Administrator's Guide

Sun™ ONE Instant Messaging

Version 6.0

816-6677-10
May 2003

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.

Copyright 2003 Sun Microsystems, Inc. All rights reserved.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, Sun ONE, iPlanet, and all Sun, Java, and Sun ONE based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

Netscape is a trademark or registered trademark of Netscape Communications Corporation in the United States and other countries.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or this document may be reproduced in any form by any means without prior written authorization of Sun Microsystems, Inc. and its licensors, if any.

THIS DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2003 Sun Microsystems, Inc. Tous droits réservés.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, Sun ONE, et iPlanet sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et d’autre pays.

UNIX est une marque enregistree aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd.

Netscape est une marque de Netscape Communications Corporation aux Etats-Unis et dans d'autres pays.

Le produit décrit dans ce document est distribué selon des conditions de licence qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ni de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit sans l’autorisation écrite préalable de Sun Microsystems, Inc. et, le cas échéant, de ses bailleurs de licence.

CETTE DOCUMENTATION EST FOURNIE “EN L'ÉTAT”, ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE GARANTIE IMPLICITE D'APTITUDE À LA VENTE, OU À UN BUT PARTICULIER OU DE NON CONTREFAÇON SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Contents

About This Guide **7**

Monospaced Font 8
Bold Monospaced Font 9
Italicized Font 9
Square or Straight Brackets 10
Command-Line Prompts 10

Chapter 1 Introduction to Instant Messaging **11**
Sun ONE Instant Messaging Components 11
Standalone Deployment Overview 13
Portal Deployment Overview 14
Sun ONE Instant Messenger 16
Sun ONE Portal Server 17
Sun ONE Portal Server, Secure Remote Access 17
Sun ONE Identity Server 18
Instant Messaging Server 18
Sun ONE Instant Messaging Multiplexor 18
Web Server 19
LDAP Directory Server 20
SMTP Server 20
Sun ONE Instant Messaging Deployment Configurations 21
The Web Server and the Instant Messenger Resources Installed on a Different Host 21
Multiple Multiplexor Hosts 23
Multiple Instant Messaging server Hosts 25
Configuration Files and Directory Structure 27
Instant Messaging server Directory Structure 27
Instant Messaging server Configuration File 28

Sun ONE Instant Messaging Data 29
Using SSL in Sun ONE Instant Messaging 29
Sun ONE Instant Messaging Access Control 30
Global Access Control 30
Conference Room and News Channel Access Controls 31
Presence Access Control 32
Instant Messaging Administrator Access Control 32
Using the Command-Line with Sun ONE Instant Messaging 32

Chapter 2 Administering Instant Messaging Server and Multiplexor 33
Administering Instant Messaging an Overview 33
User Administration 34
Stopping and Starting the Server and Multiplexor (On Unix) 35
Changing Instant Messaging Server and Multiplexor Configuration Parameters 37
Managing Logging 38
Logging Levels 38
Managing User Privileges 40
Global Access Control 40
Conference Room and News Channel Access Controls 42
Presence Access Control 43
Instant Messaging Administrator Access Control 43
Access Control File Format 44
Access Control File Examples 45
sysTopicsAdd.acl File 45
Room and News Channels ACL Files 45
Federating Deployment of Multiple Instant Messaging Servers 47
Configuring SSL 50
Configuring SSL between Sun ONE Instant Messenger and Multiplexor 56
Requesting a Certificate From The Certificate Authority 57
Installing the Certificate 58
Configuring the Instant Messaging server to enable SSL between the Multiplexor and the Instant Messenger 59
Invoking The Secure Version Of Instant Messenger 63
Managing Sun ONE Instant Messaging's LDAP Configuration 63
Searching the Directory as Anonymous Users 64
Configuring Dynamic LDAP Server Group 65
Displaying Calendar Reminders and Notifications as Instant Messenger Popups 66
Configuring Calendar Server for Displaying Calendar Reminders and Notifications as Instant Messenger popups 68
Enabling Alarms 69
Configuring Instant Messaging Server for Displaying Calendar Reminders and Notifications as Instant Messenger Popups 69
Example for Displaying Calendar Reminders and Notifications as Instant Messenger Popups . 74

Configuring Calendar Server 74
Configuring Instant Messaging Server 74
Enabling Calendar Alerts in Instant Messenger 75
Troubleshooting Display Calendar Reminders and Notifications as Instant Messenger Popups 75
Taking a Back Up of Instant Messaging Data 76
Backup Information 76
Performing Backup 77
Restoring the Back Up Information 77

Chapter 3 Managing Instant Messenger 79
Configuring Sun ONE Instant Messenger 79
Invoking Instant Messenger 80
Solving Web Server Issues 81
Changing Web Server Port 83
Customizing Instant Messenger 83
Instant Messenger Resources 84
Instant Messenger Files 84
Customizing the index.html and im.html Files (Only Standalone Deployments) 86
Customizing the Application (Java Web Start) 87
Contents Listing of imbrand.jar 89
Rebranding Instant Messenger 91
Customizing User Name Display 92
Customizing User Name Display in Search Results 92
Customizing User Name Display in Tooltip 93
Administering Instant Messenger Conference Rooms and News Channels 93
Granting Users Privilege to Create Conference Rooms and News Channels 94
Modifying Instant Messenger Proxy Settings 94
Controlling The Exposed Messenger Feature Set 95
Instant Messenger Data Stored in the End User's System 97

Chapter 4 Managing Instant Messaging Policies Using Identity Server 99
Instant Messaging Service 99
Restricting Users From Accessing Instant Messenger 100
Unassigning Instant Messaging Policy 100
To Unassign the IM Policy from the Organization 101

Chapter 5 Managing The Instant Messaging Archive 103
Instant Messaging Archive Overview 103
Archiving Instant Messages 105
Enabling the Archive Provider 106
Configuring the Archive Provider 107
Archive Provider Configuration Parameters 109

Managing Archived Data in the Portal Server Search Database 113
rdmgr Command 114
Searching Resource Descriptors (RD) 114
Deleting Resource Descriptors 115
Enabling Instant Messenger Archive Control 115
Changing the Display of the Archived Data 117
How to Use CGI Script to Submit the Data 118
Sample Deployment Scenario for Archive Provider 119

Chapter 6 Managing Instant Messaging Portal Channel 121
Introducing Instant Messaging Channel 121
Display Profile Properties for Instant Messaging Portal Channel 122
Instant Messaging Portal Channel Tag Library Reference 123
Customizing Instant Messaging Portal Channel 126
Customizing Instant Messaging Channel Content Page 128
Displaying Presence Information For All Contacts 132
Single Instant Messaging Server Deployment Configuration 134
Automatically Closing the Invite Window 134
Inserting Instant Messenger Links in an Organization 135
Enabling Secure Mode for Sun ONE Instant Messenger in Sun ONE Portal Server 137
Disallowing Users from Launching Instant Messenger 139

Appendix A Instant Messaging Configuration Parameters 141
Using the iim.conf file 141
General Configuration Parameters 143
User Source Configuration Parameters 145
Logging Configuration Parameters 148
Instant Messaging Server Configuration Parameters 150
Multiple Server Configuration Parameters 155
Multiplexor Configuration Parameters 157

Appendix B Sun ONE Instant Messaging Reference 161
imadmin 161
Synopsis 162
Options 162
Actions 162
Components 163

Appendix C Instant Messaging APIs 165
Sun ONE Instant Messaging APIs Overview 165
Instant Messaging Service API 165
Messenger Beans 166

Service Provider Interfaces 166
Archive Provider API 166
Message Conversion API 167
Authentication Provider API 168

Appendix D Troubleshooting Instant Messaging 169
The Messenger client does not load or start 170
Connection refused and timed out 171
Authentication errors 171
IM channel display error 171
Instant Messaging content is not archived 172
Server-to-server communication fails to start 172

Index 173

About This Guide

This manual describes how to administer Sun™ Open Network Environment (Sun ONE) Instant Messaging and its accompanying software components.

This preface contains the following sections:

- Who Should Read This Book
- What You Need to Know
- How This Book is Organized
- Conventions Used in This Manual

Who Should Read This Book

You should read this book if you are responsible for administering, configuring, and deploying Instant Messaging.

What You Need to Know

This book assumes that you are responsible for configuring, administering, and maintaining Instant Messaging, and you have an understanding of the following:

- JavaScript™
- HTML
- Sun ONE Portal Server
- Sun ONE Web Server Enterprise Edition
- Sun ONE Directory Server

- Sun ONE Identity Server

How This Book is Organized

This book contains the following chapters and appendices:

- About This Guide (this chapter)
- Chapter 1, “Introduction to Instant Messaging”

  This chapter describes the Instant Messaging components, architecture, and configurations.
- Chapter 2, “Administering Instant Messaging Server and Multiplexor”

  This chapter describes how to administer Sun ONE Instant Messaging server and multiplexor.
- Chapter 3, “Managing Instant Messenger”

  This chapter describes how to customize and administer the Sun ONE Instant Messenger.
- Appendix A, “”

  This appendix describes the settings you can configure for Instant Messaging.
- Appendix B, “Sun ONE Instant Messaging Reference”,

  This appendix describes the `iimadmin` command used to administer Instant Messaging.
- Appendix C, “Instant Messaging APIs”

  This appendix explains the APIs used by Sun ONE Instant Messaging.
- Appendix D, “Troubleshooting Instant Messaging”

  This appendix lists the common problems that may occur during installation and deployment of Instant Messaging.

Conventions Used in This Manual

Monospaced Font

`Monospaced font` is used for any text that appears on the computer screen or text that you should type. It is also used for file names, distinguished names, functions, and examples.

Bold Monospaced Font

Also, all paths specified in this manual are in Unix format. If you are using a Windows NT-based Instant Messaging, you should assume the Windows NT equivalent file paths whenever Unix file paths are shown in this book.

`Bold monospaced font` is used to represent text within a code example that you should type. For example, you might see something like this:

```
./setup
Copyright (c) 2001 Sun Microsystems, Inc.  All rights reserved.  Use
is subject to license terms.  Sun, Sun Microsystems, the Sun logo,
Java, Solaris and iPlanet are trademarks or registered trademarks of
Sun Microsystems, Inc.in the U.S. and other countries.  Federal
Acquisitions:  Commercial Software - Government Users Subject to
Standard License Terms and Conditions.

Copyright (c) 2001 Sun Microsystems, Inc.  Tous droits réservés.
Distribué par des licences qui en restreignent l'utilisation.  Sun,
Sun Microsystems, le logo Sun, Java, Solaris et iPlanet sont des
marques de fabrique ou des marques déposées de Sun Microsystems,
Inc.  aux Etats-Unis et dans d'autres pays.

================================================================

Verifying permissions
Verifying java available
Found java (/usr/j2se/bin/java) version (1.3.0) in the system.
Verifying installation components available
Verifying directories available
Verifying files available
Starting install wizard in graphical mode
```

In this example, **`./setup`** is what you would type from the command-line and the rest is what would appear as a result.

Italicized Font

Italicized font is used to represent text that you enter using information that is unique to your installation (for example, variables). It is used for server paths and names and account IDs.

Square or Straight Brackets

Square (or straight) brackets `[]` are used to enclose optional parameters. For example, in this document you will see the usage for the `iimadmin` command described as follows:

```
iimadmin [options] [action] [component]
```

The presence of `[options]`, `[arguments]`, and `[component]` indicates that there are optional parameters that may be added to the `iimadmin` command.

Command-Line Prompts

Command-line prompts (for example, `%` for a C-Shell, or `$` for a Korn or Bourne shell) are not displayed in the examples. Depending on which operating system environment you are using, you will see a variety of different command-line prompts. However, you should enter the command as it appears in the document unless specifically noted otherwise.

Chapter 1

Introduction to Instant Messaging

This chapter explains the Sun™ ONE Instant Messaging components, architecture, and configuration information.

The chapter contains the following sections:

- Sun ONE Instant Messaging Components
- Sun ONE Instant Messaging Deployment Configurations
- Configuration Files and Directory Structure
- Using SSL in Sun ONE Instant Messaging
- Sun ONE Instant Messaging Access Control
- Using the Command-Line with Sun ONE Instant Messaging

Sun ONE Instant Messaging Components

Instant Messaging server enables web clients to participate in real-time messaging and automatically distribute information. Sun ONE Instant Messaging allows users to participate in instant messaging and chat sessions, send alert messages to each other, and share group news instantly. It is suitable for both intranets and the Internet.

Instant Messaging can be deployed in any one of the following methods:

- As part of the Sun ONE Portal Server environment, Sun ONE Instant Messenger is made available to end users from the Sun ONE Portal Server desktop on the Solaris platform only.
- As a standalone server.

The Instant Messaging components are the same, whether you install and configure Instant Messaging with Sun ONE Portal Server, or as a standalone server. The Instant Messaging components are:

- **Sun ONE Instant Messenger Resources.** Also referred as the messenger client. This is a java-based instant messaging client that is invoked through the web, using the Java™ Web Start or the Java Plug-in.
- **Instant Messaging Server.** The Instant Messaging server serves the presence information to the messenger clients, allows users to establish instant messaging sessions, and enforces policies. The Instant Messaging server is also referred as the server.
- **Instant Messaging Multiplexor.** A scalability component that consolidates multiple messenger connections into one Transmission Control Protocol (TCP) and connects to the server. The Instant Messaging multiplexor is also referred as the multiplexor.
- **Sun ONE Identity Server Instant Messaging Service Definition.** This component can be installed only if the Identity Server or the Identity Server SDK is installed in the system
- **Sun ONE Portal Server IM Channel and Archive.** This component can be installed only if the Portal Server is installed on the system. This component consist of the Instant Messaging Portal channel on the Portal Server Desktop.

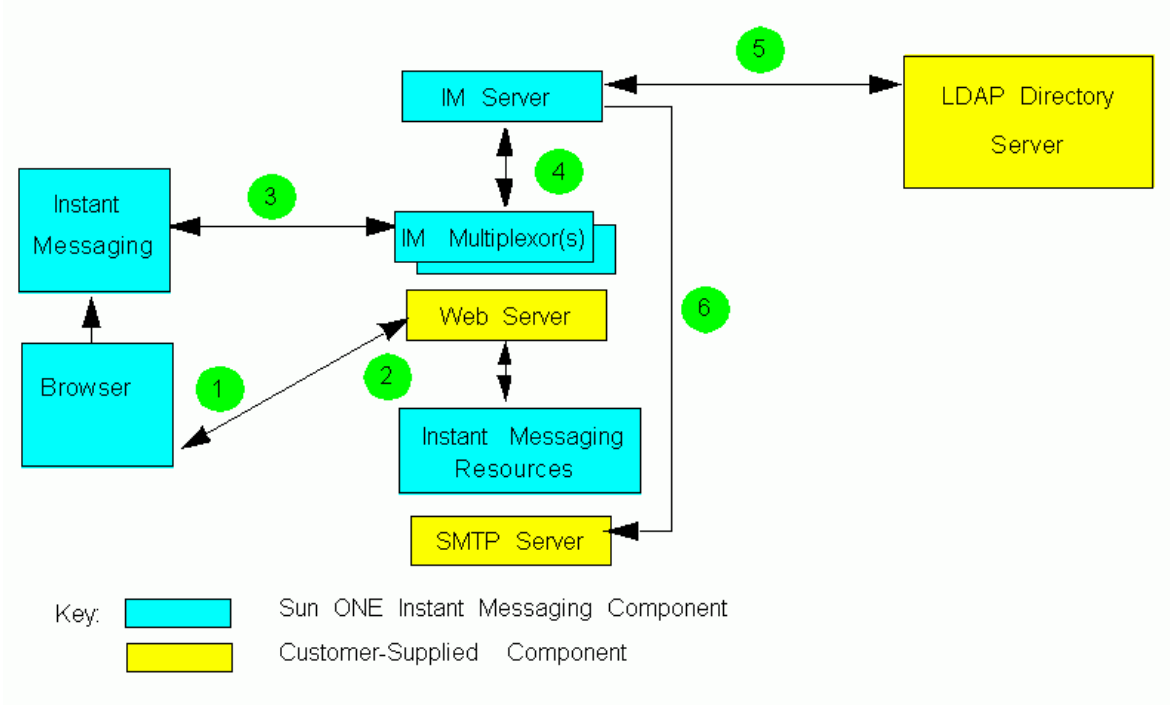
The other software components installed separately from Sun ONE Instant Messaging are:

- (Optional) Sun ONE Portal Server: Sun ONE Portal Server is installed for portal deployments.
- Web server: Portal deployments use the web server that ships with Sun ONE Portal Server. The standalone deployments need to install a web server, such as Sun ONE Web Server Enterprise Edition. In both cases, the Instant Messenger resources must reside on the web server host machine.
- LDAP directory server: Instant Messaging uses an LDAP server, such as Sun ONE Directory Server, for user authentication and user search. In a portal deployment, the LDAP server used by the Portal Server is used by the Instant Messaging server to search users.
- SMTP server: Sun ONE Messaging Server or some other SMTP server is used to forward instant messages for users who are offline.
- (Optional) Sun ONE Identity Server: Sun ONE Identity Server is installed for adding Instant Messaging service.

Standalone Deployment Overview

Figure 1-1 illustrates the interaction of the software components in a standalone configuration of Instant Messaging.

Figure 1-1 The software components in a standalone deployment.



An Instant Messaging standalone deployment works as follows:

1. The user accesses the Sun ONE Instant Messenger applet URL from a browser
2. The browser invokes Java Web Start or the Java Plug-in. Java Web Start or the Java plug-in downloads the necessary Sun ONE Instant Messenger resource files and starts the Instant Messenger.

3. The login window is displayed and the user enters the LDAP user name and password. This data is sent to the multiplexor.
4. The multiplexor routes the data received from the messenger to the backend Instant Messaging server.
5. The Sun ONE Instant Messaging server communicates with the LDAP server to authenticate the user.

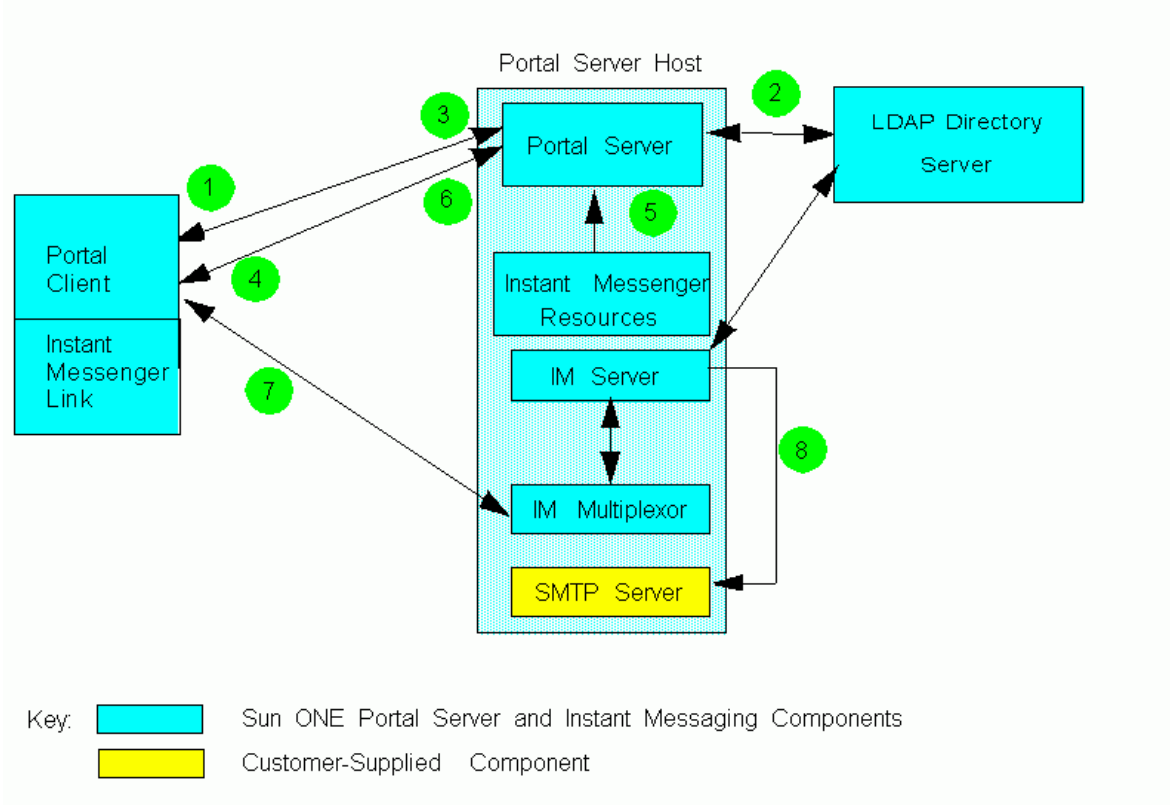
   When the user authentication is complete, the Sun ONE Instant Messaging main window is displayed with the contact list for the user. The user can now start and participate in instant messaging sessions with the other users.
6. An SMTP server, forwards the alerts as e-mails to the users when notified by the server that the users are offline.

NOTE The users should set their preferences to have alert forwarded as e-mail when they are offline.

Portal Deployment Overview

Figure 1-2 illustrates the collaboration of Sun ONE Portal Server and Instant Messaging software components.

Figure 1-2 The software components in a portal deployment.



Instant Messaging in a portal deployment works as follows:

1. The user logs in to the Sun ONE Portal Server by entering the URL in a web browser.
2. The Sun ONE Portal Server software authenticates the user with the configured authentication mechanism.
3. The Sun ONE Portal Server downloads the Sun ONE Portal Server Desktop for the user.
4. The user can access the contact list information from the Instant Messaging Contact List channel.

5. The existing Sun ONE Portal Server session ID is used to set up a session with the Instant Messaging multiplexor. The launch servlet gets the user details from Sun ONE Portal Server.

NOTE The user need not re-authenticate to Instant Messaging as the portal authentication token is used to log the user into Instant Messaging.

6. When the user clicks the Sun ONE Instant Messenger link on the Sun ONE Portal Server Desktop, the Sun ONE Instant Messaging main window is displayed with the list of available contacts for the user. The user can now start and participate in the instant messaging sessions with the other users.
7. Sun ONE Instant Messenger connects to the multiplexor.
8. An SMTP server, forwards the alerts as e-mails to the users when notified by the server that the users are offline. The users should set their preferences to have alert forwarded as e-mail when they are offline.

For more information on deploying Sun ONE Instant Messaging in the portal environment, see the *Sun ONE Instant Messaging Deployment Guide.*

Sun ONE Instant Messenger

The Java-based Sun ONE Instant Messenger is Instant Messaging's client that can be configured to be a browser-based applet using Java Plug-in, or an application independent of a browser using Java Web Start.

To run the Sun ONE Instant Messenger client on Solaris, you must use Java Web Start. On Microsoft Windows you can run Instant Messenger as an applet or a Java Web Start application. It is recommended to run Sun ONE Instant Messenger as a Java Web Start application.

For more information on customizing Sun ONE Instant Messenger, see Managing Instant Messenger.

Sun ONE Instant Messenger provides the following modes of communication:

- **Chat -** Sun ONE Instant Messenger's version of instant messaging conferences is called chat. Chat is a real-time conversation capability that enables users to complete projects, answer customer queries, and complete other time-critical assignments. Chat sessions are held either in chat rooms created on a need basis, or in pre-established conference rooms.

- **Alerts -** Alerts enable information delivery and response to users through the Instant Messenger interface. Alerts can deliver time-critical information to the user. The sender of the alert message is notified when the message is delivered, and read by the recipient. If the alert message requires a response, choose the Chat option from the Tools menu to chat with the sender.
- **Poll** - The polling function enables you to poll users for their response to a question. You can send a question and possible answers to poll recipients, and the recipients can respond with their selected answer. When recipients respond to your poll, you can view their answers in a Status window. The summary of results can also be viewed in the status window.
- **News Channels -** News channels are forums for posting and sharing information. The users can subscribe to news channels of interest to see updates using the URL of the news channels or view the news channel updates through static messages. Administrators control news channel access by assigning users to the channels they need, and deciding who can see or post information to the channels.

NOTE The instant messages can contain embedded URLs, such as `http://stocks.yahoo.com?id=sunw`. If you are using proxy servers, it might be necessary to have clients using Java Web Start modify their proxy configuration for resolving such URLs.

For more information on configuring the proxy settings manually, see Modifying Instant Messenger Proxy Settings.

Sun ONE Portal Server

Sun ONE Instant Messenger installed on the Portal Server environment can be launched from the Instant Messaging channel that appears on the Sun ONE Portal Server Desktop of the user.

Sun ONE Portal Server, Secure Remote Access

Sun ONE Portal Server, Secure Remote Access provides secure access to remote users in an intranet for Solaris-based or Windows-based systems. The user can access Secure Remote Access by logging in to the web-based Sun ONE Portal Server Desktop through the portal gateway. The authentication module configured for Sun ONE Portal Server authenticates the user. The user session is established with Sun ONE Portal Server and the access is enabled to the user's portal desktop.

In the Sun ONE Portal Server environment, you can configure Sun ONE Instant Messenger in either secure or non-secure mode. In the secure mode, communication is encrypted through the Sun ONE Portal Server Netlet. When you are accessing Sun ONE Instant Messenger in the secured mode, a lock icon appears in the Status area of the Instant Messenger. In the non-secure mode, the Sun ONE Instant Messenger session is not encrypted. For more information on Netlet, see *Sun ONE Portal Server, Secure Remote Access Administrator's Guide* at:

`http://docs.sun.com/db/doc/816-6359-10`

Sun ONE Identity Server

Sun ONE Identity Server provides user and service management, authentication and single sign-on services. It also provides policy management, logging service, debug utility, the admin console, and client support interfaces.

Instant Messaging Server

The Instant Messaging server handles tasks such as controlling Instant Messenger privileges and security, enabling Sun ONE Instant Messenger clients to communicate with each other by sending alerts, initiating chat conversations, and posting messages to the available news channels.

The Instant Messaging server supports the connection of a multiplexor that consolidates connections over one socket. For more information on the multiplexor, see Sun ONE Instant Messaging Multiplexor.

Access controls are used for administration of users, news channels, and conference rooms. These access controls are implemented by the Instant Messaging server and not the LDAP directory server.

Sun ONE Instant Messaging Multiplexor

The Instant Messaging multiplexor component connects multiple instant messenger connections into one TCP (Transmission Control Protocol) connection, which is then connected to the backend Instant Messaging server. The multiplexor reads data from the Sun ONE Instant Messenger and writes it to the server.

