Core Components machine.

Follow the step as defined in Section 6.1.5, "Post-Installation".

Step 2 Set up for monitoring.

Follow the step as defined in Section 6.1.5, "Post-Installation".

Host 2

No post-installation steps are required.

6.2.6 Verification

Host 1

Invoke `imtctl> runTests`. This runs the full verification test.

Typically, all tests succeed. However, if you run this command before setting up Host 2, the Voice Conversion Server test and the Document Conversion Server test will fail.

Run this as described in the Section 6.2.1, "Sequence of Steps".

Host 5

Invoke `imtctl> runTests`. This runs the full verification test.

Typically, all tests succeed. However, if you run this command before setting up Host 2, the Voice Conversion Server test and the Document Conversion Server test will fail.

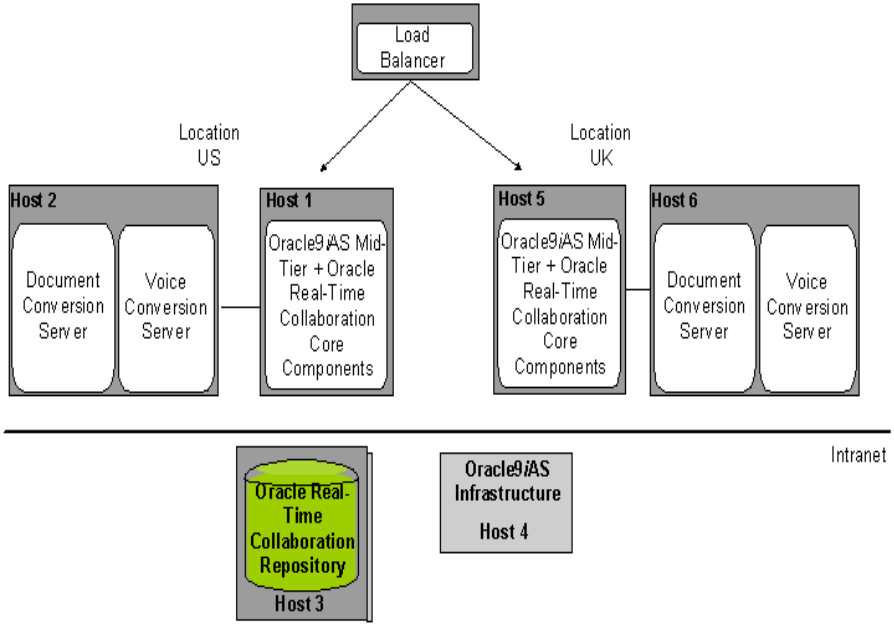
Run this as described in Section 6.2.1, "Sequence of Steps".

Host 2

Invoke `imtctl> getState` to verify that the processes are up, but the real test needs to be invoked on Host 1 and Host 5 (`imtctl> runTests`) to make sure that the services provided by Host 2 are available.

6.3 Multiple Geographical Locations and Load Balancer

Figure 6–3 Multiple Geographical Locations + LBR



In this deployment, Host 1, Host 2, Host 5, and Host 6 are deployed in the DMZ. Host 3 and Host 4 are deployed in the corporate intranet.

This is almost identical to the previous example (Section 6.2, "Multiple Real-Time Collaboration Core Components and Load Balancer") with the following key differences:

- LBR configuration: Host 1 and Host 5 are in different geographical locations, the United States and the United Kingdom. Furthermore, the LBR is configured to direct URL requests from users in the UK to the Real-Time Collaboration Core Components in the UK (Host 5), and URL requests from users in the US to the Real-Time Collaboration Core Components in the US. When users in the UK access imeeting.company.com, requests are redirected to imeeting1.company.com on Host 5. Similarly, when users in the US access imeeting.company.com, requests are redirected to imeeting2.company.com.

- Separate sets of Document and Voice Conversion Servers: The Real-Time Collaboration Core Components in Host 1 can use the Document and Voice Conversion Servers in Host 2 (location US) only, and the Real-Time Collaboration Core Components in Host 5 can use the Document and Voice Conversion Servers in Host 6 only.

Refer to Table 6–5 and Table 6–5 in the previous section for all the values for Host 1 through Host 5 and system settings. The changes are as follows:

Table 6–6 Host 5 Setting Changes from the Previous Section

Setting Name	Value	Comments	Web Conferencing Property
Location of Host 5	"UK"	Post-installation step.	InstanceLocation

Table 6–7 Host 6 Settings

Setting Name	Value	Comments	Web Conferencing Property
HTTP listening port for the Voice Conversion Server	2460	Set during installation.	
Location	"UK"	Post-installation step. Keep it the same as Host 5.	InstanceLocation

Table 6–8 Host 1 Setting Changes from the Previous Section

Setting Name	Value	Comments	Web Conferencing Property
Location	"US"	Post-installation step.	InstanceLocation

Table 6–9 Host 2 Settings

Setting Name	Value	Comments	Web Conferencing Property
HTTP listening port for the Voice Conversion Server	2460	Set during installation.	
Location	"US"	Post-installation step. Keep it the same as Host 1.	InstanceLocation

Follow all installation, prerequisite, and post-installation instructions from Section 6.2, "Multiple Real-Time Collaboration Core Components and Load Balancer" in addition to the following:

Host 1

Step 1 Set up the location.

```
imtctl> setProperty -pname InstanceLocation -pvalue "US"
```

Host 2

Set this up to serve the Real-Time Collaboration Core Components on Host 1.

```
imtctl> setProperty -pname InstanceLocation -pvalue "[\"US\"]"
```

Host 5

Step 1 Set up the location.

```
imtctl> setProperty -pname InstanceLocation -pvalue "UK"
```

