Oracle® Communications Convergence

System Administrator's Guide

Release 3.0.1

E56610-01

May 2015

ORACLE®

Oracle Communications Convergence System Administrator's Guide, Release 3.0.1

E56610-01

Contents

4 Enabling Core Services for Convergence

5 Mail Service Administration

6 Address Book Service Administration

7 Calendar Service Administration

8 Instant Messaging Service Administration

9 Configuring Convergence to Use Proxy Authentication

10 Convergence Properties Reference

11 Monitoring Convergence

12 Troubleshooting Convergence

13 Overview of Add-on Services in Convergence

14 Tuning GlassFish Server to Enhance Convergence Performance

15 Setting Up Multiple Corporate Directories

A ExpiresFilter.java Reference

Preface

This guide explains how to administer Oracle Communications Convergence and its accompanying software components.

Audience

This document is intended for Convergence system administrators. This guide assumes that you have a working knowledge of the following concepts:

- Oracle GlassFish Server administration
- Directory server management
- Structure and use of a lightweight directory access protocol (LDAP) directory
- Secure Sockets Layer (SSL) for a secured communications
- System administration and networking
- General deployment architecture

Related Documents

For more information, see the following documents:

- *Convergence Installation and Configuration Guide*: Describes the requirements for installing Convergence.
- *Convergence Security Guide*: Describes how to install and configure Convergence in a secure configuration.
- *Convergence Release Notes*: Describes any known issues for Convergence.

Document Revision History

The following table lists the revision history for this guide.

Version	Date	Description
E56610-01	May 2015	3.0.1 GA release.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1

Overview of Convergence

This chapter provides an overview of Oracle Communications Convergence.

Directory Placeholders Used in This Guide

Table 1–1 lists the directory placeholders used in this guide:

Table 1–1 Convergence Directory Placeholders

Placeholder	Description
GlassFish_Home	The directory in which the GlassFish Server software is installed. For example: **/opt/glassfish3/glassfish**.
Convergence_Domain	The directory containing the configuration files for the domain in which Convergence is Installed. *Convergence_Domain* is created in *GlassFish_Home***/domains**. By default, *Convergence_Home* is *GlassFish_Home***/domains/domain1**.
Convergence_Home	Specifies the installation location for the Convergence software. The default is **/opt/sun/comms/iwc**.
c11n_Home	The directory in which all Convergence customization files and directories are created. *c11n_Home* must be *Convergence_Domain***/docroot/iwc_static/c11n**.

Default Paths and File Names

Table 1–2 provides platform-specific information about the directories that are created when you install Convergence.

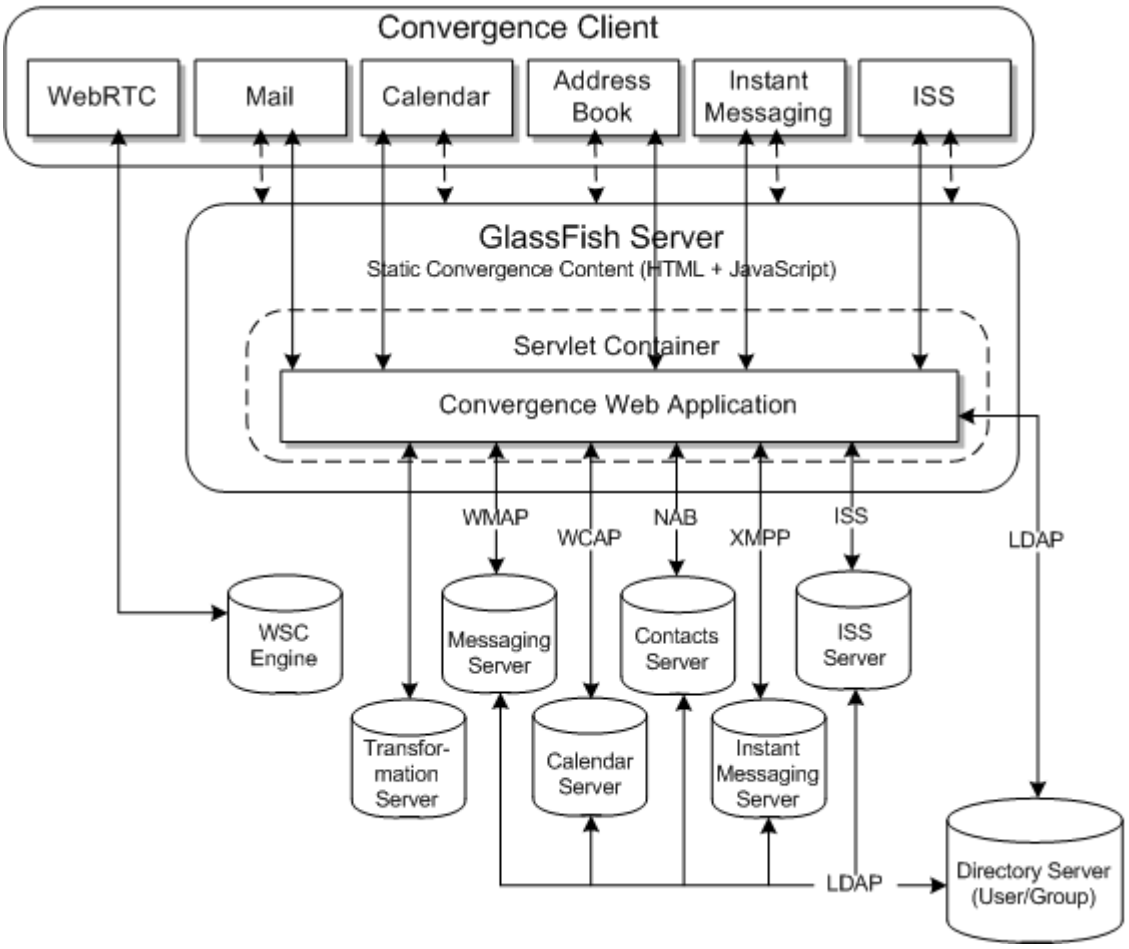
Table 1–2 Platform Convergence Directories

Description	Oracle Solaris	Red Hat Linux
Installation Directory	**/opt/sun/comms/iwc**	**/opt/sun/comms/iwc**
Data Directory	**/var/opt/sun/comms/iwc/**	**/var/opt/sun/comms/iwc/**
Binary Directory	**/opt/sun/comms/iwc/sbin**	**/opt/sun/comms/iwc/sbin**

High Level Architecture

Convergence is a web-based Java application deployed on a server in Oracle GlassFish Server. Figure 1–1 describes the Convergence architecture.

Figure 1–1 Convergence High-Level Architecture



Web Client Component

In Figure 1–1, the client component, Convergence, is a browser that uses Dojo to provide the basic infrastructure for the client components. The web client user interface provides features such as virtual list box, drag-and-drop, context menus, type ahead for address look up, and flexible layout and resizing. The client component retrieves data from the server by using protocol commands based on the AJAX technology. The client module also provides API modules for extension and customization the client.

Web Server Static Client Component

Convergence uses static content files such as HTML, CSS, and JavaScript. These static files are deployed on a web container in the *Convergence_Domain*/**docroot** directory.

Server Components

The server components of Convergence are deployed as a web application on Oracle GlassFish Server. The server components reside in the application and interact with the back-end services such as Directory Server for authentication and user preferences, Oracle Communications Messaging Server for mail-related services, Oracle Communications Calendar Server for loading the user's calendar, and Oracle Communications Instant Messaging Server for instant messaging. The Convergence

server is a servlets-based implementation. The server provides the services that are used by the client to render data on the browser. The client API communicates with the services to fetch this data.

Convergence provides the following core services:

- Authentication and authorization
- Session management and Single Sign-On (SSO)
- Protocol service
- Configuration management (XML configuration files and command line interface)
- Proxy services for mail, calendar data from back-end communications servers
- Centralized and secure request management
- Logging and basic monitoring of activities
- HTTP binding to XMPP service for instant messaging

2

Using the Convergence Administration Utility

This chapter provides an overview of how to use the Oracle Communications Convergence administration command-line utility to administer Oracle Communications Convergence.

About the Convergence Administration Utility

You can use the Convergence **iwcadmin** command-line utility to perform administrative tasks in Convergence. The following are some of the reasons you would want to use the command-line utility:

- During the initial runtime configuration, you created the runtime environment for your deployment using the initial configuration utility. While some of the many possible properties were set from your choices, many of the configuration properties were merely given default values that might not be right for your site. You can use the administration utility to change those values to ones appropriate to your site.
- In time you will need to make various changes to the configuration to accommodate changing business needs, including day-to day-operations. The configuration utility enables you to change the properties to suit your needs.
- The utility validates the values you specify. It confirms that your new values are of the proper data types, and fall within the range of valid values, if appropriate.

Administration Utility Options

The **iwcadmin** command reads or writes single or multiple configuration file properties. When the utility writes to the configuration file, it performs a validity check on the value that you provide for the property. The validity check validates data types, value limits, and ranges. The **iwcadmin** command exists in the *Convergence_Home*/**sbin** directory.

You can use the **iwcadmin** command only on the local machine on which Convergence is installed.

You must restart GlassFish server if you make any configuration changes using the **iwcadmin** command.