Conversely, when the server sends data to Sun ONE Instant Messenger, the multiplexor reads the data and writes it to the appropriate connection. The multiplexor does not perform any user authentication or parse the client-server protocol.

The multiplexor always acts as the front-end component for the Instant Messaging server. Any client-server communication must go through the multiplexor. The Instant Messaging server architecture is designed in such a way that it always uses the multiplexor along with Sun ONE Instant Messenger and Instant Messaging server to communicate with each other.

You can install multiple multiplexors based on your deployment requirements. For more information, see Sun ONE Instant Messaging Deployment Configurations.

Web Server

Instant Messaging requires a web server to serve up the Instant Messenger resources. The Instant Messenger resource files include:

- The initial `index.html` file, provided by Sun ONE Instant Messenger, or a home page with a link to invoke Sun ONE Instant Messenger.
- Sun ONE Instant Messenger jar files (`im.jar`, `imres.jar`, `imbrand.jar`, `imnet.jar`, and `imjni.jar`).
- The Sun ONE Instant Messenger Online Help.
- Embedded URLs in the instant messages and the news channels in Sun ONE Instant Messenger. Instant Messaging supports web servers, such as Sun ONE Web Server Enterprise Edition.

You must install Sun ONE Instant Messenger on the same host where the web server is installed. In a Portal deployment, Sun ONE Instant Messenger can be installed on the Sun ONE Portal Server host or on a different web server host. In most cases, the Instant Messenger will be installed on the same host where you installed the Instant Messaging server software. It is possible to locate the Sun ONE Instant Messenger software on a host other than the Instant Messaging server or multiplexor. For more information on this, see *Sun ONE Instant Messaging Installation Guide.*

NOTE Install the web server before installing Sun ONE Instant Messaging.

If you are using Sun ONE Portal Server, you can use the web server that is shipped with the product. You need not install a separate web server for Instant Messaging.

LDAP Directory Server

Sun ONE Instant Messaging in a standalone mode requires an LDAP directory server to perform user authentication and to search for users.

Sun ONE Instant Messaging in a portal deployment environment uses the directory used by Sun ONE Portal Server. In a portal deployment environment, the Instant Messaging uses the directory only for user search and not for user authentication.

The Sun ONE Instant Messaging server does not store the Instant Messenger user information instead the Instant Messenger user information is stored in the LDAP server. For performing user search in the LDAP server, the Instant Messaging server uses the LDAP `cn` and `uid` attributes.

Sun ONE Instant Messaging supports users defined and maintained in an LDAP directory, such as Sun ONE Directory Server.

Instant Messaging does not ship with an LDAP directory server. If you do not have an LDAP directory installed, you must install one. For more information, see *Sun ONE Instant Messaging Installation Guide.*

SMTP Server

Instant Messaging uses an SMTP server to forward alerts as emails to users who are offline and are therefore unable to receive alerts.

The SMTP server is not shipped with Instant Messaging. If you do not have an SMTP server installed, you must install one. For more information, see *Sun ONE Instant Messaging Installation Guide.*

Sun ONE Instant Messaging Deployment Configurations

You can install and configure Sun ONE Instant Messaging server to meet your site's requirements. The following are some of the Instant Messaging deployment scenarios:

- Sun ONE Instant Messaging deployment with separate web server host
- Sun ONE Instant Messaging deployment with multiple multiplexor hosts
- Sun ONE Instant Messaging deployment with multiple Instant Messaging server hosts

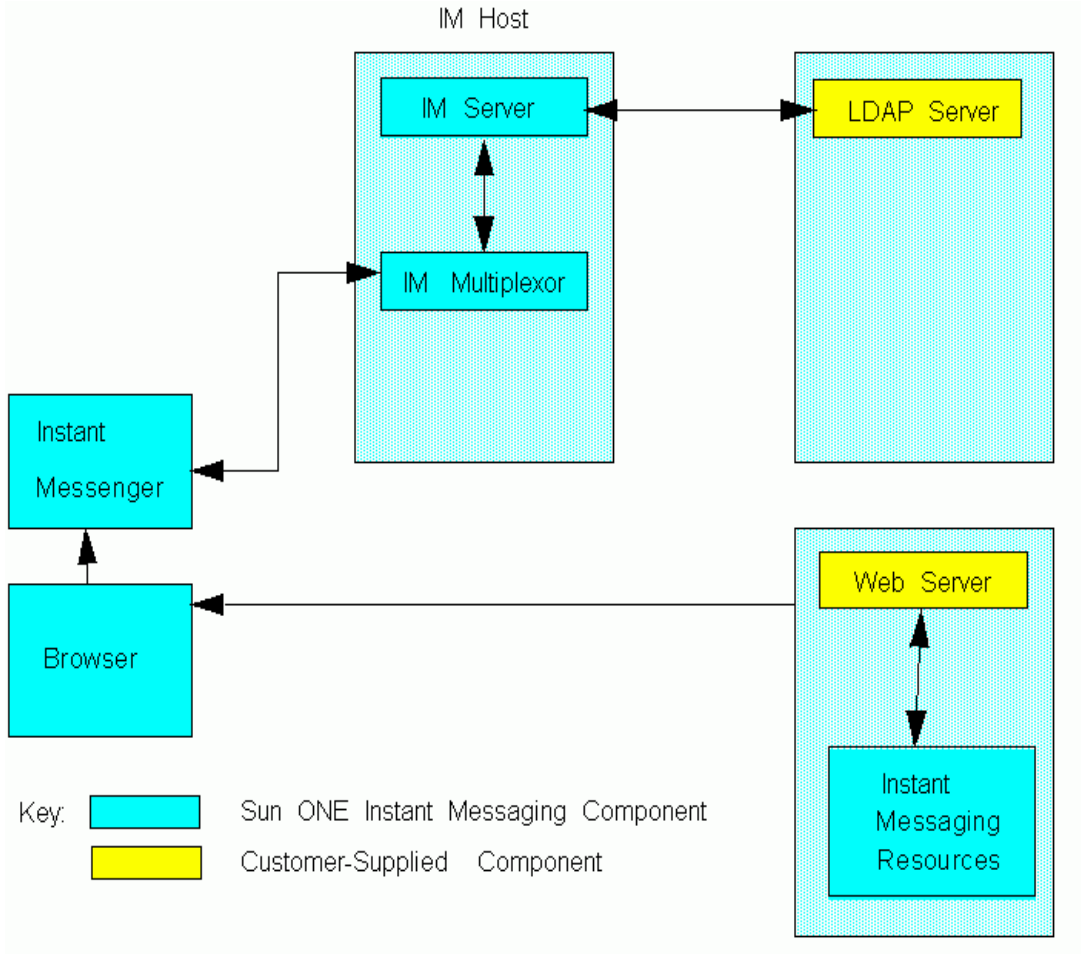
NOTE The configuration details described in this section are for standalone deployments of Instant Messaging.

For information on configuring Instant Messaging on portal deployment, see *Sun ONE Instant Messaging Deployment Guide*.

The Web Server and the Instant Messenger Resources Installed on a Different Host

Figure 1-3 shows a configuration where the Instant Messaging server and multiplexor are installed on the same host, and the web server is installed on a separate host. Use this configuration when there is an existing instance of web server and LDAP server, and you do not want to install other applications on these hosts.

Figure 1-3 The web server and the Instant Messenger installed on a separate host.



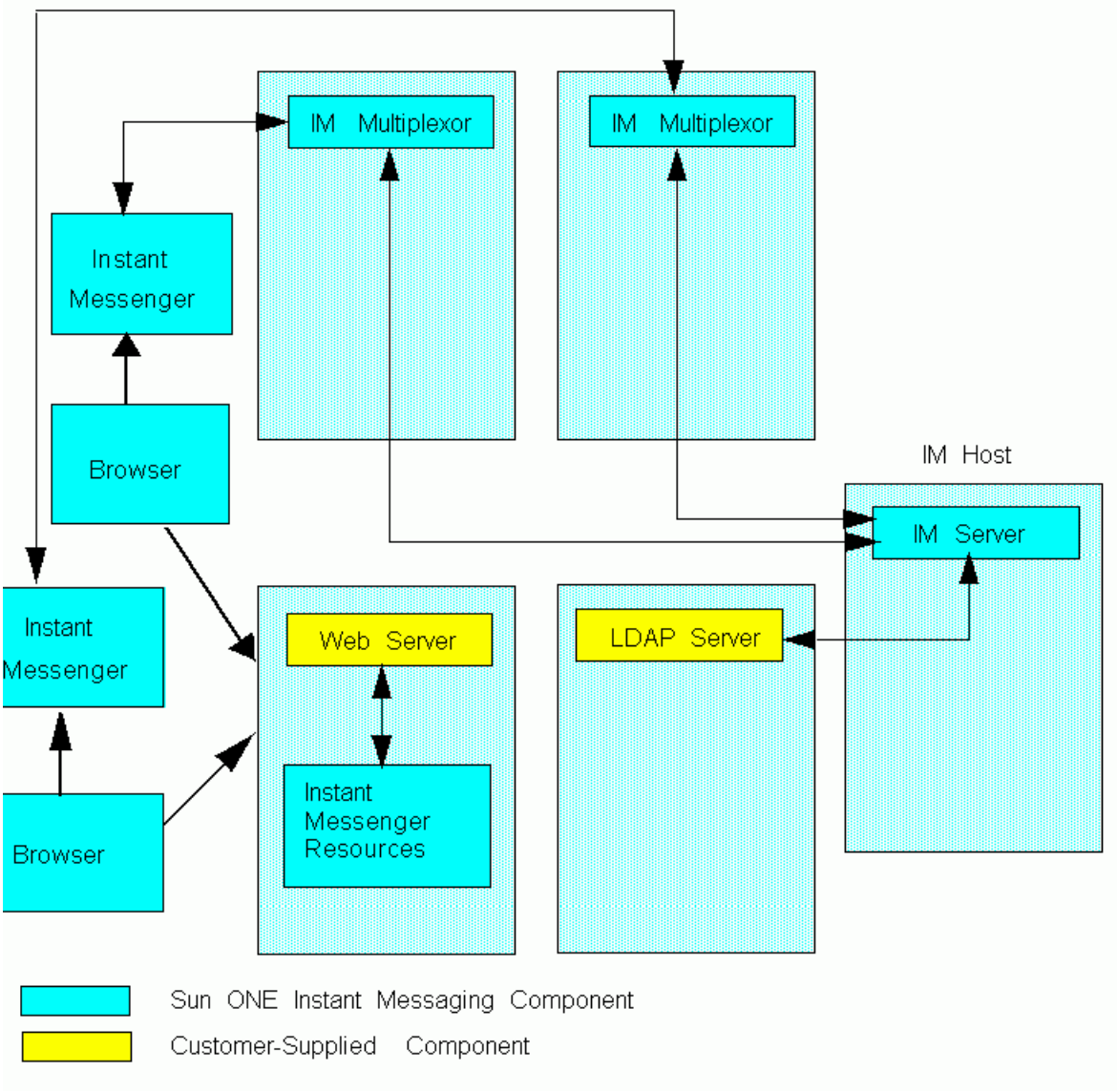
Multiple Multiplexor Hosts

Figure 1-4 shows a configuration of two multiplexors installed on separate hosts, and the Instant Messaging server on a different host. This configuration enables you to place a multiplexor outside your company's firewall. Installing multiplexors on multiple hosts distributes the load of the Instant Messaging server across multiple systems.

NOTE

- The multiplexor can be resource-intensive, so putting it on a separate host can improve the overall performance of the system.
- Windows supports only one multiplexor instance per host.

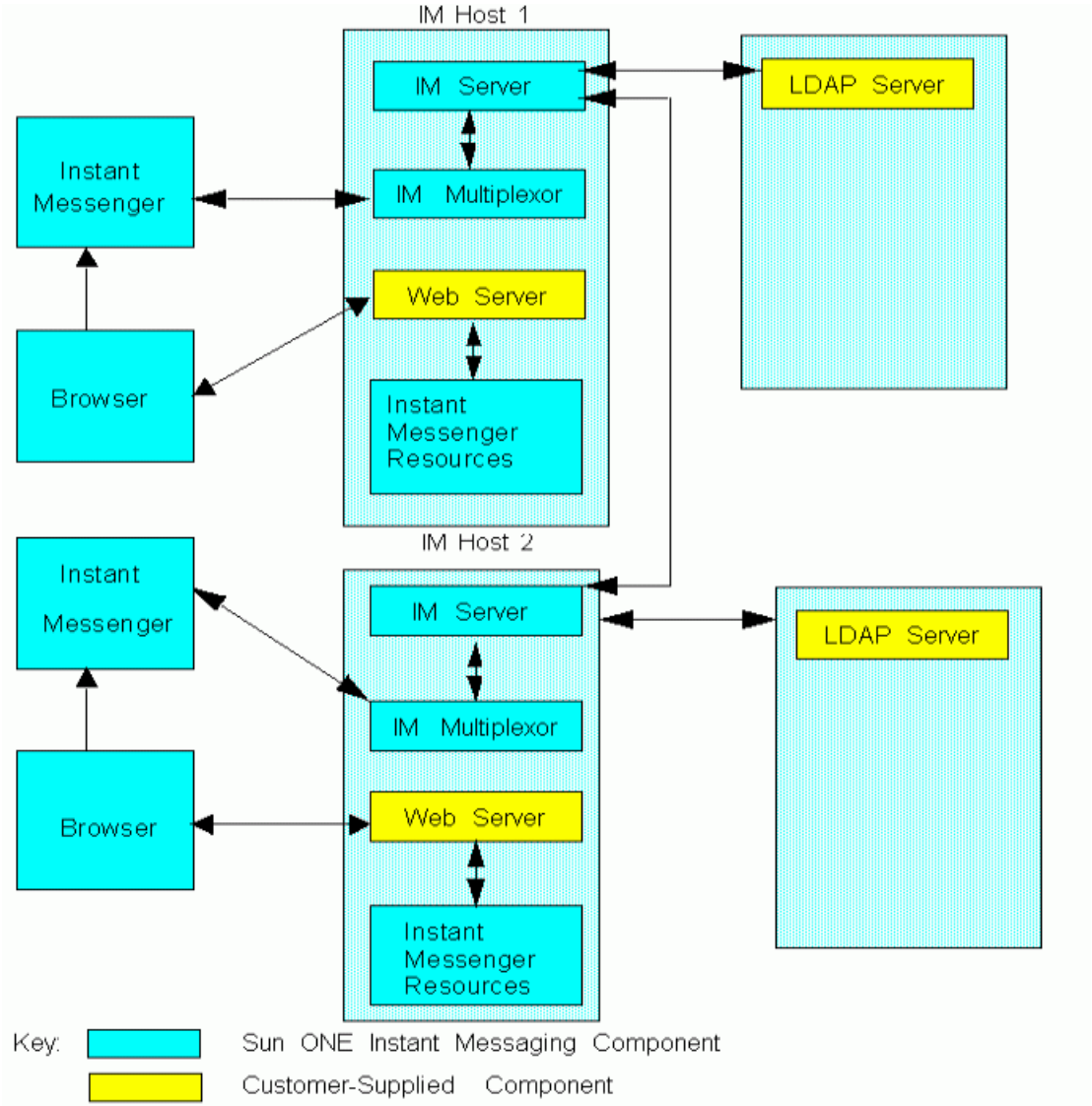
Figure 1-4 Instant Messaging Multiplexors installed on two different hosts.



Multiple Instant Messaging server Hosts

Figure 1-5 shows a configuration consisting of two Instant Messaging servers. This configuration is used when the site contains multiple administrative domains. The server configuration on each Instant Messaging server host has to be set up so that users on one Instant Messaging server can communicate with users on other Instant Messaging servers.

Figure 1-5 Multiple Instant Messaging server hosts.



Configuration Files and Directory Structure

This section describes the Instant Messaging server directory structure and the properties files used to store Instant Messaging operational data and configuration information.

Instant Messaging server Directory Structure

Table 1-1 shows the platform-specific directory structure for the Instant Messaging server.

Table 1-1 Instant Messaging server directories

Description	Solaris Location	Windows Location
Programs Files These files include the native executable files, the library files in `bin` or `lib` directory, the shell scripts in `sbin` directory, the java classes in `classes` directory, and templates files in `lib` directory.	*im_install_dir*`/SUNWiim` The default value for *im_install_dir* is `/opt`	*im_install_dir* The default value for *im_install_dir* is `c:\Program Files\Sun\InstantMessaging`
Server Configuration files These files include the `iim.conf` file, the `acl` files and a subdirectory which contains all the server-wide access control files.	`/etc/opt/SUNWiim/default/config` Note: The installer creates a symbolic link from `/etc/opt/SUNWiim/default/config` to *im_install_dir*`/SUNWiim/config`.	*im_install_dir*`\config`

Table 1-1 Instant Messaging server directories *(Continued)*

Description	Solaris Location	Windows Location
Instant Messaging Server Data. These files include the configurable directory for the files generated by the server at runtime. It includes the user data in the `db` directory, which contains information such as the user and news channels directory. It also contains the server and multiplexor log files, in the `log` directory.	*instancevardir*`/default` The default value for *instancevardir* is `/var/opt/SUNWiim`	*im_install_dir* \
Instant Messenger resources. These files contain HTML documents and `jar` files used by Sun ONE Instant Messenger. The top-most directory contains the locale-independent resources, and the locale-specific directories contain the localized resources.	*imdocroot*	*imdocroot*

NOTE On Linux, the primary server package name is `soim`, and all the above Solaris Location paths mentioned in Table 1-1 should be replaced by `soim`. For example, replace `SUNWiim` with `soim`.

Instant Messaging server Configuration File

Instant Messaging stores all configuration options in the `iim.conf` file. For more information on the parameters and their values stored in this file, see Instant Messaging Configuration Parameters.

Sun ONE Instant Messaging Data

Instant Messaging server stores the following data used by Sun ONE Instant Messenger in the runtime files directory, which you specified during the installation, and is indicated by the `iim.instancevardir` parameter in the `iim.conf` file:

- User properties, such as contact lists, messenger settings, subscribed news channels and access control.
- News channel messages and access rules.
- Alert Messages that are to be delivered. These messages are delivered and removed when the recipient logs in.
- Public conferences. This does not involve instant messages which are not persistent, but only properties of the conference objects themselves, such as access rules.

Using SSL in Sun ONE Instant Messaging

Instant Messaging supports the Secure Sockets Layer (SSL) protocol, for encrypted communications and for certificate-based authentication of Instant Messaging servers. Instant Messaging server supports SSL version 3.0.

Sun ONE Instant Messaging multiplexor and Sun ONE Instant Messenger also support SSL for encrypted communication between the client and the multiplexor.

For detailed information on SSL, see Appendix B in *Sun ONE Console and Administration Server 5.0 Server Management Guide*.

Enabling SSL for Sun ONE Instant Messaging Server necessitates the following:

1. Obtaining and installing a certificate for your Instant Messaging server, and configuring the Instant Messaging server to trust the Certification Authority's certificate.
2. Ensuring that each Instant Messaging server that needs to communicate using SSL with your server, obtains and installs a certificate.
3. Turning on SSL in the server and the multiplexor by setting the appropriate parameters in the `iim.conf` file.

Enabling the client or server SSL requires the following:

1. Obtaining and installing a certificate for your Instant Messaging multiplexor host, and configuring the Instant Messaging server to trust the Certification Authority's certificate.
2. Turning on the SSL in the multiplexor by setting the appropriate parameters in the `iim.conf` file.
3. Making sure that the end users download and use the SSL version of the Instant Messenger, such as the `imssl.jnlp` file or the `imssl.html` file.

For steps on configuring SSL, see Configuring SSL.

Sun ONE Instant Messaging Access Control

Sun ONE Instant Messaging provides the user the ability to control access to the Instant Messaging features. Based on the access control mechanisms defined in the Instant Messaging server, the users are allowed or denied access to the information.

Sun ONE Instant Messaging provides the following access control mechanisms:

- Global Access Control
- Conference and News Channel Access Controls
- Presence Access Control
- Instant Messaging Administrator Access Control

Global Access Control

Global access control specifies end user access to specific functionality in Sun ONE Instant messaging. It specifies:

- Ability to access the presence status of other users
- Ability to send alerts to other users
- Ability to save properties on the server
- Ability to create and manage conference rooms
- Ability to create and manage news channels

By default, the end user is provided with the privileges to access the presence status of other users, send alerts to users, and save properties to the server. In most of the deployments, the default values are not changed. These default values need to be changed when Instant Messaging is used exclusively for the popup functionality.

When Instant Messaging is used exclusively for the popup functionality the user will not be provided with the access privileges to presence information, chat, and news features.

If the user has privileges to create new conference rooms and news channels, make sure to enforce a policy that will limit the number of conference rooms and news channels created. As Instant Messaging does not have a search interface for conference rooms and news channels, you need to limit the number of conference rooms and news channels created.

NOTE Although certain privileges can be set globally, the administrator can also define exceptions for these privileges. For example, the administrator can deny certain default privileges to select users or groups.

For more information on configuring global access control, see Managing User Privileges.

Conference Room and News Channel Access Controls

Users can have the following access privileges on Conference rooms and News channels:

- `MANAGE` - full access, which includes the ability to set the conference room or the news channel privilege for other users.
- `WRITE` - privilege to add contents to the conference room or the news channel.
- `READ` - privilege to read the conference room or the news channel contents.
- `NONE` - no access privileges.

Users with the MANAGE privilege can set the default privilege level for all the other users. These users can also define the exception rules to grant an access level that is different from the default access level permission given to specific users or groups.

NOTE Setting the WRITE privilege, grants the users the READ privilege.

Presence Access Control

The end users can specify if other users can see their presence or not. By default, all users can access the presence information of another user. The end user can also set exceptions for denying this access to certain users and groups.

If the end user has denied other users from accessing the user's presence status, then end users availability status appears as offline in others contact lists. No alerts or chat invitations can be sent to the end user, whose presence status is offline.

Presence access control can be configured using the User Settings window in the Instant Messenger. For more information on configuring presence access control, see *Sun ONE Instant Messenger Online Help*.

Instant Messaging Administrator Access Control

The Instant Messaging Administrator has access to all Instant Messaging features. The Administrator has MANAGE access to all conference rooms and news channels, see presence information of any user, and view and modify properties such as Contact Lists and Instant Messenger Settings of any user. The global access control settings has no impact on the Administrator's privileges.

Using the Command-Line with Sun ONE Instant Messaging

The Instant Messaging server provides a command-line utility to start, stop and refresh the server and multiplexor. For more information on the imadmin command, see Stopping and Starting the Server and Multiplexor (On Unix).

Chapter 2

Administering Instant Messaging Server and Multiplexor

This chapter explains how to administer Instant Messaging server and Multiplexor, and perform other administrative tasks, such as changing configuration parameters and managing user privileges. This chapter also lists the administrative tasks for Sun ONE Portal Server deployments.

This chapter contains the following sections:

- Administering Instant Messaging an Overview
- Stopping and Starting the Server and Multiplexor (On Unix)
- Changing Instant Messaging Server and Multiplexor Configuration Parameters
- Managing Logging
- Managing User Privileges
- Federating Deployment of Multiple Instant Messaging Servers
- Configuring SSL
- Managing Sun ONE Instant Messaging's LDAP Configuration
- Displaying Calendar Reminders and Notifications as Instant Messenger Popups
- Taking a Back Up of Instant Messaging Data

Administering Instant Messaging an Overview

The administrative tasks in Instant Messaging are:

- Starting, stopping, and refreshing the server and the multiplexor with the `imadmin` command
- Changing the configuration parameters, such as host names and LDAP search filters, by manually editing the `iim.conf` file
- Managing server and multiplexor log files
- Managing user privileges (Access Control files)
- Configuring communication between multiple Instant Messaging servers
- Setting up and using Secure Sockets Layer (SSL)
- Backing up and restoring Instant Messaging server files

User Administration

Instant Messaging does not provide user provisioning tools. You need to use a directory provisioning tool for provisioning Instant Messaging users. Instant Messaging does not provide specific commands to add, modify, or delete Instant Messaging users.

Likewise in a standalone deployment, you cannot prevent a user from using Sun ONE Instant Messenger. In a standalone deployment, the only way to prevent users from using Instant Messaging is to delete them from the directory. In a portal deployment using the policy attributes, you can prevent a user from accessing Instant Messenger.

The administrator can manage Instant Messaging users, using the Instant Messaging Administrator Access Control mechanism. For more information on Instant Messaging Administrator Access Control, see Sun ONE Instant Messaging Access Control. In a portal deployment, Sun ONE Identity Server is used for provisioning Instant Messaging users. For more information, see Sun ONE Identity Server.

CAUTION If you deny users the privilege to set up watches on other users by editing the `sysWatch.acl` file, the Sun ONE Instant Messenger's Main window is not displayed for these users. This effectively denies the user the ability to send instant messages. However, users would still be able to see alerts and news channels.

Stopping and Starting the Server and Multiplexor (On Unix)

The `imadmin` command enables you to:

- Start and stop the Instant Messaging server and multiplexor
- Start and stop only the multiplexor or only the server
- Refresh the Instant Messaging server and multiplexor configuration
- Refresh only the multiplexor or server configuration

The `imadmin` command-line utility can be executed only by the user who has administration rights to the system(s) on which the Sun ONE Instant Messaging server and multiplexor are running. This user is typically the identity that the server runs as and is designated during installation:

- On Solaris - `inetuser`
- On Windows - the user with full administration privileges, such as an `administrator`.
- In a portal deployment, if the Portal Server and the Instant Messaging server are installed on the same host, the user is the one who is running the Portal Server, as `root`.

The `imadmin` command-line utility is located in the following directory:

- On Solaris: *im_install_dir*`/SUNWiim/sbin`

Starting the Sun ONE Instant Messaging server enables Sun ONE Instant Messenger to connect to it. Stopping the Instant Messaging server closes all connections and disconnects all the Instant Messengers.

If required, you can start and stop the multiplexor instance separately. For example, if you have changed a configuration parameter which only affects the multiplexor, or if you only have the multiplexor installed on a different host, you can start and stop the multiplexor instance separately.

To Start the Instant Messaging Server and Multiplexor

For a given instance, the configuration specifies whether only the multiplexor or only the server or both these components are enabled.

Use the `imadmin` command to start the Sun ONE Instant Messaging Server and / or multiplexor, depending on which component is enabled:

```
imadmin start
```

If both server and multiplexor are enabled, this command first starts the Instant Messaging server, and then starts the multiplexor.

To Stop the Instant Messaging Server and Multiplexor

Use the `imadmin` command to stop the Sun ONE Instant Messaging server and / or multiplexor, depending on which component is enabled.

```
imadmin stop
```

This command stops the server and the multiplexor, terminates all user connections, and disconnects any inbound and outbound servers configured.