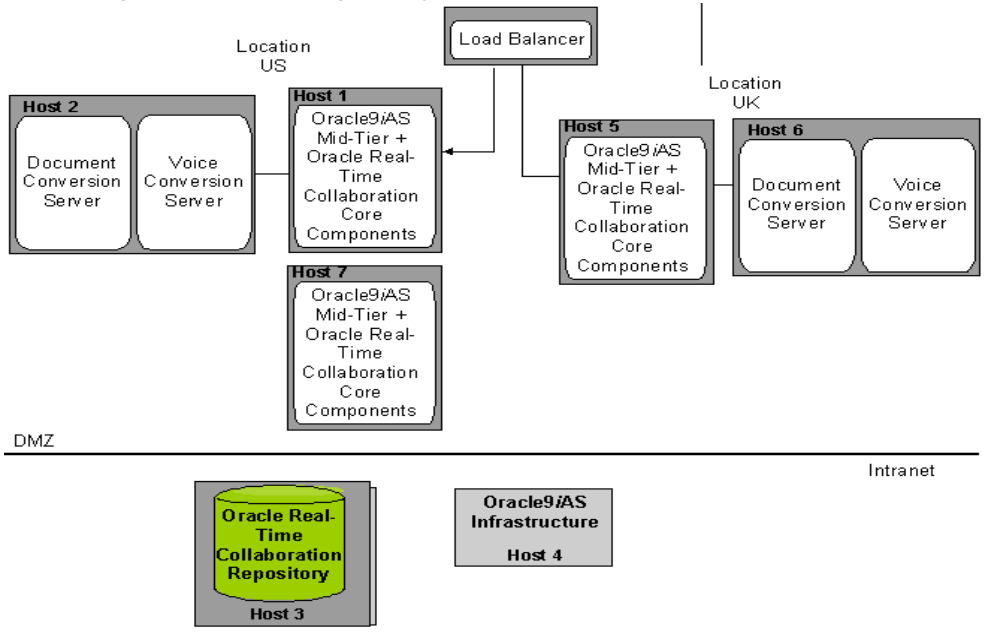
Host 6

Set this up to serve the Real-Time Collaboration Core Components on Host 5.

```
imtctl> setProperty -pname InstanceLocation -pvalue "[\"UK\"]"
```

6.4 Adding New Core Components to an Existing System

Figure 6–4 Existing Web Conferencing Deployment with Additional Core Components



This example shows the addition of a machine, Host 7, to the configuration shown in the previous example. The changes are as follows:

- Host 7 has been added in location "US"
- Host 2 will provide document and voice conversion services to the Real-Time Collaboration Core Components on Host 7.
- The LBR needs to be configured to direct all requests from the US to either Host 1 or Host 7 and to continue to direct all UK requests to Host 5.

6.4.1 Sequence of Steps

1. Install prerequisites on Host 7.
2. Install Real-Time Collaboration Core Components on Host 7.
3. Perform post-installation steps on Host 7.
4. Perform verification tests on Host 7.

5. Restart the Document and Voice Conversion Servers on Host 2.
6. Perform verification tests on Host 7.

6.4.2 Prerequisites

Host 7: Oracle9*i*AS mid-tier set up and configured to use the Oracle9*i*AS Infrastructure on Host 4.

> **See Also:** *Oracle Collaboration Suite Installation and Configuration Guide for Solaris* for details

6.4.3 Settings

Configure Host 7 as follows:

Table 6–10 Host 7 Settings

Name of setting	Value	Comments	Web Conferencing Property
Number of mx	1	Default	SrvNumProcs
Listening port for the single mx	2400	Set during installation. Keep it the same as Host 1.	MxListenPort
Actual host name of Host 7	"host7.company.com"		N/A
Alias for Host 5	"imeeting3.company.com"	Post-installation step	ApacheWebHost
Apache listen port for HTTP on Host 5	7777	Default. Set during installation.	ApacheWebPort
Apache listen port for HTTPS on Host 5	443	Default	ApacheWebSecurePort
Location of Host 5	"US"	Post-installation step. Keep it the same as Host 1.	InstanceLocation

6.4.4 Installation

On Host 1 and Host 5, install the Real-Time Collaboration Core Components, and provide the required port numbers.

6.4.5 Post-Installation

Invoke the imtctl commands in this section, while setting Web Conferencing properties. The commands in this section are based on the data in the Settings tables in Section 6.4.3, "Settings".

Host 7

It is not necessary to set up e-mail, proxy, or global Web host, because they are global settings and will apply to this instance, as well.

Step 1 Integrate Web Conferencing with Oracle HTTP Server on this Real-Time Collaboration Core Components machine.

Follow the step as defined in Section 6.1.5, "Post-Installation".

Step 2 Set up location.

```
imtctl> setProperty -pname InstanceLocation -pvalue "US"
```

Step 3 Set up for monitoring.

Follow the step as defined in Section 6.1.5, "Post-Installation".

Step 4 Restart Host 2.

Restart Host 2 so that it recognizes the Real-Time Collaboration Core Components on Host 7.

6.4.6 Verification

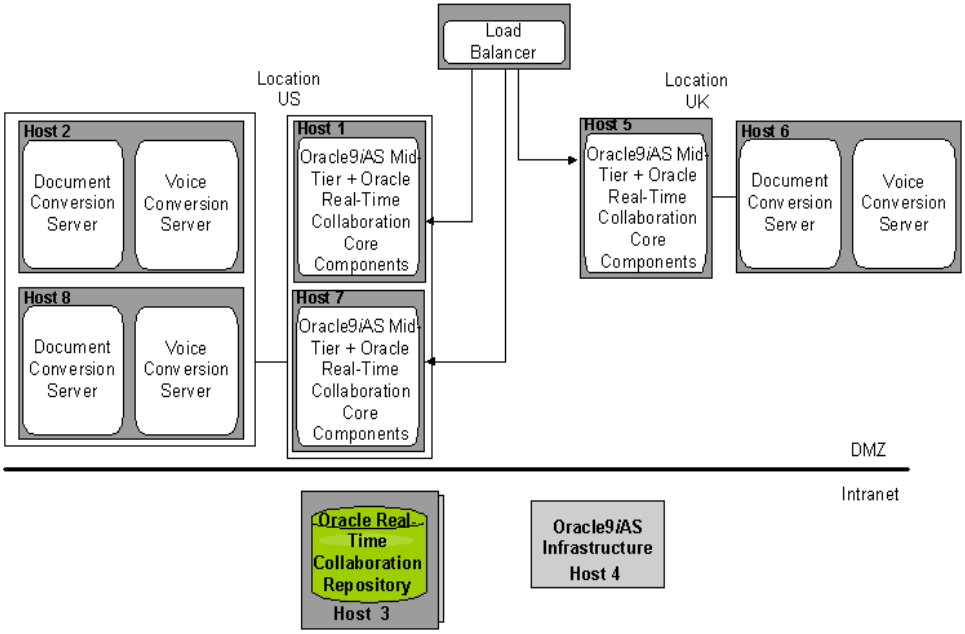
Host 7

Invoke `imtctl> runTests`. This runs the full verification test.

All tests succeed.

6.5 Adding Document and Voice Conversion Servers to an Existing System

Figure 6–5 Web Conferencing Deployment with Additional Document and Voice Conversion Servers



Another machine with Document and Voice Conversion Servers is added (Host 8).

6.5.1 Sequence of Steps

1. Install prerequisites on Host 8.
2. Install the Document and Voice Conversion Servers on Host 8.
3. Perform post-installation steps on Host 8.
4. Perform verification tests on Host 8.
5. Bring down the Document and Voice Conversion Servers on Host 2, so that Host 8 is the only one available.
6. Perform verification tests on Host 7 or Host 1.

7. Restart the Document and Voice Conversion Servers on Host 2.

6.5.2 Prerequisites

See the Oracle Collaboration Suite Installation and Configuration Guide for Solaris for additional information on the prerequisites.