Command-Line Utility Syntax

Use the **iwcadmin** command to display its usage and syntax:

```
iwcadmin -h
```

The following example shows the usage and syntax of the **iwcadmin** command:

```
USAGE: iwcadmin [-p port] [-s] [-o param_name [-v param_value]] [-l [group_name]] [-f
config_params_file] [-V]
    -u    --admin          userID of user authorized to make iwcadmin updates. Optional parameter.
If -u is not specified, userID (default: admin)
                           is pulled from the iwcadmin.properties file.
    -p    --port           Administration port of the server.
    -s    --secure         Use a secure connection (HTTPS).
    -o    --option         The configuration parameter name to read or write.
    -v    --value          The value to be set. Must be used with the -o option and must be
specified immediately after the -o option.
    -l    --list           List all the configuration parameters and their values.
    -f    --file           The file from which to read configuration property/value pairs.
    -V    --version        Display the version information of the product.
    -h    --help           Display this message.
```

- To read the value of a property:

  ```
  iwcadmin [-p port] [-s] -o option_name
  ```

- To write the value of a property:

  ```
  iwcadmin [-p port] [-s] -o option_name -v option_value
  ```

- To update multiple properties:

  ```
  iwcadmin [-p port] [-s] -f path/filename
  ```

- To read the value of all configuration properties:

  ```
  iwcadmin [-p port] [-s] -l
  ```

- To get information about configuration properties:

  ```
  iwcadmin [-p port] [-s] -o config_parameter_name -h
  ```

- To get information about configuration properties for a specific module:

  ```
  iwcadmin [-p port] [-s] -l group_name
  ```

  This option is useful when you want to see the values of the configuration parameters for a specific group

  For example:

  ```
  iwcadmin -l mail
  mail.cookiename = webmailsid
  mail.enable = true
  mail.enablessl = false
  mail.host = siroe.com
  mail.port = 8990
  mail.proxyadminid = admin
  mail.proxyadminpwd = r6iwhIcDUL6r69vu2Jt24A==
  mail.requesttimeout =
  mail.spam.enableaction =
  mail.spam.folder =
  mail.uwcsievecompatible = true
  ```

  Where *group_name* is the name of the group for which you want to list the parameters. To get a list of the groups available in the Convergence deployment, use the **-h** option.

For example:

```
iwcadmin -l -h
List all the configuration parameters and their values. Optionally takes group
name as an argument and lists the parameters that belong to the given group.
Available groups: base, ugldap, auth, mail, log, cal, caldav, nab, ISS, ab,
client, admin, sso, im, smime, user, ens, notify, oin
```

- To get the current version of the software:

  ```
  iwcadmin -V
  ```

Note: If you use **tcsh** and enter an **iwcadmin** parameter enclosed in curly braces {}, you must escape the braces by preceding each brace with a backslash (\).

For example:

```
\{ ... \}
```

Managing Security of Passwords

When using the **iwcadmin** command, you cannot include the `-W password_file` parameter unless the password file is encrypted. For this reason, the `-W` parameter is omitted from all examples in this guide.

Use the following command to retrieve an encrypt password:

```
iwcadmin -o admin.adminpwd
```

If you exclude the `-W password_file` parameter from your commands, the command-line utility asks you to provide your password.

Command-line Options

Table 2–1 lists all the command-line syntax options.

Table 2–1 Options for Configuration Utility for Convergence

Option	Long name	Description
-p	--port	Administration port on which the server listens.
-s	--secure	Optional. Ensures a secure connection.
-o	--option	Configuration property name to read or write. If you do not specify the `-v` option, the utility performs a read operation. If this option is specified in the same command with the `-f` option, the `-f` option is ignored. The `-o` option takes precedence.
-v	--value	Value to be set. Use with the `-o` option and must be specified immediately after the `-o` option.
-f	--file	The file that contains the property name value pairs. This file contains multiple pairs of properties and their values. It enables an administrator to update multiple properties in a batch mode using a single command. The format of the file is a list of option and value pairs (separated by **=**), and a line return between options.
-V	--version	Version information of the product.
-l	--list	This option has no values. It retrieves all existing configuration parameters and displays them. Optionally, this parameter also takes a group name as an argument and lists the parameters that belong to the given group.
-h	--help	Help to use this utility.

You can obtain details about the configuration parameters that you can use with the `-o` option by using this option along with the `-h` option. Before setting a configuration parameter value, you can learn about the parameter usage, the functionality, and the supported data type.

The following syntax shows the usage of the `-o` option with the `-h` option:

```
iwcadmin [-p port] [-s] -o config_params_name -h
```

The `-h` option displays the following configuration parameter details:

- Option Name: Name of the configuration parameter.
- Description: Short description.
- Syntax: Input data type.
- Allowed Pattern: Accepted parameter pattern or range of values.
- Current Value: Current value of this parameter in the Convergence deployment.

The following example displays help for the **user.mail.blockimages** configuration parameter:

```
iwcadmin -o user.mail.blockimages -h
Option Name: user.mail.blockimages
Description: Specifies if images in the incoming mail should be shown or blocked
Syntax: boolean
Current Value: false
```

Setting and Unsetting Configuration Parameters

You can set or unset configuration parameters in Convergence. If a parameter does not require mandatory values, you can unset the parameter by setting its value to a blank string. You cannot unset parameters that require mandatory values.

For example, to unset the **ab.pstore.[psidentifier1].ldaphost** parameter, type the following command:

```
iwcadmin -o ab.pstore.[psidentifier1].ldaphost -v ""
```

This parameter is unset in the configuration.

To set a parameter, type the following command:

```
iwcadmin -o ab.pstore.[psidentifier1].ldaphost -v "ldap_host_name"
```

The **iwcadmin** command checks whether the parameter that you set is valid and has acceptable values.

Running the Administration Utility in Batch Mode

To update multiple attributes or configuration parameters in your deployment, invoke the **iwcadmin** command in batch mode. The **-f** parameter in the **iwcadmin** command enables you to set multiple parameters in a file by invoking the command only once.

To run the **iwcadmin** command in the batch mode:

1. Create a file with the name-value pairs for the options that you want to set. For example, the following entries in a file set the log level for all the log related modules in Convergence to the **DEBUG** level and the log rotation policy to **2048** bytes.

   ```
   log.ADDRESS_BOOK.level = DEBUG
   ```

```
log.ADMIN.level = DEBUG
log.AUTH.level = DEBUG
log.CONFIG.level = DEBUG
log.DEFAULT.level = DEBUG
log.PROTOCOL.level = DEBUG
log.PROXY_CAL.level = DEBUG
log.PROXY_MAIL.level = DEBUG
log.SIEVE.level = DEBUG
log.sizetriggerval = 2048
```

In this example, the left hand side option is the name of the parameter that you want to set and the right hand side string is the value that you want to set it to.

2. Save the file at an appropriate location. For example, **/tmp/logLevelSetting**.
3. Type the **iwcadmin** command with the **-f** option and provide the path to the file:

```
iwcadmin -f /tmp/logLevelSetting
```

3

Convergence Administrative Tasks

This chapter explains several administrative tasks for Oracle Communications Convergence.

Authentication

This section describes administrative tasks related to authentication.

See also *Convergence Security Guide* for information about certificate-based authentication.

Setting Up Account for End User

To set up Convergence UI login for end users, evaluate if you want to use:

- UID (default), or
- Email Address Login (LDAP mail attribute)

The procedures for setting up email address login which uses the LDAP **mail** attribute are the following:

- Constructing a Filter for Email Address Login
- Enabling Email Address Login on Convergence Server
- Activating mailAlternateAddress (optional)

Constructing a Filter for Email Address Login

In order to create a filter for email address login, you need the **uid** and **mail** attributes.

The **mail** attribute identifies the primary email address for a user, calendar group, or calendar resource. This is the email address retrieved and displayed by lookup applications.

Table 3–1 lists the variables used in constructing the filter.

Table 3–1 Mail Attribute Filter Variables

Variable	Description
%U	Name part of the login name (that is, everything before the login separator stored in the servers configuration).
%V	Domain part of the login string.
%o	Original login ID entered by the user.

For more information on LDAP attributes, specifically, **inetDomainSearchFilter**, see the discussion about LDAP object classes and attributes in your Oracle Communications Messaging Server and Oracle Communications Calendar Server documentation.

Enabling Email Address Login on Convergence Server

To set up email address login, enable it on the Convergence Server:

```
iwcadmin -o ugldap.ugfilter -v "(|(uid=%U)(mail=%o))"
```

See "Convergence Properties Reference" for information on **ugldap.ugfilter**.

Activating mailAlternateAddress (optional)

mailAlternateAddress is the alternate RFC 822 email address of this recipient. A filter similar to **mail** can be performed on **mailalternateaddress**:

```
iwcadmin -o ugldap.ugfilter -v "(|(uid=%U)(mail=%o)(mailalternateaddress=%o))"
```

Configuring LDAP Authentication in Convergence

LDAP authentication is enabled by default when you configure Convergence. You can use separate LDAP servers to store authentication information and user preferences. By default, Convergence uses UG LDAP as the authentication LDAP server. You can enable LDAP authentication by using the following command line option:

```
iwcadmin -o auth.ldap.enable -v true
```

Configuring Convergence for Multiple Directory Servers

You can configure Convergence to use a separate directory server for user authentication and another for user/group information.

When LDAP authentication module is configured for authentication, the LDAP authentication module, by default, uses the UG LDAP for authentication. If you use separate LDAP servers for storing the authentication information and user preferences, the schema type and user trees should match in both the LDAP stores.

To enable your site to use a separate LDAP server for authentication, you must set the following configuration parameters.