To Refresh the Configuration (Instant Messaging Server and Multiplexor)

Use the `imadmin` command with the `refresh` parameter to refresh the server and / or multiplexor configuration, as shown in the following example:

```
imadmin refresh
```

This command stops and restarts the enabled server and / or multiplexor components.

NOTE Whenever you change a configuration parameter in the `iim.conf` file, make sure to refresh the configuration.

If necessary, you can stop, start or refresh the multiplexor or server only, regardless of which components are enabled in the configuration. To do this, use the `multiplexor` or `server` argument with the `imadmin` command.

- To Start the Multiplexor only, type:

  ```
  imadmin start multiplexor
  ```

- To Stop the Server only, type:

  ```
  imadmin stop server
  ```

To Start and Stop the Instant Messaging Server and Multiplexor (Windows Only)

On Windows, open the Services dialog box from the Control Panel to start and stop the Instant Messaging server and the multiplexor. For more instructions on starting and stopping services, refer to the documentation provided with the Windows operating system.

Changing Instant Messaging Server and Multiplexor Configuration Parameters

Instant Messaging stores configuration parameters in the `iim.conf` file. For a complete list of configuration parameters, see Instant Messaging Configuration Parameters.

To change configuration parameters, manually edit the configuration parameters and values in the `iim.conf` file, then refresh the Sun ONE Instant Messaging server configuration. If you change a multiplexor parameter, you only need to refresh the multiplexor using the following `imadmin` command:

```
imadmin refresh multiplexor
```

To Change Configuration Parameters

For a complete list of parameters and their values, see Instant Messaging Configuration Parameters.

To Change Configuration Parameters:

1. Change to the `config` directory. For example, on Solaris type:

   ```
   cd /etc/opt/SUNWiim/default/config
   ```

2. Edit the `iim.conf` file. For example:

   ```
   vi iim.conf
   ```

3. Save your changes.
4. Refresh the configuration.

CAUTION If you change the multiplexor listen port (`iim_mux.listenport`) or the multiplexor host, update the `im.html` or the `im.jnlp` files accordingly. Failure to do so disables the Sun ONE Instant Messenger from connecting to the server. For more information, see the section on Managing Instant Messenger.

Managing Logging

Instant Messaging creates log files that record events, related status of various software components, system errors, and other aspects of the server and multiplexor. By examining the log files, you can monitor many aspects of the server's operation.

You can configure the level of logging for both the Sun ONE Instant Messaging server and the multiplexor by specifying the parameters in the `iim.conf` file. For information on configuring the level of logging in the `iim.conf` file, see the section on To Change Configuration Parameters.

The location of the log files are specified during Instant Messaging installation.

- On Solaris, the default directory is:

  `/var/opt/SUNWiim/default/log`

- On Linux, the default directory is:

  `/var/opt/soim/default/log`

- On Windows, the default directory is:

  `c:\Program Files\Sun\InstantMessaging\log`

As part of the regular Instant Messaging server system maintenance, you need to periodically review and trim the log files from occupying more disk space. The server does not perform this action.

Logging Levels

The level or priority of maintaining the error log defines how detailed, or verbose, the log should be. A higher priority level implies less details as only events of high priority (high severity) are recorded in the log file. In contrast a lower priority level implies greater details as more events are recorded in the log file.

You can set the logging level separately for the Instant Messaging server and the multiplexor.

Table 2-1 contains the logging levels for the Instant Messaging server and their description. These logging levels are a subset of the levels defined by the Unix `syslog` facility.

Table 2-1 Logging Levels for the Instant Messaging Server and the Multiplexor

Level	Description
FATAL	This priority level records minimum logging details in the log file. A log record is added to the log file whenever a severe problem or critical condition occurs. If a FATAL problem occurs, the application might stop functioning.
ERROR	A log record is added to the log file whenever a recoverable software error condition occurs or a network failure is detected. For example, when the server fails to connect to a client or to another server.
WARNING	A log record is added to the log file whenever a user error is detected. For example, when the server cannot understand the communication sent by the client.
NOTICE	A periodic event is written to the log file to report the status of the server. This includes state (running), the number of clients connected, and number of inbound and outbound servers connected.
INFO	A log record is added to the log file whenever a significant action takes place. For example, when a user successfully logs in or logs out.
DEBUG	The tasks are recorded in the log file. This information is useful for debugging purposes only. Each event with individual steps within each process or task are written to the log file, to help the user identify the problems while debugging the application.

When you select a particular logging level, events corresponding to that level and to all higher and less verbose levels are logged.

`NOTICE` is the default level for both the server and the multiplexor log files.

NOTE If you specify `DEBUG` as the logging level, your log files will occupy more disk space. Monitor and trim your log files to prevent them from occupying more disk space.

To Set Log File Levels

The log file levels are set within the `iim.conf` file. The following are the two log file logging level options:

- The parameter for the logging to the server is: `iim.log.iim_server.severity`.
- The parameter for the logging to the multiplexor is: `iim.log.iim_mux.severity`.

For more information on configuring Instant Messaging, see To Change Configuration Parameters.

Managing User Privileges

The Administrator can control user access to Instant Messaging information by restricting privileges to the user. These privileges determine if the user can add and delete news channels, send alerts, and setup watches on other users. These features provide the users the access to the required features and views in the Instant Messaging. All the Instant Messaging features are controlled by the privilege system that determines what a user can view or perform on Instant Messaging.

Sun ONE Instant Messaging provides the following access control mechanisms:

- Global Access Control
- Conference and News Channel Access Controls
- Presence Access Control
- Instant Messaging Administrator Access Control

Global Access Control

Global access control provides all the users the privilege to access the following Instant Messaging features:

- Access the presence status of the other users
- Send alerts to other users
- Save properties on the server
- Create new conference rooms

- Create new news channels

By default, the end user is provided the privileges to access the presence status of other users, send alerts to users, and save properties to the server. In most of the deployments, the default values need not be changed. These default values need to be changed when Instant Messaging is used exclusively for the popup functionality.

NOTE Although certain privileges can be set globally, the administrator can also define exceptions for these privileges. For example, the administrator can deny certain default privileges to select users or groups.

The location of the global access control ACL files are:

- On Solaris:

  `/etc/opt/SUNWiim/default/config/acls`

- On Linux:

  `/etc/opt/soim/default/config/acls`

- On Windows, the default directory is:

  im_install_dir`\config\acls`

Table 2-2 contains the global access control ACL files for Sun ONE Instant Messaging and the privileges these ACL files provide the users.

Table 2-2 Global Access Control ACL Files

ACL File	Privileges
`sysSaveUserSettings.acl`	The User Settings privilege enables the users to change their preferences in the User Settings dialog.
`sysTopicsAdd.acl`	The News Channel Creation privilege enables the users to create News channels.
`sysRoomsAdd.acl`	The Conference Room Creation privilege enables the users to create Conference rooms.
`sysSendAlerts.acl`	The Send Alert privilege enables the users to send alerts.

Table 2-2 Global Access Control ACL Files

ACL File	Privileges
`sysWatch.acl`	The Watch privilege enables the users to watch changes on other users. The Sun ONE Messenger window is not displayed for users who do not have this privilege.

Conference Room and News Channel Access Controls

For each Conference room and News channel, you can define the default access the users can have. The access privileges the users can have on Conference rooms and News channels are:

- `MANAGE`
- `WRITE`
- `READ`
- `NONE`

Users with the `MANAGE` privilege can set the default privilege level for all the other users. They can also define the exception rules to grant an access level that is different from the default access level to specific users or groups.

NOTE Setting the `WRITE` privilege, grants the users the `READ` privilege.

The conference room and news channel privileges are set through Sun ONE Instant Messenger. These files are updated automatically when you use Sun ONE Instant Messenger to manage conference rooms and news channels.

Table 2-3 lists the Conference room and News channel access control ACL files and the privileges that these files provide the users. These ACL files are located in the `db/acls` directory.

Table 2-3 Conference Room and News Channel Access Control ACL Files

ACL File	Privileges
roomname`.acl`	This file sets access privileges that the users can have on conference rooms.

Table 2-3 Conference Room and News Channel Access Control ACL Files

ACL File	Privileges
news channelname.acl	This file sets access privileges that the users can have on news channels.

Presence Access Control

You can specify if other users can see your presence or not. By default, all users can see your presence status. You can also set exceptions for denying this access to certain users and groups.

If you have denied other users from accessing your presence status, then these users will see your availability status as offline in their contact lists. These users will not be able to send alerts or chat invitations to you, as your presence status is offline.

Presence access control can be configured using the User Settings window in the Instant Messenger. For more information on configuring presence access control, see Sun ONE Instant Messaging Online Help.

Instant Messaging Administrator Access Control

The Instant Messaging Administrator has access to all Instant Messaging features. The Administrator has MANAGE access to all conference rooms and news channels. They can also see the presence information of any user, view and modify properties such as Contact Lists and Instant Messenger Users Settings of any user. The global access control settings have no impact on the Administrator.

Table 2-4 lists the Administrator access control ACL file and the privileges this ACL file provides the Administrator. This ACL file is located in the config/acls directory:

Table 2-4 Administrator Access Control ACL File

ACL File	Privilege
sysAdmin.acl	This file sets administrative privileges to all Sun ONE Instant Messaging features for the users. This privilege overrides all the other privileges, and hence is reserved for the administrators.

Access Control File Format

The access control file contain a series of entries that define the privileges. Each entry starts with a tag as follows:

- `d:` - default
- `u:` - user
- `g:` - group

NOTE The `d:` tag must be the last entry in an access control file. The server ignores all entries after a `d:` tag. If the `d:` tag is `true`, then all other lines are ignored. You cannot set the `d:tag` as `true` in a ACL file and selectively disallow users that privilege.

The tag is followed by a colon (`:`). In case of the default tag it is followed by `true` or `false`.

The user and group tags are followed by the user or group name.

Multiple users and groups are specified by having multiple users (u)and groups (g) in lines.

If default is set to `true`, all other entries in the file are redundant. If default is set to `false`, only the users and groups specified in the file will have that particular privilege.

The following are the default `d:` tag entries in the ACL files for a new installation:

- `sysAdmin.acl` - Contains `d:false`
- `sysTopicsAdd.acl` - Contains `d:false`
- `sysRoomsAdd.acl` - Contains `d:false`
- `sysSaveUserSettings.acl` - Contains `d:true`
- `sysSendAlerts.acl` - Contains `d:true`
- `sysWatch.acl` - Contains `d:true`

NOTE The format and also the existence of all the ACL files might undergo changes in the future release of the product.

Access Control File Examples

This section consist of sample ACL files that show privileges set at the system level, (`sysTopicsAdd.acl`) and at the conference room or news channel level (*newschannel*`.acl`).

sysTopicsAdd.acl File

In the following example, the default `d:` tag entry for `sysTopicsAdd.acl` file is `false`. So the Add and the Delete news channels privileges are available to the users and groups that appear before the default, namely `user1`, `user2`, and the `sales` group.

```
# Example sysTopicsAdd.acl file
u:user1
u:user2
g:sales
d:False
```

Room and News Channels ACL Files

The format of the *roomname*`.acl` and *news channelname*`.acl` files is slightly different from the system level ACL files. The *roomname*`.acl` and *news channelname*`.acl` files contain an additional number entry after the user or group entry that defines the access level. The access levels are:

- 1 - None
- 2- Read
- 6 - Write
- 14 - Manage

In the following news channel ACL example, the default access is Read, with Manage access given to `user1`, Write access given to `user2`, and an access of None to `user3`.

```
# Example newschannel.acl file
v:3.0.1
u:user1:14
u:user2:6
u:user3:1
d:2
```

NOTE The line `v:3.0.1` in the `newschannel.acl` file, tells the server how to interpret the values. If this line is not included, the server will not be able to associate the value of 2 with Read access, and the value 6 with Write access.

NOTE Do not edit the *roomname*`.acl` and *news channelname*`.acl` files manually. These files are updated automatically as you use Sun ONE Instant Messenger to manage conference rooms and news channels. Because Sun ONE Instant Messaging server reads and writes these files when the users change access using Sun ONE Instant Messenger, the users can lose their changes if the files are edited manually while the server is running.

Changing User Privileges

To change user privileges:

1. Change to the `config/acls` directory. For example, on Solaris:

   ```
   cd /etc/opt/SUNWiim/default/config/acls
   ```

2. Edit the appropriate ACL file. For example:

   ```
   vi sysTopicsAdd.acl
   ```

3. Save the changes.

4. Users need to refresh the Sun ONE Instant Messenger window to see the changes.

Federating Deployment of Multiple Instant Messaging Servers

When you federate multiple Sun ONE Instant Messaging deployments, you form a larger instant messaging community with multiple domains. Users in different domains can communicate with each other, use conference rooms on other domains, and subscribe to news channels on other domains based on the access privileges.

For enabling communication between multiple Sun ONE Instant Messaging servers in your network, you need to configure your server to identify itself to the other Instant Messaging servers in the network. The Instant Messaging server identifies itself with its domain name, host and port number, serverID, and password.

Within the server configuration, you can assign each Instant Messaging server a symbolic name, consisting of letters and digits, for example, `IMserver1`.

CAUTION It is recommended that the server-to-server communication is secured using TLS (SSL). This is required to prevent third party infringement of security when data is exchanged between two servers. This precaution is extremely desirable in the case where the link between the two servers uses the public internet. Follow the instructions outlined below to configure SSL between Instant Messaging servers.

To Configure Communication Between Instant Messaging Servers

This procedure describes how to enable communication between two Instant Messaging servers, `iim.company22.com` and `iim.i-zed.com`.

1. Gather the following information listed in Table 2-5.

   Table 2-5 lists the parameters in the `iim.conf` file for server-to-server communication and the values for these parameters in the Instant Messaging servers, `iim.company22.com` and `iim.i-zed.com`.

Table 2-5 Configuration Information for Server-to-Server Communication

Parameter in iim.conf File	Value for Server iim.company22.com	Value for Server iim.i-zed.com
`iim_server.serverid`	`Iamcompany22`	`Iami-zed`
`iim_server.password`	`secretforcompany22`	`secret4i-zed`
`iim_server.coservers`	`coserver1`	`coserver1`
`iim_server.coserver1.host`	`iim.i-zed.com:9919`	`iim.company22.com:9919`
`iim_server.coserver1.serverid`	`Iami-zed`	`Iamcompany22`
`iim_server.coserver1.password`	`secret4i-zed`	`secretforcompany22`

   For more information on the configuration parameters, see Instant Messaging Configuration Parameters.

NOTE You can configure your server to communicate with other Instant Messaging servers. Each Instant Messaging server is identified by its symbolic name. The symbolic name of the server is added in the `iim_server.coservers` parameter in the `iim.conf` file. This parameter has multiple values and each value is separated by a comma.

2. Change to the `config` directory on the server `iim.company22.com`. For example, on Solaris:

   ```
   cd /etc/opt/SUNWiim/default/config
   ```

3. Edit the `iim.conf` file, for example:

   ```
   vi iim.conf
   ```

NOTE The `iim.conf` file should be owned by the Instant Messaging server account you created during installation. If the `iim.conf` file cannot be read by the Instant Messaging server account, Instant Messaging server and multiplexor would be unable to read the configuration. Additionally, you might lose the ability to edit the `iim.conf` file.

The following example shows the section of the `iim.conf` file on `iim.company22.com` corresponding to the server-to-server communications that you can modify:

```
iim_server.serverid=Iamcompany22
iim_server.password=secretforcompany22
iim_server.coservers=coserver1
iim_server.coserver1.host=iim.i-zed.com:9919
iim_server.coserver1.serverid=Iami-zed
iim_server.coserver1.password=secret4i-zed
```

4. Follow Step 2 through Step 3 for the `iim.conf` file on server `iim.i-zed.com`.

   The following example shows the section of the `iim.conf` file on `iim.i-zed.com` corresponding to the server-to-server communications that you can modify:

```
iim_server.serverid=Iami-zed
iim_server.password=secret4i-zed
iim_server.coservers=coserver1
iim_server.coserver1.host=iim.company22.com:9919
iim_server.coserver1.serverid=Iamcompany22
iim_server.coserver1.password=secretforcompany22
```

5. Save the changes and refresh the configurations on both servers.

Configuring SSL

This section describes how to set up the Secure Sockets Layer (SSL) security protocol for use between Sun ONE Instant Messaging servers. Before following the procedures in this section, become familiar with public-key cryptography concepts. See Appendix B in the *Sun ONE Console and Administration Server 5.0 Management Guide* for public-key cryptography concepts.

The high-level steps to configure SSL for Instant Messaging servers are:

1. Generating a self-signed certificate.
2. Generating a Certificate Signing Request.
3. Sending a Certificate Signing Request to a Certificate Authority (CA) and getting back a signed certificate.
4. Installing the Certificate on the Sun ONE Instant Messaging server, and the CA's certificate on other servers. This means that you also have to install the other server's CA certificate on your system.
5. Activating SSL.

To configure SSL, Instant Messaging server requires a key file that contains the public and private key installed in a directory that you specify using the `iim.conf` parameter, `iim_server.sslkeystore`. In most of the installations these files are stored in the `config` directory.

To verify certificates Sun ONE Instant Messenger server uses the keys stored in the `nlcacerts` file in its Java install directory on Solaris, *Javahome*`/lib/security/` and on Windows, *Javahome*`\lib\security\`.

You need to create the `nlcacerts` file by including the CA's certificate into the file, if the CA is not in the `nlcacerts` file. The `nlcacerts` file can be located:

On Unix - *Javahome*`/lib/security/cacerts`

On Windows - *Javahome*`\lib\security\cacerts`.

The `cacerts` file is created by the Java install.

When enabling SSL for use with the Instant Messaging server, choose one of the following methods:

- **Using a self-signed certificate** - Place your self-signed certificate in the `iimkeys` file:

  On Solaris - `/etc/opt/SUNWiim/default/config/iimkeys`

  On Linux - `/etc/opt/soim/default/config/iimkeys`

On Windows - *im_install_dir*\config\iimkeys

Also export the self-signed certificates to the other Instant Messaging servers so that they can be placed it in the servers' nlcacerts file.

- **Using a certificate signed by a CA that is not already in cacerts** - Place your certificate and your signing CA's certificate in the iimkeys file:

  On Solaris - /etc/opt/SUNWiim/default/config/iimkeys

  On Linux - /etc/opt/soim/default/config/iimkeys

  On Windows - *im_install_dir*\config\iimkeys.

  Also, export your certificate signed by CA to the other servers so that they can place the certificate in their nlcacerts file.

- **Using a certificate signed by a CA already in cacerts** - Place your certificate in the iimkeys file:

  On Solaris - /etc/opt/SUNWiim/default/config/iimkeys

  On Linux - /etc/opt/soim/default/config/iimkeys

  On Windows - *im_install_dir*\config\iimkeys)

  The other servers will already have your certificate signed by a CA in their nlcacerts file.

NOTE You can run the following command to list all the CAs in your nlcacerts file:

Javahome/keytool -list -keystore cacerts

Run this command from the directory that contains the cacerts file. Press Return when prompted for the password.

When enabling SSL for use with the Instant Messaging server remember that your Instant Messaging server is the "client" of the other servers, and therefore you might have to import the CA's certificate for that server.

To Generate a Self-Signed Certificate

1. Run the following command on the Sun ONE Instant Messaging server from its `config` directory. For example, on Solaris:

   Javahome`/bin/keytool -genkey -alias iim -keypass` *`value`* `-keystore iimkeys -storepass` *`value`* `-validity 365`

   Substitute your own values for *value*.

2. When prompted, enter the information to create a distinguished name.

   This creates a `keystore` entry containing a private key and a self-signed certificate for the public key. The certificate is signed using the corresponding private key.

NOTE If you are using the self-signed certificate for your Sun ONE Instant Messaging server, then this certificate is put in the `nlcacerts` file:

On Unix - *Javahome*`/lib/security/nlcacerts`

On Windows - *Javahome*`\lib\security\nlcacerts`).

If you are using self-signed certificate, skip the section To Get Your Self-Signed Certificate Signed by a Certificate Authority and Install it.

To Get Your Self-Signed Certificate Signed by a Certificate Authority and Install it

Getting the Instant Messaging certificate signed by a Certificate Authority enables the other Sun ONE Instant Messaging servers to verify the Instant Messaging certificate. This occurs because the servers' CA keystore (on Solaris, *Javahome*`/lib/security/cacerts`; on Windows, *Javahome*`\lib\security\cacerts`) already has the certificates from Verisign and Thawte Certificate Authorities.

Follow these steps to send the certificate information to the CA and install it:

1. Generate a Certificate Signing Request (CSR) to send to a Certificate Authority (CA). For example, on Solaris:

   Javahome`/bin/keytool -certreq -keystore iimkeys -alias iim -file iim.csr`

2. Submit the `iim.csr` file to your chosen CA.

   You can email your `iim.csr` file, or cut and paste it in a browser window, if the CA, such as Thawte, allows it.

NOTE Once you have emailed your request, you must wait for the CA to assign you with your certificate. The response time for your certificate request varies. For example, if your CA is internal to your company, it might only take a day or two to respond to your request. If your selected CA is external to your company, it could take several weeks to respond to your request.

3. When the CA sends a response, be sure to save the information in a text file. For example, a file named `careplyfile`. You will need the data when you install the certificate.

   You should also back up the certificate data in a safe location. If your system ever loses the certificate data, you can reinstall the certificate using your backup file.

   Once you receive your certificate, you are ready to install it in your Sun ONE Instant Messaging server's certificate database.

4. Import the Certificate Reply from the CA.

   This might require that you import the certificate of the CA that signed your certificate first as a "trusted certificate."

CAUTION Ensure that the CA's certificate is valid prior to importing it as a "trusted" certificate. View it first (by using the `keytool -printcert` command or the `keytool -import` command without the `-noprompt` option), and make sure that the displayed certificate fingerprints match the expected ones. You can contact the person who sent the certificate and compare the fingerprints that you see with the ones that the certificate show or that a secure public key repository shows. Only if the fingerprints match is it guaranteed that the certificate has not been replaced in transit with someone else's (for example, an attacker's) certificate.

If you trust that the certificate is valid, you can add it to your keystore.

Assuming the CA's certificate is in a file called `myfavca.cert`, run the following command on the Sun ONE Instant Messaging server in the `config` directory.

```
keytool -import -alias myfavca -file myfavca.cert -keystore
iimkeys
```

Then import your new certificate on the Sun ONE Instant Messaging server to replace your self-signed certificate. Run this command on the Sun ONE Instant Messaging server in the `config` directory:

```
keytool -import -trustcacerts -keystore iimkeys -alias iim -file
careplyfile
```

5. Your server is now ready for SSL activation. See To Activate SSL For Server To Server Communication.

To Export a Public Key Certificate and Import it on other Instant Messaging Servers

The `cacerts` file is used by Java to verify code signing but can also be used for SSL. The following is the location of the `cacerts` file:

On Unix- *Javahome*`/lib/security/cacerts`

On Windows - *Javahome*`\lib\security\cacerts`

If the `nlcacerts` file exists then that will be used by the Sun ONE Instant Messaging server. The following is the location of the `nlcacerts` file:

On Unix - *Javahome*`/lib/security/nlcacerts`

On Windows - *Javahome*\lib\security\nlcacerts

Use this procedure if you are using a self-signed certificate or a certificate that is signed by a CA that is not in the default trusted CA keystore:

1. Export the Public Key Certificate.

   If you are self-signed, export your self-signed certificate. For example, run the following command on the Sun ONE Instant Messaging server, in the config directory:

   ```
   keytool -export -keystore iimkeys -alias iim -file export.cert
   ```

   If your certificate is signed by a CA not in the existing cacerts file on the other server, then export your signing CA's certificate to the other server. For example, run the following command on the Instant Messaging server, in the config directory:

   ```
   keytool -export -keystore iimkeys -alias myfavca -file
   export.cert
   ```

   Then copy the export.cert file to the other server, in the *Javahome*/lib/security directory.

2. Import the Public Key Certificate as trusted on every other Sun ONE Instant Messaging server that will talk SSL to this server. To import Public Key Certificate create the *Javahome*/lib/security/nlcacerts file with the command:

   ```
   keytool -import -keystore nlcacerts -alias iimca -file
   export.cert
   ```

   Enter a password when prompted.

   For more information see the complete documentation for keytool at:

   ```
   http://java.sun.com/j2se/1.3/docs/tooldocs/solaris/keytool.html
   ```

   ```
   http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html
   ```

3. Your server is now ready for SSL activation. See To Activate SSL For Server To Server Communication.

To Activate SSL For Server To Server Communication

Before you can activate SSL, you must create a certificate database, obtain and install a server certificate, and trust the CA's certificate as described earlier.

1. Set these `iim.conf` parameters:
   - `iim_server.usesslport=true`
   - `iim_server.sslport=9910`

   These parameters should already be in the `iim.conf` file.
2. Set the server-to-server configurations as described in Federating Deployment of Multiple Instant Messaging Servers, and add the following:
   - `iim_server.coserver1.usessl=true`

   Change the port number of the following:
   - `iim_server.coserver1.host=hostname:9910`

   The port number should be the SSL port of the other server.

   Following is a section of `iim.conf` file with the required SSL configuration:

```
! Server to server communication port.
iim_server.port = "49919”
! Should the server listen on the server to server communication port
iim_server.useport = "True”
! Should this server listen for server-to-server communication using ssl port
iim_server.usesslport = "True”
iim_server.sslport=49910
iim_server.sslkeystore=/opt/SUNWiim/config/iimkeys
iim_server.sslkeystorepassphrase=somepassword
iim_server.coservers=coserver1
iim_server.coserver1.serverid=Iamcompany22
iim_server.coserver1.password=secretforcompany22
iim_server.coserver1.usessl=true
iim_server.coserver1.host=iim.i-zed.com:49910
iim_server.serverid=Iami-zed
iim_server.password=secret4i-zed
```

Configuring SSL between Sun ONE Instant Messenger and Multiplexor

Instant Messenger communicates with the multiplexor component of the server. For a secured connection between the Instant Messaging server and the Instant Messenger, you need to configure SSL between Sun ONE Instant Messenger and the multiplexor.

NOTE The Instant Messaging server implements SSL differently when compared to the multiplexor. For this reason the steps to configure SSL in multiplexor is enumerated in this section separately.

