# Administrator's Guide

*Sun™ ONE Instant Messaging*

**Version 6.1**

817-4113-10
December 2003

# Contents

# About This Guide

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

This preface contains the following sections:



## 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 Application Server SE (Standard 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 Sun™ ONE Instant Messaging Software”

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

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

  This chapter describes how to customize and administer the Sun ONE Instant Messenger.
- Chapter 4, “Managing Instant Messaging and Presence Policies”

  This chapter describes how to manage administrator and end user privileges, especially with policies set in the Sun ONE Identity Server.
- Chapter 5, “Managing The Instant Messaging Archive”

  This chapter explains how to manage and configure the Sun ONE Instant Messaging Archive.
- Appendix A, “Instant Messaging Configuration Parameters”

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

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

This chapter explains the APIs used by Sun ONE Instant Messaging.

- Appendix D, “Troubleshooting Instant Messaging”

  This appendix lists the common problems that might occur during installation and deployment of the Sun ONE Instant Messaging server.

- Appendix E, “Legacy Instant Messaging Service 6.0”

  This appendix describes the Instant Messaging service and lists and describes the attributes of this service, which enable administrators to enforce policy mechanisms for accessing Sun ONE Instant Messaging server.

# Document 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) 2003 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) 2003 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 `imadmin` command described as follows:

```
imadmin [options] [action] [component]
```

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

# Related Third-Party Web Site References

Third-party URLs are often referenced in Sun documentation to provide additional, related information.

**NOTE** Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources.

Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

# Accessing Related Sun Documentation Online

In addition to this guide, administrators can refer to the following related documents:

- *Sun ONE Directory Server* documentation set

  `http://docs.sun.com/db/prod/s1.s1dirs#hic`

- *Sun ONE Messaging Server* documentation set

  `http://docs.sun.com/db/prod/s1msgsrv#hic`

- *Sun ONE Calendar Server* documentation set

  `http://docs.sun.com/db/prod/s1.s1cals#hic`

- *Sun ONE Instant Messaging* server documentation set

  `http://docs.sun.com/db/prod/s1instmsg#hic`

- *Sun ONE Identity Server* documentation set

  http://docs.sun.com/db/prod/s1.s1idsrv#hic

- *Sun ONE Portal Server* documentation set

  http://docs.sun.com/db/prod/s1.s1portals#hic

- *Sun ONE Web Server* documentation set

  http://docs.sun.com/db/prod/s1websrv#hic

These documents and others can all be accessed by navigating the following Web site:

http://docs.sun.com

Chapter 1

# Introduction to Sun™ ONE Instant Messaging Software

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

The chapter contains the following sections:



## Sun ONE Instant Messaging Components

Instant Messaging server enables end users to participate in real-time interactive messaging and discussions. Sun ONE Instant Messaging allows end 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.

The components used to provide the Sun ONE Instant Messaging service to end users vary depending on the type of deployment.

# Basic Deployment Scenarios

The Sun ONE Instant Messaging server can be deployed in any one of the following scenarios:

- As a server connected only to an LDAP server.
- As a server connected to Sun ONE Identity Server.
- As a server connected to Sun ONE Identity Server and Sun ONE Portal Server. Sun™ ONE Instant Messenger is made available to end users on the Portal Server Desktop.

# Quick Reference of Core Instant Messaging Components

The core Instant Messaging components are the same, regardless of which of the preceding deployment methods you use. The Instant Messaging components are:

- **Sun ONE Instant Messenger Resources.** This is the set of files that make up the Sun ONE Instant Messenger client.
- **Sun ONE Instant Messenger.** This is a Java Instant Messaging applet. It is the Java-based Sun ONE Instant Messenger client that is invoked through the web, using Java™ Web Start or the Java™ Plug-in.
- **Sun ONE Instant Messaging Server.** The Instant Messaging server serves the presence information to the messenger clients, allows end users to establish Instant Messaging sessions, and enforces policies.
- **Instant Messaging Multiplexor.** A scalability component that consolidates multiple messenger connections into one Transmission Control Protocol (TCP) connection to the server. The Instant Messaging multiplexor is also referred to 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.

# Quick Reference of Instant Messaging Related Components

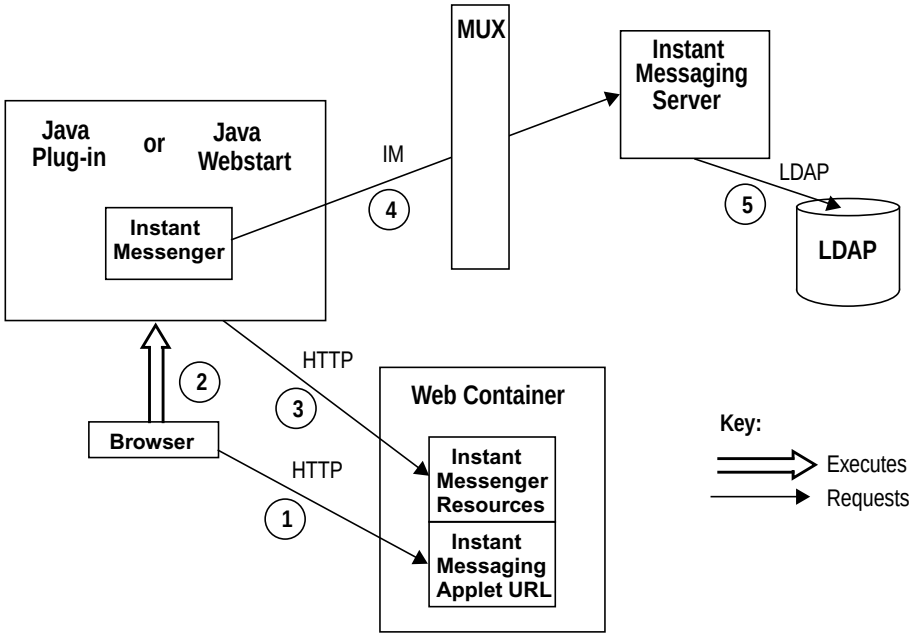
The following software components work with Sun ONE Instant Messaging server, but they are installed separately:

- Web server: Portal deployments use the web server that ships with Sun ONE Portal Server. The LDAP deployments need to install a web server, such as Sun ONE Application Server SE (Standard 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 end user authentication and end user search. In a portal deployment, the LDAP server that is used by the Portal Server is used by the Instant Messaging server to search end users.
- (Optional) SMTP server: Sun ONE Messaging Server or some other SMTP server is used to forward instant messages for end users who are offline.
- (Optional) Sun ONE Portal Server: Sun ONE Portal Server is installed for portal deployments.
- (Optional) Sun ONE Identity Server: Sun ONE Identity Server is installed for adding the Instant Messaging service.

# Deployment Overview: LDAP-Only Deployment

Figure 1-1 on page 18 illustrates the interaction of the software components in the authentication process of an LDAP-only configuration of Sun ONE Instant Messaging. The focus is on the flow of authentication requests, where the protocols used for requests are indicated above the arrows. The IM protocol is a proprietary protocol. The term MUX is an abbreviation for multiplexor. An explanation of the steps in this process follow the figure.

**Figure 1-1** Flow of Authentication Requests in an LDAP-Only Configuration




The key difference between a Sun ONE Instant Messaging LDAP-only deployment and a Sun ONE Instant Messaging deployment that uses Sun ONE Identity Server is the authentication process. The authentication process in an Instant Messaging LDAP-only deployment works as follows:

1. End user accesses the Sun ONE Instant Messenger applet URL from a browser
2. The browser invokes Java Web Start or the Java Plug-in.
3. Java Web Start or the Java plug-in downloads the necessary Sun ONE Instant Messenger resource files and starts the Instant Messenger.
4. The log-in window is displayed and the end user enters the log-in name and password. This data is sent to the Instant Messaging server via the multiplexor.

5. The Sun ONE Instant Messaging server communicates with the LDAP server to authenticate the end user and to request end-user information.

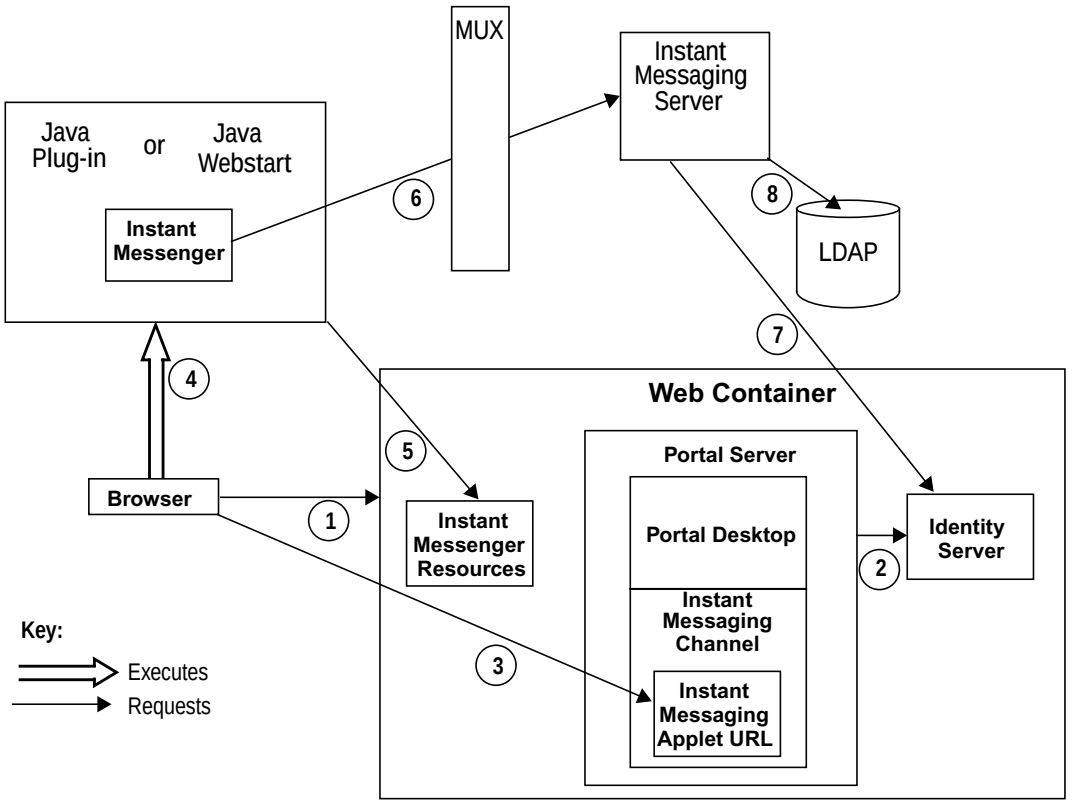
   End users should set their preferences to have alerts forwarded as email when they are offline.

   When the end-user authentication is complete, the Sun ONE Instant Messaging main window is displayed with the contact list for the end user. The end user can now start and participate in Sun ONE Instant Messaging sessions with the other end users.

# Deployment Overview: Identity Server and Portal Server in a Single Sign-On Environment

Figure 1-2 illustrates authentication process of the Sun ONE Instant Messaging software in collaboration with the Sun ONE Portal Server and Sun ONE Identity Server components in a Single Sign-On environment. As with Figure 1-1 on page 18, this figure focuses on the flow of authentication requests. An explanation of the steps in this process follows the figure.

**Figure 1-2** Flow of Authentication Requests in a Portal Server & Identity Server Configuration.



The authentication process of the Sun ONE Instant Messaging server in a Sun ONE Identity Server and Portal Server deployment within a single sign-on environment works as follows:

1. The end user logs in to the Sun ONE Portal Server by entering the URL in a web browser.
2. The Sun ONE Identity Server software authenticates the end user and returns a session token and the Sun ONE Portal Server downloads Portal Server Desktop for the end user. Portal Server Desktop is displayed in the end user's browser. See Step 6 for an explanation of the session token.
3. The end user clicks the Sun ONE Instant Messenger URL link from the Instant Messaging channel on the Portal Server Desktop.
4. The browser invokes Java Web Start or the Java Plug-in.
5. Java Web Start or the Java plug-in downloads the necessary Sun ONE Instant Messenger resource files and starts the Instant Messenger.
6. Sun ONE Instant Messenger requests authentication to the Sun ONE Instant Messaging server using the session token.

   The session token is what enables single sign-on to work. This token is provided as an applet parameter and is used throughout the authentication process. End users are not asked for their credentials again as long as the session token is present.
7. Sun ONE Instant Messaging server asks Sun ONE Identity Server to validate the session token. If the session is valid, Sun ONE Instant Messenger displays the end user's contact list and the end user can use Sun ONE Instant Messenger services: chat, alerts, polls, etc.
8. Sun ONE Instant Messaging server must query LDAP directly to get or set end-user information, such as contact lists or subscriptions.

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

# The Role of the Instant Messaging Components

## 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 that you run Sun ONE Instant Messenger as a Java Web Start application.

For more information on customizing Sun ONE Instant Messenger, see "Managing Sun™ ONE Instant Messenger" on page 77.