- **auth.ldap.enable** - Set this parameter to **true**.
- **auth.ldap.schemaversion** - Set this parameter to the schema version that you are using for the UG LDAP. The schema versions for the UG LDAP and authentication LDAP must be the same.
- **auth.ldap.dcroot** - DC (Domain Component) or user tree root node in the LDAP. This should be the same value as in the UG LDAP.
- **auth.ldap.host** - Host name of the authentication LDAP server.
- **auth.ldap.enablessl** - Set this parameter to **true** or **false** to enable or disable SSL.
- **auth.ldap.port** - Port number that the LDAP server listens to. If the LDAP server is configured in SSL mode, you must provide the SSL port.
- **auth.ldap.minpool** - Minimum number of connections that you want to have when the LDAP pool is initialized.
- **auth.ldap.maxpool** - Maximum number of connections that you want to have when the LDAP pool is initialized.

- **auth.ldap.timeout** - Set this to the maximum number seconds that the LDAP server should wait for returning search results before aborting the search.
- **auth.ldap.binddn** - The Bind DN of the user. The LDAP server privilege user ID. For example, **cn=DirectoryManager**.
- **auth.ldap.bindpwd** - The bind DN user password.

You can set the parameters in batch mode. See "Running the Administration Utility in Batch Mode".

The following configuration parameter can be set when the administrator needs to customize default values.

```
iwcadmin -o auth.ldap.ugfilter -v user_group_filter
```

This should result in unique user entry under given domain/organization. For example,**(|(uid=%U)(mail=%o))** otherwise it will cause unexpected results. If not set (uid=%U) will be used as default value.

Configuring LDAP Over SSL

If you use the same LDAP server, both for authentication and storing user preferences, you must set the **ugldap.enablessl** and **ugldap.port** configuration parameters by using the **iwcadmin** command.

```
iwcadmin -o ugldap.enablessl -v true
iwcadmin -o ugldap.port -v user_group_ldap_port
```

If your deployment uses an LDAP server other than the User/Group LDAP for authentication, you must set the following parameters by using the **iwcadmin** command:

```
iwcadmin -o auth.ldap.enablessl -v true
iwcadmin -o auth.ldap.port -v ldapport
```

Logging

Convergence creates log files that records events, status of various software components, system errors, and other aspects of the server such as session, IP addresses and so on. By examining the log files, you can monitor the server's operation.

The following are the components of Convergence for which you can set logging information.

- Address Book (ADDRESS_BOOK)
- Administration (ADMIN)
- Authentication (AUTH)
- Configuration (CONFIG)
- Default (DEFAULT)
- Event Notification System (ENS)
- Notify (NOTIFY)
- Protocol (PROTOCOL)
- Calendar Proxy (PROXY_CAL)

- Indexing and Search Service Proxy (PROXY_ISS)
- Mail Proxy (PROXY_MAIL)
- Network Address Book Proxy (PROXY_NAB)
- Outside In Proxy (PROXY_OIN)
- SIEVE filters (SIEVE)

For each component, you can set a log level. The existing log levels are described in "About Log Levels". To see the list of components for which logging can be enabled, use the following command:

```
iwcadmin -l | grep log.*.level

log.ADDRESS_BOOK.level = DEBUG
log.ADMIN.level = DEBUG
log.AUTH.level = DEBUG
log.CONFIG.level = DEBUG
log.DEFAULT.level = DEBUG
log.ENS.level = DEBUG
log.NOTIFY.level = DEBUG
log.PROTOCOL.level = DEBUG
log.PROXY_CAL.level = DEBUG
log.PROXY_ISS.level = DEBUG
log.PROXY_MAIL.level = DEBUG
log.PROXY_NAB.level = DEBUG
log.PROXY_OIN.level = DEBUG
log.SIEVE.level = DEBUG
```

Enabling Logging

Communication Center uses a set of loggers for various components of the server. You can enable and set log levels for each of the components by using the **iwcadmin** command.

For example, the following command sets the Address Book logging to the level **INFO**.

```
iwcadmin -o log.ADDRESS_BOOK.level -v INFO
```

About Log Levels

Convergence uses Apache Log4j as its underlying logging framework. All the log levels that Log4j offers are available in Convergence. The following log levels are available:

- OFF
- ERROR
- WARN
- INFO
- DEBUG

Specify the Log File Location

You can specify the following log locations:

- Application log location: All log information generated by the server are sent to the application log. This log file contains information about the behavior of the application.
- Administration log location: All log information that is generated by the **iwcadmin** command are sent to the administration log location.

To set log information for the application logger, type the following command:

```
iwcadmin -o log.location -v /data/logs/file.log
```

where *file* is the name you choose for the log file.

To set the logging information for the administration logger, use the following command:

```
iwcadmin -o log.adminloglocation -v /data/logs/file.log
```

About Log Rotation Policies

Log rotation is an approach to manage log files by renaming the existing log file and creating a new log file. All the log messages generated after creating the new file is written in this new log file.

Convergence supports log rotation based on size or time. Size-based log rotation is triggered when the log file reaches a specified size in kb (kilobytes). Time based log rotation is triggered based on the date pattern specified by the administrator.

This example shows how to set size based log rotation:

```
iwcadmin -o log.sizetriggerval -v 102400
```

This example shows how to set time based log rotation policy:

```
iwcadmin -o log.timetriggerval -v "'.'yyyy-MM"
```

For more information about frequency patterns for time based log rotation, see the apache web site:

http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/DailyRollingFileAppender.html

Logging User IP Address and Session Tracking Information

To log IP address and session tracking information, you must modify the log pattern to include the IP address and session ID of a user so that these get added into the log file. Type the following command:

```
iwcadmin -o log.pattern -v '%c: %p from %C  Thread %t ipaddress=%X{ipaddress}
sessionid=%X{sessionid} at %d - %m %n'
iwcadmin -o log.enableusertrace -v true
```

Modify the log-pattern to include the user IP address (*%X{ipaddress}*) and session id (*%X{sessionid}*) in the log messages.

Note: If the GlassFish Server hosting Convergence resides behind a front-end reverse proxy or load balancer (web server), this front-end's IP address is captured, not the browser's IP address. To overcome this situation, use the following command to set the **auth-pass-through-enabled** parameter to **true** on the GlassFish Server.

```
asadmin set
server-config.network-config.protocols.protocol.http-listener-1.htt
p.auth-pass-through-enabled=true
```

In case you are using a reverse proxy in front of Convergence, you have to configure that reverse proxy to put the original client IP address into an HTTP Header that must be called **proxy-ip**.

If you have set **auth-pass-through-enabled** to **true**, then your load balancer or reverse proxy must be passing the IP address to the client. If you do not configure the load balancer or reverse proxy in this manner, or if you bypass the load balancer, you will not be able to log into Convergence.

GlassFish Load Balancer plug-in automatically adds client original IP address to HTTP Header **proxy-ip**.

See your Oracle GlassFish Server documentation for more information.

Sample Convergence Logging Session

The following example shows a typical logging session:

```
PROTOCOL: DEBUG from com.sun.comms.client.web.IwcCookieManager Thread
httpSSLWorkerThread-80-23 ipaddress=198.51.100.0 sessionid= at 23:08:31,920-
cleaning client cookies: webmailcookiename is webmailsid
PROTOCOL: DEBUG from com.sun.comms.client.web.IwcCookieManager Thread
httpSSLWorkerThread-80-23 ipaddress=198.51.100.0 sessionid= at 23:08:31,920-
cleaning client cookies: webmailcookiepath is /
PROTOCOL: DEBUG from com.sun.comms.client.web.IwcCookieManager Thread
httpSSLWorkerThread-80-23 ipaddress=198.51.100.0 sessionid= at 23:08:31,920-
Cookie sent by client : JSESSIONID value=687380a1199c738c5165692c4587 path=null
comment=null domain=null version=0 isSecure? false maxAge=-1
PROTOCOL: DEBUG from com.sun.comms.client.web.IwcCookieManager Thread
httpSSLWorkerThread-80-23 ipaddress=198.51.100.0 sessionid= at 23:08:31,921-
Removing iwc client cookie JSESSIONID
```

These messages indicate that the user session has been invalidated by the server. There are a few reasons why a user session is invalidated:

- a logout is issued from the browser.
- a new login is initiated, but there is already active session in progress.
- the GlassFish server is shutdown. All sessions are then invalidated.

About SSL in Convergence

This section explains how to configure Convergence with SSL.

Secure connections between applications connected over the Web can be obtained by using protocols such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS).

SSL is often used to refer to either of these protocols or a combination of the two (SSL/TLS). Due to a security problem with SSLv3, Convergence recommends the use of only TLS. See *Convergence Security Guide* for information about disabling SSLv3. However, throughout this guide, secure communications may be referred to by the generic term SSL.

Configuring SSL in Convergence

SSL provides a secure means of communication between the web-browser client and the server. You can enable SSL in Convergence in two ways:

- At the time of configuring Convergence, or
- By setting the SSL configuration parameters after configuration.

To enable Convergence to use SSL, you must enable SSL at the GlassFish Server level and also set the **base.sslport** configuration parameters using the **iwcadmin** command.