6.5.3 Settings

Table 6–11 Host 8 Settings

Setting Name	Value	Comments	Web Conferencing Property
HTTP listening port for the Voice Conversion Server	2460	Set during installation	
Location	"US"	Post-installation step	InstanceLocation

6.5.4 Installation

On Host 8, install the Document and Voice Conversion Servers, and provide the required port numbers.

6.5.5 Post-Installation

Host 8

1. Set this up to serve the Real-Time Collaboration Core Components in location "US" by invoking:

```
imtctl> setProperty -pname InstanceLocation -pvalue "[\"US\"]"
```

2. Restart the Document and Voice Conversion Servers by invoking `imtctl> stop` and then `imtctl> start`.

6.5.6 Verification

Host 8

Invoke `imtctl> getState` to verify that the processes are up, but the real test needs to be invoked on Host 1 (`imtctl> runTests`) to make sure that the services provided by Host 8 are available.

Host 2

Shut down Host 2.

Host 1

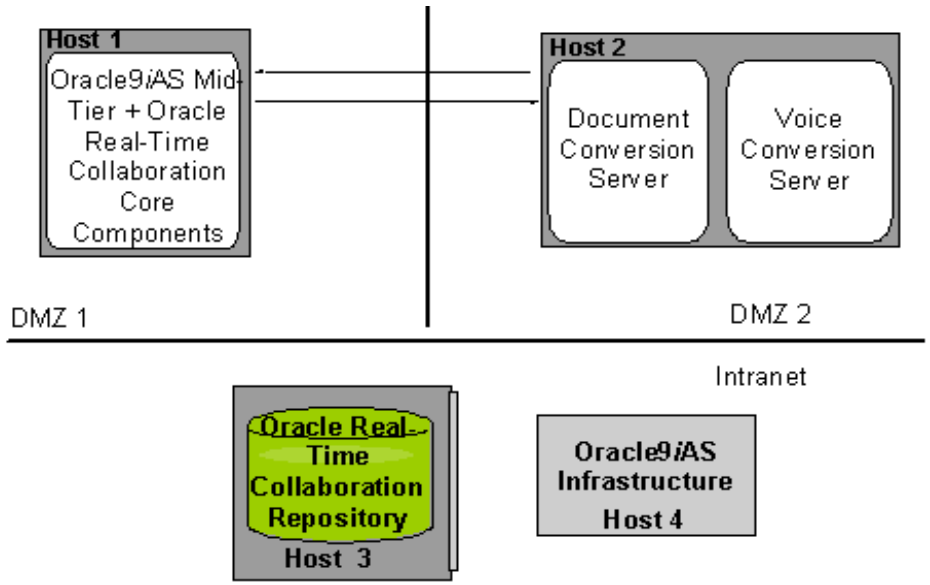
Invoke `imtctl> runTests`. This runs the full verification test.

All tests succeed.

6.6 Additional Example

The topology is identical to the basic deployment, but the Document and Voice Conversion Servers are deployed in a different DMZ.

Figure 6–6 Deployment with Core Components in a Different DMZ



The following additional configuration is required:

- From DMZ 1 to DMZ 2, enable HTTP requests on port 2450.
- From DMZ 2 to DMZ 1, enable TCP/IP requests on port 2400.

7 Monitoring

This chapter describes how Real-Time Collaboration components are monitored to provide quality of service for conferences and uninterrupted availability for conference service. This chapter includes explanations of the following:

- Process monitoring—monitors all Real-Time Collaboration processes and restart them if they are down.
- Conference monitoring—lets you monitor current conferences, including events for each conference.
- Service availability monitoring—lets you monitor whether an instance in a system is capable of providing the direct services.
- Monitoring statistics—lets you monitor performance statistics for tuning the Real-Time Collaboration system
- Configuration test—Verifies key configuration setup.
- Real-Time Collaboration Interfaces—Interfaces that allow Real-Time Collaboration services to be monitored by any monitoring system.

7.1 Process Monitoring

7.1.1 What is the Real-Time Collaboration Process Monitor?

The Real-Time Collaboration Process Monitor (imt-pm) is a Java-based process that runs as a daemon, managing all Real-Time Collaboration processes in an instance. The Real-Time Collaboration Process Monitor opens an HTTP listening point to accept requests to start and stop processes. Specifically, it does the following:

- Periodically pings all Real-Time Collaboration processes in an instance, to check if the process is active.

- Automatically restarts inactive processes (processes that do not respond to the ping) without manual intervention.

The Oracle Process Management and Notification system monitors imt-pm and serves as a watchdog for it. The Oracle Process Management and Notification system automatically restarts imt-pm if it detects it to be inactive. If imt-pm goes down, there is no impact on the processes that were being monitored by imt-pm. imt-pm can recover its state without affecting the processes that it was monitoring before it went down.

7.1.2 imt-pm and start/stop

```
imtctl> start
```

When you invoke the start command with no arguments (startall behavior) from imtctl, Oracle Process Management and Notification, and imt-pm (if necessary) automatically start, which then starts all the processes it needs to manage.

Other start commands require that imt-pm already be running.

```
imtctl >stop
```

The stop command with no arguments (stopall behavior) shuts down imt-pm and all of the processes it is managing.

7.1.3 imt-pm and Oracle Process Management and Notification

imt-pm is integrated with Oracle Process Management and Notification during the installation of Web Conferencing (Real-Time Collaboration).

7.2 Service Availability Monitoring

With Oracle Real-Time Collaboration, you can check service availability on an instance. Sometimes, it might not be enough for the processes to be up. It is also important that the instance be able to provide the conference services.

Using imtctl, determine service availability by invoking:

```
runtests -testlist mtgtest,voiceconvtest,docconvtest
```

This command checks if the instance on which the command is executed is providing the following services:

conference service—A conference test mimics the behavior of an end-user going through the same flow of Oracle Web Conferencing Application as the user, and starting an instant conference. It makes sure that the conference starts successfully on one of the available collaboration servers in an instance. It then joins another client in the same conference, and makes sure that the new client also gets the conference state consistent with the host of the conference. It finally ends the test conference. This test thus makes sure that conference service is available in a particular Real-Time Collaboration instance. If this test fails, make sure that one or more mxs, and one or more collaboration servers are up in this instance (using `imtctl> getState`).

voice conversion service—This test verifies that voice streaming service is available by mimicking the behavior of a collaboration server for connecting to an available voice conversion server, and making sure that T1 line is up, there are available voice channels, and the server is able to stream audio. If this test fails, then either no voice conversion server has been setup to service this instance, or the ones configured are not available.

document conversion service—This test verifies document conversion service availability by mimicking document conversion flow. It uploads a test document, and then tries converting that document using the document conversion servers available to the instance. If this test fails, then either no document conversion server has been setup to service this instance, or the ones configured are not available.

Real-Time Collaboration also publishes interfaces for service availability monitoring that can be integrated into any monitoring infrastructure. See Section 7.6, "Real-Time Collaboration Interfaces" for more information.