The following are the steps necessary to configure SSL between Sun ONE Instant Messenger and Multiplexor:

1. Requesting a Certificate From The Certificate Authority
2. Installing the Certificate
3. Configuring the Instant Messaging server to enable SSL between the Multiplexor and the Instant Messenger
4. Invoking The Secure Version Of Instant Messenger

Requesting a Certificate From The Certificate Authority

To enable SSL between Instant Messenger and multiplexor, you need to install the certificate and create databases for secure communication. You can request and install the certificate using Sun ONE Web Server Enterprise Edition.

To request and install a certificate using Sun ONE Web Server Enterprise Edition:

1. Type the following URL for starting the administration server in your browser:

   `http://hostname.domain-name:administration_port`

   Sun ONE Web Server then displays a window prompting you for a user name and password.

2. Type the administration user name and password you specified during the Web Server installation.

   Sun ONE Web Server displays the Administration Server page.

3. Create a separate Web Server instance. For more information on installing multiple instances of the server, see Installing Multiple Instances of the Server in *Sun ONE Web Server, Enterprise Edition Administrator's Guide* at:

   `http://docs.sun.com/source/816-5682-10/esgstart.htm#1003083`

4. Create a trust database to store the public and private keys, referred as the key-pair file. The key-pair file is used for SSL encryption.

   For information on creating a trust database, see Creating a Trust Database in *Sun ONE Web Server, Enterprise Edition Administrator's Guide* at:

   `http://docs.sun.com/source/816-5682-10/esecurty.htm#1004127`

5. Request a certificate from the Certificate Authority.

   For more information on requesting a certificate, see Requesting and Installing Other Server Certificates in *Sun ONE Web Server, Enterprise Edition Administrator's Guide* at:

   `http://docs.sun.com/source/816-5682-10/esecurty.htm#1004981`

Installing the Certificate

When you receive the server certificate from your Certificate Authority, you need to install the certificate.

To install the certificate:

1. Type the following URL for starting the administration server in your browser:

   `http://hostname.domain-name:administration_port`

   Sun ONE Web Server then displays a window prompting you for a user name and password.

2. Type the administration user name and password you specified during the Web Server installation.

   Sun ONE Web Server displays the Administration Server page.

3. Install the server certificate.

   For more information on installing the certificate, see Requesting and Installing Other Server Certificates in *Sun ONE Web Server, Enterprise Edition Administrator's Guide* at:

   `http://docs.sun.com/source/816-5682-10/esecurty.htm#1004981`

4. Change to your Web Server `alias` directory.

5. Copy the database files from your Web Server `alias` directory to the Instant Messenger `config` directory.

   To copy the database files from Web Server `alias` directory to the Instant Messenger `config` directory, type the following:

   ```
   cp https-serverid-hostname-cert7.db
   /etc/opt/SUNWiim/default/config/cert7.db
   ```

   ```
   cp https-serverid-hostname-key3.db
   /etc/opt/SUNWiim/default/config/key3.db
   ```

   ```
   cp secmod.db /etc/opt/SUNWiim/default/config/secmod.db
   ```

NOTE The user on which the Instant Messaging server runs should have Read permission on `cert7.db`, `key3.db`, and `secmod.db` files.

6. Change to your Instant Messaging `config` directory.

   ```
   cd /etc/opt/SUNWiim/default/config
   ```

7. Create the `sslpassword.conf` file. Type:

   ```
   vi sslpassword.conf
   ```

8. Enter the following line to the `sslpassword.conf` file.

   ```
   Internal (software) Token:password
   ```

   Password: The password specified during the creation of the trust database.

9. Save the file.

NOTE All Instant Messenger users should have Ownership and Read permission on the `sslpassword.conf` file.

10. After verifying the functioning of SSL, log in to Sun ONE Web Server as an administrator and remove the web server instance that you have created while requesting the certificate.

Configuring the Instant Messaging server to enable SSL between the Multiplexor and the Instant Messenger

Table 2-6 lists the parameters in the `iim.conf` file for enabling SSL between Sun ONE Instant Messenger and multiplexor. It also contains the description and the default value of these parameters:

Table 2-6 Configuration Information for enabling SSL between Sun ONE Instant Messenger and Multiplexor

Parameter	Default Value	Description
iim_mux.usessl	off	If the value is set to "on", the multiplexor requires an SSL handshake for each connection it accepts, before exchanging any application data.
iim_mux.secconfigdir	/etc/opt/SUNWiim/default/config	This directory contains the key and certificate databases. It usually contains the security module database.
iim_mux.keydbprefix	None	This value should contain the key database filename prefix. The key database file name must always end with key3.db. If the Key database contains a prefix, for example This-Database-key3.db, then value of this parameter is This-Database.
iim_mux.certdbprefix	None	This value should contain the certificate database filename prefix. The certificate database file name must always end with cert7.db. If the certificate database contains a prefix, for example Secret-stuff-cert7.db, then value of this parameter is Secret-stuff.
iim_mux.secmodfile	secmod.db	This value should contain the name of the security module file.

Table 2-6 Configuration Information for enabling SSL between Sun ONE Instant Messenger and Multiplexor

Parameter	Default Value	Description
`iim_mux.certnickname`	`Server-Cert`	This value should contain the name of the certificate you entered while installing the certificate. The certificate name is case-sensitive.
`iim_mux.keystorepasswordfile`	`sslpassword.conf`	This value should contain the relative path and the name of the file containing the password for the key database. This file should contain the following line: `Internal (software) Token:`*password* Where *password* is the password protecting the key database.

To enable SSL between Sun ONE Instant Messenger and Multiplexor:

1. Change to the `config` directory. For example, on Solaris:

   ```
   cd /etc/opt/SUNWiim/default/config
   ```

2. Edit the `iim.conf` file, for example:

   ```
   vi iim.conf
   ```

3. Add the values mentioned in the Table 2-6 to the Multiplexor configuration parameters.

   The following is an example of the `iim.conf` file with the Multiplexor configuration parameters:

```
! IIM multiplexor configuration
! ==============================
!
! Multiplexor specific options

! IP address and listening port for the multiplexor.
! WARNING: If this value is changed, the port value of '-server'
argument
! in the client's im.html and im.jnlp files should also be changed
to match th
is.
iim_mux.listenport = "siroe.com:49909"

! The IM server and port the multiplexor talks to.
iim_mux.serverport = "siroe.com:49999"

! Number of instances of the multiplexor.
iim_mux.numinstances = "1"

! Maximum number of threads per instance
iim_mux.maxthreads = "10"

! Maximum number of concurrent connections per multiplexor
process
iim_mux.maxsessions = "1000"

iim_mux.usessl = "on"
iim_mux.secconfigdir = "/etc/opt/SUNWiim/default/config"
iim_mux.keydbprefix = "This-Database"
iim_mux.certdbprefix = "Secret-stuff"
iim_mux.secmodfile = "secmod.db"
iim_mux.certnickname = "Server_Cert"
iim_mux.keystorepasswordfile = "sslpassword.conf"
```

Invoking The Secure Version Of Instant Messenger

The secure version of Instant Messenger can be invoked by accessing the `imssl.html` file or `imssl.jnlp` file from your browser. These files are located under *imdocroot*, the base directory under which all the messenger resources are stored.

The links to these applet descriptor files can also be added to `index.html` file.

Managing Sun ONE Instant Messaging's LDAP Configuration

A standalone deployment of Instant Messaging requires a directory server. In a standalone deployment, the Instant Messaging server uses the directory server to perform user authentication and to search for users.

In a portal deployment, Instant Messaging Server uses the directory used by Sun ONE Portal Server. When installed in a portal deployment environment, Instant Messaging server uses that directory used by the Portal Server to search for users, and not for user authentication. In a portal deployment, Sun ONE Portal Server performs the authentication.

If you use an LDAP directory to maintain your user namespace, the default configuration makes the following assumptions regarding the schema used by this directory:

- User entries are identified by the `inetOrgPerson` object class.
- Group entries are identified by the `groupOfUniqueNames` object class.
- The Sun ONE Instant Messenger user ID attribute of a user is provided by the `uid` attribute (from `inetOrgPerson` objectclass).
- The email address of a user is provided by the `mail` attribute.
- The display name of a user or group is provided by the `cn` attribute.
- The list of members of a group is provided by the `uniqueMember` attribute (`groupOfUniqueNames` object class).

You can change these default settings by editing the `iim.conf` file.

Searching the Directory as Anonymous Users

Instant Messaging needs to be able to search the directory to function correctly. If your directory is configured to be searchable by anonymous users, Instant Messaging has the capability to search the directory. If the directory is not readable by anonymous users, you must take additional steps to configure the `iim.conf` file with the credentials of a user ID that has at least read access to the directory.

These credentials consist of:

- A distinguished name (`dn`)
- The password of the above `dn`

To Enable Instant Messaging server to Conduct Directory Searches as a Specific User (Not Anonymous)

1. Identify values for the following parameters in the `iim.conf` file:
   - `iim_ldap.usergroupbinddn` - Specifies the distinguished name (`dn`) to use to bind to the directory for searches.
   - `iim_ldap.usergroupbindcred` - Specifies the password to use with the distinguished name (`dn`)

   For example:

   ```
   iim_ldap.usergroupbinddn="cn=iim server,o=i-zed.com"
   iim_ldap.usergroupbindcred=secret
   ```

2. In a portal deployment, the directory is in general not accessible by anonymous users. In a portal deployment, the following values are added to the `iim.conf` file:
   - `iim_ldap.usergroupbinddn=` *"bind dn"*
   - `iim_ldap.usergroupbindcred=` *ldap credentials/password*

   where:

bind dn	Specifies the value of the distinguished name (dn) the super user has to use bind to the directory from the *AMSERVER_BASEDIR*/SUNWam/lib/AMConfig.properties file.
ldap credentials/password	The password will be taken from the package parameters.

NOTE You do not have to use administrator-level credentials with write level access, as all that is necessary is read access to the domain tree. Thus, if there is an LDAP user with read level access, use its credentials instead. This is a safer alternative as it does not force you to disseminate the administrator-level credentials.

3. Edit the `iim.conf` file.

   For instructions on editing the `iim.conf` file, see To Change Configuration Parameters.

   If the `iim_ldap.usergroupbinddn` and `iim_ldap.usergroupbindcred` parameters do not appear in the `iim.conf` file, you can add them anywhere in the file.

Configuring Dynamic LDAP Server Group

In the LDAP Server, the dynamic groups filter users based on their DN and include them in a single group. The dynamic groups are defined in Sun ONE Directory Server as the `memberOfUrls` objectclass.

To enable the users to view the dynamic groups in search results and add them to their contact list, you need to include the `memberOfUrls` objects to the group search filters in the contact list.

The following modifications need to be made to the server configuration file `iim.conf`:

1. Change to the `config` directory. For example, on Solaris:

   ```
   cd /etc/opt/SUNWiim/default/config
   ```

2. Edit the `iim.conf` file. For example:

   ```
   vi iim.conf
   ```

3. Add the following information to the `iim.conf` file:

```
iim_ldap.usergroupbynamesearchfilter=(|(&(|(objectclass=groupofu
niquenames)(objectclass=groupofurls)))(cn={0}))(&(objectclass=in
etorgperson)(cn={0})))

iim_ldap.groupbrowsefilter=(|(objectclass=groupofuniquenames)(ob
jectclass=groupofurls))

iim_ldap.groupclass=groupOfUniqueNames,groupOfURLs
```

The attribute and objectclass names are configurable. The membership attribute of a dynamic group which contains the LDAP filter or LDAP URL, can be specified using the `iim_ldap.groupmemberurlattr` option.

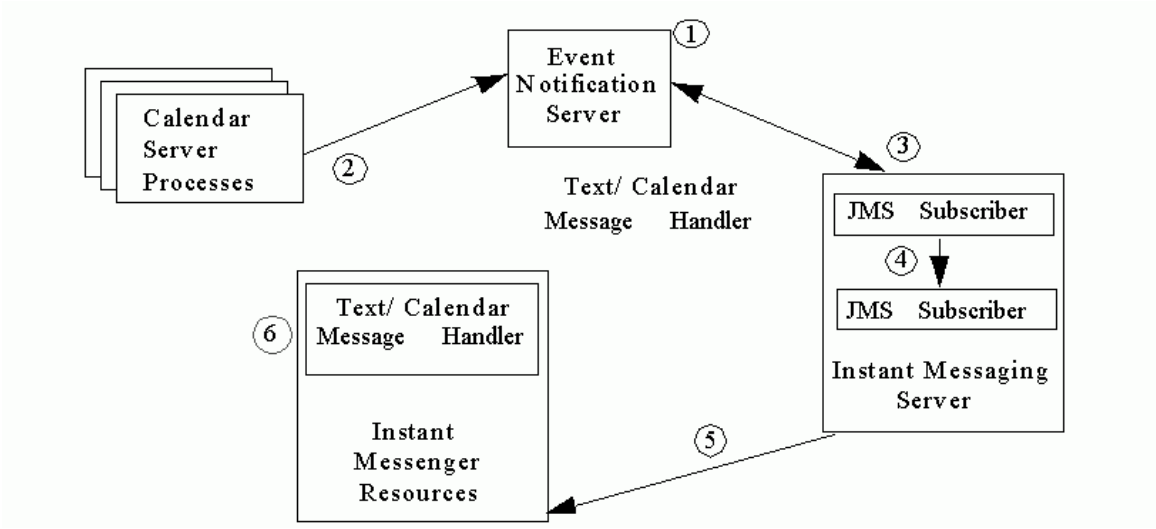
Displaying Calendar Reminders and Notifications as Instant Messenger Popups

You can configure the Instant Messenger to display calendar reminders and notifications as Instant Messenger popups if you have deployed Sun ONE Calendar Server and Sun ONE Instant Messaging server. The calendar notifications that are formatted as text/xml or text/calendar are parsed by Instant Messaging and are displayed as popups in Instant Messenger.

To access only the popup feature the calendar users can use the “POPUP” Messenger Flavor instead of using the full Instant Messenger capability. For more information on setting the Messenger Flavor, see Controlling The Exposed Messenger Feature Set.

NOTE As Single Sign-on is not provided between Sun ONE Calendar Server and Sun ONE Instant Messaging, the user has to authenticate separately on both these services. You need not reauthenticate if these services are deployed in a portal environment.

Figure 2-1 Instant Messenger Pop-up Architecture.



The Instant Messenger popups architecture for displaying calendar reminders and notifications as can be explained as follows:

1. Java Messaging Service (also referred as JMS) Subscriber subscribes for the calendar events to the Calendar Event Notification Server (ENS).
2. Calendar Server publishes events or notification to the Event Notification Server (ENS).
3. The JMS Subscriber receives these reminders and events as messages from the Calendar Event Notification Server (ENS).
4. From these messages the JMS Subscriber generates text or calendar Instant Messaging messages.

   If the owner of the calendar is online, the server sends these Instant Messaging messages to the owner.
5. The text/calendar message handler in the Instant Messenger generates the HTML alerts based on the content of these messages that is parsed by the Instant Messaging server.

Displaying calendar reminders and notifications as Instant Messenger popups contain the following components:

JMS Message Listener or Subscriber. This module implements the JMS `javax.jms.MessageListener` interface. For each JMS message received, it builds Instant Messaging notification (alert) message. The following server configuration and contents of incoming JMS messages is used:

- The Calendar ID Event URI parameter is used to determine the User ID of the calendar owner. The User ID is then used to build the recipient address of the alert.
- The `comptype` parameter determines the type of calendar object (event or task) described by the message body.
- The subject of the generated message is derived from the configuration and the component type.
- The originator of the message is provided in the configuration.
- The message contents can either be text/xml or text/calendar. If the incoming message is of type text/xml, it is converted to a text/calendar. This text/calendar representation is used to generate the Instant Messenger alert by the Text/Calendar Message Handler.

Text/Calendar Message Handler. This module is a Messenger Bean defined by the Messenger Bean specification. It intercepts all Instant Messaging messages or message type. For each message it intercepts, it generates an HTML alert and displays this alert in the Instant Messenger. It uses the following attributes of the incoming Instant Message to convert it to an HTML alert:

- The subject of the incoming message provides information on the message type, such as reminders, calendar database notifications, events and tasks. Each type of calendar event corresponds to a localized subject and is displayed in the popup.
- The text of the alert is generated from the information contained in the text/calendar message content. One template is provided for each of the event and the task.

Configuring Calendar Server for Displaying Calendar Reminders and Notifications as Instant Messenger popups

The following needs to be configured in the Calendar Server for displaying calendar reminders and notifications as Instant Messenger popups:

Enabling Alarms

Alarms need to be enabled in the Calendar server and the Calendar Event Notification Server (ENS) needs to be configured to send and receive alarm notifications.

Code Example 2-1 shows the values for alarm configuration parameters in the file `ics.conf` in the directory *calserver-root*`/cal/bin/config/`.

Code Example 2-1 Alarm configuration parameters in the file `ics.conf`.

```
caldb.serveralarms = yes
caldb.serveralarms.dispatch = yes
caldb.serveralarms.dispatchtype = ens
```

A custom alarm URL must be defined in the `ics.conf` file to enable text/xml or text/calendar formatted notifications.

Code Example 2-2 shows an example of the custom alarm URL and content type defined in the file `ics.conf`.

Code Example 2-2 Custom alarm URL defined in the file `ics.conf`.

```
caldb.serveralarms.url = enp:///ics/customalarm
caldb.serveralarms.contenttype = text/calendar
```

Configuring Instant Messaging Server for Displaying Calendar Reminders and Notifications as Instant Messenger Popups

The Instant Messaging server has to be configured for displaying calendar reminders and notifications as Instant Messenger popups. The JMS client in Instant Messaging server needs to be provided with the instructions on how to communicate to the ENS broker as the Calendar Server uses the Event Notification Server (ENS) as a JMS Bus.This is performed using the server configuration options in the file `iim.conf`.

Table 2-7 shows the server configuration options in the file `iim.conf` and their description.

Table 2-7 The Server Configuration Options in the file `iim.conf` and their description.

Option	Description
`jms.consumers`	This option contains the list of consumers identifiers separated by comma. Each consumers identifier is used to build option names whose values describe a specific consumer.
`jms.providers`	This option contains the list of JMS provider identifiers separated by comma. Each identifier is associated with a JMS provider and used in option names whose values describe the provider
`jms.consumer.`*consumer*`.provider`	Identifier of the JMS provider is associated with the JMS consumer module called consumer. Consumer is replaced by an actual consumer identifier specified in the option `iim.jms.consumers`.
`jms.consumer.`*consumer*`.name`	This option contains JMS topic or queue name associated with *consumer consumer*.
`jms.consumer.`*consumer*`.type`	This option contains the JMS *consumer consumer* type. It may contain the following values: • `topic` for JMS Topic subscription or • `queue` for JMS queue bindings. The default value is `topic`.
`jms.consumer.`*consumer*`.factory`	Using `JMSMessageListenerFactory` a Message Listener for *consumer consumer* can be instantiated and registered as a JMS callback.
`jms.consumer.`*consumer*`.param`	This option contains a free-form ascii string which is made available to the Message Listener. The string may contain additional information needed to process the incoming JMS messages that is specific to the consumer.

Table 2-7 The Server Configuration Options in the file `iim.conf` and their description.

Option	Description
`jms.provider.`*provider*`.broker`	This option contains the JMS broker host and port used while initializing the JMS provider *provider*.
`jms.provider.`*provider*`.factory`	This option contains the `ConnectionFactory` class name for the provider

Code Example 2-3 shows the JMS provider definition that is to be provided in the file `iim.conf`.

Code Example 2-3 JMS Provider definition in the file `iim.conf`.

```
jms.providers = ens
jms.provider.ens.broker = ical.example.com:7997
jms.provider.ens.factory =
com.iplanet.ens.jms.ENSConnectionFactory
```

Code Example 2-4 contains JMS consumers definition for the calendar in the file `iim.conf`.

Code Example 2-4 JMS consumers definition for the calendar.

```
jms.consumers = calendar [,...]
```

Code Example 2-5 contains the JMS consumer type and the JMS provider for the Calendar Server in the file `iim.conf`.

Code Example 2-5 JMS consumer type and the JMS provider for the Calendar Server.

```
jms.consumer.calendar.type = topic
jms.consumer.calendar.provider = ens
```

Code Example 2-6 contains the JMS consumer topic name for the Calendar Server in the file `iim.conf` file.

Code Example 2-6 JMS consumer topic name for the Calendar Server

```
jms.consumer.calendar.topic = enp:///ics/customalarm
```

Additional parameters need to be added to build the Instant Messenger message. The `jms.consumer.consumer.param` option is used for building the message. Calendar Server message listener uses a list of parameters in the following URL style:

```
params := param "=" value *("&" param "=" value)
param  := URL-ENCODED
value  := URL-ENCODED
```

The following parameters are supported by the Calendar Server message listener:

- `eventtype` contains the calendar event type. The values of the parameter `eventtype` is equivalent to the subject properties used by the Messenger Calendar Bean.
- `originator` contains the address of the originator. This value is used in the generated message

Table 2-8 contains the values for the parameter `eventtype` and their description.

Table 2-8 Values for the parameter `eventtype` and their description.

eventtype Value	Description
`calendar.alarm.event`	This value contains the event reminder.
`calendar.alarm.todo`	This value contains the task reminder.
`calendar.alarm`	This value contains both the event and the task reminder. The event or the task nature is determined from `comptype` URL parameter.
`calendar.notification.new.event`	This value contains the event creation notification.

Table 2-8 Values for the parameter eventtype and their description.

eventtype Value	Description
calendar.notification.new.todo	This value contains the task creation notification
calendar.notification.new	This value contains the component creation notification. The event or task nature is determined from the comptype URL parameter.
calendar.notification.mod.event	This value contains the event modification notification.
calendar.notification.mod.todo	This value contains the task modification notification.
calendar.notification.mod	This value contains the component modification notification. The event or the task nature is determined from the comptype URL parameter.
calendar.notification.del.event	This value contains the subject to be used for event deletion notification.
calendar.notification.del.todo	This value contains the subject to be used for task deletion notification.
calendar.notification.del	This value contains component deletion notification. The event or the task nature is determined from the comptype URL parameter.

The following example shows jms.consumer.calendar.param option with the parameters eventtype and originator:

```
jms.consumer.calendar.param =
eventtype=calendar.alarm&originator=ical
```

Example for Displaying Calendar Reminders and Notifications as Instant Messenger Popups

In this example, lets assume Sun ONE Calender Server 5.1.1 is installed on `cal.example.com` and Sun ONE Instant Messaging server 6.0 is installed on `im.example.com`. After configuring the Calendar Server and Instant Messaging server the Instant Messaging users should be able to receive reminders for calendar events and tasks.

Configuring Calendar Server

Ensure that you have installed Sun ONE calendar server with the latest patch. Based on the assumption that Calendar is installed on `cal.example.com` and Instant Messaging is installed on `im.example.com`, the following are the options that needs to be configured in the file `ics.conf`:

```
caldb.serveralarms = "yes"
caldb.serveralarms.contenttype = "text/xml"
caldb.serveralarms.dispatch = "yes"
caldb.serveralarms.dispatchtype = "ens"
caldb.serveralarms.url = "enp:///ics/customalarm"
```

If the configuration changes are made, restart the calendar server with the following commands:

```
stop-cal
```

```
start-cal
```

Configuring Instant Messaging Server

The following options needs to be configured in the file `iim.conf`:

```
! JMS Consumers
jms.consumers=cal_reminder
jms.consumer.cal_reminder.destination=enp:///ics/customalarm
jms.consumer.cal_reminder.provider=ens
jms.consumer.cal_reminder.type=topic
jms.consumer.cal_reminder.param="eventtype=calendar.alarm"
jms.consumer.cal_reminder.factory=com.iplanet.im.server.JMSCalen
darMessageListener

! JMS providers
jms.providers=ens
jms.provider.ens.broker=cal.example.com:7997
jms.provider.ens.factory=com.iplanet.ens.jms.EnsTopicConnFactory
```

After configuring the Instant Messaging server, restart the server with the following command:

```
/opt/SUNWiim/sbin/imadmin refresh
```

Enabling Calendar Alerts in Instant Messenger

To enable the Calendar alert in Sun ONE Instant Messenger:

1. Click Settings icon in the Main window or select Settings from the Tools menu to display the User Settings window.
2. Select the Alerts tab, and check “Enable calendar alerts” options.

Troubleshooting Display Calendar Reminders and Notifications as Instant Messenger Popups

If calendar alerts are not displayed, follow the steps outlined below to troubleshoot the problem.

1. Check whether the reminder was generated. The best way to do this is to check if the email reminder was received.
2. Does the Instant Messaging server receive reminders from the Calendar Server (ENS). Check the Instant Messaging server log file to see whether any data is received from calendar. In the log file, look for records which have `CalendarReminder` in them. Change the log severity in the server

(`iim.log.iim_server.severity`) to `debug` in order to gather this information. If you observe that the log file has not logged in any reminders it means that the Instant Messaging server has not received any reminders from the Calendar Server. This may have occurred due to a problem in connecting with ENS or a mismatch between ENS event references used by Calendar and Instant Messaging (`enp:///ics/customalarm` in the example above).

If calendar reminders are logged in to the log file but the events and tasks are not displayed on the Instant Messenger then move to the next step.

3. Instant Messenger may have received the calendar alert but is unable to display it. More information on this problem can be obtained by enabling the Java console. For more information on the `debug` applet parameter, set the value of the parameter to `true` in the applet page `im[ssl].html` or `im[ssl].jnlp` or `jnlpLaunch.jsp` or `pluginLaunch.jsp`.

Taking a Back Up of Instant Messaging Data

Instant Messaging does not come with any disaster recovery tools. Use your site's backup system to backup the configuration and database directories periodically.

Backup Information

The Instant Messaging information that needs to be backed up are of the following types:

- Configuration Information
- Instant Messaging user data

The configuration information is stored in the following directories:

- Solaris: `/etc/opt/SUNWiim/default/config`
- Linux: `/etc/opt/soim/default/config`
- Windows: *im_install_dir*`\config`
- (Optional) If you customized any of the files mentioned in Customizing Instant Messenger, back them up from the *$imdocroot*.

The Instant Messaging user data is stored in the following directories:

- Solaris: `/var/opt/SUNWiim/default/db`

- Linux: `/var/opt/soim/default/db`
- Windows: *im_install_dir*`\db`

Performing Backup

While the configuration information does not change frequently, the Instant Messaging user data changes rapidly and to prevent any loss of user data it is recommended that the Instant Messaging user data is backed up on a periodic basis. Backup needs to be performed before running the installation program and the uninstallation program.

To backup the user data and the configuration information you do not have to stop the Instant Messaging server as all the disk commits by the server are automatically performed.

Restoring the Back Up Information

The back up of the user data and the configuration information needs to be restored when there is a disk failure and all the user data and the configuration information is lost.