For **base.sslport** properties, refer to "Convergence Properties Reference".

```
iwcadmin -o base.sslport -v ssl_port
```

Configuring Authentication Only SSL in Convergence

Authentication-Only SSL is a mechanism in which users are authenticated by using the HTTPS protocol which prevents user authentication details from being sent unencrypted. All other requests from the client are performed using the HTTP protocol. To configure Convergence to use Authentication only SSL, you must set both the **base.sslport** to the GlassFish Server SSL port value, and the **base.enableauthonlyssl** value using the **iwcadmin** command. For example:

```
iwcadmin -o base.sslport -v ssl_port
iwcadmin -o base.enableauthonlyssl -v true
```

Enabling SSL for Back-End Servers

Use the **iwcadmin** command to enable SSL between Convergence and back-end servers.

Enabling SSL for Messaging Server

To enable SSL for mail server, set the **mail.enable** and **mail.port** configuration parameters.

```
iwcadmin -o mail.enablessl -v true
iwcadmin -o mail.port -v mail_port
```

Where *mail_port* is the SSL port open on the Messaging server.

Enabling SSL for Calendar Server

To enable SSL for Calendar Server 6.3, set the **cal.enablessl** and **cal.port** configuration properties.

```
iwcadmin -o cal.enablessl -v true
iwcadmin -o cal.port -v cal_port
```

Where *cal_port* is the SSL port open on the Calendar Server.

To enable SSL for Calendar Server 7, set the **caldav.enablessl** and **caldav.port** configuration properties.

```
iwcadmin -o caldav.enablessl -v true
iwcadmin -o caldav.port -v caldav_port
```

Where *caldav_port* is the SSL port open on the Calendar Server.

Enabling SSL for Convergence Address Book

The address book service provide by Convergence is a part of Convergence. To enable SSL for the address book service, enable SSL for Convergence.

Enabling SSL for Instant Messaging

To enable SSL between Convergence and Instant Messaging Server, you must enable TLS/SSL in Instant Messaging Server. No configuration changes are required for Convergence. See *Instant Messaging Server System Administrator's Guide* for more information.

Enabling SSL for Contacts Server Address Book

To enable SSL for Contacts Server, set the **nab.enablessl** and **nab.port** configuration properties.

```
iwcadmin -o nab.enablessl -v true
iwcadmin -o nab.port -v nab_port
```

Where *nab_port* is the SSL port open on the Contacts Server.

Enabling SSL for Indexing and Search Service

To enable SSL for Indexing and Search Service, set the **iss.enablessl** and **iss.port** configuration properties.

```
iwcadmin -o ISS.enablessl -v true
iwcadmin -o ISS.port -v iss_port
```

Where *iss_port* is the SSL port open on the Indexing and Search Service server.

Enabling SSL for LDAP

To enable SSL between Convergence and the directory server, set the **ugldap.enablessl** and **ugldap.port** configuration properties.

```
iwcadmin -o ugldap.enablessl -v true
iwcadmin -o ugldap.port -v ldap_port
```

Where *ldap_port* is the SSL port open on the directory server.

Redirecting Convergence HTTP Sessions to HTTPS

Oracle recommends setting up automatic HTTPS redirection on the Convergence web container. See the container documentation for the preferred approach for setting up HTTPS redirection.

Enabling HTTP Strict Transport Security

You can enable HTTP strict transport security (HSTS) to protect the Convergence server from dynamic content accessed in Convergence (for example, in an email). HSTS requires at least one successful HTTPS request, otherwise the HSTS header is ignored.

By default, HSTS is not enforced.

Use the **iwcadmin** command to set **base.hstsmaxage**:

```
iwcadmin -0 base.hstsmaxage -v duration
```

where *duration* represents the number of seconds that a host is remembered as a known HSTS host. A value of **0** disables HSTS. Any value greater than **0** enforces HSTS.

Configuring the Convergence Display Name to Map to the LDAP Display Name

You can configure the Convergence Display Name to map to LDAP *displayName*.

Configuring the Convergence Display Name

Table 3–2 lists the configuration parameters for mapping the LDAP display name to the Convergence display name.

Table 3–2 Configuration Parameters for Mapping the LDAP displayName

Parameter	Description
general.screenname	Defines the LDAP attribute used for the *screen name*, also referred to as *displayname*. Located in the **useroption-mappings.properties** file.
ScreenNameEditable	Determines if the display name in the Options page is editable or not. Default is *false*.

With these configuration parameters, you are able to modify the Convergence display name in the following ways:

1. If the LDAP *displayName* parameter does not contain a value, use the *cn* attribute as a fall back. If the user modifies the Convergence display name, then the LDAP *displayName* attribute is populated.
2. The ability to edit the Convergence display name is disabled by default. To enable it, set the following *iwcadmin* command:

   ```
   iwcadmin -o client.screennameeditable -v true
   ```

Configuring the Corporate Address Book to Perform *displayName* Search

To configure the corporate address book for search, modify the following parameters by editing the **config/templates/ab/corp-dir/xlate-inetorgperson.xml** file:

1. Search *displayName* and *cn* where *displayName* has a different LDAP attribute from *cn*. Modify:

   ```
   <entry>
   ...
       <displayname>db:your_ldap_displayname_attribute</displayname>
       <cn>db:your_ldap_cn_attribute</cn>
   ...
   </entry>
   ```

2. Search *displayName* only where *displayName* has a different LDAP attribute from *cn*. In this scenario, no modification is required.

3. Search *displayName* only where *displayName* has the same LDAP attribute *cn*. In this scenario, no modification is required.

About Single Sign-On

You can enhance Convergence security with single sign-on (SSO).

Oracle recommends that you deliver SSO functionality using Oracle Access Manager. See your Oracle Access Manager documentation for more information.

You can also configure Convergence for Trusted Circle SSO or write a custom SSO module.

Configuring Convergence for Trusted Circle SSO

To configure Convergence to use Trusted Circle SSO, you must enable the **sso.ms.enable** configuration parameter.

```
iwcadmin -o sso.ms.enable -v true
```

Enabling SSO, by default enables single sign-off.

To manually enable single sign off, enter the following command:

```
iwcadmin -o sso.enablesignoff -v true
```

Writing a Custom SSO Module

To enhance security in Convergence, you can write your own custom modules for authentication or single sign-on. See the discussion about security considerations for developers in *Convergence Security Guide* for more information

LDAP Service

This section explains the different LDAP services for Convergence.

Configuring LDAP Failover

To configure Convergence for LDAP failover, type the following command:

```
iwcadmin -o ugldap.host -v ldap1:port1,ldap2:port2
```

ldap1:port1 and **ldap2:port2** are the LDAP servers that are a part of the failover.

If your LDAP hosts are configured for SSL, all the failover LDAP servers in the failover mechanism are also in SSL mode. Each host does not have a separate SSL flag. All the LDAP servers should have the same privileged **userid** and **password**. All the LDAP servers should run in Master-Master replication mode.

Configuration Management

This section explains the administrative tasks pertaining to configuration management.

Configuring Convergence to use SSL for Configuration Management

To configure Convergence for SSL, you must first configure the Convergence server to accept SSL requests. Additionally, you must also configure the client utility: the **iwcadmin** command to communicate to the Convergence server in SSL mode.

To configure Convergence server administration for SSL:

1. Enable SSL by using the **iwcadmin** command.

   ```
   iwcadmin -o admin.enablessl -v true
   ```

2. Generate keystore and truststore using keytool.
3. Set the keystore password.

   ```
   iwcadmin -o admin.keystorepwd -v password
   ```

4. Copy keystore to the configuration and data files directory. The default location of this directory is **/var/opt/sun/comms/iwc/**.
5. Restart the GlassFish server.

The following log message appears indicates that the SSL configuration is a successful:

```
RMI connector server in SSL mode started successfully.
```

Set up the client to securely connect to Convergence. To do this, modify the following parameters in the **iwcadmin.properties** file. This file is available in the configuration and data files directory. The default path is: **/var/opt/sun/comms/iwc**.

1. Set the parameter **secure** to **true**. Optionally, you can use the -s option in the **iwcadmin** command.
2. Set the **truststorepath** parameter to the directory where you stored the trust store.
3. Set a password for truststorepasswd.

Changing Convergence Administrator Password

To change the Convergence administrator password, type the following command.

```
iwcadmin -o admin.adminpwd -v new_password
```

Deployment-Specific Customizable Client Options for Convergence

Customizing the Login URL and Page for a Specific Domain

Convergence enables you to configure multiple domains in a deployment. Users can login to a domain by typing the URL and suffix the domain name to the user name. For example, **user1@siroe.com**. On successful authentication, the domain information is extracted from the login name and the user is logged into the specific domain.

Convergence provides an alternative way for users to log in to a specific domain. For example, you can configure Convergence to display a customized login page based on the domain information. The Convergence server displays the login page by extracting the domain name from the URL and determining if it contains a known domain and presents the domain specific login screen for the domain. The user can then type the

user name and password and login to the domain. In this case the user will not have to suffix the domain name to the user name.

Consider an example where **siroe.com** is a configured domain for a Convergence deployment. When users access Convergence, the server presents a customized login page for the domain **siroe.com**. Convergence server determines this based on the value of the **client.{domain-name}.loginpage** property. To set a customized login page for a domain, set the **client.{domain-name}.loginpage** configuration property by typing the following command.

```
iwcadmin -o client.{siroe.com}.loginpage -v "/iwc_static/layout/
loginpage_siroe.html"
```

Setting the Auto Logout Time

Convergence enables you to set a time in minutes to automatically log out of the application in case of user inactivity in client and also when the user closes the application without logging out. By default, the time is set to zero and is disabled. To set a time and enable the automatic logout option, set the **client.autologouttime** configuration property by typing the following command.

```
iwcadmin -o client.autologouttime -v logout_time
```

Verifying passwords in Convergence

Convergence allows you to verify the administration passwords. Convergence stores all passwords in encrypted format during configuration. You can verify if the password you have set while configuring Convergence is correct by using the **EncryptPwd** utility. The utility takes the password that you want to verify, as the input, and provides an encrypted string. To verify the password, you must compare this encrypted string with the encrypted password string stored in the Convergence configuration file.