The service availability tests can be invoked periodically by a monitoring infrastructure, and the results of the tests can be plugged into an alert management system.

7.3 Component Monitoring

With Oracle Real-Time Collaboration, you can get monitoring data for the key components in the system. The monitoring data includes the following:

imt-collab: for each imt-collab on that instance, you can obtain the following data:

- number of active conferences
- number of active users
- total memory

- used memory
- total conferences since the time the process was started

Use `getMonitorStats` on the instance to get these statistics, or use the `-i` option to direct the command towards a different instance.

imt-voiceconv: for the Voice Conversion Server on that instance, you can obtain the following data:

- total number of channels
- currently used channels
- bad channels
- T1 line status

Use `getMonitorStats` on the instance to get these statistics, or use the `-i` option to direct the command toward a different instance.

Oracle Real-Time Collaboration publishes interfaces for getting this data in XML. This can then be invoked periodically by a monitoring infrastructure for historical analysis. Use getMonitorStats with the publish option.

```
imtctl> getMonitorStats -publish true
```

7.4 Conference Monitoring

The Oracle Web Conferencing Application provides Web pages to monitor the conferences that are currently running on the system. The Monitor tab in the Web Conferencing Application provides access to this functionality.

The first page in the Monitor tab lists all the conferences that are currently running on the system. For each conference, it provides the conference ID, conference title, host name, conference type, site, start time, total attendees, and current status of the conferences. It also provides a Meeting Details link, which allows the user to find out detailed information about each conference that is currently running.

In the Meeting Details pages, you can find information about the attendees participating in that conference, including details about each attendee. Conference level details like current collaboration mode, voice start time, and so on can also be found in these pages. You can also see the conference logs for this conference from this page.

7.5 Configuration Tests

These tests confirm that some important post-installation steps, like e-mail configuration, which are required for the proper functioning of the Real-Time Collaboration system have been successfully completed, and serve as a verification tool for the administrator. These tests need to be run only once after an installation, and need not be run periodically.

7.5.1 E-mail Configuration Test

This test checks whether the post-installation step required for Oracle Web Conferencing e-mail invitations to work has been performed by the administrator. For e-mail invitations to work, enterprise SMTP host and port should be specified through Oracle Web Conferencing properties. The test makes sure that these properties are set and the SMTP server is accessible from the Oracle Real-Time Collaboration mid-tier.

7.5.2 Proxy Configuration Test

This test checks whether the post-installation step required for Oracle Web Conferencing cobrowsing to work has been performed by the administrator. This test needs to succeed only if the mid-tier needs an enterprise server to access internet. This test makes sure that the properties required for proxy configuration are set, and that the proxy server is accessible from the Oracle Real-Time Collaboration mid-tier.

7.5.3 Mx Configuration Test

For supporting internet users where clients need to connect from behind their corporate proxies, mod_imeeting needs to be enabled and some properties need to be set in the Real-Time Collaboration system. This test makes sure that the required properties are set, and mod_imeeting has been enabled to handle HTTP/HTTPS requests to connect to Mx.

7.5.4 Repository (Database) Access Test

Because all Real-Time Collaboration services depend on the database as the single-point of information and backbone, a test is provided to verify access to the Real-Time Collaboration Repository. This test makes sure that database is available (that is, it is accessible and responding to queries).

7.6 Real-Time Collaboration Interfaces

This section discusses Real-Time Collaboration monitoring interfaces that can be plugged into any monitoring infrastructure. There are two types of interfaces:

- Servlet interface—for monitoring service availability
- imtctl interfaces—for getting monitoring data about key Real-Time Collaboration components.

7.6.1 Servlet Interfaces

7.6.1.1 Overview

Oracle Web Conferencing includes a suite of functionality tests that are designed to verify the availability of major Web Conferencing functionalities. The tests cover Real-Time Collaboration Repository connectivity, Voice Conversion Server availability, Document Conversion Server availability, application availability, and the ability to launch an instant conference.

The ImtTestServlet allows for these tests to be invoked by any monitoring infrastructure. ImtTestServlet acts as an adapter that makes all Real-Time Collaboration tests available as a Web application to HTTP-based Web application monitoring. The servlet is accessible to standard HTTP-based monitoring clients. Its results are designed for automated analysis. ImtTestServlet is the most flexible choice for low-effort integration of existing monitoring systems with the Real-Time Collaboration functionality tests.

7.6.1.2 ImtTestServlet API

Inputs to the Servlet

As an HTTP servlet, ImtTestServlet gets all of its input information from HTTP requests sent by the client. The servlet accepts parameters either through the URL query string or via a POST body. Either way, the same parameters are supported. Input parameters control which tests are run and the information that is returned in the case of success or failure.

Table 7–1 ImtTestServlet Inputs

Name of Test	Options	Default	Comments
alltests	true, false	false	Run all tests supported by the servlet (other test selection parameters are ignored).
mtgtest	true, false	false	Run the end-to-end conference test to verify that conference functionality is available.
voiceconvtest	true, false	false	Run the Voice Conversion Server test to verify that voice support is available for conferences.
docconvtest	true, false	false	Run the Document Conversion Server test to verify that document conversion is available.
dbtest	true, false	false	Run the Real-Time Collaboration Repository test to verify that it is available.
errorcode	any valid HTTP response code	500	Sets the HTTP response code sent when any of the selected tests fail.
successcode	any valid HTTP response code	200	Sets the HTTP response code sent when all of the selected tests succeed.
errormsg	Any string	null	Message included in the response body when any of the selected tests fail. (Note: the response body may include additional text, as well.)
successmsg	Any string	"Test(s) successful."	Message included in the response body when all of the selected tests succeed. (Note: the response body may include additional text, as well.)

7.6.1.3 Output from the Servlet

The ImtTestServlet provides its results via an HTTP response. Currently the results are either that all of the selected tests succeeded or that some test failed. If multiple tests are selected via input parameters, the result provides no details about which particular tests failed. Furthermore, no messages associated with the failure are returned.

The result of the test or tests is reflected in both the HTTP response code and, optionally, in a static string returned in the response body. It is anticipated that these

two mechanisms will be sufficient to allow integration products to detect whether the test succeeded or failed.

Because the test servlet reports only aggregate results when running multiple tests, Oracle Corporation advises that the servlet be called multiple times independently when fine-grained failure detection is desired (one request for each test to be run). However, you might choose to have a single indicator of system health, in which case, running all of the tests in a single request is an optimal configuration.