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 end users to complete projects, answer customer queries, and complete other time-critical assignments. Chat sessions (two or more participants) are held in chat rooms created on a need basis.
- **Conference Rooms -** Conference rooms are persistent chat rooms that work similarly to regular chat sessions, but offer:
  - Access Control
  - Moderated Chats
- **Alerts -** Alerts enable information delivery and response to end users through the Instant Messenger interface. Alerts can deliver time-critical information to the end 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 ask end 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 -** News channels are forums for posting and sharing information. End 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 end 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 Sun ONE Instant Messenger Proxy Settings.

## Sun ONE Portal Server

### *Portal Server Desktop*

Sun ONE Instant Messenger installed on the Portal Server environment can be launched from the Instant Messaging channel that available to end users on Portal Server Desktop.

### *Sun ONE Portal Server, Secure Remote Access*

Sun ONE Portal Server, Secure Remote Access enables remote end users to securely access their organizations network and its services over the Internet for Solaris-based or Windows-based systems. The end user can access Secure Remote Access by logging in to the web-based Portal Server Desktop through the portal gateway. The authentication module configured for Sun ONE Portal Server authenticates the end user. The end-user session is established with Sun ONE Portal Server and the access is enabled to the end user's Portal Server 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*

## Sun ONE Identity Server

Sun ONE Identity Server provides end 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 “Instant Messaging Multiplexor”.

Access control files and Sun ONE Identity Server policies are used for administration of end users, news channels, and conference rooms.

## 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 end user authentication or parse the client-server protocol (IM protocol).

You can install multiple multiplexors based on your deployment requirements. For more information, see “Sun ONE Instant Messaging Deployment Configurations” on page 26.

## Web Server

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

- The `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 (`messenger.jar`, `imres.jar`, `imbrand.jar`, `imdesktop.jar`, `imnet.jar`, and `imjni.jar`).
- The Sun ONE Instant Messenger Online Help.

You must install Instant Messenger resources on the same host where the web server is installed. In an Identity Server deployment, Sun ONE Instant Messenger can be installed on the Sun ONE Identity Server host or on a different web server host. In most cases, the Instant Messenger resources will be installed on the same host where you installed the Instant Messaging server software. It is possible to locate the Instant Messenger resources 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

The Sun ONE Instant Messaging server requires an LDAP directory server to perform end user authentication, search for end users, and access end user and group information.

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

The Sun ONE Instant Messaging server relies on common end-user attributes to search for end-user and group information. The configuration allows the system administrator to specify attribute names and search folders used by the server. Sun ONE Instant Messaging properties (Sun ONE Instant Messenger properties and subscriptions) can be stored in files on the Sun ONE Instant Messaging server or in the LDAP server.

Sun ONE Instant Messaging supports end users that are defined and maintained in an LDAP directory, such as Sun ONE 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 end 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

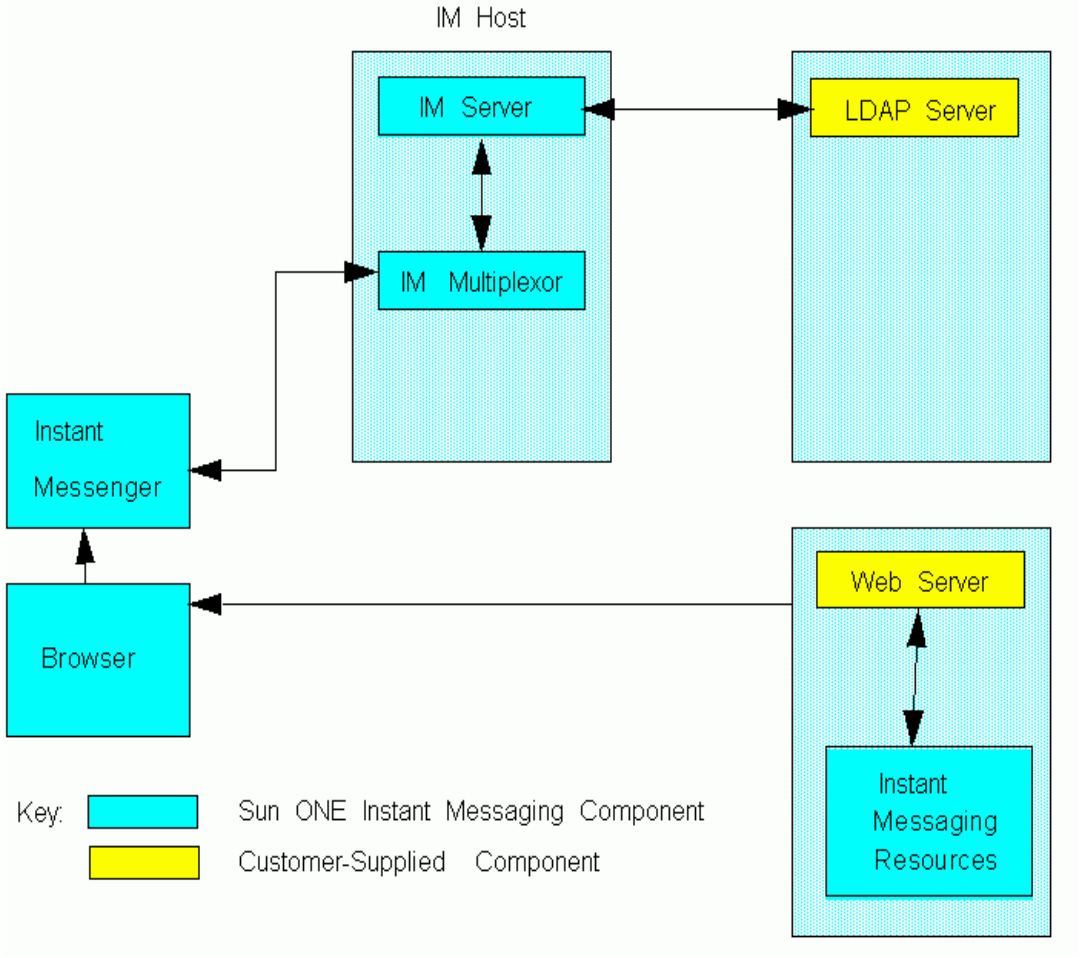
**NOTE** The configuration details described in this section are for LDAP deployments of Instant Messaging server.

For information on configuring Instant Messaging server on a 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. The Instant Messenger resources are also present on the web server host. Use this configuration when there is an existing instance of a web server and an 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.



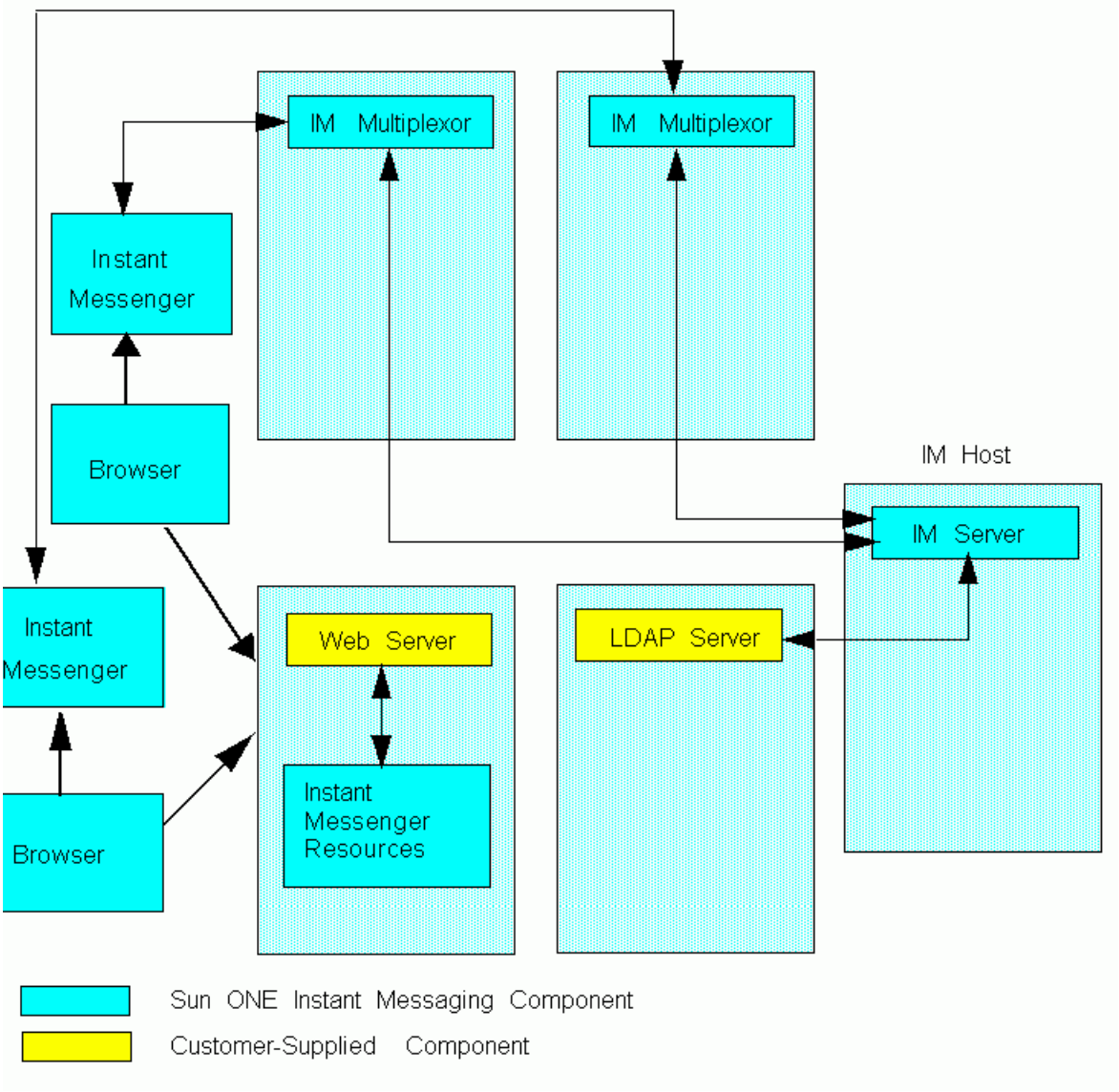
# Multiple Multiplexor Hosts

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.

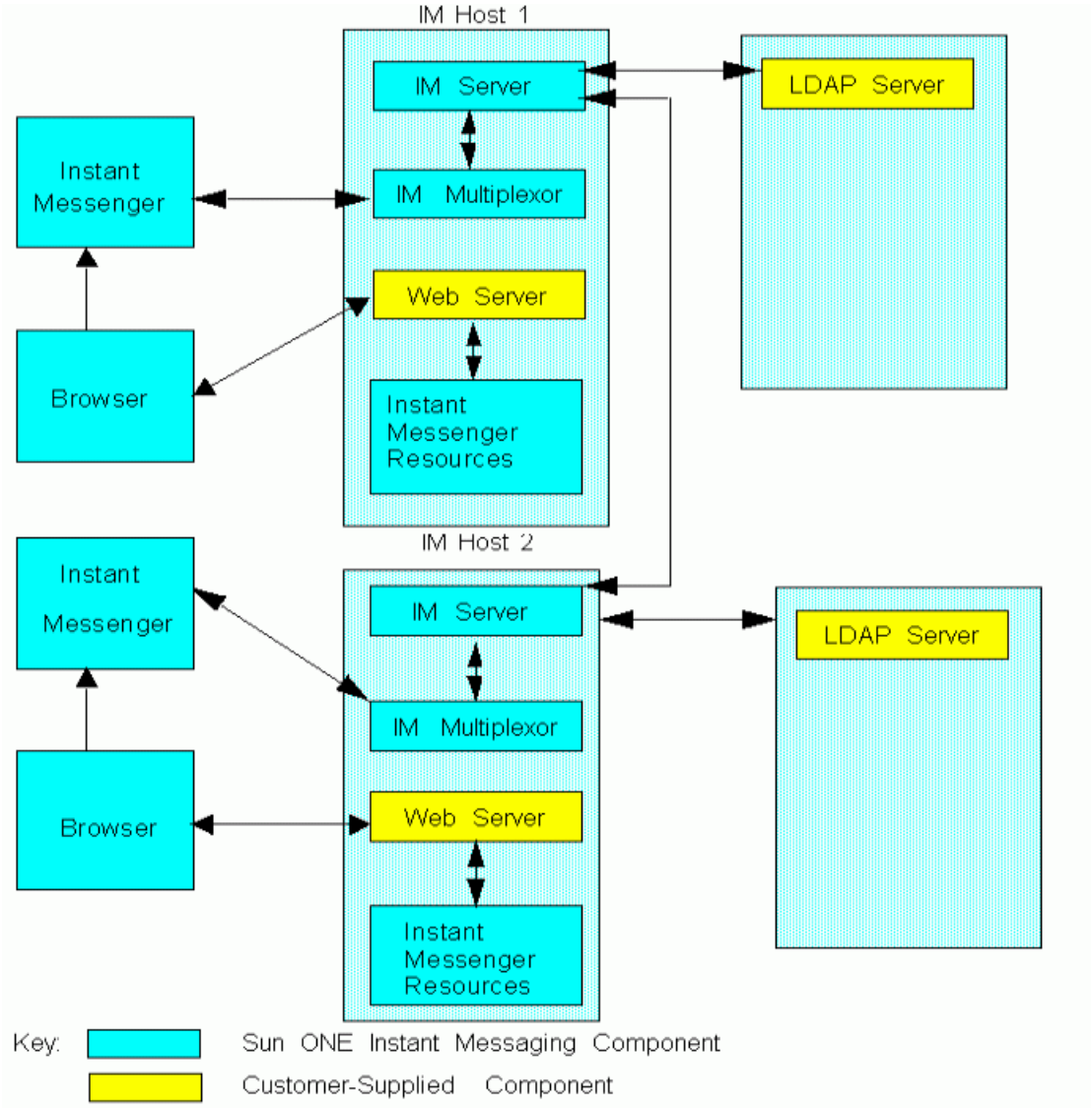
**Figure 1-4** Instant Messaging Multiplexors Installed on Two Different Hosts.