To restore the backed up user data:

1. Change to the runtime directory. For example:

   `cd` *`$runtimedir`*

2. Grant read-only permission to the db directory, type:

   `chmod -R 400 db`

3. Stop the Instant Messaging server, type:

   `imadmin stop`

4. Grant Write permission on the user data files for the server user, type:

   `chmod -R 600` *`$runtimedir`*`/db/.`

5. To restore the data, copy the backed up data to the `db` directory.

6. Start the Instant Messaging server, type:

   `imadmin start`

   On Solaris - the default value of *$runtimedir* is `/var/opt/SUNWiim/default.`

On Linux - the default value of *$runtimedir* is `/var/opt/soim/default`.

Chapter 3

Managing Instant Messenger

This chapter describes how to customize and administer Sun ONE Instant Messenger.

This chapter contains the following:

- Configuring Sun ONE Instant Messenger
- Invoking Instant Messenger
- Solving Web Server Issues
- Customizing Instant Messenger
- Administering Instant Messenger Conference Rooms and News Channels
- Modifying Instant Messenger Proxy Settings
- Controlling The Exposed Messenger Feature Set
- Instant Messenger Data Stored in the End User's System

Configuring Sun ONE Instant Messenger

There are two ways to configure and invoke Instant Messenger:

- **Using Java Web Start** - In this configuration, Instant Messenger is launched as an application from the Java Web Start. The browser dependency is removed once the Instant Messenger is launched.
- **Using the Java Plug-in** - In this configuration, Instant Messenger is run as a Java applet. To keep the Instant Messenger session active, the browser window from which the applet was launched must remain open and cannot be used to locate any other URL.

For more information on how to configure the Java software that enables Instant Messenger, see the *Sun One Instant Messaging Installation Guide.*

Invoking Instant Messenger

You can invoke Instant Messenger using:

- The `index.html` file provides you the options to launch both the Java Web Start and Java Plug-in versions of the Instant Messenger. This file also contains links to Instant Messenger documentation.
- The web page that you have designed with a a link to the Instant Messenger.
- Providing a direct URL for either the `im.html` or `im.jnlp` files.

To Invoke the Instant Messenger

Use the following URL to invoke the Instant Messenger.

NOTE Do not confuse between the multiplexor listen port and the web server port mentioned below. The installation refers to the `iim_mux.listenport` as the "web client port" whose default value is 49909. Do not include this port in the URL to invoke Instant Messenger.

`http://`*webserver***`:`***webserverport***`/`***subdirectory***`/`***filename*

In this URL,

webserver	Specifies the name of the web server where you have installed the Instant Messenger resources.
webserverport	(Optional) Specifies the web server port. The default value is `80`.
subdirectory	(Optional) Specifies the directory where the client files are installed. If the default web server document root directory is selected during the installation, then no subdirectory is required to store the client files.

filename	Specifies the Instant Messenger file to use: `index.html` - This file is provided by the product. The file contains the links to launch both the Java Web Start and Java Plug-in versions of Instant Messenger. `im.jnlp` - The jnlp file to launch only the Java Web Start version of Instant Messenger. `im.html` - The page to launch only the Java Plug-in version of Instant Messenger.

You can also, do the following:

- Add the URL to your favorites.
- Launch the application using the Java Web Start icon on your desktop.
- Use the shortcut on your desktop. You can also create a shortcut by setting the target value as `$Java-Web-Start/javaws.exe 'URL'.`
- On Solaris, to invoke Instant Messenger from the command-line, type:

  ```
  $Java-Web-Start/javaws URL
  ```

Solving Web Server Issues

This section contains web server issues that apply to standalone deployments and also to portal deployments, where the Instant Messenger is not installed on the Portal Server host and uses a different Web Server to serve Instant Messenger.

Changing the Codebase

By default, Instant Messaging fetches the Instant Messenger resources installed in the web server's document root. If you have installed the Instant Messenger files in a directory other than the web server document root, you will need to know the following details:

- Web server - You can configure the web server to enable access to the directory where the Instant Messenger files are installed, or create a symbolic link in the web server's document root.

For example, if Instant Messaging server host is `iim.i-zed` and the Instant Messenger files are installed in the `/opt/SUNWiim/html` directory, you need to create a symbolic link such as `iim` which points to the `/opt/SUNWiim/html` directory, in the web server's document root.

NOTE If you are using a symbolic link, you need not change the web server configuration.

- URL to launch Instant Messenger - The URL that is used by the users to access the `index.html` (and `im.html` and `im.jnlp` files). This URL needs to reference the Instant Messenger installation directory.

  For example, if the Sun ONE Instant Messaging server host is `iim.i-zed` and the Instant Messenger files are installed in the `/opt/SUNWiim/html` directory, you need to create a symbolic link such as `iim`, which points to the web server's document root `/opt/SUNWiim/html`. The users can access the Instant Messenger main page `index.html` using the following URL:

  `http://iim.i-zed.com/iim/`

  Users can also type the following URLs to launch Instant Messenger directly:

  For Java Plug-in:

  `http://iim.i-zed.com/iim/im.html`

  For Java Web Start:

  `http://iim.i-zed.com/iim/im.jnlp`

- Launch the Instant Messenger using Java Web Start- If the Instant Messaging Codebase specified during the installation is changed then you need to change the `codebase` parameter in the `im.jnlp` file to reference the web server and the Instant Messenger path. The following is the change to the `codebase` parameter:

  `codebase= http://`*`servername:port/path/`*

  You need to include the port number of the web server if it is not configured to the default value `80`.

  For example, if the Sun ONE Instant Messaging server host is `iim.i-zed` and the Instant Messenger files are installed in the `/opt/SUNWiim/html` directory, you can create a symbolic link such as `iim` which points to `/opt/SUNWiim/html` in the web server's document root. Then change the `codebase` parameter in the `im.jnlp` file to the following:

```
codebase="http://iim.i-zed.com/iim/"
```

NOTE The `im.jnlp` file is used for Java Web Start. If you are using Java Plug-in to launch Instant Messenger, then you need not modify these files.

Changing Web Server Port

If your web server is installed on a port other than the default (80), you need to know the following details:

- Launching Instant Messenger using Java Web Start - Edit the `im.jnlp` file and change the `codebase` parameter to:

  `codebase="http://`*webserver*`:`*webserverport*`"`

  For example, if the Sun ONE Instant Messaging server host is `iim.i-zed` and the web server is running on port 8080, the `codebase` parameter in the `im.jnlp` file should be:

  ```
  codebase="http://iim.i-zed.com:8080"
  ```

- Launching Instant Messenger using URL - The URL for the `index.html` and `im.html` and `im.jnlp` files needs to reference the web server port.

  For example, if the Sun ONE Instant Messaging server host is `iim.i-zed` and the web server port is `8080`, the URL to access the Instant Messenger main page `index.html` will be:

  ```
  http://iim.i-zed.com:8080
  ```

Customizing Instant Messenger

While designed to work right out of the box, Instant Messenger is also customizable. Instant Messenger consists of HTML and JNLP files that can be customized to suit an organization's specific needs.

You can customize the Instant Messenger to meet your requirements in the following ways:

- Customizing the index.html and im.html Files (Only Standalone Deployments)
- Customizing the Application (Java Web Start)

- Customizing User Name Display

This section describes the Instant Messaging server files you can modify to customize the Instant Messenger. The files that you can customize are all located in the `html` directory. For example, on Solaris the HTML files are located in the *imdocroot* directory.

Instant Messenger Resources

Instant Messenger Files

The Instant Messenger files are located in the *imdocroot* directory.

Table 3-1 contains the list of Instant Messenger files in the *imdocroot* directory. It also contains the description and the customization information of these files. The imdocroot directory contains *lang* which is the locale subdirectory. For example, `en_US`, `jp`, `fr_FR`.

Table 3-1 Instant Messenger Files

File	Description	Customizable?
lang/`im.html`	The initial page that launches the Java Plug-in version of Instant Messenger.	Yes.
`im.html.template`	The template version of `im.html`.	No. This file is used by the installation program to generate the `im.html` file.
`im.jar`	The client `jar` file, downloaded by `im.html` or `im.jnlp` files.	No.
lang/`im.jnlp`	The `jnlp` file to launch Java Web Start version of Instant Messenger.	Yes.
`im.jnlp.template`	The template version of `im.jnlp`.	No.
`imjni.jar`	The client `jar` file, downloaded by `im.html` or `im.jnlp`.	No.

Table 3-1 Instant Messenger Files *(Continued)*

File	Description	Customizable?
`imnet.jar`	The client `jar` file, downloaded by `im.html` or `im.jnlp`.	No.
lang / imbrand.jar	This file contains customizable properties, stylesheets, images and audio files.	Yes
lang/`imssl.html`	The Initial page that launches Java Plug-in version of Instant Messenger. This file is used for running SSL between the client and the multiplexor.	Yes.
`imssl.html.template`	The template version of `imssl.html`	No.
lang/`imssl.jnlp`	This file launches Java Web Start version of Instant Messenger. This file is used for running SSL between the client and the multiplexor.	Yes.
`imssl.jnlp.template`	The Template version of `imssl.jnlp` file.	No.
lang/`index.html`	The splash page for standalone deployment. It contains links to `im.html` and `im.jnlp`, as well as documentation links to `windows.htm`, `solaris.htm`, and `quickref.htm`. You can customize this page for your site's requirement.	Yes.
`index.html.template`	The template version of `index.html`.	No.
`SunONE.gif`	The image used by `solaris.htm` and `windows.htm`.	Can be replaced.
`SunONE.jpg`	The image used by `quickref.htm`, `solaris.htm`, and `windows.htm`.	Can be replaced.

Table 3-1 Instant Messenger Files *(Continued)*

File	Description	Customizable?
`javaws_not_installed.html`	The page that appears when a user tries to launch the Instant Messenger by using Java Web Start and Java Web Start has not been installed on the user's system.	Yes.

`quickref.html`
`solaris.html`
`windows.html`
	This document provides documentation on getting started with Instant Messenger.	Yes.

lang`/imhelp`	Instant Messenger Online Help directory.	No.
`icalendar.jar`	This jar file contains files that are used to display calendar notifications.	No.

Customizing the index.html and im.html Files (Only Standalone Deployments)

The Instant Messenger allows you to modify the "static" portion of the `index.html` and `im.html` files to produce a fully customized user interface. These HTML files contain both text and markups describing how the text is formatted and handled. Markup is implemented through a set of tags, which specify formats for headers, indents, font size, and font style.

Some of the page elements that can be modified are:

- Images and Banner
- Text on screen including title and field labels
- Background schemes

The `index.html` file launches both the Instant Messenger applet and the Java Web Start application. If you are running the Instant Messenger applet, modify the `im.html` file. The `im.html` file is called by `index.html`, and invokes the Instant Messenger applet. The `im.html` is generated during the installation and contains an applet argument that points to the multiplexor.

NOTE The argument "<PARAM NAME="server" VALUE="*servername*">" represents the Sun ONE Instant Messaging multiplexor and its port in the `im.html` file. If you change the `iim_mux.listenport` parameter's default value, you need to change the *servername* value to *host.domain*:*port*.

Customizing the Application (Java Web Start)

If you are running Instant Messenger using Java Web Start, you can modify the `im.jnlp`, `imres.jnlp`, and `imres.jar` files to customized the user interface. The following are modifications that can be made to these HTML files:

- `im.jnlp` - this file invokes the Java Web Start version of the Instant Messenger application. You can modify the codebase, title, vendor, and descriptions in the file.

  Table 3-2 shows the sample `im.jnlp` file, with the HTML code that can be customized in bold typeface.

Table 3-2 The sample `im.jnlp` file..

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Sun ONE Instant Messenger -->
<jnlp
  spec="1.0+"
  codebase="INSERT_CODEBASE_HERE"
  href="INSERT_LOCALE_HERE/im.jnlp">
  <information>
    <title>Title</title>
    <vendor>Name</vendor>
    <homepage href="http://home.htm"/>
    <description>Description</description>
    <description kind="short">Description Kind</description>
    <icon href="IM_JLF32x.gif"/>
    <offline-allowed/>
  </information>
  <security>
    <all-permissions/>
  </security>
  <resources>
    <j2se version="1.3+">
      <resources>
        <jar href="INSERT_LOCALE_HERE/imres.jar"/>
        <jar href="INSERT_LOCALE_HERE/imbrand.jar"/>
      </resources>
    </j2se>
    <jar href="messenger.jar"/>
    <jar href="imdesktop.jar"/>
    <jar href="imnet.jar"/>
    <jar href="icalendar.jar"/>
    <nativelib href="imjni.jar"/>
  </resources>
  <application-desc main-class="com.iplanet.im.client.iIM">
    <argument>server=INSERT_SERVER_HERE</argument>

<argument>help_codebase=INSERT_CODEBASE_HERE/INSERT_LOCALE_HERE<
/argument>
  </application-desc>
</jnlp>
```

NOTE In the `im.jnlp file`, the argument `<argument>`*servername*`</argument>` represents the Sun ONE Instant Messaging multiplexor host and port. If you change the default value of `iim_mux.listenport` parameter, you need to change the *servername* value to *host.domain*:*port*.

- `imbrand.jar` - This file contains the image and audio files, and the properties that can be customized. You need Java Developers Kit 1.3(JDK) to extract the contents from the imres.jar file using the `jar` command. For more information of the `imbrand.jar` file contents, see Contents Listing of `imbrand.jar` file.

  The following is the syntax for the `jar` command:

  ```
  jar xvf imbrand.jar
  ```

  This command creates a directory tree where the resource files are copied. This directory structure has to be maintained when you modify the individual files in the `jar` file.

  You can substitute your version of `.gif` files or `.au` files, without changing the file names and then place the changed files back to the directory using the following `jar` command:

  ```
  jar -uf imbrand.jar com/Sun/im/client/images/*.gif
  ```

  This command updates the `imbrand.jar` file with the modified `.gif` files. You can also do the same with the audio files (`.au` files).

Contents Listing of imbrand.jar

Table 3-3 lists the files in the `imbrand.jar` file and their description. The `imbrand.jar` file contains the image and audio files that can be used to re-brand the Instant Messenger.

Table 3-3 Contents Listing of `imbrand.jar` file.

File Name	Description
`Angry_16.gif`	An emoticon used to show angry emotion graphically.
`Devil_16.gif`	An emoticon used to show devilish emotion graphically.
`Laugh_16.gif`	An emoticon used to show laugh emotion graphically.

Table 3-3 Contents Listing of `imbrand.jar` file.

File Name	Description
`Angel_16.gif`	An emoticon used to show angelic emotion graphically
`Smiley_16.gif`	An emoticon used to show smile emotion graphically.
`Love_16.gif`	An emoticon used to show love emotion graphically
`Grin_16.gif`	An emoticon used to show grin emotion graphically.
`Wink_16.gif`	An emoticon used to show wink emotion graphically.
`Sad_16.gif`	An emoticon used to show sad emotion graphically.
`Suprise_16.gif`	An emoticon used to show suprise emotion graphically.
`Away_13.gif`	Icon for away status that appears in the Change Status menu.
`Online_13.gif`	Icon for online status that appears in the Change Status menu.
`Offline_13.gif`	Icon displayed when the user is away or not connected that appears in the Change Status menu.
`Idle_13.gif`	Icon for idle status that appears in the Change Status menu.
`Forwarded_13.gif`	Icon displayed against users when they are offline and have set alerts to be forwarded to email. This icon appears in the Change Status menu.
`Away_13Padded.gif`	Icon for away status that appears in the contact list.
`Idle_13Padded.gif`	Icon for idle status that appears in the contact list.
`Online_13Padded.gif`	Icon for online status that appears in the contact list.
`Offline_13Padded.gif`	Icon for offline status that appears in the contact list.
`Forwarded_13Padded.gif`	Icon displayed against users when they are offline and have set alerts to be forwarded to email. This icon appears in the contact list.
`Away_24.gif`	Icon for away status that appears in the status bar.
`Online_24.gif`	Icon for online status that appears in the status bar.
`Offline_24.gif`	Icon for offline status that appears in the status bar.

Table 3-3 Contents Listing of `imbrand.jar` file.

File Name	Description
`tray_icon.ico`	Instant Messenger icon that appears on the task bar.
`app_icon.gif`	The Instant Messenger application icon.
`logon_splash.gif`	The Sun ONE logo displayed in the Login and About boxes
`alert.au`	Sound when user receives an alert.
`away.au`	Sound when the user changes status to away.
`soundon.au`	Sound when user exits Instant Messenger.
`soundoff.au`	Sound when user logs on to Instant Messenger.
`send.au`	Sound when user sends an instant message.
`receive.au`	Sound when user receives an instant message.

Rebranding Instant Messenger

The `imbrand.jar` file contains all images and the properties that control the look and feel of the Instant Messenger. You can customize the appearance of the Instant Messenger by modifying the images and the properties in the `imbrand.jar` file.

To re-brand the Instant Messenger:

1. Copy `imbrand.jar` file to a working directory and change to this directory. For example:

   `cp` *`imdocroot/lang/`*`imbrand.jar` *`working_directory`*

2. Extract the `imbrand.jar` file.

   ```
   jar xf imbrand.jar
   ```

   This command creates a directory tree where the resource files are copied. This directory structure has to be maintained when you modify the individual files in the `jar` file.

3. Update the `imbrand.jar` file with the modified `.gif` files and `.au` files.

   ```
   jar cf imbrand.jar .
   ```

4. Copy the imbrand.jar file to the *imdocroot* directory. For example:

   `cp imbrand.jar imdocroot/lang/.`

NOTE If multiple locales are supported, the procedure for re-branding Instant Messenger should be followed for every supported locale.

Customizing User Name Display

The User Name display can be customized in the tooltip and the search results.

Customizing User Name Display in Search Results

When two users have the same first name and last name, it is impossible to know which user has to be added to the contact list. You can customize the Instant Messenger to display more information in the search results for the user search. For displaying more information in the user search results, in the `imbrand.jar` file you need to add `dialogs.searchresults.format` attribute to the `brand.properties` file at:

`com/sun/im/client/`

For more information on how to modify `imbrand.jar`, see Customizing the Application (Java Web Start).

More information can be displayed in the user search results by including additional LDAP attribute values in the `dialogs.searchresults.format` attribute.

The LDAP attributes are specified in the following format:

`${attr:attribute-name}`

The following example shows the LDAP attribute in `dialogs.searchresults.format` attribute:

`dialogs.searchresults.format=(${attr:title})`

To use random attributes from the LDAP user entry, the list of these custom attributes needs to be specified in the server configuration file `iim.conf`. These custom attributes need to be specified as values for the attribute `iim_ldap.userattributes`.

The following example shows the `iim_ldap.userattributes` with the list custom attributes as value:

`iim_ldap.userattributes=title,department,telephonenumber`

Customizing User Name Display in Tooltip

You can customize the Instant Messenger to display additional information in the Contact tooltip.

For example, to display the phone number of the Contact when the mouse is placed over the Contact:

1. Change to the following directory:

   `com/sun/im/client/`

2. Open the `brand.properties` file.
3. Add the `contact.tooltip.format.html` attribute to the file.
4. Save the changes to the file
5. Change to the following directory.

   cd *imdocroot*

6. Add the `contact.tooltip.format.html` attribute and the `telephonenumber` attribute as its value in the HTML code of the `imbrand.jar` file:

   ```
   contact.tooltip.format.html=mailto: ${attr:mail} tel:
   ${attr:telephonenumber}
   ```

For information on customizing the `imbrand.jar` file, see Customizing the Application (Java Web Start).

Administering Instant Messenger Conference Rooms and News Channels

Listed below are tasks that you can perform in the Instant Messenger to administer the conference rooms and the news channels. For more information on performing these tasks, see the *Instant Messenger Online Help*.

- Administering conference rooms
- Administering and managing news channels
- Assigning conference room access levels to users
- Assigning news channel access levels to users
- Assigning users to conference rooms

- Assigning users to news channels (subscribing)
- Creating new conference rooms
- Creating new news channels
- Configuring user settings
- Deleting conference rooms
- Deleting messages from news channels
- Deleting news channels
- Posting messages in news channels
- Removing users from conference rooms
- Removing users from news channels

Granting Users Privilege to Create Conference Rooms and News Channels

The administrator can create conference rooms and news channels for users. However, individual users can also create conference rooms and news channels, if they have the privileges to add conference rooms and news channels. The users who create a conference room or a news channel by default have Manage access, enabling them to administer the conference room or the news channel.

- To permit users to create conference rooms, add the user(s) or group(s) to the `sysRoomsAdd.acl` file.
- To permit users to create news channels, add the user(s) or group(s) to the `sysTopicsAdd.acl` file.

For more information on managing user privileges, see Managing User Privileges.

Modifying Instant Messenger Proxy Settings

The instant messages can contain embedded URLs. For example, `http://stocks.yahoo.com?id=sunw`. If you are using proxy servers, you need to resolve such embedded URLs by modifying the Instant Messenger proxy settings in the Java Web Start configuration.

This is likely to happen if your organization has a firewall, and you need to go through the proxy server before connecting your client hosts to internet, and if Java Web Start has not been configured with the right proxy settings.

To Modify Instant Messenger Proxy Settings

Java Web Start can automatically configure the proxy settings by querying the system or the default browser. However, it is not possible for the Java Web Start to automatically configure these settings if the proxy settings are configured using a JavaScript file.

To set the proxy settings manually:

1. Invoke Java Web Start.
2. From the File menu, choose Preferences.
3. Select Manual option in the Preferences dialog.
4. Enter the following details:

   HTTP Proxy. Enter the Name or the IP address of the proxy server.

   HTTP Port. Enter the port number of the proxy server.

   No Proxy_Hosts. Enter the name of any domain that you can connect directly, bypassing the proxy server. Use commas to separate multiple host names.
5. Click OK to save the proxy settings.

Controlling The Exposed Messenger Feature Set

The exposed feature set of the Instant Messenger can be controlled by the administrator by configuring the Instant Messaging applet parameters in the applet descriptor files.

Table 3-4 shows the Instant Messenger applet parameters in the applet descriptor files. It also contains the description and the default values of these parameters.

Table 3-4 Instant Messenger Applet Parameters

Parameter	Default Value	Description
`server`	`127.0.0.1`	The Instant Messaging server host and port.

Table 3-4 Instant Messenger Applet Parameters

Parameter	Default Value	Description
debug	FALSE	If this parameter is set to true, the applet records all the task performed on java console.
uid		This parameter is used for SSO.
token		This parameter contains SSO token and is used for auto-logon.
secure	FALSE	Indicates to the messenger that it is run in SRA mode. It displays secure indicator.
usessl	FALSE	Tells messenger to use SSL when connecting to server.
allow_alert_only	FALSE	Tells messenger to let user display neither the contact list nor the news channel. This parameter is used in CHAT and POPUP flavors.
allow_file_transfer	TRUE	Allows file attachment and transfer.
flavor	ALL	Flavor to be used. The Supported values are ALL(default), IM, NEWS, CHAT and POPUP. ALL: All services are enabled. IM: presence, conferencing and alert (notification) services are enabled. NEWS: Only news and alert services are enabled. Presence/contact list are not. CHAT: only the private conference service is accessible. POPUP: the messenger is only capable of receiving alerts.
enable_moderator	TRUE	If set true, enables the moderated conference feature.

Table 3-4 Instant Messenger Applet Parameters

Parameter	Default Value	Description
`messenger_bean`		This parameter contains a list of messenger beans to be used. You can enter multiple factory class names separated by comma.
`gateway_url`		This parameter contains the URL of the gateway component of portal SRA.

Instant Messenger Data Stored in the End User's System

Instant Messenger caches limited amount of information on the end user's system for auto-login. This information can be located at:

${HOME}`/.sunmsgr`

${HOME} is the user's home directory. The home directory of the user can be obtained from the `user.home` parameter in the Java system property.

Table 3-5 shows the directories and files containing the cached data. It also contains the description of the files and the directories.

Table 3-5 The directory and files containing the Cached Data

File/Directory Name	Type	Description
`.sunmsgr/messenger.properties`	file	The file containing the auto-logon properties
`.sunmsgr/`<*user_domain*>`/`	directory	Directory containing data specific to a particular {loginname-domainname} combination.
`.sunmsgr/`<*user_domain*>`/messenger.properties`	file	This file contains auto-logon options specific to particular <*user_domain*>. This file is not used.
`.sunmsgr/`<*user_domain*>`/messages/`	directory	This directory contains cached messages. This directory is not used.

Table 3-6 shows the auto-logon properties for Instant Messaging. It also contains the description and the default values of these properties.

Table 3-6 The Auto-logon Properties

Parameter	Default Value	Description
`net.server`	`127.0.0.1`	Instant Messaging server host name and port.
`net.server.`*n*		The secondary servers' host names and port numbers.
`net.user`		The default user id
`net.pass`		The encoded user password that enables auto-logon.

Managing Instant Messaging Policies Using Identity Server

This section describes what you need to know about administering Instant Messaging server when it is deployed with Sun ONE Identity Server. This section contains:

- Instant Messaging Service
- Restricting Users From Accessing Instant Messenger

Instant Messaging Service

When you deploy Instant Messaging with Sun ONE Identity Server, an Instant Messaging service is added to the Sun ONE Identity Server. The Instant Messaging service enables the administrator to enforce policy mechanisms for accessing Sun ONE Instant Messaging.

Table 4-1 lists the Instant Messaging service attributes.

Table 4-1 Instant Messaging service attributes

Service Attributes	Description
`sunIMEnable`	This is a boolean attribute. When enabled it has Access and Deny permissions for an organization. These attributes will be added as dynamic attributes.
`sunIMAllowAlertOnly`	This is a boolean attribute. When enabled the instant messenger only displays the alerts. The contact list or the news is not displayed. This attribute is used in CHAT and POPUP flavors. By default this attribute is disabled.
`sunIMAllowFileTransfer`	This is a boolean attribute. When enabled it allows files to be attached to the messages. By default this attribute is enabled.

Table 4-1 Instant Messaging service attributes

Service Attributes	Description
`sunIMEnableModerator`	This is a boolean attribute. It enables the moderated conference feature in Instant Messenger. By default, this attributed is enabled.
`sunIMFlavor`	This attribute can be selected from a drop down list. It describes the message type to be enabled. The values are `ALL`, `IM`, `NEWS`, CHAT and `POPUP`. The default selected value is `ALL`.

Restricting Users From Accessing Instant Messenger

You can restrict the users from running Instant Messenger by unassigning the Instant Messaging policy for the users.

Unassigning Instant Messaging Policy

This approach is recommended if the roles to which Instant Messaging policies should be assigned or unassigned are already present in the organization.

By default the Instant Messaging policy is added to the organization with the permission to run Instant Messenger. To disallow certain users in the organization from accessing Instant Messenger, you need to:

1. Create a new policy. For example: “Disable IM”.
2. Unassign the Instant Messaging Policy for the organization.
3. Add Disable IM policy to the role for which you want to disallow access to Instant Messenger.
4. Add users to whom you want to disallow access to Instant Messenger to this role.

To Create A New Policy

1. Login to Sun ONE Identity Server as Administrator.
2. Select Policy Management from View drop down list.
3. Click the organization link in which you want to create the policy. For example: `siroe.com`.
4. Select Policies from the Show drop down list

5. Click New. The Create Policy page is displayed in the right panel.
6. Specify the following details:

   Name. Enter the policy name in the field. For example: **`Disable IM`**.

   Service. Select Instant Messaging from the drop down list.
7. Click Create. The policy is created.
8. To add a rule, click Add Rule.
9. To disable the access to Instant Messenger, uncheck “Has Privilege to Run IM” check box. To allow access to Instant Messenger select “Has Privilege to Run IM” check box.
10. Click Save.

To Unassign the IM Policy from the Organization

1. Select User Management from View drop down list.
2. Click the organization link in which you want to unassign Instant Messaging policy. For example: `siroe.com`.
3. Select Policies from Show drop down list.
4. Select the name of the policy. For example, Enable IM.
5. Click Unassign.

Chapter 5

Managing The Instant Messaging Archive

This chapter explains how to manage and configure the Sun ONE Instant Messaging Archive.

This chapter contains the following sections:

- Instant Messaging Archive Overview
- Archiving Instant Messages
- Enabling the Archive Provider
- Configuring the Archive Provider
- Managing Archived Data in the Portal Server Search Database
- Enabling Instant Messenger Archive Control
- Changing the Display of the Archived Data
- How to Use CGI Script to Submit the Data
- Sample Deployment Scenario for Archive Provider

Instant Messaging Archive Overview

The Instant Messaging archive captures instant messages and archives these messages in a Portal Server Search database. It enables the user to query and retrieve these archived messages using the Search page on the Portal Server desktop.

Sun ONE Instant Messaging Archive contains the following components:

Archive and Retrieval Component. Sun ONE Portal Server Search 6.0 component also known as Archive and Retrieval component is used to store the archived Instant Messages. The Instant Messaging archive data is indexed and can be assigned to categories and stored in the Portal Server Search database. For example, alert messages can be stored under the Alert category.

NOTE Storing data in separate categories helps in simplifying the search operation and enables quick retrieval of the archived data.

Instant Messaging Archive Search or Display. When the user performs a search operation for documents matching certain criteria, the Portal Server Search fetches pages matching this criteria. These pages can be remote web pages or they could be Instant Messaging archive data also referred as Instant Messaging resource descriptors.

- For the remote web pages, the URL of the pages matching the criteria is listed in the Search Results List. When the user clicks the URL of a web page in the Search Results List, the browser fetches this page from the remote web server.
- For the Instant Messaging Resource Descriptor, the archive data is stored in the Portal Server Search database and is not available as downloadable documents from the web server.

When the user clicks the URL of the Instant Messaging resource descriptors to view the archive data, the Instant Messaging Archive Search or Display JSP is invoked. The Instant Messaging Archive Search JSP retrieves the information from the Portal Server Search database and generates a text or HTML response containing the Instant Messaging Archive data.

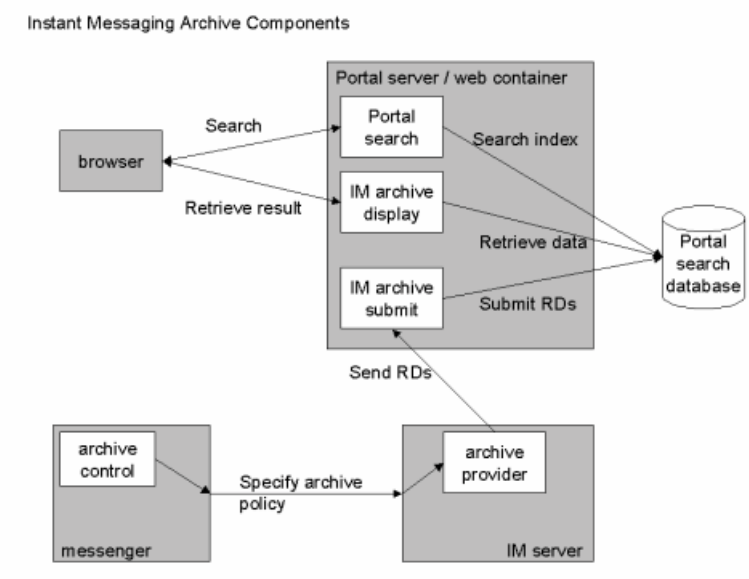
Instant Messaging Archive Submit Servlet. Sun ONE Portal Server Search 6.0 does not provide a method for adding records to the Portal Server Search database from a remote server. A command-line utility `rdmgr` is used to add new records to the Portal Server Search database. The Instant Messaging Archive Submit Servlet enables addition of records to the Portal Server Search database from a remote Instant Messaging server.

Instant Messaging Archive Provider. This component is invoked by the Instant Messaging server whenever Instant Messages are to be archived. The Instant Messaging Archive Provider builds the Summary Object Interchange Format (SOIF) complaint Resource Descriptors (RD) based on the data provided by the Instant Messaging server. It sends these Resource Descriptors to the Instant Messaging Archive Submit servlet for them to be added to the Portal Server Search database. It also maintains a buffer of the records to be submitted to the Portal Server Search database, to reduce the performance hit.

Instant Messenger Archive Control. Instant Messaging data can be archived automatically without any interaction from the end user. To control the archive functionality the end user needs to enable the Instant Messenger Archive Control component. This component allows the user to set default archive options, such as “archive all conferences”, and change the default on a per-transaction basis. For example, the user can choose not to archive the content of the conferences.

Figure 5-1 illustrates Sun ONE Instant Messaging Archive components.

Figure 5-1 Sun ONE Instant Messaging Archive components



Archiving Instant Messages

All instant messages are divided into the following categories for the purpose of archiving:

Chat. All the messages in the private conference rooms.

Conference. All the messages in the public conference rooms.

Alerts. This category contains all the alert messages.

Poll. This category contains all poll messages.

News. This category contains all messages posted in the news channels.

The following are the features of Sun ONE Instant Messaging Archive Provider:

- It captures all the Instant Messaging traffic passing through the server.
- The archived data can be stored under separate categories on the Portal Server Search.
- Storing the data as separate categories helps in simplifying the search and retrieval of the archived data.
- The search can be performed using the Portal Server desktop.
- The security feature of Portal Server Search can be used to provide an access control list. The archive provider provides security features by which only a set of admin users can be allowed to access the archived data.
- The data can be managed using the Portal Server Search database management tools.

Enabling the Archive Provider

To enable archive provider in Instant Messaging:

1. Change to the `Config` directory. For example, on Solaris:

   cd `/etc/opt/SUNWiim/default/config`

2. Open the `iim.conf` file.

   For example:

   `vi iim.conf`

3. Add the following line to the `iim.conf` file:

   For the default archive provider, add the following line:

   ```
   iim_server.msg_archive = true
   ```

   For a custom archive provider, add the following line:

   `iim_server.msg_archive.provider =` *provider_name*

   To use the Portal Server Search based archive, replace the *provider_ name* with the following:

   ```
   com.iplanet.im.server.IMPSArchive
   ```

   It is adequate to perform the above steps to enable the archive provider in Instant Messaging, provided the messenger archive control is enabled. Refer to the section on Enabling Instant Messenger Archive Control for further details.

   If the messenger archive control is disabled, it is possible to enable archiving by setting the following option:

   ```
   iim_server.msg_archive.auto = true
   ```

   This setting directs the server to ignore the end-user's archive control settings.

   By default this setting is not added by the installer during the install. This is done on purpose, to avoid archiving data by mistake before completing the archive setup. The administrator needs to add this setting to ignore the end user's archive control settings.

4. Save the file.

5. Refresh the Instant Messaging server configuration. To refresh type:

   ```
   imadmin refresh
   ```

NOTE It is recommended to define the organization of the archive before enabling it.

The rest of this chapter contains information that will help you organize the IM archive data.

Sun ONE Instant Messaging provide the APIs and SPIs that can be used to write custom archive providers. For more information on Instant Messaging APIs, see Instant Messaging APIs.

Configuring the Archive Provider

The archive provider stores the archived messages as resource descriptors (RD) in the Portal Server Search database. The archive provider uses the following fields of the Portal Server Search schema:

Title. This field contains the names of the public conference rooms for Conference category, names of the participants in a chat session for the Chat category, subject of the Alert messages and the names of the News Channels for alerts and news categories. The title field will contain "`Poll from $Sender`" for poll category, where `$Sender` is the display name of the sender of the poll.

Keyword. For conference and chat categories, this field will contain a list of all the participants in the conference room. For a public conference room, it will also contain the name of the conference room. For the Alert category, it will contain the display names of the sender and the recipients. For the News category, it will contain the name of the channel. For the Polls category, it will contain the list of sender and recipients. For all categories, in addition to the above values this field also contains a unique ID for the categories.

Table 5-1 shows the unique ID for each category and their description

Table 5-1 Unique ID for each category and their description

Category	Unique ID
Conference	`RoomName-StartTime`
Chat	Where: `RoomName` - Name of the public or private conference room `StartTime` - Is the timestamp of the creation of RD
Alert	`Alert-messageID` Where: `messageID` - Message Id of the message which will be archived. Message Id has importance when the RD contains only one message. For example, News message and Alert message.
Poll	`Poll-pollID`
News	`TopicName-messageID`

ReadACL. For the Conference and News categories, the value for this field is taken from the ACL files of the respective conference rooms and news channels. For the Chat category, this field contains the DN of the participants. For the Alert category, this field contains the sender's DN and the recipient's DN. For the Poll category, the archiver will provide a new ACL file.

The search access to the RDs is controlled by the value in the ReadACL field. If the document level security is enabled, the user has access to the search results only if the ReadACL field has the user's DN. If the Instant Messenger Archive control is enabled, for the chat messages, the user DN added to the ReadACL field depends on the user selection.

Description. This field contains the archived message without the HTML formatting.

Full-Text.This field contains the HTML formatted archived messages.

Classification. This field contains the category of the archived message.

Archive Provider Configuration Parameters

The Archive Provider configuration parameters need to be manually added to the `iim.conf` file in the directory `/etc/opt/SUNWiim/default/config`.

The following are the optional Archive Provider configuration parameters which can be added to the `iim.conf` file:

Table 5-2 Optional Archive Provider parameters which can be added to the `iim.conf` file.

Parameter	Default Value	Description
`iim_arch.title.attr`	`Title`	This parameter contains the name of the field equivalent to the `Title` field in the default schema of the Portal Server Search.
`iim_arch.keyword.attr`	`Keyword`	This parameter contains the name of the field equivalent to the `Keyword` field in the default schema of the Portal Server Search.
`iim_arch.readacl.attr`	`ReadACL`	This parameter contains the name of the field equivalent to the `ReadACL` field in the default schema of the Portal Server Search.

Table 5-2 Optional Archive Provider parameters which can be added to the `iim.conf` file.

Parameter	Default Value	Description
`iim_arch.description.attr`	`Description`	This parameter contains the name of the field equivalent to the `Description` field in the default schema of the Portal Server Search.
`iim_arch.fulltext.attr`	`Full-Text`	This parameter contains the name of the field equivalent to the `Full-Text` field in the default schema of the Portal Server Search.
`iim_arch.category.attr`	`Category`	This parameter contains the name of the field equivalent to the `Category` field in the default schema of the Portal Server Search.
`iim_arch.readacl.admin`	None	This parameter contains the administrator's DN. Multiple values should be separated by ";"
`iim_arch.readacl.adminonly`	`false`	This parameter will contain `true` or `false`. `true` - Only the administrator's DN specified by the parameter `iim_arch.readacl.admin` will be added to the `ReadACL` field overwriting the default behavior of the `ReadACL` field. `false` - The administrator's DN specified by the parameter `iim_arch.readacl.admin` will be added to the ReadACL field in addition to the default behaviour.

Table 5-2 Optional Archive Provider parameters which can be added to the `iim.conf` file.

Parameter	Default Value	Description

`iim_arch.categories`	`all`	This parameter contains a list of message types that can be archived.
		The value can be:
		`poll`
		`alert`
		`chat`
		`conference`
		`news`
		Multiple values can be specified separated by commas(“`,`”).

`iim_arch.categoryname`	None	If a category name is not assigned for any of the categories then the value of this parameter is taken as the category name.
`iim_arch.alert.categoryname`	None	This parameter contains the name of the category containing the archived alert messages. Note: It is not required to dedicate a category to alert messages.
`iim_arch.poll.categoryname`	None	This parameter contains the name of the category containing the archived poll messages. Note: It is not required to dedicate a category to poll messages.
`iim_arch.conference.categoryname`	None	This parameter contains the name of the category containing the archived conference messages. Note: It is not required to dedicate a category to conference messages.

Table 5-2 Optional Archive Provider parameters which can be added to the `iim.conf` file.

Parameter	Default Value	Description
`iim_arch.chat.categoryname`	Name	This parameter contains the name of the category containing the archived chat messages. Note: It is not required to dedicate a category to chat messages.
`iim_arch.news.categoryname`	None	This parameter contains the name of the category containing the archived news messages. Note: It is not required to dedicate a category to news messages.
`iim_arch.conference.quiettime`	`5`	This parameter contains the maximum duration of silence between two consecutive messages in a room (both public and private) after which the RD expires and a new RD is created for archiving the message. The value is in minutes.
`iim_arch.poll.maxwaittime`	`15`	This parameter contains the (maximum) time for which poll data is buffered in the server. The value is in minutes. Note: The value of this parameter should be greater than the sum of `iim_arch.submit_timer` + `iim.arch.invokerdmgr` + delay time. The default value of the delay time is 120 secs.
`iim_arch.submit_timer`	`120`	This parameter contains the maximum time before which the the Portal Server Search will be updated.

Table 5-2 Optional Archive Provider parameters which can be added to the `iim.conf` file.

Parameter	Default Value	Description
`iim_arch.ignoreexplicitdeny`	`true`	This parameter will contain `true` or `false`. `true` - For Poll and Conference category the data with explicit deny access will not be archived. Each time when these messages are not archived this information will be logged into the `server.log` file. `false` - For Poll and Conference category the data with explicit deny access will not be archived and the message will be added to the Portal Server Search database. Note: If you do not explicitly deny access to a room or a news channel then the default access is either READ or WRITE or MANAGE. Some users can also be granted NONE access.
`iim.arch.invokerdmgr`	`300`	This parameter specifies the time after which the `rdmgr` command line interface will be called to submit the data to the portal search server. The value for this parameter is in seconds.
`iim_arch.portal.deployuri`	/portal	This parameter contains the deploy URI of the Portal Server.
`iim_arch.portal.channelname`	`IMChannel`	This parameter contains the name of the Instant Messaging Channel.

Table 5-2 Optional Archive Provider parameters which can be added to the `iim.conf` file.

Parameter	Default Value	Description
`iim_arch.backup_file`		This parameter contains the name of the file which will contain the archived messages in the SOIF format. This file is cleared when the submission of data to the Portal Server Search database is successful. If the Instant Messaging server crashes or fails to submit the data to the Portal Server Search then the data from this file will be used to submit the data.
`iim_arch.submit_cgi`		This parameter contains the URL of the CGI script which will be used by the submit servlet to submit the data to the Portal Server Search database.

Managing Archived Data in the Portal Server Search Database

NOTE These instructions are Solaris specific as Sun ONE Portal Server is supported on Solaris only.

The Instant Messaging data is archived in the form of Resource Descriptors (RDs) in the Portal Server Search database. The individual entries in the Portal Server Search database are called resource descriptors (RDs). An RD is a specific set of information about a single resource. The fields of each RD are determined by the Portal Server Search database schema.

To manage the archived data, you need to manage the Resource Descriptors (RDs) in the Portal Server Search database. This section explains some of the frequently performed maintenance tasks on the Portal Server Search database.

For more information on managing data in the Portal Server Search database, see the *Sun ONE Portal Server 6.0 Administrator's Guide* at:

http://docs.sun.com/source/816-6359-10/

rdmgr Command

The rdmgr command is the main command used to work with the Search service. It gives the administrator two types of subcommands: one that is used to work with resource descriptors (RDs); and the other that is used for database maintenance. The rdmgr command is normally run in a search-enabled Portal Server instance directory.

To invoke the rdmgr command:

1. Change to the following directory:

   $cd /var/opt/SUNWps/https-servername/

2. Type the following in the command-line:

   run-cs-cli *$portalbasedir/*SUNWps/bin/rdmgr args

   Where *$portalbasedir* is the directory in which Portal Server is installed.

For more information on rdmgr command, see Command-Line Utilities in *Sun ONE Portal Server 6.0 Administrator's Guide* at:

http://docs.sun.com/source/816-6359-10/cliadm.html#19141

Searching Resource Descriptors (RD)

Running rdmgr command with the argument value -Q generates a list of RDs that refines the search operation.

For example:

- To search for resource descriptors (RDs) containing the text testing, type:

  run-cs-cli $portalbasedir/SUNWps/bin/rdmgr -Q testing

- To search for resource descriptors (RDs) belonging to a particular category, type:

```
run-cs-cli $portalbasedir/SUNWps/bin/rdmgr -Q
"classification=Archive:Chat:January"
```

Deleting Resource Descriptors

The following are the examples for deleting resource descriptors (RDs) from the Portal Server Search database:

To delete all resource descriptors (RD) containing the text `testing`, type:

```
run-cs-cli $portalbasedir/SUNWps/bin/rdmgr -d -Q testing
```

To delete all resource descriptors (RD) from a category `Archive:Chat:January`, type:

```
run-cs-cli $portalbasedir/SUNWps/bin/rdmgr -d -Q
"classification=Archive:Chat:January"
```

Enabling Instant Messenger Archive Control

The Instant Messenger Archive Control component enables the end user to control the archived instant messages. This component allows the user to search for the archived instant messages stored in the Portal Server Search database by clicking the Archive button in the Instant Messenger main window. It also enables the user to set default archive options, such as “archive all conferences” in the Archive tab of the Instant Messenger. The Instant Messenger Archive Control feature is provided by two optional Instant Messenger modules.

The Instant Messager Archive Control component can be enabled by setting the `archive_control` applet parameter in the applet descriptor file.

The applet descriptor files for the Instant Messaging standalone deployment that need to be changed are:

- `im.jnlp, imssl.jnlp and jnlpLaunch.jsp (portal only)` for Java Web Start
- `im.html, imssl.html and pluginLaunch.jsp (portal only)` for Java Plugin

Changes for JNLP files and jnlpLaunch.jsp files:

If you are using Java Web Start to launch the Instant Messenger, perform the following steps to enable the Instant Messenger Archive Control feature in the Instant Messenger:

1. Go to the messenger documentation root directory to locate the `im.jnlp` and `imssl.jnlp` files

   The `jnlpLaunch.jsp` file can be found at:

   `/etc/opt/SUNWps/desktop/default/IMProvider`

2. Edit the `jnlp or jsp` file, and add or edit the following line:

```
<argument>archive_control=true</argument>
```

Changes for html applet pages and pluginLaunch.jsp files:

If you are using Java Plug-in to launch the Instant Messenger, perform the following steps to enable the Instant Messenger Archive Control feature in the Instant Messenger:

1. Go to the messenger documentation root directory to locate the `im.jnlp` and `imssl.jnlp` files

   The `jnlpLaunch.jsp` file can be found at:

   `/etc/opt/SUNWps/desktop/default/IMProvider`

2. Edit the `jnlp or jsp` file, and add or edit the following line:

```
<PARAM NAME="archive_control" VALUE="true" />
<EMBED archive_control=true;/>
```

NOTE Instant Messenger Archive Control should not be enabled if the value of `iim_server.msg_archive.auto` is set to true in the `iim.conf` file of the Instant Messaging Server, as users messenger settings will not have any effect.

Changing the Display of the Archived Data

The data that is archived is deployed using the `IMArchiveDisplay.jsp` file. The `IMArchiveDisplay.jsp` file is installed in the folder `/etc/opt/SUNWps/desktop/default/IMProvider` by default. The file can be modified to change the style and the resource strings of the archived data.

For example, to replace the default system message displayed when a user joins the room “joe has joined the room” to “joe has entered the room;” perform the following:

1. Edit the `IMArchiveDisplay.jsp` file. For example:

   ```
   vi IMArchiveDisplay.jsp
   ```

2. Replace the code line in Code Example 5-1 with Code Example 5-2 in the file `IMArchiveDisplay.jsp`:

Code Example 5-1 Modifying the default system message.

```
....
ht.put("has_joined_the_room","<span class='user'> {0} </span>
<span class='headervalue'> has joined the room.</span>");
....
```

Code Example 5-2 After replacing the default system message.

```
....
ht.put("has_joined_the_room","<span class='user'> {0} </span>
<span class='headervalue'> has entered the room.</span>");
....
```

Similarly, the resource strings for the other keys and the style for displaying the key information can also be modified.

If the attribute name of Title and Full-Text in the default schema of the Portal Server Search is changed, then these changes should also be reflected in the `IMArchiveDisplay.jsp` file.

How to Use CGI Script to Submit the Data

The `Submit` servlet uses the `Runtime.exec()` method of Java to submit the data to the Portal Server Search database. When the Portal Server is under a high load this operation might take a long time. To avoid this delay you need to configure the Instant Messaging server to use CGI script for submitting the data to the Portal Server search database.

To configure the Instant Messaging server to use CGI script to submit the data:

1. Copy the CGI script `IMArchiveSubmit.cgi` from `/etc/opt/SUNWps/desktop/default/IMProvider/` to the appropriate directory in your Web Server instance.

2. Configure the Web Server to run the `IMArchiveSubmit.cgi` script.

   For more information on configuring the Web Server to the CGI script, see *Sun ONE Web Server, Enterprise Edition Administrator's Guide* at:

   `http://docs.sun.com/source/816-5691-10/esprgrm.htm#21236`

3. Add the following parameter `iim_arch.submit_cgi` in the `iim.conf` file.

```
iim_arch.submit_cgi = $URL_OF_THE CGI_SCRIPT
```

Where:

$URL_OF_THE CGI_SCRIPT is the URL of the CGI script.

4. Restart the Instant Messaging server.

Sample Deployment Scenario for Archive Provider

This sample deployment scenario explains how to archive the related Instant Messaging data collectively.

To archive the related Instant Messaging data collectively:

Create separate categories for each type of data. For example, in the Archive category where all the archive Instant Messaging data are stored, create a sub-category Chat for storing chat messages. You can also create subcategories for archiving data based on time. For example, to archive chat data for the month of December 2002 the subcategory will be:

`Archive:Chat:2002:12`

To archive all the chat data based on time, do the following:

1. Change to the `config` directory. For example, on Solaris type:

   `cd /etc/opt/SUNWiim/default/config`

2. Edit the `iim.conf` file. For example:

   `vi iim.conf`

3. Add the following value for the parameter `iim_arch.chat.categoryname`:

   ```
   iim_arch.chat.categoryname = Archive:Chat:%Y:%M
   ```

   The archive provider automatically assigns the current year for %Y and current month for %M. These values are taken from the system date and time.

To archive and back up the chat data the month of December 2002 to the subcategory, type:

1. `rdmgr -Q "classification=Archive:Chat:2002:12" > archive.soif`
2. Store the `.soif` file to your backup system.

To remove the archived chat data for the month of December 2002 from the Portal Server Search database, type:

```
rdmgr -d "classification=Archive:Chat:2002:12"
```

Chapter 6

Managing Instant Messaging Portal Channel

This chapter explains how to manage and customize the Instant Messaging channel on the Portal Server Desktop.

The chapter contains the following sections:

- Introducing Instant Messaging Channel
- Display Profile Properties for Instant Messaging Portal Channel
- Instant Messaging Portal Channel Tag Library Reference
- Customizing Instant Messaging Portal Channel
- Inserting Instant Messenger Links in an Organization
- Enabling Secure Mode for Sun ONE Instant Messenger in Sun ONE Portal Server
- Disallowing Users from Launching Instant Messenger

Introducing Instant Messaging Channel

The Instant Messaging channel displays the user's personalized Contact List and the presence status of these contacts on the Portal Server Desktop. The user can initiate instant messaging sessions with the contacts listed in the Instant Messaging channel.

The Instant Messaging channel is based on a Portal Server content provider called `IMProvider`. The `IMProvider` is an extension of the `JSPProvider` in the Portal Server. As an extension of the `JSPProvider`, `IMProvider` uses the JSP files to generate the content page and the edit page for the Instant Messaging channel. The

JSP files are also used to generate the pages used to launch the Instant Messenger. The `IMProvider` also defines an instant messaging-specific tag library and this tag library is used by the JSP files. The JSP files and the tag library use the channel properties that are defined by the `IMProvider`.

The sections below describe the channel properties, the tag library. It also describes how to customize the JSP files that are used by the `IMProvider`.

Display Profile Properties for Instant Messaging Portal Channel

The Instant Messaging channel uses `IMProvider` as its content provider. The `IMProvider` display profile element is installed in the global desktop service display profile. This display profile defines the properties that are used by the `IMProvider`. The display profile includes properties that are required by `JSPProvider` as well as properties that are specific to the `IMProvider`.

Table 6-1 contains the `IMProvider` specific Display Profile Properties for the Instant Messaging Portal channel

Table 6-1 `IMProvider` specific Display Profile Properties for the Instant Messaging Channel

Property Name	Description	Value
`server`	The host name of the Instant Messaging Server used by the Instant Messaging Portal channel	
`port`	The port of the Instant Messaging Server used by the Instant Messaging Portal channel	`49999`
mux	The hostname of the Instant Messaging multiplexor to be used when the Instant Messenger is launched by the channel.	
muxport	The port of the IM multiplexor to be used when the Instant Messenger is launched by the channel.	49909
`netletRule`	The name of the Netlet rule that is used with the Instant Messenger in secure mode through the SRA gateway.	`49917`
`codebase`	The URL prefix from where the Instant Messenger files are downloaded.	
`clientRunMode`	The method for invoking Instant Messenger. Instant Messenger can be invoked as a plug-in or JNLP.	`plugin`

authMethod	The authentication method used by the Instant Messaging server. The method used for authenticating can be `idsvr` (Identity Server authentication) or `ldap` (LDAP server authentication.)	`idsvr`
authUsernameAttr	The name of the attribute used to authenticate the user name when using the `idsvr` authentication method.	`uid`
username	The username to use when authenticating using the `ldap` authentication method.	*userID*
password	The password to use when authenticating using the `ldap` authentication method.	*Password*
contactGroup	The name of the contact group that is displayed in the channel.	`My Contacts`

Code Example 6-1 shows a channel that is created using the `IMProvider`. This channel customizes several Display Profile properties. If the sample portal is installed, a channel called `IMChannel` is automatically created in the default organization. The `IMChannel` does not customize any of the Display Profile properties.

Code Example 6-1 Display Profile Definition for the Instant Messaging Channel