7.6.1.4 Samples

The following examples are designed to illustrate some of the input parameter combinations that might be useful. The examples are templated to account for variable host names, ports, and application root contexts.

http://<host>:<port>/<root-context>/servlet/ImtTestServlet

Comments: This confirms that the test servlet has been properly installed. It does not run any tests.

http://<host>:<port>/<root-context>/servlet/ImtTestServlet?alltests=true

Comments: This runs all of the tests returning the standard error (500) and success (200) codes.

http://<host>:<port>/<root-context>/servlet/ImtTestServlet?mtgtest=true&errorcode=404

Comments: This runs only the end-to-end conference test and returns 404 if the test fails.

http://<host>:<port>/<root-context>/servlet/ImtTestServlet?mtgtest=true&voiceconvtest=true&errormsg=mtgorvoicefailed

Comments: This runs the conference and voice tests and reports a custom message on failure, in addition to a standard 500 response code.

7.6.1.5 Limitations

The ImtTestServlet is currently limited in the following way. Because the servlet runs in an OC4J_imeeting, the inaccessibility of the OC4J_imeeting will prevent granular detection of failures in other Real-Time Collaboration components, such as the Web Conferencing Servers and Document Conversion Servers. However, this limitation is minor, because OC4J_imeeting is the gatekeeper for all Web

Conferencing services. Inaccessibility of OC4J_imeeting is equivalent to inaccessibility of all Web Conferencing services from a client's perspective.

8

Reports

Oracle Real-Time Collaboration supports various reporting capabilities, including e-mailed reports and usage trend information available within the Oracle Web Conferencing Application. Some aspects of these features require post-installation configuration, such as including sender's and receiver's e-mail addresses. The chapter explains the properties that must be configured.

8.1 Usage Report

The usage report is designed to give high-level Web Conferencing usage metrics to business administrators and business managers. The report contains data which is based on one of the following two metrics:

- Time (minutes/hours)
- Usage numbers (count of conferences, users, and so on)

The Key Performance Indicators (KPI) are then summarized with a trending pattern that displays the count and percentage change over the previous time period. The report gives business administrators and business managers an idea of what collaborative modes and features are used, top users, and line of business information. The usage report is computed for a weekly period. Any user who has access to the report can view historical data in addition to the current data by keying in the relevant dates.

The following metrics are captured in the usage report:

- Conference minutes summary:
  - Total user minutes
  - Total conference minutes
  - Total voice minutes

    - Total recorded minutes
    - Total playback minutes
    - Host rating
    - Average conference length
    - Longest conference (minutes)
- Conference usage summary:
    - Conferences
    - Voice conferences
    - Total users
    - Total recordings
    - Total playbacks
    - Users in conferences
    - Voice users
    - Average users per conference
    - Largest conference (users)
- Top 10 lists
    - Users
    - Sites

In addition to the weekly KPIs table, the conference minutes and usage details are displayed in a time series of week, month, quarter, and year. The weekly average over the highest period is also computed.

8.2 Feedback Reports

The feedback report is designed to give high-level system performance feedback via user ratings to the business administrator and to business managers. The report contains data which is based on performance ratings given by conference hosts at the end of every conference.

The host of the conference can give a comment (raw text feedback) in addition to rating a conference excellent, good, or poor. Optionally, the host can even decide to not rate the conference, in which case the conference is classified with no response.

The KPIs are then summarized with a trending pattern displaying the count and percentage change over the previous time period.

Below the KPIs table, the top ten conferences with poor, good, excellent, and no response ratings with user comments are listed. These are grouped into four individual tables. Only conferences with host comments are listed in the tables. This convenient format allows administrators to proactively respond to the users who rate the conferences poor.

8.3 Uptime Reports

Oracle Real-Time Collaboration provides tests that can be used by monitoring infrastructures to periodically monitor Web Conferencing for service availability. This data can then be used produce uptime reports. See Section 7.6, "Real-Time Collaboration Interfaces" for more information on the tests and how to integrate them with your monitoring infrastructure.

9

Web Conferencing Sites

9.1 Overview

As an enterprise class platform for real-time collaboration, Oracle Web Conferencing allows a single deployment instance to be configured easily to meet the requirements of various lines of business (LOB). This objective can be achieved by creating individual sites and customizing system, application, and conference level properties.

Web Conferencing sites are created by the business administrator. For every site created, the integrating application or line of business uses a unique authentication token along with the site ID to communicate with the Web Conferencing Application. Creating a site provides the following benefits:

- Customized system, application, and conference properties.
- Support for custom integrated flows which can bypass the Web Conferencing Application.
- Support for custom Web Conferencing Console behavior based on the properties set for the site.
- Site-level reports which give the same depth and breakdown of data as the global Web Conferencing site reports.
- Custom user interfaces, including a separate Oracle Web Conferencing prelogin application page with full support for listing public conferences only for the specific site.
- Custom branding to uniquely identify the integrating site/line of business.

The global site is pre-created and cannot be deleted. By default, all users belong to site "iMeeting," which is a global site. However, a user can belong to more than one site.

9.2 Site ID

In an enterprise, a single deployment of the Real-Time Collaboration system can be used by various LOB applications. These integrating applications are called sites. An identifier for such an integrating application is called a site ID. The Real-Time Collaboration system uses the concept of site to provide a customized "view" of the Oracle Web Conferencing deployment, for each of the integrating applications.

Applications in an enterprise can integrate with Oracle Web Conferencing in two ways:

- It can programmatically invoke Web Conferencing functionality using Oracle Real-Time Collaboration Integration Service calls or,
- If the application has a browser-based interface, it can invoke the Oracle Web Conferencing Application pages directly.

In either case, a site ID enables the Real-Time Collaboration system to provide a customized service for each integrating application.

9.2.1 Creating a Site

The Oracle Web Conferencing administrator can create a site by providing a unique site name, display name, and description for the site. The Real-Time Collaboration system then registers the site and provides the administrator with a site ID and an authentication token, both of which are system generated. An application that wants to integrate with the Real-Time Collaboration system can use the new site ID to make the appropriate calls. You can then set up site-specific properties to control the interaction of the integrating site with Oracle Web Conferencing. For all properties that you do not change, the values of the corresponding default Web Conferencing site apply.

9.2.2 Using a Site ID

For Oracle Real-Time Collaboration Integration Service calls, the site ID is passed as one of HTTP header fields in the HTTP request to invoke the service. For any application Web page that provides a link to the Oracle Web Conferencing Application page, it can provide the site ID as a URL parameter. For example, the following URL can be used to provide a customized view of Oracle Web Conferencing for site 123456.

http://<hostname>:<port-number>/imtapp/app/prelogin.uix?siteID=123456

Invoking this URL returns a page that has the look and feel and contents specified for site 12345. Also, any conferences created from the application pages will have the attributes defined for conferences for that site.

9.2.3 Customizing Site Properties

Once a site ID is created in the Oracle Real-Time Collaboration system, an administrator can set different site-level properties to customize its behavior for that site. There are two broad classes of properties:

- Application Properties: Change the look and feel of the Oracle Web Conferencing Application pages. Examples of such properties are the co-branding name that appears at the top of each application page, the e-mail address of the Contact Us page, the number of rows displayed for each table in the application page, and so on.
- Conference Properties: Change the attributes of any conference, whether it is invoked from the application pages or from the integration service calls. Examples of such properties include the Startup (collaboration) Mode of the conference, the availability of chat during the conference, and so on.