# Federation of Multiple Instant Messaging Deployments

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 end users on one Instant Messaging server can communicate with end users on other Instant Messaging servers. For more information on federating multiple Instant Messaging deployments, see "Federating Deployment of Multiple Instant Messaging Servers" on page 49.

**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<br>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. | *instant-messaging-installation-directory*`/SUNWiim`<br>The default value for the instant-messaging-installation directory is `/opt` | *instant-messaging-installation-directory*<br>The default value for the instant-messaging-installation directory is `c:\Program Files\Sun\InstantMessaging` |
| Server Configuration files<br>These files are in the instant-messaging-configuration directory and include the `iim.conf` file and a subdirectory which contains all the server-wide access control files. | By default, the instant-messaging-configuration directory is located at:<br>`/etc/opt/SUNWiim/default/config` Note: The installer creates a symbolic link from `/etc/opt/SUNWiim/default/config` to *instant-messaging-installation-directory*`/SUNWiim/config`. | *instant-messaging-installation-directory*`\config` |

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

| Description | Solaris Location | Windows Location |
|---|---|---|
| Instant Messaging Server Data.<br>These files include the configurable directory for the files generated by the server at runtime. It includes the end user data in the `instant-messaging-`database 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`<br>The default value for *instancevardir* is `/var/opt/SUNWiim` | *instant-messaging-installation-directory* \ |
| Instant Messenger resources.<br>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. | instant-messaging-resource directory<br>The default value for this resource directory is:<br>`/opt/SUNWiim/html` | instant-messaging-resource directory |

**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`.

# Sun ONE 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:

- End user properties, such as contact lists, messenger settings, subscribed news channels and access control (alternatively, these properties can be stored in LDAP).
- 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 by setting the appropriate parameters in the `iim.conf` file.

Enabling SSL between the multiplexor and Sun ONE Instant Messenger 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 Privacy, Security, and Site Policies

Sun ONE Instant Messaging provides the ability to control access to Instant Messaging features and preserve end-user privacy.

## Site Policies

Site policies specify end-user access to specific functionality in Sun ONE Instant Messaging. It specifies:

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

The Instant Messaging administrator has access to all Instant Messaging features. The administrator has `MANAGE` access to all conference rooms and news channels, can view presence information of any end user, and can view and modify properties such as Contact Lists and Instant Messenger Settings of any end user. The site policy settings have no impact on the administrator’s privileges.

By default, the end user is provided with the privileges to access the presence status of other end users, send alerts to end 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 pop-up functionality.

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

**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 end users, roles, or groups.

For more information on configuring site policies, see "Managing Instant Messaging and Presence Policies" on page 97.

## Conference Room and News Channel Access Controls

End 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 end 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.

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

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

## User Privacy

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

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

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

Chapter 2

# Administering Sun™ ONE Instant Messaging Server and Multiplexor

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

This chapter contains the following sections, which describe the various administrative tasks in Instant Messaging:

# Administering Instant Messaging: End Users

The administrative tasks in Instant Messaging are listed in the preceding section and are described throughout the rest of this chapter. Take note of the methods—as explained subsequently—for provisioning and managing end users.

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

Likewise in an LDAP-only deployment, you cannot prevent an end user from using Sun™ ONE Instant Messenger. In an LDAP-only deployment, the only way to prevent end users from using Instant Messaging is to delete them from the directory. In an Identity deployment using the policy attributes, you can prevent an end user from accessing Sun ONE Instant Messenger.

The administrator can manage Instant Messaging end users, using the Instant Messaging Administrator Access Control mechanism. For more information on Instant Messaging Administrator Access Control, see "Sun ONE Privacy, Security, and Site Policies" on page 35. In an Identity deployment, Sun ONE Identity Server is used for provisioning Instant Messaging end users. For more information, see "Sun ONE Identity Server" on page 24.

**CAUTION** If you deny end users the privilege to set up watches on other end users by editing the `syswatch.acl` file, the Sun ONE Instant Messenger's Main window is not displayed for these end users. This effectively denies end users the ability to send instant messages. However, end 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 end user who has administration rights to the system(s) on which the Sun ONE Instant Messaging server and multiplexor are running. This end user is typically the identity that the server runs as and is designated during installation:

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

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

- On Solaris: *instant-messaging-installation-directory*`/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 end 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 Sun ONE Instant Messenger from connecting to the server. For more information, see the section on Managing Sun™ ONE 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 an end 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 end 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 End-User Privileges

The Administrator can control end-user access to Instant Messaging information by restricting privileges to the end user. These privileges determine if the end user can add and delete news channels, send alerts, and setup watches on other end users. These features provide the end 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 end user can view or perform on Instant Messaging.

Sun ONE Instant Messaging provides the following access control mechanisms:

- Conference and News Channel Access Controls
- User privacy

## Conference Room and News Channel Access Controls

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

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

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

**NOTE** Setting the WRITE privilege, grants end 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 2-2 lists the Conference room and News channel access control files and the privileges that these files provide end users. These access control files are located in the db/acls directory.

**Table 2-2** Conference Room and News Channel Access Control Files

| ACL File | Privileges |
|---|---|
| *roomname*.acl | This file sets access privileges that end users can have on conference rooms. |
| *news channelname*.acl | This file sets access privileges that end users can have on news channels. |

## Room and News Channels Access Control File Format

The format of the *roomname*.acl and *news channelname*.acl files is slightly different from the system level access control files. For more information on the system level access control files, see "Access Control File Format" on page 101. 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 access control file 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
g:cn=group1,ou=groups,o=example:6
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 end users change access using Sun ONE Instant Messenger, end users can lose their changes if the files are edited manually while the server is running.

# User Privacy

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

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

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

# Federating Deployment of Multiple Instant Messaging Servers

In an LDAP-only deployment, when you federate multiple Sun ONE Instant Messaging deployments you form a larger Instant Messaging community. End users from different servers can communicate with each other, user conference rooms on other domains, and subscribe to news channels on remote servers based on the access privileges.

In an Identity deployment, a single Sun ONE Instant Messaging server can host multiple domains. You can designate a single domain as the default domain for a Sun ONE Instant Messaging server instance. End users in different domains hosted by the same server cannot interact with each other. When you federate multiple Sun ONE Instant Messaging deployments, end users in default domains can see the end users in default domains of other remote Sun ONE Instant Messaging servers.

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 Sun ONE Instant Messaging servers in the network. A Sun ONE Instant Messaging server identifies itself with its domain name, host and port number, serverID, and password.

Within the server configuration, you can assign each Sun ONE 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 2-3.

   Table 2-3 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 2-3** 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

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

**NOTE** For more information about managing certificates, see *Sun ONE Web Server Administrator's Guide.*

3. Configuring the Instant Messaging Server to Enable SSL between the Multiplexor and the Instant Messenger
4. To Activate SSL for Server to Server Communication
5. 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.

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

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 end 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 using an editor of your choice. For example, you could 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 end 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 2-4 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 2-4** 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.`<br>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`.<br>If the certificate database contains a prefix, for example `Secret-stuff-cert7.db`, then value of this parameter is `Secret-stuff`. |

**Table 2-4** Configuration Information for enabling SSL Between Sun ONE Instant Messenger and Multiplexor

| Parameter | Default Value | Description |
|---|---|---|
| `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.<br>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:<br>`Internal (software) Token:`*password*<br>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 2-4 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 the resource directory, the base directory under which all the Sun ONE Instant Messenger resources are stored.

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

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

## Configuring the Instant Messaging server to enable SSL between two Instant Messaging servers

Table 2-5 on page 59 lists the parameters in the `iim.conf` file for enabling SSL between two Sun ONE Instant Messaging servers. It also contains the description and the default value of these parameters:

**Table 2-5** Configuration Information for Enabling SSL Between Two Sun ONE Instant Messaging Servers

| Parameter | Default Value | Description |
|---|---|---|
| `iim_server.secconfigdir` | `/etc/opt/SUNWiim/default/config` | This directory contains the key and certificate databases. It usually contains the security module database. |
| `iim_server.keydbprefix` | None | This value should contain the key database filename prefix. The key database file name must always end with `key3.db.`<br>If the Key database contains a prefix, for example `This-Database-key3.db`, then value of this parameter is `This-Database`. |
| `iim_server.certdbprefix` | None | This value should contain the certificate database filename prefix. The certificate database file name must always end with `cert7.db`.<br>If the certificate database contains a prefix, for example `Secret-stuff-cert7.db`, then value of this parameter is `Secret-stuff`. |
| `iim_server.secmodfile` | `secmod.db` | This value should contain the name of the security module file. |
| `iim_server.certnickname` | `Server-Cert` | This value should contain the name of the certificate you entered while installing the certificate.<br>The certificate name is case-sensitive. |

**Table 2-5** Configuration Information for Enabling SSL Between Two Sun ONE Instant Messaging Servers

| Parameter | Default Value | Description |
|---|---|---|
| `iim_server.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:<br>`Internal (software) Token:`*password*<br>Where *password* is the password protecting the key database. |
| `iim_server.trust_all_cert` | `false` | If this value is true than the server will trust all certificates and will also add the certificate information into the log files. |

# Managing Sun ONE Instant Messaging's LDAP Configuration

An LDAP-only deployment of Sun ONE Instant Messaging server requires a directory server. In an LDAP-only deployment, the Instant Messaging server uses the directory server to perform end-user authentication and to search for end users.

In an Identity deployment, Sun ONE Instant Messaging server uses the directory used by Sun ONE Portal Server. When installed in an Identity deployment environment, Sun ONE Instant Messaging server uses the directory used by the Sun ONE Identity Server to search for end users, and not for end-user authentication. In an Identity deployment, Sun ONE Identity 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:

- End user entries are identified by the `inetOrgPerson` object class.
- Group entries are identified by the `groupOfUniqueNames` object class.
- Sun ONE Instant Messenger user ID attribute of an end user is provided by the `uid` attribute (from `inetOrgPerson` objectclass).
- The email address of an end user is provided by the `mail` attribute.

- The display name of an end 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 End 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
```

| 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. |
|---|---|

2. In an Identity deployment, the directory is generally not searchable by anonymous users. In an Identity deployment set the `iim_ldap.useidentityadmin` configuration parameter to `true`. Also you can delete or comment out the following configuration parameters:
   - `iim_ldap.usergroupbinddn`
   - `iim_ldap.usergroupbindcred`.
3. Edit the `iim.conf` file.

   See "To Change Configuration Parameters" on page 43 for instructions on editing the `iim.conf` file.

   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 end users based on their DN and include them in a single group. The dynamic groups are defined in Sun ONE Directory Server as the `groupOfUrls` objectclass.

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

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=groupofuniquenames)
(objectclass=groupofurls)))(cn={0}))(&(objectclass=inetorgperson)(cn={0})))

iim_ldap.groupbrowsefilter=(|(objectclass=groupofuniquenames)(objectclass=g
roupofurls))

iim_ldap.groupclass=groupOfUniqueNames,groupOfURLs
```

The attribute and objectclass names are configurable.By default, the `memberOfUrls` attribute is used as the membership attribute of a dynamic group. If you want to use an attribute name other than `memberOfUrls`, set the `iim_ldap.groupmemberurlattr` option to the attribute name you want to use.