```
<Channel name="SampleIM" provider="IMProvider">
    <Properties>
        <String name="server" value="budgie.siroe.com"/>
        <String name="port" value="49909"/>
        <String name="netletPort" value="49917"/>
        <String name="codebase" value="http://budgie.siroe.com"/>
    </Properties>
</Channel>
```

Instant Messaging Portal Channel Tag Library Reference

This section describes the tag library and provides a reference of the available tags for the Instant Messaging channel. The tag library is defined in a file `im.tld` in the directory `/etc/opt/SUNWps/desktop/default/tld`

Table 6-2 contains the tag libraries used by the file `IMContent.jsp`

Table 6-2 Tag Library Reference

Tag Name	Description	Attribute/Description
`getContactGroups`	Returns the list of contact groups that the user has defined.	`id` (optional) `scope` (optional)
`getContactGroup`	Returns the list of contacts in the named contact group. The list is returned as a Collection of internal objects that can be passed to the `obtainContact` tag.	`group` (required) The name of the contact group. `id` (optional) `scope` (optional)
`getUsername`	Returns the Instant Messaging user name for the contact.	`id` (optional) `scope` (optional)
`getToken`	Returns the login token for the user. The login token is either an Identity Server `SSOToken` or the password of the user	`id` - (optional) `scope` - (optional)
`obtainContact`	A context setup tag that is used to obtain the presence information for the indicated contact. The other tags can be used inside this tag.	`contact` (required) The internal identifier of the contact obtained from the `getContactGroup` tag.
`getContactPresence`	Returns the current presence status for the contact. The value can be `AWAY, BUSY, CLOSED, FORWARDED, IDLE, OPEN, OTHER`. These values are obtained from the `PresenceSession` class in the Instant Messaging API.	`id` (optional) `scope` (optional)
`getContactName`	Returns the common name for the contact.	`id` (optional) `scope` (optional)
`getContactUsername`	Returns the IM user name for the contact.	`id` - (optional) `scope` - (optiona
`getDateTime`	Returns the update time.	`format` (required) `id` (optional) `scope` (optional)
`isSecureMode`	Returns boolean value indicating whether the channel is accessed through the SRA gateway and if the Netlet is loaded.	`id` (optional) `scope` (optional)

Table 6-2 Tag Library Reference

Tag Name	Description	Attribute/Description
`getCodebase`	Returns the codebase used to download the Instant Messenger applet. This tag takes into account whether the Instant Messaging channel is accessed through the SRA gateway. If the channel is accessed through SRA gateway the Netlet is loaded.	`id` - (optional) `scope` - (optional)
`getIMServer`	Returns the “-server” argument to be passed to the Instant Messenger. This tag takes into account whether the Instant Messaging channel is accessed through the SRA gateway. If the channel is accessed through SRA gateway the Netlet is loaded.	`id` (optional) `scope` (optional)
`debug`	When the debug level of Identity Server is at message level then the messages are added to the log files.	id - (optional) `message` - (required) This message which will be printed in the log file. `logFile` - (optional)
`substitute`	Replaces the locale specific contents of the archive data based on the user’s locale.	`id` - (optional) `data` - (required) The data that will be substituted based on the resource table. `ht` - (optional) `locale` - (optional)

Code Example 6-2 shows an example of how the tags are defined in a JSP file. The library's symbol (`%@ taglib`) is used with the tag name to retrieve the appropriate information or function.

Code Example 6-2 An example of how the tags are defined in a JSP file.

```
<%@ taglib uri="/tld/im.tld" prefix="im" %>
     <%@ taglib uri="/tld/jx.tld" prefix="jx" %>
<im:getContactGroup id="contacts" group="My Contacts"/>
     <jx:forEach var="contact" items="$contacts">
        <im:obtainContact contact="$contact">
<br>
         <im:getPresence id="p">
<jx:choose>
             <jx:when test="$p=='online'">
                 <img src="im_online.gif">
             </jx:when>
             <jx:when test="$p=='idle'">
                 <img src="im_idle.gif">
             </jx:when>
         </jx:choose>
         <im:getName>
        </im:obtainContact>
</jx:forEach/>
```

Customizing Instant Messaging Portal Channel

While designed to work right out of the box, the Instant Messaging Portal channel is also customizable. It consists of JSP files that can be customized to an organization's specific needs. These JSP files are installed in the `/etc/opt/SUNWps/desktop/default/IMProvider` directory.

Table 6-3 contains the JSP files that can be modified to customize the Instant Messaging channel's user interface look and feel.

Table 6-3 The JSP files that can be modified to customize the Instant Messaging channel .

File Name	Description
`IMContent.jsp`	This file can be modified to customize the Instant Messaging channel content page.
`IMEdit.jsp`	This file can be modified to customize the Edit page in the Instant Messaging channel.
`pluginLaunch.jsp`	This file can be modified to customize the Instant Messenger launch window.

Table 6-3 The JSP files that can be modified to customize the Instant Messaging channel

File Name	Description
`jnlpLaunch.jsp`	This file can be modified to customize the Java Web Start window that is used to launch the Instant Messenger.
`invite.jsp`	This file can be modified to customize the Conference Invite window in the Instant Messenger.

Perform the following, if you have the Instant Messaging channel provider referred as `IMProvider` and you want to customize the `IMContent.jsp` file for this channel:

1. Create a directory `IMChannel` under `/etc/opt/SUNWps/desktop/default/`. Type:

   ```
   mkdir /etc/opt/SUNWps/desktop/default/IMChannel
   ```

2. Copy the file `IMContent.jsp` from the `IMProvider` directory to the `IMChannel` directory you have created.

```
cp /etc/opt/SUNWps/desktop/default/IMProvider/IMContent.jsp
/etc/opt/SUNWps/desktop/default/IMChannel
```

CAUTION Do not modify the original version of the JSP or template files in the Instant Messaging channel. Create a new version of the JSP or the template files to maintain the original version of these files. The Instant Messaging channel will automatically use the new version of the JSP or the template files.

3. Modify the `IMContent.jsp` file according to your requirement. For example:

   ```
   vi /etc/opt/SUNWps/desktop/default/IMChannel/IMContent.jsp
   ```

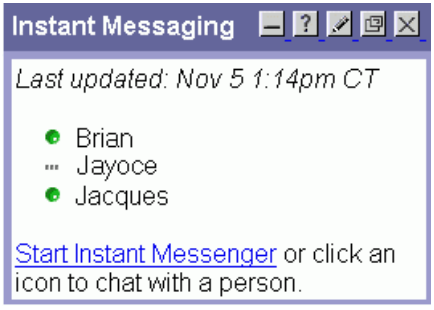
4. Save the changes made to the file.

Customizing Instant Messaging Channel Content Page

`IMContent.jsp` is the Instant Messaging Portal channel content page. It contains the list of contacts and the presence status of the contacts. Instant Messenger can be lauched by clicking the link “Start Instant Messenger” or by clicking the presence status of the contacts.

Figure 6-1 shows the Instant Messaging channel content page example.

Figure 6-1 The Instant Messaging Channel Content Page Example



Code Example 6-3 contains the code example for generating the contact list in the above Instant Messaging Channel content page.

Code Example 6-3 The Instant Messaging Channel content page example.

```
<%@ page import="java.util.*" %>
<%@ taglib uri="/tld/im.tld" prefix="im" %>
<%@ taglib uri="/tld/jx.tld" prefix="jx" %>
<%@ taglib uri="/tld/desktop.tld" prefix="dt" %>
<%@ taglib uri="/tld/desktopProviderContext.tld" prefix="dtpc" %>
<%
    if (request.getAttribute("invite") != null ||
    request.getParameter("invite") != null) { %> <jsp:forward
page="invite.jsp"/> <% }
    String launchType = request.getParameter("launch");
    if (launchType != null) {
    if (launchType.equals("jnlp")) { %> <jsp:forward
page="jnlpLaunch.jsp"/> <% }
    if (launchType.equals("plugin")) { %> <jsp:forward
page="pluginLaunch.jsp"/> <% }
    }
    else {
%>
<dt:obtainChannel channel="$JSPProvider">
function launch_<dt:getName/>(user) {
  var launchURL =
"IMLaunch?provider=<dt:getName/>&launch=<dtpc:getStringProperty
key='clientRunMode'/>&msg=Welcome&last=false";
  if (user.length > 0) {
    launchURL = launchURL + "&username=" + user;
  }
  var myURL="<%= HttpUtils.getRequestURL(request) %>";
  var urlprefix = window.location.protocol + "//" +
window.location.hostname;
  if (urlprefix != myURL.substr(0, urlprefix.length) {
    launchURL = launchURL + "&secure=yes";
  }
  window.open(launchURL, '_blank',
'width=320,height=200,hotkeys=no,status=no,resizable=no,scrollba
rs=yes');
  return false;
}
  <jx:declare id="fontFace" type="java.lang.String"/>
  <dtpc:getStringProperty id="fontFace" key="fontFace1"/>
  <font face="<%=fontFace%>">
  <font size=-1><i>Last updated: <im:getDateTime
format="{0,date,long} {0,time,short}"/></i></font>
  <p>
```

```
<jx:set var="nogroups" value="true"/>
  <dtpc:getStringProperty id="groups" key="contactGroup"/>
  <jx:if test="$groups == ''">
    <im:getContactGroups id="groups"/>
  </jx:if>
  <jx:forEach var="cg" items="$groups">
    <jx:set var="nogroups" value="false"/>
    <img src="im/images/folder_col.gif" border=0 align=absmiddle>
<b><jx:expr value="$cg"/></b><br>
    <im:getContactGroup id="contacts" group="$cg"/>
      <jx:set var="noneonline" value="true"/>
      <jx:forEach var="contact" items="$contacts">
         <im:obtainContact contact="$contact">
           <im:getContactPresence id="p"/>
           <jx:declare id="p" type="java.lang.String"/>
       <jx:set var="anchortag">
<a onClick="launch_<dt:getName>('<im:getContactUsername/>')">
       </jx:set>
           <jx:choose>
              <jx:when test="$p=='OPEN'">
                  <jx:expr value="$anchortag"/>
      <img src="im/images/Online_13Padded.gif" alt="Open"
border=0 align=absmiddle></a>
              <im:getContactName/></br>
<jx:set var="noneonline" value="false"/>
              </jx:when>
              <jx:when test="$p=='ONLINE'">
                  <jx:expr value="$anchortag"/>
      <img src="im/images/Online_13Padded.gif" alt="Online"
border=0 align=absmiddle></a>
              <im:getContactName/></br>
                  <jx:set var="noneonline" value="false"/>
</jx:when>
              <jx:when test="$p=='IDLE'">
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Idle_13Padded.gif"
alt="Idle" border=0 align=absmiddle></a>
              <im:getContactName/></br>
                  <jx:set var="noneonline" value="false"/>
              </jx:when>

              <jx:when test="$p=='BUSY'">
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Online_13Padded.gif"
alt="Busy" border=0 align=absmiddle></a>
```

```
              <im:getContactName/></br>
                  <jx:set var="noneonline" value="false"/>
              </jx:when>
<%-- Offline States - Uncomment this if you want users to see
offline contacts --
              <jx:when test="$p=='CLOSED'">
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Offline_13Padded.gif"
alt="Offline" border=0 align=absmiddle></a>
              <im:getContactName/></br>
              </jx:when>

              <jx:when test="$p=='AWAY'">
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Away_13Padded.gif"
alt="Away" border=0 align=absmiddle></a>
              <im:getContactName/></br>
              </jx:when>
              <jx:when test="$p=='FORWARDED'">
<jx:expr value="$anchortag"/>
                  <img src="im/images/Forwarded_13Padded.gif"
alt="Offline" border=0 align=absmiddle></a>
              <im:getContactName/></br>
              </jx:when>
              <jx:when test="$p=='OTHER'">
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Offline_13Padded.gif"
alt="Other" border=0 align=absmiddle></a>
              <im:getContactName/></br>
              </jx:when>
            <jx:otherwise>
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Offline_13Padded.gif"
alt="<%=p%>" border=0 align=absmiddle></a>
              <im:getContactName/></br>
</jx:choose>
        </im:obtainContact>
      </jx:forEach>
      <jx:if test="$noneonline == 'true'">
      <i>No on-line
         </jx:otherwise>
--  End of offline states --%>
contacts.</i><br>
      </jx:if>
  </jx:forEach>
<jx:if test="$nogroups == 'true'">
```

```
<p>
<font color="red">Warning</font>: Unable to get contact
information from Instant Messaging server.
</jx:if>
<p>
Start the <a href="#" onClick="launch_<dt:getName>('')">
Instant Messenger</a> or click an icon to chat with a person.
</dt:obtainChannel>
<% } %>
```

Displaying Presence Information For All Contacts

By default, the `IMContent.jsp` file only displays presence information for on-line contacts. The `IMContent.jsp` file can be modified to display the presence information for all contacts in the selected contact group(s).

To display the presence information for all contact in the selected contact group:

1. Create a directory `IMChannel` under `/etc/opt/SUNWps/desktop/default/`. Type:

   ```
   mkdir /etc/opt/SUNWps/desktop/default/IMChannel
   ```

2. Change to the `IMChannel` directory. Type:

   ```
   cd /etc/opt/SUNWps/desktop/default/IMChannel
   ```

3. Copy the file `IMContent.jsp` from the `IMProvider` directory to the `IMChannel` directory you have created.

```
cp /etc/opt/SUNWps/desktop/default/IMProvider/IMContent.jsp .
```

CAUTION Do not modify the original version of the JSP or template files in the Instant Messaging channel. Create a new version the JSP or the template files to maintain the original version of these files. The Instant Messaging channel will automatically use the new version of the JSP or the template files.

4. Modify the `IMContent.jsp` file according to your requirement. For example:

   `vi IMContent.jsp`

5. Uncomment the following Java Script in the `IMContent.jsp` file.

```
<%-- Offline States - Uncomment this if you want users to see
offline contacts --
              <jx:when test="$p=='CLOSED'">
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Offline_13Padded.gif"
alt="Offline" border=0 align=absmiddle></a>
                  <im:getContactName/></br>
              </jx:when>
              <jx:when test="$p=='AWAY'">
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Away_13Padded.gif"
alt="Away" border=0 align=absmiddle></a>
                  <im:getContactName/></br>
              </jx:when>
              <jx:when test="$p=='FORWARDED'">
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Forwarded_13Padded.gif"
alt="Offline" border=0 align=absmiddle></a>
                  <im:getContactName/></br>
              </jx:when>
              <jx:when test="$p=='OTHER'">
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Offline_13Padded.gif"
alt="Other" border=0 align=absmiddle></a>
                  <im:getContactName/></br>
              </jx:when>
              <jx:otherwise>
                  <jx:expr value="$anchortag"/>
                  <img src="im/images/Offline_13Padded.gif"
alt="<%=p%>" border=0 align=absmiddle></a>
                  <im:getContactName/></br>
              </jx:otherwise>
--  End of offline states --%>
```

6. Save the changes.

Single Instant Messaging Server Deployment Configuration

When your site contains a single Instant Messaging server, the users need not edit the server information in the channel's Edit page. You can edit `IMEdit.jsp` to only display a part of the Edit page. The change made to the Edit page depends on the type of authentication used by the Instant Messaging channel.

If you are using Identity Server for user authentication, the entire Instant Messaging Server Information section can be removed from the Edit page. If you are using LDAP authentication, the Instant Messaging Server Information section can contain only the Username and Password fields.

Automatically Closing the Invite Window

The `invite.jsp` file generates the contents for the Invite window that is created when a user is invited to a conference in the Instant Messenger. This window is opened before the server checks whether the Instant Messenger is running. By modifying the `invite.jsp` the Invite window can be closed automatically.

To automatically close the Invite window:

1. Create a directory `IMChannel` under `/etc/opt/SUNWps/desktop/default/`. Type:

   ```
   mkdir /etc/opt/SUNWps/desktop/default/IMChannel
   ```

2. Change to the `IMChannel` directory. Type:

   ```
   cd /etc/opt/SUNWps/desktop/default/IMChannel
   ```

3. Copy the file `invite.jsp` from the `IMProvider` directory to the `IMChannel` directory you have created.

```
cp /etc/opt/SUNWps/desktop/default/IMProvider/invite.jsp .
```

CAUTION Do not modify the original version of the JSP or template files in the Instant Messaging channel. Create a new version the JSP or the template files to maintain the original version of these files. The Instant Messaging channel will automatically use the new version of the JSP or the template files.

4. Open the `invite.jsp` file. For example:

   ```
   vi invite.jsp
   ```

5. Uncomment the following Java Script in the `invite.jsp` file to close the Invite window automatically.

```
<%-- uncomment this script if you want the window to close
automatically
<script>window.close();</script>
```

6. Save the changes.

Inserting Instant Messenger Links in an Organization

By default the Instant Messenger links are added to the Application channel in the default organization. The Instant Messenger links allows the Instant Messenger to be launched from the Application channel. You need to add Instant Messenger links manually, if:

- You want to add these links for another organization
- You do not have the sample portal installed

- You are using the `AppList` provider for another channel.

The contents for the Instant Messenger links are in the file `/opt/SUNWps/samples/desktop/dp-IMChannel.xml`. The `dp-IMChannel.xml` file also contains the sample `IMChannel`.

Edit a copy of the file `dp-IMChannel.xml` to add the Instant Messenger links information to the display profile for another organization and install the file using the `dpadmin` command:

1. Change to the following directory:

   $PORTAL_BASEDIR`/SUNWps/bin/`

2. Create a copy of the file `dp-org-im.xml`.

   ```
   cp dp-org-im.xml newfile.xml
   ```

3. To modify the Application channel, type the following `dpadmin` command:

```
dpadmin modify -u ADMIN_DN -w PASSPHRASE -d ORG_DN -m
newfile.xml
```

Where,

ADMIN_DN - Replace with LDAP administrator DN. For example: `amadmin`

PASSPHRASE - Replace with the administrator's password.

ORG_DN - Replace with the DN of the Organization where the links are to be added. For example: `o=siroe.com, o=isp`

The URL for launching the Instant Messenger using Java Plug-in will be a reference to the Instant Messaging channel with a launch argument. For example:

```
/portal/dt?action=content&provider=IMChannel&launch=plugin&usern
ame=sam
```

The URL for launching the Instant Messenger applet with Java Web Start will be:

```
/portal/imlaunch?channel=IMChannel&launch=jnlp&username=sam
```

Enabling Secure Mode for Sun ONE Instant Messenger in Sun ONE Portal Server

Netlet facilitates secure communication between the Instant Messenger and the server.

NOTE The Instant Messaging channel automatically uses the secured mode when accessed through the Secure Remote Access gateway. The Instant Messaging channel does not use the secured mode when it is not accessed through the gateway.

To enable the secure mode, you need to add the Netlet Rule.

To add the Netlet Rule:

1. Login to Sun ONE Directory Server Access Management Edition as Administrator
2. Click the default organization link. Example: `siroe.com`.
3. Select Services from the Show drop down list
4. Scroll down to SRA Configuration and select Netlet.
5. Click the arrow icon beside Netlet. The Netlet Rules are displayed in the right panel.
6. Click Add under Netlet Rules.
7. Type `IM` in the Rule Name field.

NOTE The Netlet rule name can be different. You can configure the Instant Messaging channel to use a different Netlet rule.

8. Remove the default value in the URL field and leave the field blank.

9. Uncheck Download Applet check box, if the Instant Messaging resources are installed on the Portal Server host. Otherwise if the Instant Messaging resources and Portal Server are installed on different hosts, Select the Download Applet check box and enter the following string:

   ```
   $IM_DOWNLOAD_PORT:$IM_WEBSERVER_HOST:$IM_WEBSERVER_PORT
   ```

   For example:

   ```
   49916:company22.siroe.com:80
   ```

   where,

   IM_DOWNLOAD_PORT. The port on which Instant Messaging resources are downloaded using Netlet.

   IM_WEBSERVER_HOST. The host name of the web server serving the Instant Messenger. For example. `company22.siroe.com`

   IM_WEBSERVER_PORT. The port number of the web server serving the Instant Messenger. For example, `80`.

10. Select the default value in the Port-Host-Port List and click Remove.
11. Enter the localhost port on which Netlet will run in the Client Port field. For example: `49916`.
12. Enter the Instant Messenger host name in the Target Host(s) field.
13. Enter the Instant Messenger port in the Target Port(s) field.

NOTE	The values for Netlet Port, Instant Messaging Host and Instant Messaging Port should be the same as the Instant Messaging service attributes mentioned in the Instant Messaging service panel.

14. Click Add to List.
15. Click Save to save the Netlet Rule.

Disallowing Users from Launching Instant Messenger

You can remove the ability for users to use the Instant Messaging channel by removing the channel from the user's display profile. For example, to remove the sample IMChannel that is automatically installed, do the following:

1. Login to Sun ONE Directory Server Access Management Edition as Administrator.
2. Click the organization link. Example: `siroe.com`.
3. Select Services from the Show drop down list
4. Click the arrow icon next to the Desktop service.
5. Click the Container and Channel Management Link.
6. Select the check box to the left of the IMChannel channel.
7. Click Delete to delete the channel.

Appendix A

Instant Messaging Configuration Parameters

This chapter explains the Instant Messaging configuration parameters.

This chapter contains the following sections:

- Using the iim.conf file
- General Configuration Parameters
- User Source Configuration Parameters
- Logging Configuration Parameters
- Instant Messaging Server Configuration Parameters
- Multiple Server Configuration Parameters
- Multiplexor Configuration Parameters

Using the iim.conf file

Instant Messaging stores configuration settings in the following file:

- On Solaris:

  `/etc/opt/SUNWiim/config/iim.conf`

- On Windows:

  iim_install_dir`\config\iim.conf`

This file is a plain ASCII text file, with each line defining a server parameter and its value(s):

- A parameter and its value(s) are separated by an equal sign (=) with spaces and tabs allowed before or after the equal sign.
- A value can be enclosed in double quotes (" "). If a parameter allows multiple values, the entire value string must be enclosed in double quotes.
- A comment line must have an exclamation point (!) as the first character of the line. Comment lines are for informational purposes and are ignored by the server.
- If a parameter appears more than once, the value of the last parameter listed overrides the previous value.
- A backslash (\) is used for continuation and indicates the value(s) are longer than one line.
- Each line is terminated by a line terminator (\n, \r, or \r\n).
- The key consists of all the characters in the line starting with the first non-whitespace character and up to the first ASCII equal sign (=) or semi-colon (;). If the key is terminated by a semi-colon, it is followed by "lang-" and a tag that indicates the language in which this value is to be interpreted. The language tag is followed by an equal sign (=). All whitespace characters before and after the equal sign are ignored. All remaining characters on the line become part of the associated value string.
- Multiple values in the value string are separated using commas (,).
- Within a value, if any special characters like comma, space, newline, tab, double quotes, or backslash are present, the entire value needs to be within double quotes. In addition, every carriage return, line feed, tab, backslash, and double quotes within the value must specified with a backslash (\).
- If you make changes to the `iim.conf` file, you must refresh the Sun ONE Instant Messaging Server in order for the new configuration settings to take effect.

NOTE The `iim.conf` file is initialized by the installation process and should be modified only as described in this guide.

General Configuration Parameters

Table A-1 describes the general configuration parameters.

Table A-1 General Configuration Parameters

Parameter	Default Value	Description
`iim.comm.modules`	`iim_server,iim_mux`	The communication modules used. The values are `iim_server` and `iim_mux`. The default value is `iim_server, iim_mux`, which means both the server and multiplexor are used. The `iim_mux` value is useful for multiplexor.
`iim.smtpserver`	`localhost`	SMTP server to send mail to users who have set the option for forwarding their messages as emails or to pagers.

`iim.instancedir`	On Unix:
	`/opt`
	On Windows:
	`c:\Program Files\Sun\InstantMessaging`
		The installation directory root.

`iim.instancevardir`	On Solaris:
	`/var/opt/SUNWiim/default`
	On Linux:
	`/var/opt/soim/default`
	On Windows:
	im_install_dir`\`
		Sets the directory to contain runtime files, including the user profile database, logs, and other files created by the server and multiplexor at runtime.

`iim.user`	`inetuser` for Standalone deployment. `root` for portal deployment.	The user name using which the server processes run. Used only on Unix platforms.
`iim.group`	`inetgroup` for Standalone deployment. `root` for portal deployment.	The group using which the server processes run. Used only on Solaris platform.

Table A-1 General Configuration Parameters

Parameter	Default Value	Description
iim.jvm.maxmemorysize	256	The maximum number heap size in MB the JVM running the server is allowed to use. Used to construct the -mx argument of the Java command.
iim.mail.charset	None	This parameter specifies if the headers of the mail are in ascii and not encoded. It contains the name of the charset to be used to encode the headers of the mail message sent out for offline alerts. For example: iim.mail.charset=iso-2022-jp
iim.jvm.command	/usr/j2se/bin/java	The location of the Java Runtime Executable (JRE).
iim.identity.basedir	/opt	The base directory of the Sun ONE Identity Server installation.
iim.identity.jre	/usr/java_1.3.1_04	The location of the JRE used by the Identity Server to run all it's processes.
iim.portal.deployuri	/portal	The URI using which the Portal Server war files are deployed in the Identity Server.
iim.portal.host	imhostname	The host name of the server on which the Portal Server is running. Specify the port number if a non default port number is used.
iim.portal.protocol	http	The protocol used to access the Portal Server.
iim.policy.resynctime	720	The Instant Messaging server clears all cached user information on a regular basis in order to eliminate old user information. This parameter specifies the frequency, in minutes, at which the cached user information is cleared.

User Source Configuration Parameters

Table A-2 describes the user source configuration parameters.

Table A-2 User Source Configuration Parameters *(1 of 3)*

Parameter	Default Value	Description
`iim_ldap.host`	`localhost:389`	LDAP server name and port used by IM server for user authentication.
`iim_ldap.searchbase`	`o=internet`	The string used as base to search for the users and groups on the LDAP server.
`iim_ldap.usergroupbinddn`	None (the server performs anonymous searches)	Specifies the `dn` to use to bind to the LDAP server for searches.
`iim_ldap.usergroupbindcred`	None (the server performs anonymous searches)	Specifies the password to use with the `iim_ldap.usergroupbinddn` dn for LDAP searches.
`iim_ldap.loginfilter`	`(&(|(objectclass=inetorgperson)(objectclass=webtopuser))(uid={0}))`	Search filter used during user login.
`iim_ldap.usergroupbyidsearchfilter`	`(|(&(objectclass=groupofuniquenames)(uid={0}))(&(|(objectclass=inetorgperson)(objectclass=webtopuser))(uid={0})))`	The search filter used to search for users and groups in the directory, under the base specified by ID.
`iim_ldap.usergroupbynamesearchfilter`	`(|(&(objectclass=groupofuniquenames)(cn={0}))(&(|(objectclass=inetorgperson)(objectclass=webtopuser))(cn={0})))`	The search filter used to search for users and groups in the directory, under the base specified by name.

Table A-2 User Source Configuration Parameters *(2 of 3)*

Parameter	Default Value	Description
iim_ldap.allowwildcardinuid	False	Determines if wildcards should be enabled for UIDs while performing a search. As most directory installations have UIDs indexed for exact searches only, the default value is False. Setting this value to True can impact performance unless UIDs are indexed for substring search.
iim_ldap.userclass	inetOrgPerson,webtopuser	The LDAP class that indicates that an entry belongs to a user.
iim_ldap.groupclass	groupOfUniqueNames	The LDAP class that indicates that an entry belongs to a group.
iim_ldap.groupbrowsefilter	(objectclass=groupofuniqu enames)	The search filter used to browse all groups in the directory, under the specified search base.
iim_ldap.searchlimit	40	Maximum number of entries to be returned by a search. A value of -1 means search is disabled on this server and a value of 0 indicates unlimited search.
iim_ldap.userdisplay	cn	LDAP attribute to use for display name of users.
iim_ldap.groupdisplay	cn	LDAP attribute to use for display name of groups.
im_ldap.useruidattr	uid	LDAP attribute used as users' UID.
im_ldap.groupmemberattr	uniquemember	LDAP attribute that gives the list of members of a group.

Table A-2 User Source Configuration Parameters *(3 of 3)*

Parameter	Default Value	Description
iim_ldap.usermailattr	mail	LDAP attribute that should contain users' provisioned email addresses. Used when the email message sent to an offline user.
iim_ldap.userattributes	None	LDAP attribute that contains the list of custom attributes from the LDAP user entry.
iim_ldap.groupattributes	None	LDAP attribute that contains the list of custom attributes from the LDAP group entry.
iim_ldap.groupmemberurlattr	None	The membership attribute of a dynamic group, which contains the LDAP filter or the LDAP URL.
iim_ldap.useidentityadmin	The default value is true, if Sun ONE Identity Server Instant Messaging Service Definition component is installed. The default value is false, if Sun ONE Identity Server Instant Messaging Service Definition component is not installed.	If the value is true then the Identity Server Administrator credentials will be used to bind to the Directory Server.

Logging Configuration Parameters

Table A-3 describes the logging configuration parameters.

Table A-3 Logging Configuration Parameters

Parameter	Default Value	Description
`iim.log.iim_server.severity`	`NOTICE`	Level of logging required for the server module. The possible values from highest to lowest are: FATAL, ERROR, NOTICE, WARNING, INFO, and DEBUG. If a lower level of logging is chosen, it is implied that you get the higher levels too. That is, if you choose WARNING you get FATAL, ERROR, NOTICE, and WARNING.
`iim.log.iim_server.url`	On Solaris: `/var/opt/SUNWiim/default/log/server.log` On Linux: `/var/opt/soim/default/log/server.log` On Windows: *im_install_dir*`\log\server.log`	Location of the server log file. This file needs to be periodically trimmed to prevent disk space from filling up.
`iim.log.iim_mux.severity`	`NOTICE`	Level of logging required for the multiplexor module. The possible values from highest to lowest are: FATAL, ERROR, NOTICE, WARNING, INFO, and DEBUG. If a lower level of logging is chosen, it is implied that you get the higher levels too. That is, if you choose WARNING you get FATAL, ERROR, NOTICE, and WARNING.

Table A-3 Logging Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim.log.iim_mux.url`	On Solaris: `/var/opt/SUNWiim/default/log/mux.log` On Linux: `/var/opt/soim/default/log/mux.log` On Windows: *im_install_dir*`\log\mux.log`	Location of the multiplexor log file. This file needs to be periodically trimmed to prevent disk space from filling up.
`iim.log.iim_server.maxlogsize`		This parameter contains the maximum size of a server log file. If the log files exceeds the size specified in this parameter then server creates a new file to log in the details.

Instant Messaging Server Configuration Parameters

Table A-4 describes the Sun ONE Instant Messaging Server configuration parameters.

Table A-4 General IM Server Configuration Parameters

Parameter	Default Value	Description
`iim_server.domainname`	*host's domain name*	The logical IM Server domain name you want this server to support. This is the name that is used by other servers in the network to identify this server. It is also the name used by this server to identify its users to other servers. This is not necessarily the Fully Qualified Domain Name of the system running the Sun ONE Instant Messaging Server. For example, if the system `iim.xyz.com` is the only Sun ONE Instant Messaging Server for a company `xyz.com`, then the domain name is likely to be `xyz.com`.
`iim_server.port`	`49919`	IP address and port for the server to bind to, and to listen for connections from other servers. IP address setting is useful for multi homed machines when you want to use only one particular IP address. If no IP address is listed, this indicates a value of `INADDR_ANY` on `localhost`.

Table A-4 General IM Server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
iim_server.useport	TRUE	Indicates whether the server should listen on the server-to-server communication port. The possible values are TRUE and FALSE. If TRUE, the server listens on the port defined by iim_server.port or on port 9919, if that is not explicitly defined.
iim_server.sslport	49910	Server's SSL port used for secure server-to-server communication. Note: The value format is IPaddress:port. If no IP address is listed, this indicates a value of INADDR_ANY on localhost.
iim_server.usesslport	FALSE	Indicates if the server should listen on the server-to-server SSL communication port. The possible values are TRUE and FALSE. If TRUE, the server listens on the port defined by iim_server.sslport or on port 9910, if that is not explicitly defined.
iim_server.sslkeystore	None	File name containing the certificate to use for SSL connection.
iim_server.sslkeystorepassphrase	None	Password of file name containing the SSL certificate.

Table A-4 General IM Server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim_server.clienttimeout`	`15`	Specifies the time, in minutes, before the server discards client connections that are no longer active. For example, when a machine is turned off. The minimum accepted value is 5.
`iim_server.usesso`	`0`	This parameter tells the server whether or not depend on the SSO provider during authentication. An SSO provider is a module which the server uses to validate a session id with a SSO service. In portal deployment, Portal Server Session API provides the IM server with the ability to validate session ids sent by the client. The value for this parameter can either be `0`, `1`,or `-1`. `0` - do not use the SSO provider (default). `1` - use the SSO provider first and default to LDAP when the SSO validation fails. `-1`- use SSO provider only without attempting LDAP authentication even when the SSO validation fails. The `iim_server.usesso` parameter is used in conjunction with the `iim_server.ssoprovider` parameter.

Table A-4 General IM Server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
iim_server.ssoprovider	None	This parameter specifies the class implementing the SSO Provider. If iim_server.usesso is not equal to 0 and this option is not set, the server uses the default Portal Server based SSO Provider.
iim_server.msg_archive	false	This parameter specifies whether the archive provider should be enabled or disabled.
iim_server.msg_archive.provider	None	This parameter contains the list of custom archive providers. This parameter allows multiple values and each value is separated by a comma(,).
iim_server.msg_archive.auto	false	This parameter tells the server whether the users' archive control settings can be considered. If the value for this parameter is true, it is equivalent to selecting archive everything option in the User Settings.
iim_server.conversion	false	This parameter specifies whether message conversion should be enabled. It specifies whether the configured list of Message Conversion Providers should be used for message conversion.

Table A-4 General IM Server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
iim_server.conversion.provider	None	This parameter contains the list of Message Conversion Providers to be used for message conversion. This parameter allows multiple values with each value is separated by a comma(,).
iim_server.servertimeout	-1	The server can be configured to automatically close the connection opened by a remote server, if the remote server is inactive. This is performed by periodically measuring the time the last request was made by the remote server to the server. The connection to the remote server is terminated, if the time of the last request made by the remote server exceeds the value of the iim_server.servertimeout parameter. The parameter value is in minutes.
iim_server.enable	true	This value should contain whether or not the Instant Messaging server should be enabled. This parameter is set false to enable the Instant Messaging multiplexor.
iim_server.conversion.external.command	None	This parameter contains the external command used for message conversion.

Table A-4 General IM Server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
iim_server.stat_frequency	1	This parameter contains the frequency at which the server logs the summary of activities to the log file. The server logs the summary of activities to the log file only if the server minimum log severity is set to NOTICE or lower. The value is in minutes.
iim_server.nodisplay	false	If the value for this parameter is true then the server will not load swing or awt classes, which require Windows Operating system. To run the Instant Messaging server on a system which does not have X Windows, set this option to true.

Multiple Server Configuration Parameters

For communication between multiple Sun ONE Instant Messaging Servers in your network, you need to configure your server to identify itself with the other servers and identify itself with each coserver, or cooperating server, which will have a connection to your server. The coserver identifies itself with its IM domain name, host and port number, serverID, and password.

Each cooperating server is given a symbolic name, which is a string consisting of letters and digits, for example, coserver1. Using the symbolic naming convention you can specify multiple servers.

When Sun ONE Instant Messaging Servers are configured in this manner, you can form a larger instant messaging community. Thus:

- Users on each server can communicate with users on every other server
- Use conferences rooms on other servers
- Subscribe to news channels on other servers (subject to access privileges)

Table A-5 describes the multiple server configuration parameters.

Table A-5 Multiple Server Configuration Parameters

Parameter	Default Value	Description
`iim_server.serverid`	None	String used by this server to identify itself to all other servers.
`iim_server.password`	None	Password used by this server to authenticate itself to all other servers.

`iim_server.coservers`	None	Comma separated list containing symbolic names of the servers that can connect to this server. Any meaningful names are allowed, but they must match what you use for the `.serverid`, `.password`, and `.host` parameters. Examples:
		`iim_server.coservers=coserver1,coserver2`
		or
		`iim_server.coservers=abc,xyz,ntc`

`iim_server.`*coserver1*`.serverid`	None	String that identifies the cooperating server represented by the name, *coserver1* to authenticate to this server. Note: If you had used `abc` in the `iim_server.coservers` list, then the corresponding name for its *serverid* would be `iim_server.abc.serverid`.
`iim_server.`*coserver1*`.password`	None	Password used by cooperating server represented by the name, *coserver1* to authenticate to this server. Note: If you had used `abc` in the `iim_server.coservers` list, then the corresponding name for its password would be `iim_server.abc.password`.
`iim_server.`*coserver1*`.host`	None	IP address and the port to connect to, for users on this server to communicate to users on the server represented by the name `coserver1`. Note: If you had used `abc` in the `iim_server.coservers` list, then the corresponding name for its host would be `iim_server.abc.host`. Note: The value format is `name:port` or `IPaddress:port`.

Table A-5 Multiple Server Configuration Parameters *(Continued)*

Parameter	Default Value	Description
`iim_server.`*coserver1*`.usessl`	`False`	Indicates if this server should use SSL to talk to the server identified by *coserver1*. The possible values are TRUE and FALSE.

Multiplexor Configuration Parameters

Table A-6 describes the multiplexor configuration parameters.

Table A-6 Multiplexor Configuration Parameters

Parameter	Default Value	Description
`iim_mux.listenport`	`49909`	IP address and listening port for the multiplexor, to listen for Sun ONE Instant Messenger. The value format is *IP_address:port*. If no IP address is listed, this indicates a value of `INADDR_ANY` on localhost. Note: If you change this value, also change the `im.html` and `im.jnlp` files so that they match the port value.
`iim_mux.serverport`	`49999`	The IM server and port the multiplexor talks to. The value format is *servername*`:`*port* or *IP_address*`:`*port*.
`iim_mux.numinstances`	`1`	Number of instances of the multiplexor. This parameter is valid only for Solaris platforms.
`iim_mux.maxthreads`	`5`	Maximum number of threads per instance of the multiplexor.
`iim_mux.maxsessions`	`2000`	Maximum number of concurrent connections per multiplexor process.
`iim_mux.usessl`	`off`	If the value is set to `on`, the multiplexor requires an SSL handshake for each connection it accepts, before exchanging any application data.

Table A-6 Multiplexor Configuration Parameters *(Continued)*

Parameter	Default Value	Description
iim_mux.secconfigdir	/etc/opt/SUNWiim/default/config	The /etc/opt/SUNWiim/default/config is the value of the iim_mux.secconfigdir parameter. This directory contains the key and certificate databases. It usually contains the security module database.
iim_mux.keydbprefix	None	This value should contain the key database filename prefix. The key database file name must always end with key3.db. If the Key database contains a prefix, for example This-Database-key3.db, then value of this parameter is This-Database.
iim_mux.certdbprefix	None	This value should contain the certificate database filename prefix. The certificate database file name must always end with cert7.db. If the certificate database contains a prefix, for example Secret-stuff-cert7.db, then value of this parameter is Secret-stuff.
iim_mux.secmodfile	secmod.db	This value should contain the name of the security module file.
iim_mux.certnickname	Server-Cert	This value should contain the name of the certificate you entered while installing the certificate. The certificate name is case-sensitive.
iim_mux.keystorepasswordfile	/etc/opt/SUNWiim/default/config/sslpassword.conf	This value should contain the relative path and the name of the file containing the password for the key database. This file should contain the following line: Internal (software) Token:*password* Where *password* is the password protecting the key database.

Table A-6 Multiplexor Configuration Parameters *(Continued)*

Parameter	Default Value	Description
iim_mux.stat_frequency	600	This value should contain the frequency at which the multiplexor logs the summary of activities to the log file.The minimum value is 10 seconds.
iim_mux.enable	true	If the value is true then the multiplexor will run for this instance. If the value is false then the multiplexor will not run for this instance.

Sun ONE Instant Messaging Reference

This chapter explains the `imadmin` command used to administer Sun ONE Instant Messaging.

imadmin

You can use the `imadmin` utility to start, stop, and refresh the Sun ONE Instant Messaging Server and Multiplexor. On the Solaris platform, run `imadmin` as `root` or the user specified during the installation.

Requirements: You must invoke the `imadmin` utility from the host on which Sun ONE Instant Messaging Server is installed.

Location:

- On Solaris: *`im_install_dir`*`/SUNWiim/sbin`
- On Windows: *im_install_dir*`\sbin`

Table B-1 The `imadmin` command with description:

Command	Description
`imadmin start`	Starts the enabled server and / or multiplexor component(s).
`imadmin stop`	Stops the enabled server and / or multiplexor component(s).
`imadmin refresh`	Refreshes the enabled server and / or multiplexor component(s).
`imadmin start server`	Starts only the server.
`imadmin stop server`	Stops only the server.
`imadmin refresh server`	Refreshes only the server.
`imadmin start multiplexor`	Starts only the multiplexor.
`imadmin stop multiplexor`	Stops only the multiplexor.

Table B-1 The `imadmin` command with description:

Command	Description
`imadmin refresh multiplexor`	Refreshes only the multiplexor.

Synopsis

`imadmin [`*options*`] [`*action*`] [`*component*`]`

Options

The options (Solaris platform only) for the `imadmin` command are:

Table B-2 Options for `imadmin` Command

Option	Description
`-c` *alt-config-file*	Used with the `start` and `refresh` actions, to specify a different configuration file other than `/etc/opt/SUNWiim/config/iim.conf` file.
`-h`	Displays help on the `imadmin` command.

Actions

The actions for the `imadmin` command are:

Table B-3 Actions for `imadmin` Command

Option	Description
`start`	Sets the `classpath`, the Java heap size and starts all the specified components.
`stop`	Stops all the specified component's daemons.
`refresh`	Stops and starts the specified component(s). Useful after a configuration change.

Components

The components for the `imadmin` command are:

Table B-4 Components for `imadmin` Command

Option	Description
`server`	Indicates the Sun ONE Instant Messaging Server.
`multiplexor`	Indicates the Sun ONE Instant Messaging Multiplexor alone.

Appendix C

Instant Messaging APIs

This chapter explains the APIs used by Sun ONE Instant Messaging.

Sun ONE Instant Messaging APIs Overview

Sun ONE Instant Messaging provides Java APIs which can be used to develop extension or integration modules. Detailed documentation of these APIs are provided with the installed Instant Messenger component, in the form of HTML files generated by Javadocs. The Javadoc files are installed in the *imdocroot*/`apidocs/` directory. To view the API documentation, point your browser to *imcodebase*/`apidocs` where the `codebase` is the Instant Messenger resources codebase.

The following are the Instant Messaging APIs:

- Instant Messaging Service API
- Messenger Beans
- Service Provider Interfaces
- Authentication Provider API

Instant Messaging Service API

The Instant Messaging API is used by the applications located on the same host or in the remote host to access Sun ONE Instant Messaging services, such as Presence, Conference, Notification, Polls and News channels.

The Instant Messaging Service API can be used for:

- A Java-based or web-based client, such as a portal channel

- A Bridge or a Gateway to enable another class of clients.
- Integration of Instant Messenger and Presence in to the existing applications
- Displaying news feeds as Sun ONE Instant Messenger news.

Messenger Beans

A Messenger bean is a dynamically loaded module used to extend the messenger functionality. Messenger beans can add action listeners, such as buttons and menu items, and item listeners, such as check boxes and toggle buttons in the existing Instant Messenger window. The item listeners are invoked when a user input is received and bean-specific actions are based on the user input. Beans have the ability to add their own settings panel and save bean-specific properties on the server. Beans can be notified of any event received by the Instant Messenger. for example, a new alert message.

The applications that use Messenger Beans are:

- Ability for users to share application and conference along with voice or video.
- Ability to retrieve and process the transcript of a conference For example, the contents of a received or sent alert, for archiving purposes.

NOTE The Sun ONE Instant Messenger Archive control functionality is provided through a Messenger Bean.

Service Provider Interfaces

The Service Provider Interface APIs provide the ability to extend the Sun ONE Instant Messaging server functionality. The Service Provider Interface is composed of the following independent APIs:

- The Archive Provider API
- The Document Converter API
- The Authentication Provider API

Archive Provider API

An Archive Provider is a software module usually providing integration with the archive or auditing system. Each configured Archive Provider is invoked for each server process.

The Archive Provider is invoked for the following server processes:

- When a instant message is sent. The Instant Messages, such as alert, poll, chat, news or conference.
- During an authentication event, such as login or logout.
- When there is a change in the presence status.
- During a subscription event. For example, when someone joins or leaves a conference, or subscribes or unsubscribes to a news channel.

The applications that use the Archive Provider API are:

- Instant Messaging Archive

NOTE The default Instant Messaging archive in Sun ONE Instant Messaging is based on the Archive Provider API. For more information on Instant Messaging Archive, see Managing The Instant Messaging Archive.

- The application that records the usage statistics for sizing purposes

Message Conversion API

A Message Converter is invoked for every message or each message part going through the server. The Message Converter may leave the message part intact or modify or remove the message part. The text parts are processed as Java String Objects. The Message Converter processes other attachment as a stream of bytes and returns a potentially different stream of bytes, or nothing at all if the attachment is to be removed.

The applications that uses Message Conversion API are:

- Virus checking and removal
- Translation engine integration
- Message content filtering

Authentication Provider API

The Authentication Provider API provides ability to deploy Sun ONE Instant Messaging in environments that are not using Sun ONE Identity Server password-based or token-based authentication service. This API is invoked whenever a user requests authentication, and it can be used in conjunction with the LDAP authentication.

The application that uses Authentication Provider API is:

- Single Sign-on (SSO) with Sun ONE Identity Server is performed using the Authentication Provider API. This API can also be used to integrate with other authentication systems.

Appendix D

Troubleshooting Instant Messaging

This chapter lists the common problems that may occur during installation and deployment of Instant Messaging. The log information generated by the various system components on their operation can be extremely useful when trying to isolate or troubleshoot a problem. For details and more information on logging, refer to the section Managing Logging. This section lists the various log files with their default locations on Solaris.

The Multiplexor and Server logs are in the files `mux.log` and `server.log` respectively, by default these files are in the directory `/var/opt/SUNWiim/default/log`. The logging level for Multiplexor and Server log files is controlled in the `iim.conf` configuration file using the properties `iim.log.iim_mux.severity` and `iim.log.iim_mux.severity`. These properties can have the following values:

- `fatal`
- `error`
- `warning`
- `notice`
- `info`
- `debug.`

Logging configuration in portal deployments is determined by the `com.iplanet.services.debug.level` property. This property can contain the following values:

- `message`
- `warning`
- `error`

- `off`

Table D-1 shows the location for the desktop and archive log files.

Table D-1 The location for the desktop and archive log files

Log File	Default Location
`desktop.debug`	`/var/opt/SUNWam/debug/`
`IMArchiveSearch.log`	`/var/opt/SUNWam/debug/`
`IMArchiveSubmit.log`	`/var/opt/SUNWam/debug/`

Logging information for the Messenger client can be obtained by enabling logging output from the Java Web Start application manager or the Java plug-in manager.

Listed below are some problems and their possible causes and the clues for troubleshooting these problems:

The Messenger client does not load or start

Connection refused and timed out

Authentication errors

IM channel display error

Instant Messaging content is not archived

Server-to-server communication fails to start

The Messenger client does not load or start

The following are the possible causes for this problem:

- Wrong codebase in the applet page.
- Application/x-java-jnlp-file MIME type not defined in the web server configuration.
- Plug-in of Java Web Start not installed or functional.
- No compatible Java version available.

Where to get the necessary information:

- In the Java Web Start or plug-in errors (exception stack trace, launch page.)
- In the applet page source on the browser.

Connection refused and timed out

The following are the possible causes for this problem:

- Either the Instant Messaging server or the Multiplexor is not running.
- Incorrect Multiplexor host or port names used in the Applet descriptor file (`.jnlp` or `.html`.)
- Different SSL settings used between the Instant Messenger and the Multiplexor.
- Client and server version mismatch.

Where to get the necessary information:

- Instant Messaging server and Multiplexor log files.

Authentication errors

The following are the possible causes for this problem:

- Problems while accessing the LDAP server.
- User not found.
- Invalid credentials.
- Invalid Identity Server session.

Where to get the necessary information:

- Instant Messaging server, Identity authentication and LDAP log files.

IM channel display error

The following are the possible causes for this problem:

- Authentication error when the server cannot validate the session token.
- Instant Messaging channel is not configured properly. For example, incorrect Instant Messaging server host and / or port.
- Plug-in or Java Web Start is not installed or is not functional.
- User not found and the Instant Messaging server cannot find the user in the LDAP lookup.

Where to get the necessary information:

- Instant Messaging server and Instant Messaging channel logs.

Instant Messaging content is not archived

The following are the possible causes for this problem:

- Content is actually archived but the user has insufficient rights to access it.
- The content has not yet been committed to the Compass database.
- The archive provider has been disabled in the Instant Messaging server.

Where to get the necessary information:

- In the Instant Messaging server and the archive log files.

Server-to-server communication fails to start

The following are the possible causes for this problem:

- Incorrect server identification.
- Mismatch in the SSL settings.

Where get the necessary information:

The necessary information can be obtained from the two Instant Messaging server log files.

Index

A

access control files
  default privileges 44
  examples 45
  format 44
activating SSL 55
administering
  conference rooms 93
  news channels 93
  using the command-line interface 32

B

backing up Sun ONE Instant Messaging Server 76

C

changing
  configuration parameters 37
  user privileges 46
chat 16
components
  LDAP directory server 20
  multiplexor 18
  SMTP server 20
  Sun ONE Instant Messaging Server 11
  Sun ONE Instant Messenger 16
  web server 19
conference rooms
  administering 93
  controlling access to 45
configuration files 27
configuration parameters
  general 143, 150
  logging 148
  multiple servers 155
  multiplexor 157
  SSL 151
  user source 145
configurations 21
configuring
  sever-to-server communications 47
  SSL 50
  SSL parameters 151
customizing `index.html` and `im.html` files 86

D

directory server 12, 20
directory structure 27

E

embedded URLs 17, 19
exporting and importing a public key certificate 54

F

forwarding alerts 20

G

generating a self-signed certificate 52
getting a self-signed certificate signed by a CA and installing it 52
granting users privilege to create conference rooms and news channels 94

I

`iim.conf` file 28, 37, 48, 61, 66, 115, 116, 117, 119, 141
`imadmin` command 35, 161
`imres.jnlp` file 87
`index.html` file 19

J

Java Web Start 79

L

LDAP directory server
  enable IM to search as a specific user 64
  requirements 12
  standalone and portal modes 20
logging
  monitoring and trimming log files 39
  overview 38
  setting levels 40
logging levels 38

M

managing
  logging 38
  user privileges 40
multiplexor 18
  listenport parameter 38, 89
  logging levels 39
  overview 18

N

news channels
  administering 93
  controlling access to 45

P

privileges 40
proxy settings 94
public key certificate
  exporting and importing 54
  `keytool` command 55

S

self-signed certificate
  generating 52
  getting it signed by a CA and installing it 52
server
  changing configuration parameters 37
  components 11
  configurations 21
  logging levels 39
server-to-server communications 47
setting log file levels 40
SMTP server 20
SSL
  activating 55

configuration parameters 151
configuring 50
starting
server and multiplexor 35
server and multiplexor (Windows only) 37
stopping
server and multiplexor 36
server and multiplexor (Windows only) 37
Sun ONE Instant Messaging Server
access control 30
administration overview 33
backing up 76
components 11
configuration file 28
logging overview 38
SSL 50
using the command-line interface to administer 32
Sun ONE Instant Messaging server
configurations 21
directory structure 27
server-to-server communications 47
Sun ONE Instant Messenger
communication modes 16
customizing 84
enabling secure mode in Sun ONE Portal Server 137
overview 16
proxy settings 94
Sun ONE Portal Server configuration
overview 14
Sun ONE Web Server Enterprise Edition 19
`sysRoomsAdd.acl` file 94
`sysTopicsAdd.acl` file 45, 94

U

URLs, embedded 17
user administration 34
user interface 126
user privileges
changing 46
creating conference rooms 94
creating news channels 94

W

web server 19, 21