Figure 9–1 Site ID

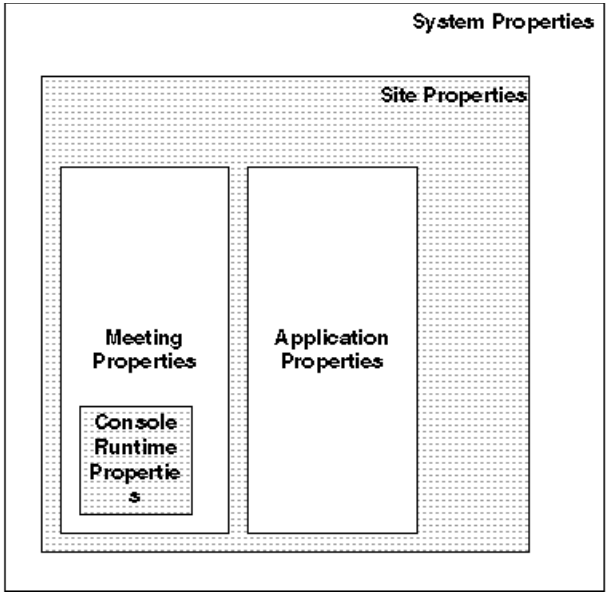


Figure 9–1, "Site ID" shows how the various property values in Oracle Web Conferencing are inherited and set. When a site is created, it inherits the default values of all the properties from the system settings. For each site the administrator can set various conference-level and application-level properties to override the default values. Some of the conference-level properties can be changed from inside the Web Conferencing Console when the conference is running. For example, the default setting for the system is to set the start mode of each conference to be Cobrowse mode. When a site is created, this default value is inherited for that site. The site administrator can override this default value by changing this property for that time to say Desktop Sharing mode.

10

imtctl Command Line Utility

The imtctl utility provides a command-line interface for administering and configuring the Real-Time Collaboration system. The utility supports a variety of commands to support actions such as starting and stopping Real-Time Collaboration processes, viewing their current state, and configuring them. You can run imtctl in an interactive shell mode, or it can take a single command as a command-line argument, in which case it will execute the command and exit.

The utility is available on all platforms under <ORACLE_HOME>/imeeting/bin. It is invoked as "imtctl" on all platforms. Invoking imtctl without any arguments opens the interactive shell.

```
$ORACLE_HOME/imeeting/bin> imtctl
```

A list of supported commands can be obtained by executing the help command:

```
imtctl> help
```

You can get specific instructions for a particular command by invoking the command followed by the "-help" option:

```
imtctl> <COMMAND> -help
```

The exit command ends the shell session:

```
imtctl> exit
```

10.1 Common Commands

10.1.1 Help

Gets the list of commands available in the shell.

10.1.2 listInstances

List all the instances in the Real-Time Collaboration system.

Examples:

It can only be invoked in one way:

```
imtctl> listInstances
```

Table 10–1 Real-Time Collaboration Instances

Instance Name	Host Name	Oracle Web Conferencing Home
instance1.oracle.com	host1.oracle.com	C:/core/imeeting
instance2.oracle.com	host2.oracle.com	/u02/ 90200b/imeeting

10.1.3 listComponents

List the components in any instance. The different forms of this command are:

- listComponents [-i <instance-name>]:

  List all the components in the current instance. Use -i to list components in an instance with the given name.

- listComponents - cid <component-id>

  List the component with the given ID.

- listComponents -cname <component-name> [-i <instance-name>]

  List the components of the given name. If -i is used, the component with the given name in the given instance is listed.

- listComponents -ctype <component-type> [-i <instance-name>]

  List the components with the given type. If -i is used, the components with the given type in the given instance are listed.

10.1.3.1 Examples

Example 1

To get the components in the current instance (ap079sun.us.oracle.com), invoke the following:

```
imtctl> listComponents
```

Table 10–2

ID	Name	Type	Description	num_procs
10001	imt-collab	clbsvr	Collaboration Server	4
10003	imt-starter	starter	Web Conferencing Starter	1
10002	OC4J_imeeting	oc4j	OC4J	1
10000	mx	mxcomm	Multiplexer	1

Example 2

To get the components in a different instance (instance1.company.com), invoke the following:

```
imtctl> listComponents -i instance1.company.com
```

Table 10–3

ID	Name	Type	Description	num_procs
10006	imt-voice	voiceconv	Voice Conversion Server	1
10005	imt-starter	starter	Web Conferencing Starter	1
10004	imt-converter	docconv	Document Conversion Server	1

To list the component with the ID 10006, invoke the following:

```
imtctl> listComponents -cid 10006
```

Table 10–4

ID	Name	Type	Description	num_procs
10006	imt-voice	voiceconv	Voice Conversion Server	1

Example 3

To list the components on the current instance with the type "clbsvr", invoke the following

```
imtctl> listComponents -ct clbsvr
```

Table 10–5

ID	Name	Type	Description	num_procs
10001	imt-collab	clbsvr	Collaboration Server	4

10.1.4 start

Start the whole (current) instance, or individual components in the current instance. The different forms of this command are:

- start:

  Start all the components in the current instance. This is the recommended way of starting an instance. This takes care of startup dependencies.

- start - cid <component-id>

  Start the component with the given ID.

- start -cname <component-name>

  Start the component of the given name.

- start -ctype <component-type>

  Start the components with the given type.

10.1.4.1 Examples

Example 1

To start a component with component ID 10001 (which needs to be on the current instance) invoke:

```
imtctl> start -cid 10001
```

Example 2

To start a component with name imt-collab (which needs to be on the current instance) invoke:

```
imtctl> start -cname imt-collab
```

10.1.5 stop

Stop the whole (current) instance, or individual components in the current instance. The different forms of this command are:

- stop:

  Stop all the components in the current instance. This is the recommended way of stopping an instance. This takes care of stopping all components in the instance in appropriate order.

- stop - cid <component-id>

  Stop the component with the given ID.

- stop -cname <component-name>

  Stop the component with the given type.

- stop -ctype <component-type>

  Stop the component with the given type.

10.1.5.1 Examples

Example 1

To start a component with component ID 10001 (which needs to be on the current instance) invoke:

```
imtctl> stop-cid 10001
```

To start a component with name imt-collab (which needs to be on the current instance) invoke:

```
imtctl> stop -cname imt-collab
```

10.1.6 getState

Shows the state of components in any instance. The different forms of this command are:

- imtctl> getState [-i <instance-name>]:

  Show state of all the components in the current instance. Use -i to show the state of components in an instance with the given name.

- imtctl> getState - cid <component-id>

  Show state of the component with the given ID.