To verify a password:

1. Enter the following command:

   ```
   java -cp /var/opt/sun/comms/iwc/WEB-INF/lib/iwc-shared-util.jar
   com.sun.comms.shared.util.EncryptPwd
   ```

   You will be prompted to provide the encryption key.

   > **Note:** Where **/var/opt/sun/comms/iwc/WEB-INF** refers to the default deploy directory to which Convergence is deployed.

2. Type the encryption key. By default the encryption key is available in the file: **/var/opt/sun/comms/iwc/config/.ngc_enc**.

   ```
   Enter the encryption key ( To generate a new key press Enter ):
   ```

   You will be prompted to enter a string to encrypt.

3. Type the password that you guess is the right password. For example.

   ```
   Enter string to encrypt: admin123
   ```

   The password you guess is encrypted and displayed at the prompt.

   ```
   admin123 ---> rE9ZIq6H0r49RgsQrKHXsw==
   ```

4. Compare the encrypted password (rE9ZIq6H0r49RgsQrKHXsw==) with the encrypted password available in the configuration file to verify if the password you provided is correct. If the encrypted password strings match, the password you guessed is correct.
5. If the encrypted password strings do not match you can provide another string, or type **quit** to exit.

```
Enter string to encrypt: quit
Bye...
```

Creating a Directory Server User in LDAP to Manage Convergence

A user must have a minimum set of LDAP privileges to manage the LDAP tasks for a Convergence deployment. Instead of using **cn=Directory Manager**, create an administrator user with a set of privileges that can enable him to manage a Convergence installation. The following privileges must be available for the user:

- Read
- Write
- Search
- Add
- Delete
- Update

The following LDIF file contains the ACIs assignments for Schema 1 for a user named **convergenceAdminUser**.

```
# Sample for Schema 1
# Adding ACIs to DC Tree
dn: o=internet
changetype: modify
add: aci
aci: (targetattr="*") (version 3.0; acl "foo"; allow (read,search) userdn="ldap://
/uid=convergenceAdminUser, ou=people, o=siroe.sun.com,dc=siroe,dc=sun,dc=com";)

# Adding ACIs to Organization Tree
dn: dc=siroe,dc=sun,dc=com
changetype: modify
add: aci
aci: (targetattr="*") (version 3.0; acl "foo"; allow (all) userdn="ldap:///
uid=convergenceAdminUser, ou=people, o=siroe.sun.com,dc=siroe,dc=sun,dc=com";)

# Adding ACIs to Address Book BaseDN
dn: o=PiServerDb
changetype: modify
add: aci
aci: (targetattr="*") (version 3.0; acl "foo"; allow (all) userdn="ldap:///
uid=convergenceAdminUser, ou=people, o=siroe.sun.com,dc=siroe,dc=sun,dc=com";)
```

The following LDIF file contains the ACIs assignments for Schema 2 for a user named **convergenceAdminUser**:

```
# Sample for Schema 2
# Adding ACIs to Organization Tree
dn: dc=siroe,dc=sun,dc=com
changetype: modify
add: aci
```

```
aci: (targetattr="*") (version 3.0; acl "foo"; allow (all) userdn="ldap:///
uid=convergenceAdminUser, ou=people, o=siroe.sun.com,dc=siroe,dc=sun,dc=com";)

# Adding ACIs to Address Book BaseDN
dn: o=PiServerDb
changetype: modify
add: aci
aci: (targetattr="*") (version 3.0; acl "foo"; allow (all) userdn="ldap:///
uid=convergenceAdminUser, ou=people, o=siroe.sun.com,dc=siroe,dc=sun,dc=com";)
```

Using the LDAP modify command, create the user:

```
ldapmodify -h hostname -p port -D "cn=Directory Manager" -w pwd -f add_acis.ldif

modifying entry o=internet

modifying entry o=usergroup

modifying entry o=PiServerDb
```

Additionally, you must also set the **ugldap.binddn** and **ugldap.bindpwd** parameters in Convergence to reflect the user credentials:

```
iwcadmin -o ugldap.binddn -v uid=convergenceAdminUser, ou=people,
o=siroe.com,o=usergroup

iwcadmin -o ugldap.bindpwd -v ug_ldap_bindpassword
```

Configuring VLV Browsing Indexes for Directory Server

Directory Server provides a mechanism to create indexes. These indexes improve the turnaround time at the time of searching for entries in the directory server instance. You must set the following parameters to enable VLV indexes in Directory Server.

- **search_base**
- **vlv_search_filter**
- **vlv_sort_attribute**
- **vlv_scope**

> **Note:** If you have multiple back-end Directory Servers that store user group information, you must create the indexes on all the instances.

Before setting the VLV Browsing indexes, you must have information about the directory server settings. The directory server settings are available in the **dse.ldif** file under the *directory_server_root*/**config** directory. Specifically, you would need the value of the **cn** attribute. The following example shows the **dse.ldif** file:

```
dn: cn=isp,cn=ldbm database,cn=plugins,cn=config

objectClass: top
objectClass: extensibleObject
objectClass: nsBackendInstance
cn: isp
creatorsName: cn=directory manager
modifiersName: cn=directory manager
entrydn: cn=isp,cn=ldbm database,cn=plugins,cn=config
numSubordinates: 4
```

```
nsslapd-suffix: o=isp
nsslapd-cachesize: -1
nsslapd-cachememsize: 10485760
nsslapd-readonly: off
nsslapd-require-index: off
nsslapd-directory: /var/opt/SUNWdsee/dsins1/db/isp
```

Applying the VLV Browsing Index Settings

Use the **ldapmodify** command to specify the Directory Server browsing search indexes. For example:

```
ldapmodify -h directory.aus.sun.com -p 389 -D "cn=Directory Manager"
dn: cn=Browsing isp,cn=isp,cn=ldbm database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvSearch
cn: Browsing isp
vlvbase: o=aus.sun.com,o=isp
vlvscope: 2
vlvfilter: (&(mail=*)(cn=*))
aci: (targetattr="*")(version 3.0; acl "VLV for Anonymous";
allow (read,search,compare) userdn="ldap:///anyone";)

dn: cn=Sort by cn,cn=Browsing isp,cn=isp,cn=ldbm database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvIndex
cn: Sort by cn
vlvSort: cn
```

Generate the Indexes

In the previous section, we provided the information about the search indexes that we want to create for your search base. For the settings to take effect, the indexes must be generated. It is recommended that these steps should be performed during a scheduled change window. This is because the Directory Server needs to be restarted.

The following commands describes the steps to create the indexes:

1. Change directory to the directory server installation. For example:

   ```
   cd /opt/SUNWdsee/ds6/bin
   ```

2. Stop the directory server instance:

   ```
   dsadm stop /var/opt/SUNWdsee/dsins1/
   ```

3. Populate the index entries using the **dsadm reindex** command. The **reindex** option requires you to provide the **vlv_sort_attribute**, the path to the directory server instance, and the value of the user group base.

   ```
   dsadm reindex -l -t "Sort by cn" /var/opt/SUNWdsee/dsins1/ "o=isp"
   ```

4. Start the directory server instance.

   ```
   dsadm start /var/opt/SUNWdsee/dsins1/
   ```

Handling Invalid Session Redirects in Convergence

The Convergence client sends AJAX requests to communicate with the server. If these requests are redirected for any reason, you must take special care with the redirects. With AJAX requests, redirects are automatically handled by the browser. The contents of the redirected page are handed over as the AJAX response. But, when you look at the response headers, you cannot determine if the request was successful or if the request was redirected. If the request is redirected, then the application may not understand the response. As a result, you must configure Convergence to understand the contents of a redirected page.

When there is a security agent in between the Convergence client and server, problems occur when the agent intercepts every request while looking for a valid session. If the session is invalid, the request is redirected to a login page configured in security agent. Because Convergence does not understand the contents of the login page, it displays a response parsing error, such as a syntax error. To get around this problem, the security agent should redirect to a page that Convergence is able to understand, instead of redirecting to a custom login page.

Convergence expects session time out error messages to be in specific format. When the agent encounters session time out, it needs to redirect the request to a page that generates this error message instead of its login page. Sample error messages are provided in Table 3–3 and can be copied to the policy agents deployment location.

Convergence uses different protocols for each service. For Mail: the **wmap** protocol, for Calendar: the **wcap** protocol, for Address book: **wabp** protocol, and for Options: the **iwcp** protocol.

The agent should be configured to differentiate between the kinds of requests it receives and correspondingly send the error response specific to that service.

For example, if the agent receives **/iwc/svc/wmap/*** request, the error response should be as mentioned in *Convergence_Domain***/jsp/samplefiles/MailServiceErrorJSON.jsp**.

Table 3–3 lists the requests that are redirected, the URL patterns, and appropriate error responses.