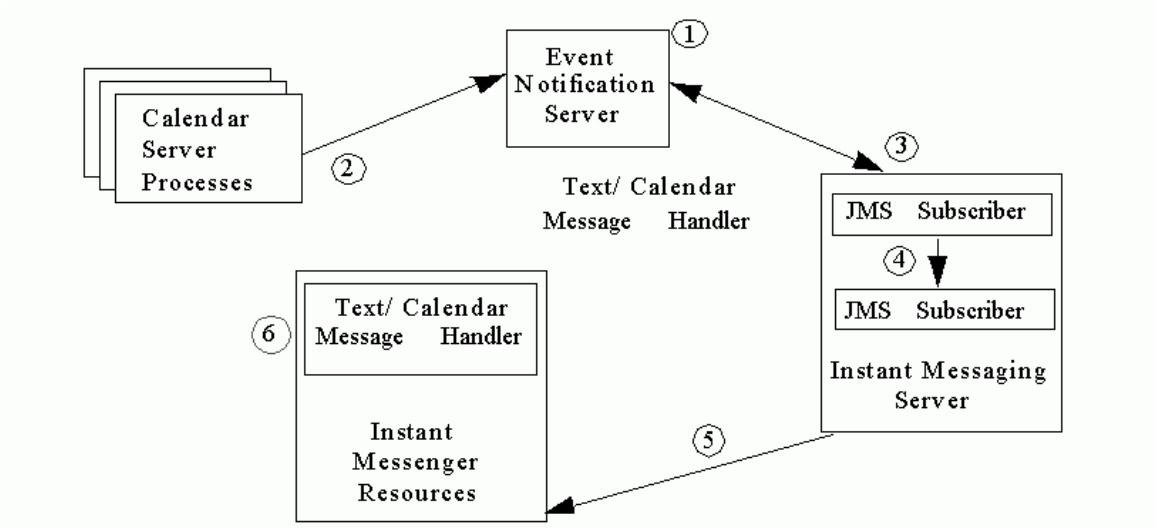
# Displaying Calendar Reminders and Notifications as Instant Messenger Popups

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” on page 94.

**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 *calendar-server-install-dir*`/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 2-6 shows the server configuration options in the file `iim.conf` and their description.

**Table 2-6** 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:<br>• `topic` for JMS Topic subscription or<br>• `queue` for JMS queue bindings.<br>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 2-6** 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 2-7 contains the values for the parameter `eventtype` and their description.

**Table 2-7** 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 2-7** 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`.

# Backing Up 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 end user data
- Instant Messenger resources

The configuration information is stored in the Instant Messaging configuration directory as follows:

- Solaris: `/etc/opt/SUNWiim/default/config`
- Linux: `/etc/opt/soim/default/config`
- Windows: *instant-messaging-installation-directory*`\config`
- (Optional) If you customized any of the files mentioned in Customizing Sun ONE Instant Messenger, back them up from the resource directory.

The Sun ONE Instant Messaging end user data is stored in the following database directories:

- Solaris: `/var/opt/SUNWiim/default/db`
- Linux: `/var/opt/soim/default/db`
- Windows: *instant-messaging-installation-directory*`\db`

The Instant Messenger resources must be backed up if they have been customized. The location of the Instant Messenger resources are provided during installation.

# Performing Backup

While the configuration information does not change frequently, the Instant Messaging end-user data changes rapidly and to prevent any loss of end-user data it is recommended that the Instant Messaging end-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 end 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 end-user data and the configuration information needs to be restored when there is a disk failure and all the end-user data and the configuration information is lost.

To restore the backed up end-user data:

1. Change to the runtime directory. For example:

   `cd` *runtime-directory*

2. Grant read-only permission to the instant-messaging-database directory, type:

   `chmod -R 400 db`

3. Stop the Instant Messaging server, type:

   `imadmin stop`

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

   `chmod -R 600` *runtime-directory*`/db/.`

5. To restore the data, copy the backed up data to the instant-messaging-database directory.
6. Start the Instant Messaging server, type:

   ```
   imadmin start
   ```

   On Solaris - the default value of *runtime-directory* is `/var/opt/SUNWiim/default.`

   On Linux - the default value of *runtime-directory* is `/var/opt/soim/default.`

Chapter 3

# Managing Sun™ ONE Instant Messenger

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

This chapter contains the following:



## Configuring Sun ONE Instant Messenger

There are two ways to configure and invoke Sun ONE Instant Messenger:

**Using Java Web Start** In this configuration, Sun ONE Instant Messenger is launched as an application from the Java Web Start. The browser is no longer necessary once Sun ONE Instant Messenger is launched.

**Using the Java Plug-in** In this configuration, Sun ONE 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 Sun ONE Instant Messenger, see the *Sun One Instant Messaging Installation Guide*.

# Invoking Instant Messenger

You can invoke Sun ONE Instant Messenger using:

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

## To Invoke Sun ONE Instant Messenger

Use the following URL to invoke the 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-resource-directory* is selected during the installation, then no subdirectory is required to store the client files. |

| | |
|---|---|
| *filename* | Specifies the Sun ONE Instant Messenger file to use:<br>`index.html` - This file is provided with the product. The file contains the links to launch both the Java Web Start and Java Plug-in versions of Instant Messenger.<br>`im.jnlp` - The jnlp file to launch only the Java Web Start version of Sun ONE Instant Messenger.<br>`im.html` - The page to launch only the Java Plug-in version of Sun ONE 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 describes web server issues that apply to LDAP deployments and also to portal deployments, where Instant Messenger is not installed on the Portal Server host and uses a different Web Server to provision Sun ONE Instant Messenger.

## Changing the Codebase

The web-server-resource directory can be the same directory as the instant-messaging-resource directory. However, they do not need to be the same. If they are not the same for your site, use the appropriate method from the following to enable the web server to download the Sun ONE Instant Messenger resources:

- Web server - You can configure the web server to enable access to the directory where the Sun ONE Instant Messenger files are installed, or create a symbolic link in the web-server-resource directory.

For example, if Instant Messaging server host is `iim.i-zed` and the Sun ONE 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 within the web-server-resource directory.

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

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

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

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

  End users can also type the following URLs to launch Sun ONE 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 Sun ONE 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 Instant Messaging server host is `iim.i-zed` and the Sun ONE 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-resource directory. 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, 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 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 Sun ONE 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 Instant Messaging server host is `iim.i-zed and` the web server port is `8080`, the URL to access the Sun ONE Instant Messenger main page `index.html` will be:

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

# Customizing Sun ONE Instant Messenger

Sun ONE Instant Messenger is customizable. HTML and JNLP files 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 LDAP Deployments)
- Customizing the Application (Java Web Start)
- Customizing User Name Display

This section describes the Instant Messaging server files you can modify to customize Sun ONE 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 instant-messaging-resource directory.

# Instant Messenger Resources

## Sun ONE Instant Messenger Files

The Sun ONE Instant Messenger files are located within the directory referred to as the instant-messaging-resource directory, which is also simply referred to as the resource directory.

Table 3-1 contains the list of Sun ONE Instant Messenger files in the instant-messaging-resource directory. It also contains the description and the customization information of these files.Within the instant-messaging-resource directory, is the locale subdirectory represented generically in a directory path as *lang*, but specifically as abbreviations of languages, such as, `en_US`, `jp`, and `fr_FR`.

**Table 3-1** Sun ONE Instant Messenger Files

| File | Description | Customizable? |
|---|---|---|
| *lang*/`im.html` | The initial page that launches the Java Plug-in version of Sun ONE 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. |

**Table 3-1** Sun ONE Instant Messenger Files

| File | Description | Customizable? |
|---|---|---|
| `imdesktop.jar` | A 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 Sun ONE Instant Messenger. | Yes. |
| `im.jnlp.template` | The template version of `im.jnlp`. | No. |
| `imjni.jar` | A client `jar` file, downloaded by `im.html` or `im.jnlp`. | No. |
| `messenger.jar` | The main client jar file, downloaded by im.html or im.jnlp. | No |
| `icalendar.jar` | The icalendar parser used to process calendar reminders. | No |
| `imnet.jar` | A 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 Sun ONE Instant Messenger. It 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 Sun ONE 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. |
| `jnlpLaunch.jsp` | If an end user is already logged onto Sun ONE Identity Server, then this file can be used to allow single sign-on and to launch Sun ONE Instant Messenger using Java Web Start. | Yes |

**Table 3-1** Sun ONE Instant Messenger Files

| File | Description | Customizable? |
|---|---|---|
| `pluginLaunch.jsp` | If an end user is already logged onto Sun ONE Identity Server, then this file can be used to allow single sign-on and to launch Sun ONE Instant Messenger using Java Plug-in. | Yes |
| `index.html` | The splash page for an LDAP 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. |
| *lang*`/imhelp/SunONE.jpg` | The image used by `quickref.htm`, `solaris.htm`, and `windows.htm`. | Can be replaced. |
| `javaws_not_installed.html` | The page that appears when an end user tries to launch the Sun ONE Instant Messenger by using Java Web Start and Java Web Start has not been installed on the end user's system. | Yes. |
| `quickref.html`<br>`solaris.html`<br>`windows.html` | Located in `lang/imhelp/`, they provide documentation on getting started with Sun ONE 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 LDAP 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 Sun ONE Instant Messenger applet and the Java Web Start application. If you are running the Sun ONE 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` file 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*.

---

# Launching Instant Messenger Using Sun ONE Identity Server SSO:

To launch the Sun ONE Instant Messenger client using single sign-on with Identity Server use `jnlpLaunch.jsp` and `pluginLaunch.jsp`. These files are in the resource directory. To launch the Instant Messaging server enter the following in the browser:

**`instant-messaging-codebase/jnlpLaunch.jsp?server=muliplexor-hostname:mulip lexor-port`**

or

**`intant-messaging-codebase/pluginLaunch.jsp?server=www.example.com:49909`**

where,

**intant-messaging-codebase** is the codebase from which the Instant Messenger resources are downloaded. For example, `http://www.example.com`.

**(multiplexor)-hostname** is the name for the mulitplexor. For example, http://www.compnay22.com.

**(multiplexor) port** is the multiplexor port number. For example, 49909

**jnlpLaunch.jsp** is used for launching Instant Messenger using Java Web Start.

**pluginLaunch.jsp** is used for launching Instant Messenger using Java Plug-in.

**NOTE**

- The `jnlpLaunch.jsp` and `pluginLaunch.jsp` files require an argument for the server.
- The `jnlpLaunch.jsp` and `pluginLaunch.jsp` files can be customized similar to `im.jnlp` and `im.html` files.

# Customizing the Application (Java Web Start)

If you are running Sun ONE Instant Messenger using Java Web Start, you can modify the `im.jnlp`, `imres.jnlp`, and `imres.jar` files to customize 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.

  Code Example 3-1 shows a sample `im.jnlp` file with the HTML code that can be customized in bold typeface.

**Code Example 3-1** 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 on 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. The same is possible with the audio files (`.au` files).

## Contents Listing of imbrand.jar

Table 3-2 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 Sun ONE Instant Messenger.

**Table 3-2** 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. |
| `Angel_16.gif` | An emoticon used to show angelic emotion graphically |
| `Smiley_16.gif` | An emoticon used to show smile emotion graphically. |

**Table 3-2** Contents Listing of imbrand.jar file.

| File Name | Description |
|---|---|
| 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 an end user is away or not connected that appears in the Change Status menu. |
| Idle_13.gif | Icon to show idle status, appears in the status bar and contact list. |
| Forwarded_13.gif | Icon displayed against end 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. |
| 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 an end user receives an alert. |
| away.au | Sound when an end user changes status to away. |
| soundon.au | Sound when an end user exits Instant Messenger. |
| soundoff.au | Sound when an end user logs on to Instant Messenger. |
| send.au | Sound when an end user sends an instant message. |
| receive.au | Sound when an end 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` *instant-messaging-resource-directory/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 resource directory. For example:

   `cp imbrand.jar` *instant-messaging-resource-directory/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 end users have the same first name and last name, it is impossible to know which end 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 arbitrary 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` *instant-messaging-resource-directory*

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 Sun ONE Instant Messenger Conference Rooms and News Channels

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

- Administering conference rooms
- Administering and managing news channels
- Assigning conference room access levels to end users
- Assigning news channel access levels to end users
- Assigning end users to conference rooms
- Assigning end users to news channels (subscribing)
- Creating new conference rooms
- Creating new news channels
- Configuring end user settings
- Deleting conference rooms
- Deleting messages from news channels
- Deleting news channels
- Posting messages in news channels
- Removing end users from conference rooms
- Removing end users from news channels

## Granting End Users the Privilege to Create Conference Rooms and News Channels

The administrator can create conference rooms and news channels for end users. However, with the proper privileges, end users can do this also. For more information about adding policies to give end users access to create conference rooms and news channels, see Chapter 4, “Managing Instant Messaging and Presence Policies” on page 97. End 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. For more information on managing end-user privileges, see “Managing End-User Privileges” on page 46.

# Modifying Sun ONE Instant Messenger Proxy Settings

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

**Table 3-3** Instant Messenger Applet Parameters

| Parameter | Default Value | Description |
|---|---|---|
| `server` | `127.0.0.1` | The Instant Messaging server host and port. |
| `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 Instant Messenger that it is run in SRA mode. It displays a security indicator. |
| `usessl` | FALSE | Tells Instant Messenger to use SSL when connecting to server. |
| `allow_alert_only` | FALSE | Tells Instant Messenger to let end user display neither the contact list nor the news channel.<br>This parameter is used in CHAT and POPUP flavors. |
| `allow_file_transfer` | TRUE | Allows file attachment and transfer. |

**Table 3-3** Instant Messenger Applet Parameters

| Parameter | Default Value | Description |
|---|---|---|
| `enable_moderator` | `TRUE` | If set to true, enables the moderated conference feature. |
| `messenger_bean` | | This parameter contains a list of messenger beans to be used. You can enter multiple factory class names with each separated by a comma. |
| `domain` | null | This parameter is used in multi-domain Sun ONE Identity Server deployments. The value of this parameter should be the logical domain name of the organization in which this end user is present. |
| `gateway_url` | null | 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 a limited amount of information on the end user's system for auto-login. This information can be located at:

*home-directory*`/.sunmsgr`

*home-directory* is the end user's home directory. The home directory of the end user can be obtained from the `user.home` parameter in the Java system property.