- imtctl> getState -cname <component-name> [-i <instance-name>]

  Show state of the components of the given name in the current instance. If -i is used, it shows state of the components with the given name in the given instance.

- imtctl> getState -ctype <component-type> [-i <instance-name>]

  Show state of the components with the given type in the current instance. If -i is used, it shows state of the components with the given type in the given instance.

Examples

To get the state of all the components in the current instance, invoke:

```
imtctl> getState
```

Table 10–6

ID	Name	Type	Status	num_procs
10001	imt-collab	clbsvr	UP	4
10000	mx	mxcomm	UP	1

10.1.7 setProperty

Set a property at a specified scope. Oracle Real-Time Collaboration supports system, instance (default), component, site, and conference type scopes. See Chapter 5, "Configuration" for more information on these scopes.

The different forms of this command are:

- imtctl> setProperty -pname <property_name> -pvalue <property_value>:

Set the value for the property at instance level.

- imtctl> setProperty -cid <component-id> -pname <prop_name> -pvalue <prop_value>

  Set the value of the property for the component with the given ID.

- imtctl> setProperty -cname <component-name> [-i <instance-name>] -pname <prop_name> -pvalue <prop_value>

  Set the value of the property for the component of the given name in the current instance. If `-i` is used, it sets the property for the component of the given name in the given instance.

- imtctl> setProperty -ctype <component-type> [-i <instance-name>] -pname <prop_name> -pvalue <prop_value>

  Set the value of the property for the components with the given type in the current instance. If `-i` is used, it sets the property for the components with the given type in the given instance.

- imtctl> setProperty -system true -pname <property_name> -pvalue <property_value>:

  Set the value for the property at system level.

- imtctl> setProperty - siteId <site-id>-pname <property_name> -pvalue <property_value>:

  Set the value for the property at site level.

Examples

To set the ApacheWebHost property for the current instance to imeeting4.company.com, invoke:

```
imtctl> setProperty -pname ApacheWebHost -pvalue "imeeting4.company.com"
```

To set the system-wide log-level to SEVERE, invoke:

```
imtctl> setProperty -system true -pname LogLevel -pvalue SEVERE
```

See Chapter 5, "Configuration" and Chapter 6, "Sample Deployments" for more examples.

10.1.8 getProperty

Get a property at the a specified scope. Oracle Real-Time Collaboration supports system, instance (default), component, site, and meeting type scopes. See Chapter 5, "Configuration" for more information on these scopes.

The different forms of this command are:

- imtctl> getProperty -pname <property_name>

  Get the value for the property at instance level.

- imtctl> getProperty -cid <component-id> -pname <prop_name>

  Get the value of the property for the component with the given ID.

- imtctl> getProperty -cname <component-name> [-i <instance-name>] -pname <prop_name>

  Get the value of the property for the component of the given name in the current instance. If `-i` is used, it gets the property for the component of the given name in the given instance.

- imtctl> getProperty -ctype <component-type> [-i <instance-name>] -pname <prop_name>

  Get the value of the property for the components with the given type in the current instance. If `-i` is used, it gets the property for the components with the given type in the given instance.

- imtctl> getProperty -system true -pname <property_name>

  Get the value for the property at system level.

- imtctl> getProperty - siteId <site-id>-pname <property_name>

  Get the value for all the properties at site level for the specified site

- imtctl> getProperty - mtgType <meeting type> -siteId <site-id> -pname <property_name>

  Get the value for all the properties at meeting type level for the specified meeting type and the site level properties in the specified site.

Examples

To get the ApacheWebHost property for the current instance, invoke:

```
imtctl> getProperty -pname ApacheWebHost
```

The effective value for instance 90200b.ap079sun.us.oracle.com of the property "ApacheWebHost" is "imeeting4.company.com"

To get the system-wide log-level, invoke:

```
imtctl> getProperty -system true -pname LogLevel
```

10.1.9 getProperties

Gets all properties at the a specified scope *and* above. Oracle Real-Time Collaboration supports system, instance (default), component, site, and meeting type scopes. See Chapter 5, "Configuration" for more information on these scopes.

The different forms of this command are:

- imtctl> getProperties

  Get the values for the properties at instance level and system.

- imtctl> getProperties -ct <component-type> [-i <instance-name>]

  Get the value of all the properties for the component with the given type in the current instance. If `-i` is used, it gets the property for the components with the given type in the given instance. It will return properties for the component, instance and system.

- imtctl> getProperties -system true

  Gets all system properties only

- imtctl> getProperties - siteId <site-id>

  Gets the value for all the properties at site level for the specified site and the system.

10.1.10 runTests

Runs Real-Time Collaboration tests (all tests or a specified set of tests) on a specific instance or all instances in the system.

- runTests [-i <instance-name>]:

  Run tests in the current instance. Use -i to run all the tests in an instance with the given name.

- runTests [-testlist <test-list separated by commas>]

  Different tests are:

  - mtgtest—conference service test
  - voiceconvtest—voice conversion service test

- docconvtest—document conversion service test
- dbtest—database connectivity test
- modtest—mx/mod configuration test
- proxytest—proxy configuration test
- emailtest—e-mail configuration test

- runTests - system true

  Run the tests on all the instances in the system.

Examples

After doing an installation and during the post-installation steps, the following can be invoked to see if things are configured properly and working.

```
imtctl> runTests
```

To run the conference test alone, invoke the following

```
imtctl> runTests -testlist "mtgtest"
```

To run the conference test and voice conversion service test, invoke the following

```
imtctl> runTests -testlist "mtgtest,voiceconvtest"
```

11
Troubleshooting

After Oracle Web Conferencing has been successfully configured and is working, there are two kinds of problems that require an administrator intervention:

- User complaints—Some users are unable to use Oracle Web Conferencing, or they are unable to log in, etc.
- Alerts—An administrator receives alerts after integrating Oracle Real-Time Collaboration monitoring into the company's monitoring infrastructure.

In addition, an administrator could experience problems while configuring the system.

For the exact steps to resolve these problems, and for other common guidelines, see the Troubleshooting document at http://otn.oracle.com/products/webconferencing/.

A

Time Zones

The following table contains a listing of time zones supported by Oracle Real-Time Collaboration.