Table 3–3 Requests that are Redirected, URL Patterns, and Error Responses

Service Request	URL Pattern	Redirect to File
Mail	**/iwc/svc/wmap/***	**MailServiceErrorJSON.jsp**
Calendar	**/iwc/svc/wcap/***	**CalServiceErrorJSON.jsp**
Address Book	**/iwc/svc/wabp/***	If the expected response type is JSON: **AddressBookErrorJSON.jsp**; If the expected response type is XML: **AddressBookErrorXML.jsp**
Options	**/iwc/svc/iwcp/**	**IwcProtocolErrorJSON.jsp**

4

Enabling Core Services for Convergence

You can integrate Oracle Communications Convergence with other Oracle Communications products to provide the following core services:

- Email and messaging, provided by Oracle Communications Messaging Server.
- Calendar, provided by Oracle Communications Calendar Server.
- Address book, provided by Convergence or Oracle Communications Contacts Server.
- Instant messaging, provided by Oracle Communications Instant Messaging Server.
- Indexing and search, provided by Oracle Communications Indexing and Search Service.

Convergence allows you to provide services for a specified set of users or domains. You might want to provide or disable services at the following levels:

- The entire Convergence installation
- An individual domain (or set of domains)
- An individual user (or set of users)

Enabling Services for the Entire Convergence Installation

The address book service is enabled by default. You can enable or disable any of the other services without customizing Convergence.

After you install Convergence, you must initially configure the software by running the **init-config** utility. When you run **init-config**, you can enable and configure mail, calendar, and instant messaging services for the entire installation. You can enable any combination of these services. Thus, the "default" setting for whether a service is enabled or not depends on whether you select it for configuration when you run **init-config**. See *Convergence Installation and Configuration Guide* for more information.

After the initial configuration, you can enable or disable a service for your entire Convergence deployment. This encompasses all domains in the deployment and all users under the domains.

Use the Convergence **iwcadmin** command-line utility to set the following options to either **true** or **false**:

- **mail.enable**
- **cal.enable** (for Calendar Server 6.3) or **caldav.enable** (for Calendar Server 7)
- **ab.enable** (for Convergence address book service) or **nab.enable** (for Contacts Server address book service)

- **im.enable**
- **ISS.enable**

> **Note:** In a calendar server co-existence scenario, be sure to set both **cal.enable** (Calendar 6.3) and **caldav.enable** (Calendar 7) parameters.
>
> Only one address book service parameter can be set to true, either **ab.enable** or **nab.enable**.

Enabling Services for an Individual User or Domain

To enable or disable a service for a user or domain, you must set the appropriate LDAP attributes for that service in the user entry or domain entry in the LDAP. You can use Oracle Communications Delegated Administrator to set the LDAP attributes that determine service availability.

For detailed descriptions of the Delegated Administrator command-line utility, see *Delegated Administrator System Administrator's Guide*.

Managing Service Access Through LDAP

Managing services through LDAP affects user access to Convergence and to the software products that deliver the Convergence services. This is a very different conceptual territory than controlling the services available through Convergence, the client. When you disable LDAP service attributes, user access to the software that provides service is also disabled. All clients are disabled for those users, not only Convergence.

To manage access to services in the LDAP:

1. Install and configure the Oracle Communications software that delivers services in Convergence: Messaging Server, Calendar Server, Contacts Server (optional), and Instant Messaging (optional).
2. Manage the services available to users and domains in the LDAP directory. When you change a user's access to a service in the LDAP directory, you affect that user's access to Messaging Server, Calendar Server, or Instant Messaging, no matter which clients that user may use to access these services. Similarly, when you change domain-level services in LDAP, you affect the access to services for all users in the domain.
3. Manage the services available in Convergence. This affects Convergence users only.

To enable a service for an individual domain or user, you must perform all three preceding tasks. To make a service available to one Convergence user, you must enable that service for the entire Convergence installation. See "Enabling Services for the Entire Convergence Installation" for more information.

See "Enabling Services for an Individual User or Domain" for more information.

Using the Delegated Administrator Command-Line Utility to Manage Services

You can use the Delegated Administrator **commadmin** command-line utility to manage services in Convergence.

For detailed information about the **commadmin** command, see *Delegated Administrator System Administrator's Guide*.

The following examples manage Convergence services using the **example.com** domain.

- To create a domain with the mail and calendar services:

```
commadmin domain create -D username -d example.com -n example.com -w bolton -S
mail,cal -H mailhost.example.com
```

- To add the mail and calendar services to an existing domain:

```
commadmin domain modify -D username -w bolton -n example.com -d example.com -S
mail, cal
```

- To delete the mail and calendar services from an existing domain:

```
commadmin domain delete -D username -w bolton -d example.com -n example.com -S
mail,cal
```

- To create a user with the mail and calendar services in an existing domain:

```
commadmin user create -D username -n example.com -w secret -F last_name -l
first_name -L major -W secret -S mail,cal -H mailhost.example.com
```

- To enable the mail and calendar services for an existing user:

```
commadmin user modify -D username -n example.com -w bolton -l user_name -A
description:"description" -S mail,cal -H mailhost.example.com
```

- To disable the mail and calendar services for an existing user:

```
commadmin user delete -D chris -n example.com -w bolton -l user_name -S mail,
cal
```

Using Delegated Administrator to Manage Services

In the Delegated Administrator Administration Console, you can manage services by service packages. Sets of service packages are allocated to an organization or domain, and then the service packages are assigned to individual users. The service packages provide mail and calendar services to users.

To manage domain-level or organization-level services with the Delegated Administrator Administration Console, you must log in as a top-level administrator.

For information about service packages and how to use them, see *Delegated Administrator System Administrator's Guide*.

Enabling and Disabling Services with Direct LDAP Provisioning

You can configure mail, calendar, and instant messaging services by setting the appropriate LDAP user and domain attributes. You can use direct LDAP tools or provisioning scripts (if they have been developed at your site).

LDAP Attributes for Mail Service

To enable mail service to an individual user, set the following attribute in the user's entry in the User/Group tree:

```
mailUserStatus: active
```

To disable a user's mail service, set:

```
mailUserStatus: deleted
```

To enable mail service to an individual domain, set the following attribute in the domain entry:

```
mailDomainStatus: active
```

To disable access to mail service for all users in the domain, set:

```
mailDomainStatus: deleted
```

LDAP Attributes for Calendar Service

To enable calendar service to an individual user, set the following attribute in the user's entry in the User/Group tree:

```
icsStatus: active
```

> **Note:** When the **icsStatus** attribute is used in a user entry, it must be associated with the **icsCalendarUser** object class.

To disable a user's calendar service, set:

```
icsStatus: deleted
```

To enable the calendar service to an individual domain, set the following attribute in the domain entry:

```
icsStatus: active
```

> **Note:** When the **icsStatus** attribute is used in a domain entry, it must be associated with the **icsCalendarDomain** object class.

To disable access to calendar service for all users in the domain, set:

```
icsStatus: deleted
```

LDAP Attributes for Instant Messaging Service

To enable instant messaging service to an individual user, you can use the **imadmin assign services** command, or you can add the following instant messaging object classes in the user's LDAP entry in the User/Group tree:

```
sunIMUser
sunPresenceUser
```

To disable access to instant messaging service for a user, remove the above object classes from the user's LDAP entry.

Enabling and Configuring IM Service After Initial Configuration of Convergence

To enable IM service after having configured Convergence, perform the following steps:

1. Set the **im.enable=true** using the **iwcadmin** command.
2. In **/var/opt/sun/comms/iwc/config**, edit **httpbind.conf**. Set the IM server name, domain name, component JIDs, and password for **httpbind** and **avatar**.

   Note: The component **JID** and password should match the ones specified in the **iim.conf** file that you specified when you configured Convergence with the Instant Messaging Server during the installation.

   The passwords for the **httpbind** and the avatar component must be encrypted. See "Verifying passwords in Convergence" for information on generating the encrypted password.

3. Restart the GlassFish server.
4. Type the following:

   ```
   /opt/sun/comms/im/sbin/imadmin assign_services
   ```

   to add IM object classes to the users.

5

Mail Service Administration

This chapter explains how to administer the mail service in Oracle Communications Convergence.

See "Enabling Core Services for Convergence" for information about enabling services.

About S/MIME

Support for Secure/Multipurpose Internet Mail Extension (S/MIME) 3.1 is available in Convergence. Convergence users who are set up to use S/MIME can exchange signed or encrypted messages with other users of Convergence, Microsoft Outlook Express, and Mozilla mail systems.

The Convergence online help instructs end users in how to configure and send encrypted mail.

S/MIME provides a consistent way for email users to send and receive secure MIME data, using digital signatures for authentication, message integrity and non-repudiation and encryption for privacy and data security. S/MIME version 3.1 (RFC 3851) is supported.

You can deploy a secure mail solution using Oracle Communications Messaging Server that is configured with S/MIME. Convergence users who are set up to use S/MIME can exchange signed or encrypted messages with other users of Convergence, Microsoft Outlook Express, and Mozilla mail systems. A messaging proxy can provide an additional layer of security at the firewall to further protect information assets within Messaging Server.