Table 3-4 shows the directories and files containing the cached data. It also contains the description of the files and the directories.

**Table 3-4** 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 {log-in name, domain name} combination. |

**Table 3-4** The directory and files containing the cached data

| File/Directory Name | Type | Description |
|---|---|---|
| `.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 3-5 shows the auto-logon properties for Instant Messaging. It also contains the description and the default values of these properties.

**Table 3-5** The Auto-logon Properties

| Parameter | Default Value | Description |
|---|---|---|
| `net.server` | `127.0.0.1` | Instant Messaging server host name and port. |
| `net.server.`*n*<br>(Where *n* is a digit used to distinguish one entry from another) | | The secondary servers' host names and port numbers. |
| `net.user` | | The default user id |
| `net.pass` | | The encoded user password that enables auto-logon. |

Chapter 4

# Managing Instant Messaging and Presence Policies

The Sun™ ONE Instant Messaging server provides various functional features such as chat, conferencing, polls, presence access, etc. A policy describes a set of access control privileges that can be associated with these features. In turn, end users and groups can be assigned to policies according to the needs of an organization.

This chapter describes how to define and use policies to manage the access that end users and administrators have to the Sun ONE Instant Messaging server features and privileges:



## Methods for Controlling End User and Administrator Privileges

Different sites using Sun ONE Instant Messaging server have different needs in terms of enabling and restricting the type of access end users have to the Instant Messaging service. The process of controlling end user and administrator Sun ONE Instant Messaging server features and privileges is referred to as policy management. There are two methods of policy management available: through access control files or through Sun™ ONE Identity Server.

## Introduction to Managing Policies Using Access Control Files

The access control file method for managing policies allows you to adjust end-user privileges in the following areas: news channel management, conference room management, the ability to change preferences in the User Settings dialog, and ability to send alerts. It also allows specific end users to be assigned as system administrators.

## Introduction to Managing Policies Using Sun ONE Identity Server

Managing policies through Sun ONE Identity Server gives you control of the same privileges available with the access control file method; however it additionally allows more fine-tuned control over various features, such as: the ability to receive alerts, send polls, receive polls, etc. For a complete list, please refer to table Table 4-4 on page 105. Furthermore, managing policies using Sun ONE Identity Server gives you finer-tuned control over privileges.

Two types of policies exist: Instant Messaging policies and Presence policies. The Instant Messaging policies govern general Instant Messaging features, such as the ability to send or receive alerts; the ability to manage public conferences and news channels; and the ability to send files. Presence policies govern the control end users have over changing their online status, and in allowing or preventing others from seeing their online or presence information.

## Managing Policies: The Method to Use

When choosing which method to use to manage policies, it is also necessary to choose where they will be stored. You select the method for managing policies by editing the `iim.conf` file and setting the `iim.policy.modules` parameter to either `identity` for the Sun ONE Identity Server method or `iim_ldap` for the access control file method, which is also the default method.

If you will use an LDAP-only deployment—therefore, you will not be using Sun ONE Identity Server—you must use the access control file method. If you are using Sun ONE Identity Server with the Sun ONE Instant Messaging server, and you have installed the Instant Messaging and Presence services components, you can use either policy management method. Please note that managing policies using Sun ONE Identity Server is a more comprehensive method. One advantage of this method is that it allows you to store all end-user information in the directory.

The specific steps for setting which method you want to use to manage policies are as follows:

1. Change directories to the directory that contains the `iim.conf` file.
2. Open the `iim.conf` file using an editor of your choice.
3. Edit the `iim.policy.modules` parameter by setting it to one of the following:
   - `iim_ldap` (the access control file method)
   - `identity` (the Sun ONE Identity Server method)
4. Edit the `iim.userprops.store` parameter and set it to either:
   - ldap (to store user properties in LDAP)
   - file (default) (to store user properties in files)
5. Save your changes.
6. Refresh the configuration.

# Policy Configuration Parameters

Table 4-1 lists and describes the new parameters available in the `iim.conf` file that relate to the increased role that Sun ONE Identity Server can play in Instant Messaging deployments:

**Table 4-1** New Parameters Related to Identity Server in iim.conf File

| Parameter Name | Use | Values |
|---|---|---|
| `iim.policy.modules` | Indicates if Identity Server is used for policy storage | iim_ldap (default)<br>identity |
| `iim.userprops.store` | Indicates if the user properties are in user properties file or from LDAP | file (default)<br>ldap |

**NOTE** Currently the `iim.userprops.store` parameter is only significant when the service definitions for the Presence and Instant Messaging services have been installed.

# Managing Policies Using Access Control Files

By editing access control files you control the following end-user privileges:

- To access the presence status of the other end users
- To send alerts to other end users
- To save properties on the server
- To create new conference rooms
- To create new news channels

By default, end users are provided the privileges to access the presence status of other end users, send alerts to end users, and save properties to the server. In most of the deployments, the default values need not be changed.

**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 end users or groups.

The location of the access control files are:

- On Solaris:

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

- On Linux:

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

- On Windows, the default directory is:

  *instant-messaging-installation-directory*`\config\acls`

Table 4-2 lists the global access control files for Sun ONE Instant Messaging and the privileges these files provide end users.

**Table 4-2** Access Control Files

| ACL File | Privileges |
|---|---|
| `sysSaveUserSettings.acl` | Defines who can and cannot change their own preferences. |
| `sysTopicsAdd.acl` | Defines who can and cannot create News channels. |
| `sysRoomsAdd.acl` | Defines who can and cannot create Conference rooms. |
| `sysSendAlerts.acl` | Defines who can and cannot send alerts. |
| `sysWatch.acl` | Defines who can and cannot watch changes of other end users. The Sun ONE Instant Messenger window is not displayed for end users who do not have this privilege. |
| `sysAdmin.acl` | Reserved for administrators only.This file sets administrative privileges to all Sun ONE Instant Messaging features for all end users. This privilege overrides all the other privileges and gives the administrator MANAGE access to all conference rooms and news channels as well as to end user presence information, settings, and properties. |

# Access Control File Format

The access control file contains 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 an access control file and selectively disallow end users that privilege.

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

End-user and group tags are followed by the end-user or group name.

Multiple end users and groups are specified by having multiple end 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 end 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 access control files might change in future releases of the product.

# Access Control File Examples

This section shows a sample access control file that shows privileges set for, the `sysTopicsAdd.acl` file. For information about access control files at the conference room and news channel level (Therefore, *roomname*`.acl` and *newschannel*`.acl`) see "Conference Room and News Channel Access Controls" on page 46.

## 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 end users and groups that appear before the default, namely `user1`, `user2`, and the `sales` group.

```
# Example sysTopicsAdd.acl file
u:user1
u:user2
g:cn=sales,ou=groups,o=siroe
d:False
```

## Changing End User Privileges

To change end user privileges:

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

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

2. Edit the appropriate access control file. For example:

   ```
   vi sysTopicsAdd.acl
   ```

3. Save the changes.

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

# Managing Policies using Sun ONE Identity Server

The Instant Messaging and Presence services in Sun ONE Identity Server provide another way to control end user and administrator privileges. Each service has three types of attributes: dynamic, user, and policy. A policy attribute is the type of attribute used to set privileges.

Policy attributes become a part of the rules when rules are added to a policy created in Identity Server to allow or deny administrator and end-user involvement in various Instant Messaging features, such as receiving poll messages from others.

When Sun ONE Instant Messaging server is installed with Sun ONE Identity Server, several example policies and roles are created. See the *Sun ONE Identity Server Getting Started Guide* and the *Sun ONE Identity Server Administration Guide* for more information about policies and roles.

Furthermore, if the example policies are not sufficient, you can create new policies and assign those policies to a role, group, organization, or end user as needed to match your site's needs.

When the Instant Messaging service or the Presence service are assigned to end users, they receive the dynamic and user attributes applied to them. The dynamic attributes can be assigned to a Sun ONE Identity Server configured role or organization.

When a role is assigned to an end user or an end user is created in an organization, the dynamic attributes then become a characteristic of the end user. The user attributes are assigned directly to each end user. They are not inherited from a role or an organization and, typically, are different for each end user.

When end users log on, they get all the attributes that are applicable to them depending upon which roles are assigned to them and how the policies are applied.

Dynamic, user or policy attributes are associated with end users after assigning the Presence and Instant Messaging Services to these end users.

# Instant Messaging Service Attributes

Table 4-3 lists the policy, dynamic, and user attributes that each service has:

**Table 4-3** Sun ONE Identity Server Attributes for Sun ONE Instant Messaging

| Service | Policy Attribute | Dynamic Attributes | User Attributes |
|---|---|---|---|
| sunIM | sunIMAllowChat | sunIMProperties | sunIMUserProperties |
| | sunIMAllowChatInvite | sunIMRoster | sunIMUserRoster |
| | sunIMAllowForumAccess | sunIMConferenceRoster | sunIMUserConferenceRoster |
| | sunIMAllowForumManage | sunIMNewsRoster | sunIMUserNewsRoster |
| | sunIMAllowForumModerate | | |
| | sunIMAllowAlertsAccess | | |
| | sunIMAllowAlertsSend | | |
| | sunIMAllowNewsAccess | | |
| | sunIMAllowNewsManage | | |
| | sunIMAllowFileTransfer | | |
| | sunIMAllowContactListManage | | |
| | sunIMAllowUserSettings | | |
| | sunIMAllowPollingAccess | | |
| | sunIMAllowPollingSend | | |
| sunPresence | sunPresenceAllowAccess | sunPresenceDefaultAcess | sunPresenceEntityDefaultAccess |
| | sunPresenceAllowPublish | sunPresenceAccessDenied | sunPresenceEntityAccessDenied |
| | sunPresenceAllowManage | sunPresenceAccessPermitted | sunPresenceEntityAccessPermitted |
| | | sunPresenceDevices | sunPresenceEntityDevices |

For each attribute in the preceding table, a corresponding label appears in the Identity Server admin console. The two following tables list each attribute with its corresponding label and a brief description. Table 4-4 lists and describes the policy attributes and Table 4-5 lists and describes the dynamic and user attributes.

**Table 4-4** Identity Server Policy Attributes for Instant Messaging

| Policy Attribute | Admin Console Label | Attribute Description |
|---|---|---|
| sunIMAllowChat | Ability to Chat | End users can be invited to join chat room and access normal chat functionality |
| sunIMAllowChatInvite | Ability to Invite others to Chat | End users can invite others to chat |
| sunIMAllowForumAccess | Ability to Join Conference Rooms | A conference tab shows up in Sun ONE Instant Messenger, allowing end users to join conference rooms |
| sunIMAllowForumManage | Ability to Manage Conference Rooms | End users are able to create, delete, and manage conference rooms |
| sunIMAllowForumModerate | Ability to Moderate Conference Rooms | End users can be conference moderators |
| sunIMAllowAlertsAccess | Ability to Receive Alerts | End users can receive alerts from others |
| sunIMAllowAlertsSend | Ability to Send Alerts | End users can send alerts to others |
| sunIMAllowNewsAccess | Ability to Read News | A News button is displayed in Sun ONE Instant Messenger that enables end users to list news channels in order to receive and send news messages |
| sunIMAllowNewsManage | Ability to Manage News Channels | End users can manage news channels and create, delete, and assign privileges to news channels |
| sunIMAllowFileTransfer | Ability to Exchange Files | End users can add attachments to alert, chat, and news messages |
| sunIMAllowContactListManage | Ability to Manage one's Contact List | End users can manage their own contact lists; they can add and delete users or groups to and from the list; they can rename the folder in their contact list |
| sunIMAllowUserSettings | Ability to Manage Messenger | A Settings button is displayed in the Sun ONE Instant Messenger that enables end users to change their own Sun ONE Instant Messenger settings |

**Table 4-4** Identity Server Policy Attributes for Instant Messaging

| Policy Attribute | Admin Console Label | Attribute Description |
|---|---|---|
| sunIMAllowPollingAccess | Ability to Receive Polls | End users can receive poll messages from others, and they can respond to polls |
| sunIMAllowPollingSend | Ability to Send Polls | A Poll button is displayed in Sun ONE Instant Messenger that enables end users to send poll messages to others and to receive the responses |
| sunPresenceAllowAccess | Ability to Access other's Presence | End users can watch the presence status of others. The contact list, in addition to showing the contact, reflects contacts' presence status changes by changing the status icon |
| sunPresenceAllowPublish | Ability to Publish Presence | End users can click to select their status (online, offline, busy, etc.) for others to watch |
| sunPresenceAllowManage | Ability to Manage Presence Access | An Access tab is displayed in the Settings of the Sun ONE Instant Messenger; end users can set up their own default presence access, presence permitted, or presence denied list |

# Modifying Attributes Directly

An end user can log into Sun ONE Identity Server admin console and view the values of attributes in the Instant Messaging and Presence service attributes. If the attributes have been defined as modifiable, end users can alter them. However, by default no attributes in the Instant Messaging service are modifiable, nor is it recommended that end users be allowed to modify them. However, from the standpoint of system administration, manipulating attributes directly can be useful.