Table A–1 Real-Time Collaboration Time Zones

Internal Name	External User Visible Name
Pacific/Pago_Pago	(-11:00) Pago Pago
Pacific/Honolulu	(-10:00) Hawaii
America/Anchorage	(-09:00) Alaska
America/Vancouver	(-08:00) Canada Pacific Time
America/Los_Angeles	(-08:00) US Pacific Time
America/Tijuana	(-08:00) Tijuana
America/Edmonton	(-07:00) Canada Mountain Time
America/Denver	(-07:00) US Mountain Time
America/Phoenix	(-07:00) Arizona
America/Mazatlan	(-07:00) Mazatlan
America/Winnipeg	(-06:00) Canada Central Time
America/Regina	(-06:00) Saskatchewan
America/Chicago	(-06:00) US Central Time
America/Mexico_City	(-06:00) Mexico City
America/Guatemala	(-06:00) Guatemala
America/El_Salvador	(-06:00) El Salvador

Table A–1 Real-Time Collaboration Time Zones

Internal Name	External User Visible Name
America/Managua	(-06:00) Managua
America/Costa_Rica	(-06:00) Costa Rica
America/Montreal	(-05:00) Canada Eastern Time
America/New_York	(-05:00) US Eastern Time
America/Indianapolis	(-05:00) East Indiana
America/Panama	(-05:00) Panama
America/Bogota	(-05:00) Bogota
America/Lima	(-05:00) Lima
America/Halifax	(-04:00) Canada Atlantic Time
America/Puerto_Rico	(-04:00) Puerto Rico
America/Caracas	(-04:00) Caracas
America/Santiago	(-04:00) Santiago
America/St_Johns	(-03:30) Newfoundland
America/Sao_Paulo	(-03:00) Sao Paulo
Atlantic/Azores	(-01:00) Azores
Etc./UTC	(00:00) Universal Time
UTC	(00:00) Universal Time
Atlantic/Reykjavik	(00:00) Reykjavik
Europe/Dublin	(00:00) Dublin
Europe/London	(00:00) London
Europe/Lisbon	(00:00) Lisbon
Africa/Casablanca	(00:00) Casablanca
Africa/Nouakchott	(00:00) Nouakchott
Europe/Oslo	(+01:00) Oslo
Europe/Stockholm	(+01:00) Stockholm
Europe/Copenhagen	(+01:00) Copenhagen
Europe/Berlin	(+01:00) Berlin

Table A–1 Real-Time Collaboration Time Zones

Internal Name	External User Visible Name
Europe/Amsterdam	(+01:00) Amsterdam
Europe/Brussels	(+01:00) Brussels
Europe/Luxembourg	(+01:00) Luxembourg
Europe/Paris	(+01:00) Paris
Europe/Zurich	(+01:00) Zurich
Europe/Madrid	(+01:00) Madrid
Europe/Rome	(+01:00) Rome
Africa/Algiers	(+01:00) Algiers
Africa/Tunis	(+01:00) Tunis
Europe/Warsaw	(+01:00) Warsaw
Europe/Prague	(+01:00) Prague Bratislava
Europe/Vienna	(+01:00) Vienna
Europe/Budapest	(+01:00) Budapest
Europe/Sofia	(+02:00) Sofia
Europe/Istanbul	(+02:00) Istanbul
Europe/Athens	(+02:00) Athens
Asia/Nicosia	(+02:00) Nicosia
Asia/Beirut	(+02:00) Beirut
Asia/Damascus	(+02:00) Damascus
Asia/Jerusalem	(+02:00) Jerusalem
Asia/Amman	(+02:00) Amman
Africa/Tripoli	(+02:00) Tripoli
Africa/Cairo	(+02:00) Cairo
Africa/Johannesburg	(+02:00) Johannesburg
Europe/Moscow	(+03:00) Moscow
Asia/Baghdad	(+03:00) Baghdad
Asia/Kuwait	(+03:00) Kuwait

Table A–1 Real-Time Collaboration Time Zones

Internal Name	External User Visible Name
Asia/Riyadh	(+03:00) Riyadh
Asia/Bahrain	(+03:00) Bahrain
Asia/Qatar	(+03:00) Qatar
Asia/Aden	(+03:00) Aden
Africa/Khartoum	(+03:00) Khartoum
Africa/Djibouti	(+03:00) Djibouti
Africa/Mogadishu	(+03:00) Mogadishu
Asia/Dubai	(+04:00) Dubai
Asia/Muscat	(+04:00) Muscat
Asia/Yekaterinburg	(+05:00) Yekaterinburg
Asia/Tashkent	(+05:00) Tashkent
Asia/Calcutta	(+05:30) India
Asia/Novosibirsk	(+06:00) Novosibirsk
Asia/Almaty	(+06:00) Almaty
Asia/Dacca	(+06:00) Dacca
Asia/Krasnoyarsk	(+07:00) Krasnoyarsk
Asia/Bangkok	(+07:00) Bangkok
Asia/Saigon	(+07:00) Vietnam
Asia/Jakarta	(+07:00) Jakarta
Asia/Irkutsk	(+08:00) Irkutsk
Asia/Shanghai	(+08:00) Beijing, Shanghai
Asia/Hong_Kong	(+08:00) Hong Kong
Asia/Taipei	(+08:00) Taipei
Asia/Kuala_Lumpur	(+08:00) Kuala Lumpur
Asia/Singapore	(+08:00) Singapore
Australia/Perth	(+08:00) Perth
Asia/Yakutsk	(+09:00) Yakutsk

Table A–1 Real-Time Collaboration Time Zones

Internal Name	External User Visible Name
Asia/Seoul	(+09:00) Seoul
Asia/Tokyo	(+09:00) Tokyo
Australia/Darwin	(+09:30) Darwin
Australia/Adelaide	(+09:30) Adelaide
Asia/Vladivostok	(+10:00) Vladivostok
Australia/Brisbane	(+10:00) Brisbane
Australia/Sydney	(+10:00) Sydney Canberra
Australia/Hobart	(+10:00) Hobart
Asia/Magadan	(+11:00) Magadan
Asia/Kamchatka	(+12:00) Kamchatka
Pacific/Auckland	(+12:00) Auckland

Glossary

DMZ

The "demilitarized zone" is a subnetwork located between a company's corporate intranet and the Internet.

Document Conversion Server

Document Conversion Server enables the Web Conferencing Application to automatically convert documents for the Document Presentation mode.

Oracle Web Conferencing Application

The set of components used to administer the Web Conferencing system. It consists of the application Web pages, Web listener, and the Document Conversion Server.

Oracle Web Conferencing Console

The console where real-time conferences take place.

Load Balancer

A very fast network device which can distribute Web requests to a large number of physical servers. The purpose of a load balancer (LBR) is to provide a single published address to the client browsers, and, in the case of Real-Time Collaboration, provide multiple Real-Time Collaboration Core Components mid-tiers which actually service the requests, based on the distribution of the requests done by the LBR.

Web Conferencing Server

The backend server that manages all of the runtime state associated with a conference. Clients never directly communicate with the Web Conferencing Server.

Instead, clients communicate with the HTTP Listener, which uses mod_imeeting to communicate with the appropriate Web Conferencing Server.

Voice Conversion Server

The Voice Conversion Server enables listen-only voice streaming by capturing voice stream from the Dialogic card, encoding it using GSM codec, and streaming it to the Web Conferencing Server for recording or listen-only voice support.

Index

P

Q

R

S

U

V