The Convergence client supports S/MIME with these features:

- Create a digital signature for an outgoing mail message to assure the message's recipient that the message was not tampered with and is from the person who sent it
- Encrypt an outgoing mail message to prevent anyone from viewing, changing or otherwise using the message's content before the message arrives in the recipient's mailbox
- Verify the digital signature of an incoming signed message with a process involving a certificate revocation list (CRL)
- Automatically decrypt an incoming encrypted message so the recipient can read the message's contents
- Exchange signed or encrypted messages with other users of an S/MIME compliant client such as Convergence, Communications Express Mail, and Mozilla mail systems

To properly administer S/MIME, you need to be familiar with the following concepts:

- Digitally signed email messages
- Encrypted email messages
- Local key store of a browser
- Smart cards and the software and hardware to use them
- Private-public key pairs and their certificates
- Certificate authorities (CA)
- Verifying keys and their certificates
- Certificate revocation list (CRL)

Configuring Convergence with S/MIME

To support S/MIME, you must configure and store certificate information in Messaging Server and Directory Server.

In a typical deployment, these products run on server machines separate from the clients on which Convergence is running.

Table 5–1 lists the requirements for supporting S/MIME in Convergence.

Table 5–1 Requirements for S/MIME in Convergence on Client Machines

Component	Description
Private-public keys with certificates	One or more private-public key pair with certificates. Certificates are required and they must be in standard X.509 v3 format. Obtain keys and certificates from a CA for each Convergence user who will use the S/MIME features. The keys and their certificates are stored on the client machine or on a smart card. The public keys and certificates are also stored in an LDAP directory that can be accessed by Messaging Server and Convergence. A certificate revocation list (CRL), maintained by the CA, must be part of your system if you want key certificates checked against it to further ensure that the keys are valid. See "Accessing a CRL" for more information.
Smart card software (only required when keys and certificates are stored on smart cards)	ActivIdentity ActiveClient, Version 6.2, or Litronic NetSign 215 Reader CAC Compliant
Smart card reader	Any model of smart card reading device complying with ISO 7816 supported by the client machine and smart card software.

Certificate Requirements for Using S/MIME in Convergence

The signature and encryption features are not immediately available to Convergence users after you install Messaging Server. Before a user can take advantage of S/MIME, the requirements described in this information must be met.

Private and Public Keys

At least one private and public key pair, including a certificate in standard X.509 v3 format, must be issued to each Convergence user who will use S/MIME. The certificate, used in a verification process, assures other mail users that the keys really

belong to the person who uses them. A user can have more than one key pair and associated certificate.

Keys and their certificates are issued from within your organization or purchased from a third-party vendor. Regardless of how the keys and certificates are issued, the issuing organization is referred to as a certificate authority (CA).

Key pairs and their certificates are stored in two ways:

- On a smart card

  These cards are similar to commercial credit cards and should be used and safeguarded by the mail user as they do their own credit cards. Smart cards require special card readers attached to the mail user's computer (client machine) to read the private key information. See "Keys Stored on Smart Cards" for more information.

- In a local key store on the mail user's computer (client machine)

  A mail user's browser provides the key store. The browser also provides commands to download a key pair and certificate to the key store. See "Keys Stored on the Client Machine" for more information.

Keys Stored on Smart Cards

If the private-public key pair, with its certificate, is stored on a smart card, a card reader must be properly attached to the mail user's computer. The card reading device also requires software; the device and its software are supplied by the vendor from whom you purchase this equipment.

There are actually two parts to a system with card reading capabilities. One part is the hardware card reader and it's driver. The second part is the actual card, which is usually provided by a different vendor and requires drivers for reading the cards. Not all cards are supported. Refer to "Configuring Convergence with S/MIME" to see a list of the supported smart cards (ActiveCard, now renamed ActiveIdentity, and NetSign).

When properly installed, a mail user inserts their smart card into the reading device when they want to create a digital signature for an outgoing message. After verification of their smart card password, the private key is accessible by Convergence to sign the message. See "Configuring Convergence with S/MIME" for information on supported smart cards and reading devices.

Libraries from the vendor of the smart card are required on the user's computer. See "Key Access Libraries for the Client Machines" for more information.

Keys Stored on the Client Machine

If key pairs and certificates are not stored on smart cards, they must be kept in a local key store on the mail user's computer (client machine). Their browser provides the key store and also has commands to download a key pair and certificate to the key store. The key store may be password-protected; this depends on the browser.

Libraries from the vendor of the browser are required on the user's computer to support a local key store. See "Key Access Libraries for the Client Machines" for more information.

Publish Public Keys in LDAP Directory

All public keys and certificates must also be stored to an LDAP directory, accessible by Messaging Server and Convergence. This is referred to as publishing the public keys so they are available to other mail users who are creating S/MIME messages. Public

keys of the sender and receiver are used in the encrypting-decrypting process of an encrypted message.

Public key certificates are used to validate private keys that were used for digital signatures.

See "Managing Certificates for S/MIME" for more information on using **ldapmodify** to publish the public keys and certificates.

Give Mail Users Permission to Use S/MIME

To create a signed or encrypted message, a valid Convergence user must have permission to do so. This involves using the **mailAllowedServiceAccess** or **mailDomainAllowedServiceAccess** LDAP attributes for a user's LDAP entry. These attributes can be used to include or exclude mail users from S/MIME on an individual or domain basis.

See "Granting Permission to Use S/MIME Features" for more information.

Multi-language Support

A Convergence user who only uses English for their mail messages might not be able to read an S/MIME message which contains non-Latin language characters, such as Chinese. One reason for this situation is that the Java 6 Runtime Environment (JRE) installed on the user's machine does not have the **charsets.jar** file in the **/lib** directory.

The **charsets.jar** file is not installed if the English version of JRE was downloaded using the default JRE installation process. However, **charsets.jar** is installed for all other language choices of a default installation.

To ensure that the **charsets.jar** file is installed in the **/lib** directory, alert your users to use the custom installation to install the English version of JRE. During the installation process, the user must select the "Support for Additional Languages" option.

Wildcard SSL Certificates: Not Supported

While Wildcard SSL certificates enable SSL encryption on multiple subdomains with a single certificate, there are a number of security, certificate management, compatibility, and protection issues. Therefore, Wildcard SSL certificates are *NOT* supported in Convergence.

Configuring and Sending Encrypted Mail: Instructions for Convergence End Users

This section contains information to be made available to Convergence users for the sending of encrypted email messages.

Logging In for the First Time

When mail users log in to Convergence for the first time, they encounter special prompts relating to the S/MIME applet. For example:

- If the Java Runtime Environment (JRE) is not installed, users receive a popup asking them to install it. If users want to use S/MIME, they should accept and follow the subsequent prompts.

  Also, if users desire English language support and also want to read incoming S/MIME messages that contain non-Latin characters, such as Chinese, the **charsets.jar** file must be in the **/lib** directory on their computer.

To ensure that the **charsets.jar** file is installed in the **/lib** directory, use the custom installation to install the English version of JRE. During the installation process, select the "Support for Additional Languages" option.

See "Multi-language Support" for more information.

- If the signed applet certificates have not been accepted, users receive one or more prompts to accept signed and verified certificates. If users want to use S/MIME, they should accept the certificates.

Signature and Encryption Settings

There are initial signature and encryption settings that you can set to control whether all users' outgoing messages are:

- Automatically signed, or
- Automatically encrypted, or
- Automatically signed and encrypted

The initial settings also control whether the signature and encryption check boxes located at the top of a Convergence window and in the Options - Security window are displayed as checked (feature turned on) or unchecked (feature turned off). Use the **alwaysencrypt** and **alwayssign** parameters in the **smime.conf** file to specify the initial settings.

Let your mail users know that they can change the initial settings for their mail messages. After they log in to Convergence, a user can temporarily override a setting for one message, or for all their messages on an on-going basis.

Table 5–2 summarizes the use of the check boxes.

Table 5–2 Signature and Encryption Check Boxes in Convergence

Text for Check Box	Location	What Convergence User Does
Sign Message	At the top of the Convergence Compose tab (used for composing, forwarding, or replying to a message).	■ Check the box to sign the current message. ■ Uncheck the box not to sign the current message.
Encrypt Message	At the top of the Convergence Compose tab (used for composing, forwarding, or replying to a message).	■ Check the box to encrypt the current message. ■ Uncheck the box not to encrypt the current message.
Sign all outgoing Messages	In Convergence Options - Security dialog, under the Default Sending Settings heading:	■ Check the box to sign all your messages automatically. ■ Uncheck the box not to sign all your messages automatically. **Note:** You can override the setting of "Sign all messages during send" on a message-by-message basis with the "Sign Message" check box.
Encrypt all outgoing Messages	In the Convergence Options - Security dialog, under the Default Sending Settings heading:	■ Check the box to encrypt all your messages automatically ■ Uncheck the box not to encrypt all your messages automatically. **Note:** You can override the setting of "Encrypt all messages during send" on a message-by-message basis with the "Encrypt Message" check box.

Enabling the Java Console

A variety of operating messages can be written to the Java Console by the S/MIME applet as a Convergence user processes signed and encrypted messages. The Java Console messages can be helpful when troubleshooting a problem reported by a mail user. However, operating messages are only generated when the Java Console is enabled for the user by adding a **nswmExtendedUserPrefs** attribute to the **inetMailUser** object class of their LDAP entry. For example:

```
nswmExtendedUserPrefs: mesmimedebug=on
```

Do not enable the Java Console for all mail users all the time because this significantly decreases the performance of Convergence.

Securing Internet Links With SSL

The Messaging Server supports the use of the Secure Sockets Layer (SSL) for Internet links affecting Convergence, as summarized in the following table.

For the link between Messaging Server and Convergence: securing this link with SSL requires administrative work for the Messaging Server. The Convergence user must use the HTTPS protocol, rather than HTTP, when entering the URL information for the Messaging Server in their browser. See "Securing the Link Between Messaging Server and Convergence" for more information.