For example, since roles do not affect some system attributes, such as setting conference subscriptions, system administrators might want to modify the values of these attributes by copying them from another end user (such as a from a conference roster) or modifying them directly. These attributes are listed in Table 4-5 on page 107.

In reference to table Table 4-5,user attributes can be set by end users through the Sun ONE Identity Server admin console. Dynamic attributes are set by the administrator. A value set for a dynamic attribute overrides or is combined with the corresponding user attribute value.

The nature of corresponding dynamic and user attributes influences how conflicting and complementing information is resolved. For example, Conference Subscriptions from two sources (dynamic and user) complement each other; therefore, the subscriptions are merged. Neither attribute overrides the other.

**Table 4-5** Identity Server User and Dynamic Attributes for Instant Messaging

| Admin Console Label | User Attribute | Dynamic Attribute | Attribute Description | Conflict Resolution |
|---|---|---|---|---|
| Messenger Settings | `sunIMUser Properties` | `sunIMProperties` | Contains all the properties for Sun ONE Instant Messenger and corresponds to the `user.properties` file in the file-based user properties storage | Merge-however, if a particular property has a value from both the user and dynamic attribute, the dynamic attribute overrides |
| Subscriptions | `sunIMUserRoster` | `sunIMRoster` | Contains subscription information (not in use yet) | The dynamic information is taken |
| Conference Subscriptions | `sunIMUser ConferenceRoster` | `sunIMConference Roster` | Contains conference room subscription information | Merge-dynamic and user subscriptions are merged |
| News Channel Subscriptions | `sunIMNewsRoster` | `sunIMUserNews Roster` | Contains news channel subscription information | Merge-dynamic and user subscriptions are merged |
| Default Presence Visibility | `sunPresenceEntity DefaultAccess` | `sunPresenceDefault Access` | Corresponds to the access setting in Sun ONE Instant Messenger. If checked, the presence status can be viewed by everyone, and if not checked, the presence status can be viewed by no one | The dynamic information is taken |

**Table 4-5** Identity Server User and Dynamic Attributes for Instant Messaging

| Admin Console Label | User Attribute | Dynamic Attribute | Attribute Description | Conflict Resolution |
|---|---|---|---|---|
| Presence Deny List | `sunPresenceEntity AccessDenied` | `sunPresenceAccess Denied` | If the *default presence visibility* label (see preceding table entry) in the admin console is checked (viewed by everyone), end users can enter others to this list to deny them access to presence status | The dynamic information is taken |
| Presence Allow List | `sunPresenceEntity AccessPermitted` | `sunPresenceAccess Permitted` | If the *default presence visibility* label (see preceding two table entries) in the admin console is unchecked (viewed by nobody), end users can enter others to this list to allow them access to presence status | The dynamic information is taken |
| Presence Agents | `sunPresenceEntity Devices` | `sunPresenceDevices` | Not used in this release (for future use) | The dynamic information is taken |

# Pre-Defined Examples of Instant Messaging and Presence Policies

Table 4-6 lists and describes the seven example policies and roles that are created in Sun ONE Identity Server when the Instant Messaging service component is installed. You can add end users to different roles according to the access control you want to give them.

A typical site might want to assign the role IM Regular User (a role that receives the default Instant Messaging and Presence access) to end users who simply use Instant Messenger, but have no responsibilities in administering Instant Messaging policies. The same site might assign the role of IM Administrator (a role associated with the ability to administer Instant Messaging and Presence services) to

particular end users with full responsibilities in administering Instant Messaging policies. Table 4-7 lists the default assignment of privileges amongst the policy attributes. If an action is not selected in a rule, the values *allow* and *deny* are not relevant as the policy then does not affect that attribute.

**Table 4-6** Default Policies and Roles for Identity Sever

| Policy | Role the Policy Applies to | Service the Policy Applies to | Policy Description |
|---|---|---|---|
| Default Instant Messaging and presence access | IM Regular User | sunIM, sunPresence | The default access that a regular Instant Messaging end user should have. |
| Ability to administer Instant Messaging and Presence Service | IM Administrator | sunIM, sunPresence | The access that an Instant Messaging Administrator has, which is access to all Instant Messaging features. |
| Ability to manage Instant Messaging news channels | IM News Administrator | sunIM | End users can manage news channels by creating, deleting, etc. |
| Ability to manage Instant Messaging conference rooms | IM Conference Rooms Administrator | sunIM | End users can manage conference rooms by creating, deleting, etc. |
| Ability to change own Instant Messaging user settings | IM Allow User Settings Role | sunIM | End users can edit settings by clicking the Setting button in theSun ONE Instant Messenger. |
| Ability to send Instant Messaging alerts | IM Allow Send Alerts Role | sunIM | End users can send alerts in Sun ONE Instant Messenger. |
| Ability to watch changes on other Instant Messaging end users | IM Allow Watch Changes Role | sunIM | End users can access the presence status of other Instant Messaging end users. |

**Table 4-7** Assignment of the Default Policies

| | Policy | | | | | | |
|---|---|---|---|---|---|---|---|
| **Attribute** | Default Instant Messaging and presence access | Ability to administer Instant Messaging and Presence Service | Ability to manage Instant Messaging news channels | Ability to manage Instant Messaging conference rooms | Ability to change own Instant Messaging end-user settings | Ability to send Instant Messaging alerts | Ability to watch changes on other Instant Messaging end-users |
| sunIMAllowChat | allow | allow | | | | | |
| sunIMAllowChatInvite | allow | allow | | | | | |
| sunIMAllowForumAccess | allow | allow | | allow | | | |
| sunIMAllowForumManage | deny | allow | | allow | | | |

**Table 4-7** Assignment of the Default Policies

| | **Policy** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Attribute** | Default Instant Messaging and presence access | Ability to administer Instant Messaging and Presence Service | Ability to manage Instant Messaging news channels | Ability to manage Instant Messaging conference rooms | Ability to change own Instant Messaging end-user settings | Ability to send Instant Messaging alerts | Ability to watch changes on other Instant Messaging end-users |
| sunIMAllowForumModerate | deny | allow | | allow | | | |
| sunIMAllowAlertsAccess | allow | allow | | | | allow | |
| sunIMAllowAlertsSend | allow | allow | | | | allow | |
| sunIMAllowNewsAccess | allow | allow | allow | | | | |
| sunIMAllowNewsManage | deny | allow | allow | | | | |
| sunIMAllowFileTransfer | allow | allow | | | | | |
| sunIMAllowContactListManage | allow | allow | | | | | |
| sunIMAllowUserSettings | allow | allow | | | allow | | |
| sunIMAllowPollingAccess | allow | allow | | | | | |
| sunIMAllowPollingSend | allow | allow | | | | | |
| sunPresenceAllowManage | allow | allow | | | | | |
| sunPresenceAllowAccess | allow | allow | | | | | allow |
| sunPresenceAllowPublish | allow | allow | | | | | |

# Creating New Instant Messaging Policies

You can create new policies to fit the specific needs of your site.

## To Create a New Policy

1. Log on to the Sun ONE Identity Server admin console at http://hostname:port/amconsole, for example http://imserver.company22.example.com:80/amconsole
2. With the Identity Management tab selected, select Policies in the View drop down list in the navigation pane (the lower-left frame).
3. Click New to bring up the New Policy page in the data pane (the lower-right frame).
4. Select Normal for the Type of Policy.

5. Enter a policy description in the Name field, such as Ability to Perform IM Task.
6. Click Create to make the name of the new policy appear on the policy list in the navigation pane and to make the page in the data pane change to the Edit page for your new policy.
7. In the Edit page, select Rules in the View drop down list to bring up the Rule Name Service Resource panel inside the Edit page.
8. Click Add to bring up the Add Rule page.
9. Select the Service that applies, either Instant Messaging Service or Presence Service.

   Each service enables you to allow or deny end users the ability to perform specific actions. For example, Ability to Chat is an action specific to the Instant Messaging service while Ability to Access other's Presence is an action specific to the Presence service.
10. Enter a description for a rule in the Rule Name field, such as Rule 1.
11. Enter the appropriate Resource Name (`IMResource` or `PresenceResource`):
    - `IMResource` for Instant Messaging Service
    - `PresenceResource` for Presence Service
12. Select the Actions that you want to apply.
13. Select the Value for each action: Allow or Deny.
14. Click Create to display this proposed rule in the list of saved rules for that policy.
15. Click Save to make this proposed rule a saved rule.
16. Repeat steps 8-15 for any additional rules that you want to apply to that policy. For each new rule, click Save to save the changes to the policy.

# Assigning Policies to a Role, Group, Organization, or User

You can assign policies—the default policies for Instant Messaging or Instant Messaging policies that might have been created after Instant Messaging was installed—to a role, group, organization, or user.

## To Assign a Policy

1. Log on to the Sun ONE Identity Server admin console at http://hostname:port/amconsole, for example http://imserver.company22.example.com:80/amconsole
2. With the Identity Management tab selected, select Policies in the View drop down list in the navigation pane (the lower-left frame).
3. Click the arrow next to the name of the policy you want to assign in order to bring up the Edit page for that policy in the data pane (the lower-right frame).
4. In the Edit page, select Subjects in the View drop down list.
5. Click Add to bring up the Add Subject page, which lists the possible subject types:
   - Identity Server Roles
   - LDAP Groups
   - LDAP Roles
   - LDAP Users
   - Organization
6. Select the subject type that matches the policy, such as Organization.
7. Click Next
8. In the Name field, enter a description of the subject.
9. If desired, select the Exclusive check box.

   The Exclusive check box is not selected as the default setting, which means that the policy applies to all members of the subject.

   Selecting the Exclusive check box applies the policy to everyone who is not a member of the subject.
10. In the Available field, search for entries that you want to add to your subject.
    a. Type a search for the entries you want to search for. The default search is *, which displays all the subjects for that subject type.
    b. Click search.
    c. Highlight entries in the Available text box that you want to add to the Selected text box.
    d. Click Add or Add All, whichever applies.

e. Repeat steps a-d until you have added all the names you want to the Selected text box.

11. Click Create to display this proposed subject in the list of saved subjects for that policy.

12. Click Save to make this proposed subject a saved subject.

13. Repeat steps 5-12 for any additional subjects that you want to add to the policy. For each new subject, click Save to save the changes to the policy.

# Creating New Suborganizations Using Identity Server

The ability to create suborganizations using Sun ONE Identity Server enables organizationally separate populations to be created within the Sun ONE Instant Messaging server. Each suborganization can be mapped to a different DNS domain. End users in one suborganization are completely isolated from those in another. The following describes minimal steps to create a new suborganization for Instant Messaging.

## To Create a New Suborganization

1. Log on to the Sun ONE Identity Server admin console at `http://hostname:port/amconsole`, for example `http://imserver.company22.example.com:80/amconsole`

2. Create a new organization:

   a. With the Identity Management tab selected, select Organizations in the View drop down list in the navigation pane (the lower-left frame).

   b. Click New to bring up the New Organization page in the data pane (the lower-right frame).

   c. Enter the following in the appropriate fields:

      - A suborganization name, such as `sub1`
      - A domain name, such as `sub1.company22.example.com`,

   d. Click Create.

3. Register services for the newly created suborganization.

**a.** Click the name for the new suborganization, such as `sub1`, in the navigation pane (Be certain to click the name, not the property arrow at the right.).

**b.** Select Services from the View drop down list in the navigation pane

**c.** Click Register to bring up the Register Services page in the data pane.

**d.** Select the following services:

Under the Authentication heading:

- Core
- LDAP

Under the Instant Messaging Configuration heading:

- Instant Messaging Service
- Presence Service

**e.** Click Register to bring up the newly selected services for this suborganization in the navigation pane.

**4.** Create service templates for the newly selected services:

**a.** In the navigation pane, click the property arrow for a service, starting with the Core service.

The Create Service Template page appears in the data pane.

**b.** In the data pane, click Create, which replaces the Create Service Template page with a page of template options for the service you have selected.

You should click Create for each service even when you do not want to modify the template options.

**c.** Modify the options for the service template of each service as follows:

**I.** **Core:** Generally, no options need to be modified; go to Step d.

**II.** **LDAP:** Perform the following actions before going to Step d:

- Add the prefix of the new suborganization to the *DN to Start User Search* field. After adding the prefix, the final DN should be in this format:

```
o=sub1,dc=company22,dc=example,dc=com
```

- Enter the LDAP password in the *Password for Root User Bind* and *Password for Root User Bind (confirm)* fields.

III. **Instant Messaging Service:** Generally, no options need to be modified; go to Step d.

IV. **Presence Service:** If you would like to make end-user presence information available to others by default (sites tend to choose this option), select the *Dynamic Default Presence Visibility* check box before going to Step d.

**d.** Click Save.

**e.** Repeat steps a through d until you have created service templates for each service.

# Adding End Users to New Suborganizations

After new end users have been created in a suborganization they need to be assigned roles. Roles can be inherited from the parent organization as described in the following section.

## To Add End Users to a New Suborganization:

1. Go to the parent organization and select Roles from the View drop down list. The specific steps are:

   **a.** Log on to the Sun ONE Identity Server admin console at `http://hostname:port/amconsole`, for example `http://imserver.company22.example.com:80/amconsole`

   **b.** With the Identity Management tab selected, select Roles in the View drop down list in the navigation pane (the lower-left frame).

2. Click on the property arrow to the right of the role you wish to assign in order to bring up a page for that role in the data pane (the lower-right frame).
3. Select Users from the View drop down list in the data pane.
4. Click Add to bring up the Add Users page.
5. Enter a matching pattern to identify users. For example, in the `UserId` field an asterisk,*, lists all users.
6. Click Filter to bring up the Select User page.
7. Display the parentage path in the Select User page:

   **a.** Select the *Show parentage path* check box.

   **b.** Click Refresh.

8. Select the users to be assigned to this role.
9. Click Submit.

# Migrating from the Instant Messaging Service of Sun ONE Instant Messaging 6.0 Server

## Non-Migration Option

If your site used the Sun ONE Instant Messaging 6.0 server with the Sun ONE Identity Server 5.1 software to deploy the Instant Messaging service, the old attributes will be honored by the Sun ONE Instant Messaging 6.1 software. Policy attributes from the Sun ONE Instant Messaging 6.0 server, such as `sunIMAllowFileTransfer` and `sunIMEnableModerator` will override the same policy attributes set in the Sun ONE Instant Messaging 6.1 server.

## Migration Option

However, the preferable method for handling the differences in the two Instant Messaging services is to migrate from the Instant Messaging service used for the Sun ONE Instant Messaging 6.0 software and to modify or create a Sun ONE Identity Server policy which uses the Instant Messaging Service and Presence Service from the Sun ONE Instant Messaging 6.1 software. You should define the new policy in such a way that it provides the same access control to your site as the old policy did.

For example, you can modify a rule in the *Default Instant Messaging and presence access* policy to set the deny or allow status of each of the policy's attributes in order for the policy to demonstrate the same behavior that it demonstrated in the Sun ONE Instant Messaging 6.0 server or you can create a new policy with rules that will allow it to behave in the same manner as it did previously.

# Migrating Access Control Files

If your site has been using an earlier version of Sun ONE Instant Messaging server (6.0 or earlier), but you have not used an Instant Messaging service—therefore, you have not set end-user privileges by setting policies through the Sun ONE Identity Server— but have instead set end-user privileges by editing access control files, two methods are available to you for replicating the policy set within the access control files and using this information to create Sun ONE Identity Server policies:

Migrate Access Control File Information Manually

Migrate Access Control File Information Automatically

## Migrate Access Control File Information Manually

The high-level steps for this method are as follows:

1. Open each access control file (one at a time). For example, `sysTopicsAdd.acl` and `sysRoomsAdd.acl`.

   For more information about the location and format of access control files, see "Managing Policies Using Access Control Files" on page 100.

2. In each file, read the value for the default line. The default line starts with the letter `d` followed by a colon (`d:`).

3. In the Sun ONE Identity Server admin console within the *Default instant messaging and presence access* policy, set a rule to the same default value you read from the access control file.

4. Assign all the regular Instant Messaging end users the role of IM Regular User

5. For end users listed in these access control files who have different privileges, such as the ability to manage conference rooms or news channels, add them to the corresponding roles that have those privileges. See Table 4-6 on page 109 for the role that each default policy applies to.

## Migrate Access Control File Information Automatically

Instead of transferring the access control file information manually, you can perform a one-time migration of this information by issuing a command.

Type the following command:

```
imadmin migrate
```

This command will transfer information from the global access control files to the corresponding policy and its subjects. See table Table 4-8 for a list of the global access control files and the policies to which they map.

**Table 4-8** Access Control Files and the Policies They Map to

| Access Control File | Policy |
|---|---|
| sysSaveUserSettings.acl | Ability to change own Instant Messaging user settings |
| sysTopicsAdd.acl | Ability to manage Instant Messaging news channels |
| sysRoomsAdd.acl | Ability to manage Instant Messaging conference rooms |
| sysSendAlerts.acl | Ability to send Instant Messaging alerts |
| sysWatch.acl | Ability to watch changes on other Instant Messaging end users |
| sysAdmin.acl | Ability to administerInstant Messaging and Presence Service |

## Migrate Sun ONE Instant Messenger Settings

For Sun ONE Instant Messaging 6.1 server, when the parameter `iim.userprops.store` is set to `ldap` in the `iim.conf` file, the Sun ONE Instant Messenger settings for end users is stored in the `sunIMUserProperties` user attribute.

If your site has used an earlier version of Sun ONE Instant Messaging server and the Sun ONE Instant Messenger settings have been stored in the `user.properties` file, after installing the Sun ONE Instant Messaging 6.1 server, the old settings will automatically be migrated to the `sunIMUserProperties` user attribute as end users log on, as long as the `iim.userprops.store` parameter is set to `ldap` in the `iim.conf` file.

When an end user first logs onto Sun ONE Instant Messaging 6.1 server, the server checks if the `sunIMUserProperties` user attribute exists and if it is storing the end user's settings. If the end user's settings are not found at that location, the server checks if a `user.properties` file exists for that end user. If the file exists, the server transfers information from the `user.properties` file to the `sunIMUserProperties` user attribute. However, if the `user.properties` file does not exist, the default Sun ONE Instant Messenger setting is the value assigned in the `sunIMUserProperties` user attribute for that end user.

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

The Instant Messaging archive captures instant messages and archives these messages in a Portal Server Search database. It enables the end 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 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 Servlet**. When the end 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 end 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.

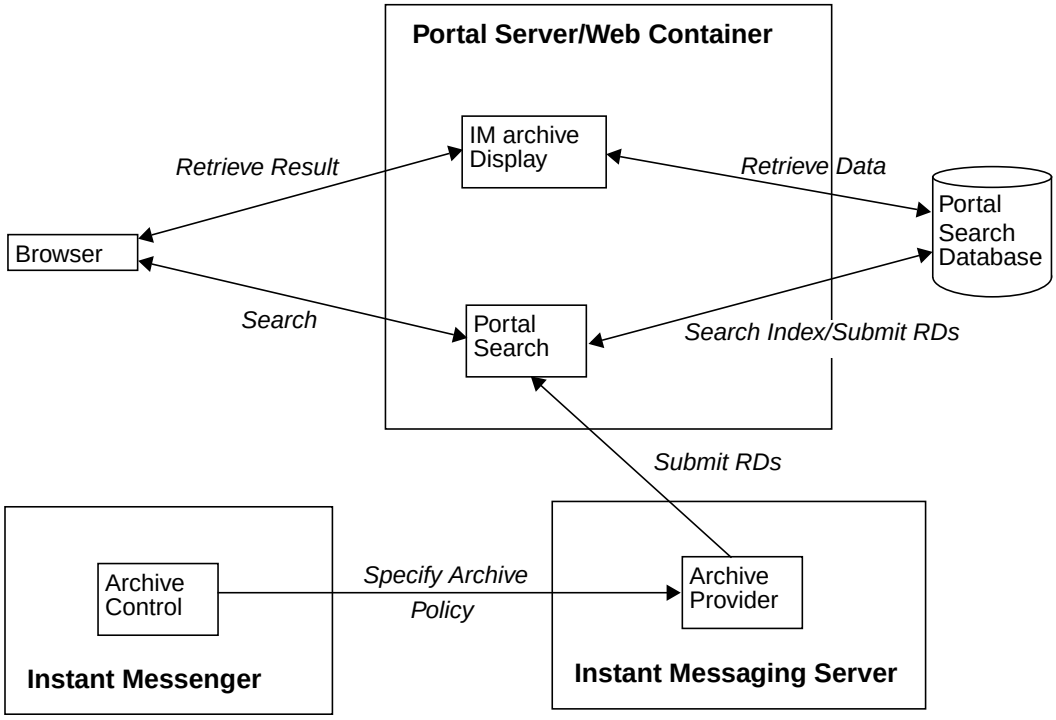
When the end user clicks the URL of the Instant Messaging resource descriptors to view the archive data, the Instant Messaging Archive Search or Display servlet is invoked. The Instant Messaging Archive Search servlet retrieves the information from the Portal Server Search database and generates a text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

**Instant Messaging Archive 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 uses Portal Server Search APIs to send these Resource Descriptors to 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 end user to set default archive options, such as “archive all conferences”, and change the default on a per-transaction basis. For example, the end 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
   ```

4. Save the file.

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

   ```
   imadmin refresh
   ```

Sun ONE Instant Messaging server provides the APIs and SPIs that can be used to write custom archive providers. For more information on Instant Messaging APIs, see "Instant Messaging APIs" on page 159.

# 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 the poll category, where *Sender* represents 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 and gives a description for each category in the archive provider.

**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 access control 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 access control file.

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

Table 5-2 lists and describes the archive provider configuration parameters that can be added to the `iim.conf` file:

**Table 5-2** Archive Provider parameters 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. |
| `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 ";" |

**Table 5-2** Archive Provider parameters added to the `iim.conf` file.

| Parameter | Default Value | Description |
|---|---|---|
| `iim_arch.readacl.adminonly` | `false` | This parameter will contain `true` or `false`.<br>`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.<br>`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. |
| `iim_arch.categories` | `all` | This parameter contains a list of message types that can be archived.<br>The value can be:<br>`poll`<br>`alert`<br>`chat`<br>`conference`<br>`news`<br>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.<br>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.<br>Note: It is not required to dedicate a category to poll messages. |

**Table 5-2** Archive Provider parameters added to the `iim.conf` file.

| Parameter | Default Value | Description |
|---|---|---|
| `iim_arch.conference.categoryname` | None | This parameter contains the name of the category containing the archived conference messages.<br>Note: It is not required to dedicate a category to conference messages. |
| `iim_arch.chat.categoryname` | Name | This parameter contains the name of the category containing the archived chat messages.<br>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.<br>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. |
| `iim_arch.ignoreexplicitdeny` | `true` | This parameter will contain `true` or `false`.<br>`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.<br>`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.<br>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 end users can also be granted NONE access. |

**Table 5-2** Archive Provider parameters added to the `iim.conf` file.

| Parameter | Default Value | Description |
|---|---|---|
| `iim_arch.portal.search` | `None` | The value of the this parameter should be the URL of the Portal Server Search servlet. For example: `http://www.example.com/portal/search`<br><br>If this parameter is not present then the Archive Provider determines the value of the Portal Server Search URL based on the `AMConfig.properties` file present on the system. |
| `iim_arch.portal.admindn` | `None` | The value of this parameter should be the dn of the admin user. For example: `uid=amadmin,ou=People,o=internet`<br><br>This parameter is required when the Document level Security in the Portal Search Server is on. |
| `iim_arch.portal.adminpassword` | `None` | The value of this parameter should be the password of the admin user as specified by the `iim_arch.portal.admindn` parameter.<br><br>This parameter is required when the Document level Security in the Portal Search Server is on. |
| `iim_arch.portal.search.database` | `None` | The value of this parameter should be the name of the database where Sun ONE Instant Messaging server stores archived messages. If this parameter is not defined then all messages are stored in the default database of Sun ONE Portal Server Search. |
| `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. |

# Storing Sun ONE Instant Messaging Archived Messages in a non-default database

To store Sun ONE Instant Messaging archived messages in a non-default Portal Server Search database instead of a default database, follow these steps:

1. Modifying the iim.conf file

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

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

   b. Open the `iim.conf` file using an editor of your choice.

   For example, you could type:

   ```
   vi iim.conf
   ```

   c. For the default archive provider, add the following line:

   **`iim_arch.portal.search.database =`** ***database-name***

   where ***database-name*** is the name of your non default database.

   d. Save the file.

2. Modify the Portal Server Search Channel

Change the Portal Server Search Channel to add an option for searching the data in another database. See the *Sun ONE Portal Server Desktop Customization Guide* for more information.

3. Modify the `IMArchiveDisplay.jsp` file:

   a. Change to the following directory:

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

   b. Create a back up of the `IMArchiveDisplay.jsp` file.

   c. Edit the `IMArchiveDisplay.jsp` file with an editor of your choice. For example, you could type the following:

   ```
   vi IMArchiveDisplay.jsp
   ```

   d. Search through the `IMArchiveDisplay.jsp` file and locate the following two lines of code:

**Code Example 5-1** Search Code from `IMArchiveDisplay.jsp` File, Before Editing

```
<search:setQuery query = "<%= scope %>"/>
<search:setRDMType rdmType = "rd-request"/>
```

**e.** Between the two lines of code shown in Code Example 5-1, add the following line of code:

```
<search:setDatabase database = "database-name"/>
```

After you add the new line of code, that section of code should look like Code Example 5-2:

**Code Example 5-2** Search Code from `IMArchiveDisplay.jsp` File, After Editing

```
<search:setQuery query = "<%= scope %>"/>
 <search:setDatabase database = "database-name"/>
<search:setRDMType rdmType = "rd-request"/>
```

where ***database-name*** is the name of the non-default 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 Administrator's Guide.*

# 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` *portal-server-install-dir/*`SUNWps/bin/rdmgr args`

   where *portal-server-install-dir* is the directory in which Portal Server is installed.

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

## 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` *portal-server-install-dir/*`SUNWps/bin/rdmgr -Q testing`

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

```
run-cs-cli portal-server-install-dir/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` *portal-server-install-dir*`/SUNWps/bin/rdmgr -d -Q testing`

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

```
run-cs-cli portal-server-install-dir/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 end 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 end 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 Messenger 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 LDAP 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 Instant 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 end 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 an end user joins the room “joe has joined the room” to “joe has entered the room;” perform the following:

1. Edit the `IMArchiveDisplay.jsp` file with an editor of your choice. For example, you could type the following:

   ```
   vi IMArchiveDisplay.jsp
   ```

2. Replace the code line in Code Example 5-3 with Code Example 5-4 in the file `IMArchiveDisplay.jsp`:

**Code Example 5-3** 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 5-4** 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.

# 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 archived 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"
```

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

Instant Messaging stores configuration settings in the `iim.conf` file within the instant messaging configuration directory as follows:

- On Solaris:

  `/etc/opt/SUNWiim/config/iim.conf`

- On Windows:

  *instant-messaging-installation-directory*`\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 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 lists and 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 end users who have set the option for forwarding their messages as emails or to pagers. |
| iim.instancedir | On Unix:<br>/opt<br>On Windows:<br>c:\Program Files\Sun\InstantMessaging | The installation directory root. |
| iim.instancevardir | On Solaris:<br>/var/opt/SUNWiim/default<br>On Linux:<br>/var/opt/soim/default<br>On Windows:<br>iim.conf\ | Sets the directory to contain runtime files, including the end-user profile database, logs, and other files created by the server and multiplexor at runtime. |
| iim.user | inetuser for LDAP deployments.<br>root for portal deployment. | The end-user name which the server processes run. Used only on Unix platforms. |
| iim.group | inetgroup for LDAP deployments.<br>root for portal deployment. | The group using which the server processes run. Used only on Solaris platform. |
| 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.<br>It contains the name of the charset to be used to encode the headers of the mail message sent out for offline alerts.<br>For example:<br>iim.mail.charset=iso-2022-jp |

**Table A-1** General Configuration Parameters

| Parameter | Default Value | Description |
|---|---|---|
| iim.jvm.command | /usr/j2se/bin/java | The location of the Java Runtime Executable (JRE). |
| iim.identity.basedir | /opt | The default installation directory—or what's also referred to as the base directory—for Sun ONE Identity Server. |
| 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 end-user information on a regular basis in order to eliminate old end-user information. This parameter specifies the frequency, in minutes, at which the cached end-user information is cleared. |

# User Source Configuration Parameters

Table A-2 lists and describes the user source configuration parameters.

**Table A-2** User Source Configuration Parameters

| Parameter | Default Value | Description |
|---|---|---|
| iim_ldap.host | localhost:389 | LDAP server name and port used by Sun ONE Instant Messaging server for end-user authentication. |
| iim_ldap.searchbase | o=internet | The string used as base to search for the end 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. |

**Table A-2** User Source Configuration Parameters

| Parameter | Default Value | Description |
|---|---|---|
| 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 end-user login. |
| iim_ldap.usergroupbyidsearchfilter | (\|(&(objectclass=groupofuniquenames)(uid={0}))(&(\|(objectclass=inetorgperson)(objectclass=webtopuser))(uid={0}))) | The search filter used to search for end 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 end users and groups in the directory, under the base specified by name. |
| 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 an end user. |
| iim_ldap.groupclass | groupOfUniqueNames | The LDAP class that indicates that an entry belongs to a group. |
| iim_ldap.groupbrowsefilter | (objectclass=groupofuniquenames) | The search filter used to browse all groups in the directory, under the specified search base. |

**Table A-2** User Source Configuration Parameters

| Parameter | Default Value | Description |
|---|---|---|
| `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 end users. |
| `iim_ldap.groupdisplay` | `cn` | LDAP attribute to use for display name of groups. |
| `im_ldap.useruidattr` | `uid` | LDAP attribute used as end users' UID. |
| `im_ldap.groupmemberattr` | `uniquemember` | LDAP attribute that gives the list of members of a group. |
| `iim_ldap.usermailattr` | `mail` | LDAP attribute that should contain end users' provisioned email addresses. Used when the email message sent to an offline end 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.<br>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 lists and 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`<br>On Linux: `/var/opt/soim/default/log/server.log`<br>On Windows: *instant-messaging-installation-directory*`\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`<br>On Linux: `/var/opt/soim/default/log/mux.log`<br>On Windows: *instant-messaging-installation-directory*`\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 lists and describes the Instant Messaging server configuration parameters.

**Table A-4** General Instant Messaging server Configuration Parameters

| Parameter | Default Value | Description |
|---|---|---|
| `iim_server.domainname` | *host's domain name* | The logical Instant Messaging 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 end users to other servers. This is not necessarily the Fully Qualified Domain Name of the system running the Instant Messaging server.<br>For example, if the system `iim.xyz.com` is the only 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`. |
| `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. |

**Table A-4** General Instant Messaging server Configuration Parameters *(Continued)*

| Parameter | Default Value | Description |
|---|---|---|
| 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.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. |

**Table A-4** General Instant Messaging server Configuration Parameters *(Continued)*

| Parameter | Default Value | Description |
|---|---|---|
| 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.<br>In portal deployment, Portal Server Session API provides the IM server with the ability to validate session ids sent by the client.<br>The value for this parameter can either be 0, 1,or -1.<br>0 - do not use the SSO provider (default).<br>1 - use the SSO provider first and default to LDAP when the SSO validation fails.<br>-1- use SSO provider only without attempting LDAP authentication even when the SSO validation fails.<br>The iim_server.usesso parameter is used in conjunction with the iim_server.ssoprovider parameter. |
| 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. |

**Table A-4** General Instant Messaging server Configuration Parameters *(Continued)*

| Parameter | Default Value | Description |
|---|---|---|
| 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 end-users' archive control settings can be considered.<br>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. |
| iim_server.conversion.provider | None | This parameter contains the list of Message Conversion Providers to be used for message conversion.<br>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.<br>The parameter value is in minutes. |

**Table A-4** General Instant Messaging server Configuration Parameters *(Continued)*

| Parameter | Default Value | Description |
|---|---|---|
| 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. |
| 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.secconfigdir | /etc/opt/SUNWiim/default/config | This directory contains the key and certificate databases. It usually contains the security module database. |
| iim_server.keydbprefix | None | This value should contain the key database filename prefix. The key database file name must always end with key3.db.<br>If the Key database contains a prefix, for example This-Database-key3.db, then value of this parameter is This-Database. |
| iim_server.certdbprefix | None | This value should contain the certificate database filename prefix. The certificate database file name must always end with cert7.db.<br>If the certificate database contains a prefix, for example Secret-stuff-cert7.db, then value of this parameter is Secret-stuff. |
| iim_server.secmodfile | secmod.db | This value should contain the name of the security module file. |

**Table A-4** General Instant Messaging server Configuration Parameters *(Continued)*

| Parameter | Default Value | Description |
|---|---|---|
| `iim_server.certnickname` | `Server-Cert` | This value should contain the name of the certificate you entered while installing the certificate.<br>The certificate name is case-sensitive. |
| `iim_server.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:<br>`Internal (software) Token:`*password*<br>Where *password* is the password protecting the key database. |
| `iim_server.trust_all_cert` | `false` | If this value is true than the server will trust all certificates and will also add the certificate information into the log files. |

# Multiple Server Configuration Parameters

For communication between multiple 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 Sun ONE Instant Messaging 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 Instant Messaging servers are configured in this manner, you can form a larger Instant Messaging community. Thus:

- End users on each server can communicate with end 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 lists and 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:<br>`iim_server.coservers=coserver1,coserver2`<br>or<br>`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 end users on this server to communicate to end 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`.<br>Note: The value format is `name:port` or `IPaddress:port`. |
| `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 lists and 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. |
| 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.<br>If the Key database contains a prefix, for example This-Database-key3.db, then value of this parameter is This-Database. |

**Table A-6** Multiplexor Configuration Parameters *(Continued)*

| Parameter | Default Value | Description |
|---|---|---|
| iim_mux.certdbprefix | None | This value should contain the certificate database filename prefix. The certificate database file name must always end with cert7.db.<br>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.<br>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:<br>Internal (software) Token:*password*<br>Where *password* is the password protecting the key database. |
| 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. |

Appendix B

# Instant Messaging Reference

This chapter explains the `imadmin` command used to administer Instant Messaging.

## imadmin

You can use the `imadmin` utility to start, stop, and refresh the Instant Messaging server and Multiplexor. On the Solaris platform, run `imadmin` as `root` or the end user specified during the installation.

### Requirements:

You must invoke the `imadmin` utility from the host on which Instant Messaging server is installed.

### Location:

- On Solaris: *instant-messaging-installation-directory*`/SUNWiim/sbin`
- On Windows: *instant-messaging-installation-directory*`\sbin`

Table B-1 lists and describes commands related to the `imadmin` command.

**Table B-1** The imadmin Commands with Descriptions

| 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. |

**Table B-1** The imadmin Commands with Descriptions

| Command | Description |
|---|---|
| `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. |
| `imadmin refresh multiplexor` | Refreshes only the multiplexor. |
| `imadmin migrate` | Generates Sun ONE Identity Server policies based on current policy access control files. |
| `imadmin version` | Prints version |

# Synopsis

`imadmin` [*options*] [*action*] [*component*]

# Options

Table B-2 lists and describes options (Solaris platform only) for the `imadmin` command.

**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

Table B-3 lists and describes actions performed after various `imadmin` commands are issued.

**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

Table B-4 lists and describes the components for the `imadmin` command.

**Table B-4** Components for `imadmin` Command

| Option | Description |
|---|---|
| `server` | Indicates the Instant Messaging server. |
| `multiplexor` | Indicates the 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 *instant-messaging-resource-directory*`/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 an end-user input is received and bean-specific actions are based on the end-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 end 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 an end 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 might occur during installation and deployment of Sun ONE 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

Catastrophic Error Leaves Server in an Inconsistent State

## 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.
- End 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.
- End user not found and the Instant Messaging server cannot find the end 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 end 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.

## Catastrophic Error Leaves Server in an Inconsistent State

If a catastrophic error occurs while installing or uninstalling Sun ONE Instant Messaging, the system might be left in an inconsistent state. This results in both install and uninstall being unable to complete. In this circumstance, you must manually remove all the Sun ONE Instant Messaging components so that a fresh install can be attempted. The clean up procedure consists of removing packages and registry information.

1. Back up any information you might need in a future installation. See "Backing Up Instant Messaging Data" on page 73.
2. Manually edit the product registry information.
   - For Solaris 9, issue the following command:

     **`prodreg(1)`**

   - For all other systems:

   a. Access and edit the `productregistry` XML file from the following locations:

      - Solaris: `/var/sadm/install/productregistry`

- Linux: `/var/tmp/productregistry`
- Windows: `%SystemRoot%/system32/productregistry`

**b.** From the preceding files, perform the following:

on all platforms, remove

- all Sun ONE Instant Messaging XML elements:

on Unix, remove the following packages or RPMs if they are still present:

- SUNWiim
- SUNWiimm
- SUNWiimd
- SUNWiimid
- SUNWiimc
- SUNWiimjd

on Windows, remove the following registry key and its subkeys:

```
HKEY_LOCAL_MACHINE\\Software\\Sun Microsystems\\Instant
Messaging\\6.
```

Appendix E

# Legacy Instant Messaging Service 6.0

This service is deprecated. If you had Sun ONE Instant Messaging software, and are now installing version 6.1, you might want to be aware that the service attributes listed in this section are still effective and supersede version 6.1 attributes.

When you deploy Sun ONE Instant Messaging server 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 server.

For more information on policies managed through Sun ONE Identity Server, see "Managing Policies using Sun ONE Identity Server" on page 103.

Table E-1 lists and describes the Instant Messaging service attributes.

**Table E-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. |
| `sunIMEnableModerator` | This is a boolean attribute. It enables the moderated conference feature in Sun ONE Instant Messenger. By default, this attributed is enabled. |

**Table E-1** Instant Messaging Service Attributes

| Service Attributes | Description |
|---|---|
| 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. |

# Index

## A



## B



## C

## D



## E



## F



## G

# I



# J



# L



# M

# N



# P



# S

## U



## W

W