For the link between Messaging Server and the S/MIME applet: When checking public keys certificates against a CRL, the S/MIME applet must communicate directly with the Messaging Server. Securing this link with SSL requires administrative work for the Messaging Server in addition to setting **sslrootcacertsurl** and **checkoverssl** in the **smime.conf** file. See "Securing the Link Between the Messaging Server and S/MIME Applet" for more information.

Securing the Link Between Messaging Server and Convergence

The Messaging Server supports the use of SSL for the Internet link between it and Convergence. Once you have set up Messaging Server for SSL, configure Convergence for SSL. See "About SSL in Convergence" for more information.

A Convergence user specifies the Convergence URL in their browser with the HTTPS protocol:

```
HTTPS://hostname.domain:SSL_port
```

instead of the HTTP protocol (HTTP://hostname.domain:port). When the Convergence login window displays, the user sees a lock icon in a locked position at the bottom of their window to indicate they have a secure link.

See *Messaging Server System Administrator's Guide* for information about configuring encryption and certificate-based authentication for SSL.

Securing the Link Between the Messaging Server and S/MIME Applet

When checking the certificate of a public key against a CRL, the S/MIME applet must communicate directly with the Messaging server.

To Secure the Communications Link with SSL

1. Configure the Messaging server for SSL. See the discussion about configuring encryption and certificate-based authentication in See *Messaging Server System Administrator's Guide* for information.

2. Set the **sslrootcacertsurl** parameter in the **smime.conf** file to specify the information to locate the root SSL CA certificates. These CA certificates are used to verify the Messaging Server's SSL certificates when the SSL link is established between the Messaging Server and the S/MIME applet.
3. Set the **checkoverssl** parameter in the **smime.conf** file to **1**. This Messaging Server option determines whether SSL is used for the link between the Messaging Server and the S/MIME applet. Regardless of how a Convergence user specifies the URL for the Messenger Server (HTTP or HTTPS), the link between the Messaging Server and the S/MIME applet is secured with SSL when **checkoverssl** is set to **1**.

> **Note:** A proxy server can be used between the Messaging Server and client applications such as Convergence. See "Proxy Server and CRL Checking" using a proxy server with and without a secured communications link.

Key Access Libraries for the Client Machines

Whether your mail users keep their private-public key pairs and certificates on a smart card or in a local key store of their browsers, key access libraries must be present on the client machines to support the storage methods.

The libraries are supplied by vendors of the smart cards and browsers. You must ensure that the correct libraries are on the client machines and specify the library name(s) with the appropriate platform parameter in the **smime.conf** file. The parameter is **platformwin**.

You can specify only the libraries you know are installed on the client machines or you can specify all the library names for a given platform and vendor if you are not sure what is installed. If the S/MIME applet does not find the library it needs among the names you specify, the S/MIME features do not work.

The syntax to specify one or more library filenames is:

```
platform_parameter==vendor:library=
library_name;...
```

where:

- *platorm_parameter* is the parameter name for the platform of the client machine where Convergence is accessed. Enter **platformwin**
- *vendor* specifies the vendor of the smart card or browser. Choose one of these literals:
  - **CAC** (for an ActivCard or NetSign smart card)
  - **CAPI** (for Internet Explorer with CAPI)
  - **MOZILLA** (for Mozilla with Network Security Services)
- *library_name* specifies the library filename.

The following libraries are required on the client machines when using Internet Explorer with Cryptographic Application Programming Interface (CAPI) on any Windows OS:

- **acpkcs211.dll**
- **capibridge.dll**
- **softokn3.dll**

- **core32.dll**

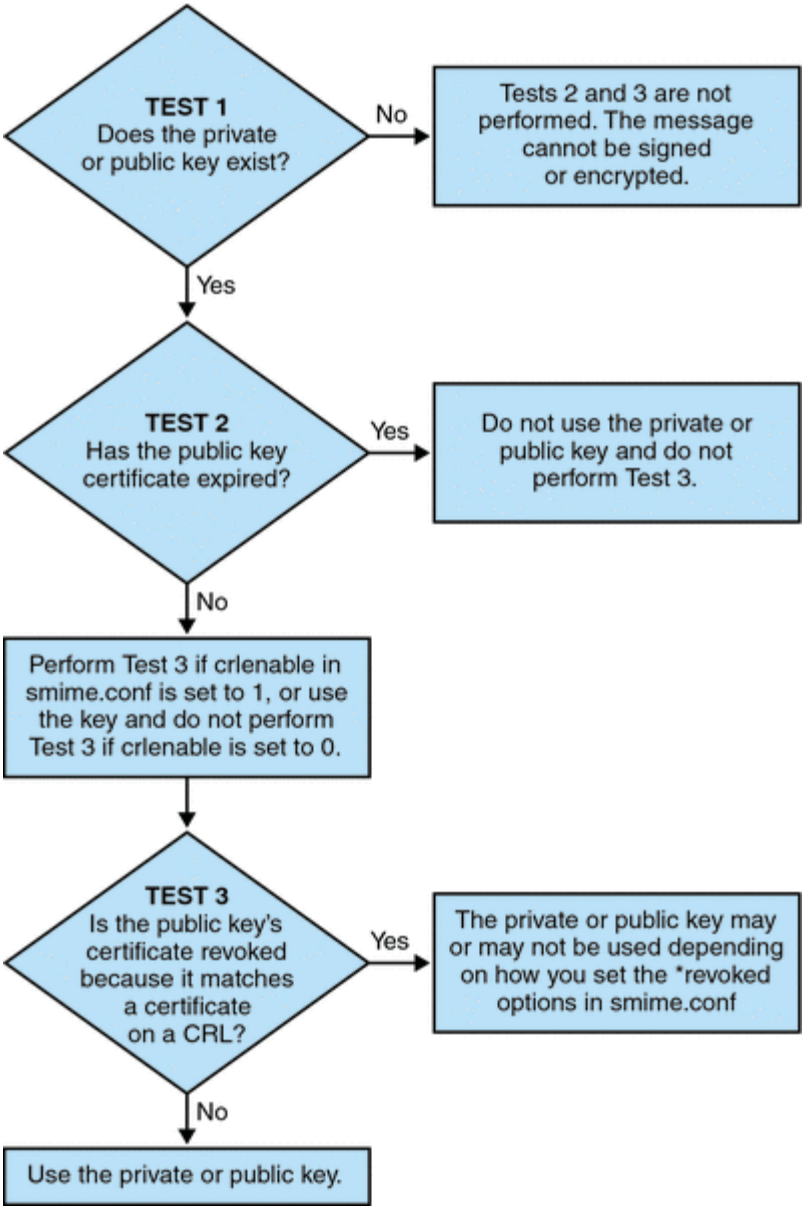
The following example specifies one smart card library and one Internet Explorer library, and one Mozilla library for a Windows platform:

```
platformwin==CAC:library=acpkcs211.dll;CAPI:library=capibridge.dll;
MOZILLA:library=softokn3.dll;
```

Verifying Private and Public Keys

Before Convergence Mail uses a private or public key, it must pass the verification tests shown in Figure 5–1. The remainder of this section describes the details of checking a public key's certificate against a CRL.

Figure 5–1 Private and Public Key Verification



Finding a User's Private or Public Key

When a Convergence Mail user has multiple private-public key pairs and multiple email addresses (primary, alternate, or alias addresses), it is possible that their keys are associated among their addresses. In this case, it is important that the S/MIME applet finds all the keys for verification purposes. Use the **usercertfilter** parameter in the **smime.conf** file to define a filter that creates a list of mail addresses for a key's owner at the time the public key's certificate is checked against a CRL. See the discussion about the **smime.conf** file and its parameters in the Messaging Server documentation for more information.

About Certificate Checking Against a CRL

A certificate revocation list, or CRL, is a list of revoked certificates maintained by the CA who issues the key pairs and certificates. When CRL checking is enabled, it causes the system to check the CRL whenever a certificate request has been made to see whether or not that certificate has been revoked.

When **crlenable** is set to **1** in the **smime.conf** file, a CRL test is performed after an unexpired key is found. The public key's certificate is checked against a CRL. There can only be one CRL for each CA, however the same CRL can be located in different places.

Checking a certificate against a CRL is done by the Messaging Server after the S/MIME applet sends it a request to do so. A public key certificate is used to validate a public key. Because a private key is kept secret, only used by the person who owns it, a private key cannot be checked directly against a CRL. To determine if a private key is good, the public key certificate of the key pair is used. When the public key's certificate passes the CRL test, the associated private key passes the test too.

Revocation of a certificate can happen for a variety of reasons, such as its owner has left your organization or lost the smart card. There are three situations for checking a certificate against a CRL:

- When an outgoing message is signed:

  The S/MIME applet always does this check unless you set **sendsigncert** to **0** or **crlenable** to **0**.

- When an incoming signed message is read:

  The S/MIME applet always does this check unless you set **readsigncert** to **0** or **crlenable** to **0**.

- When an outgoing message is encrypted:

  The S/MIME applet always does this check unless you set **sendencryptcert** to **0** or **crlenable** to **0**.

Accessing a CRL

A certificate contains zero or more URLs, known as distribution points, that are used by Messaging Server to locate a CRL. If the certificate does not have a CRL URL, it cannot be checked against a CRL and the private or public key is used to sign or encrypt a message without knowing its true status.

If Messaging Server fails to locate or gain access to a CRL after trying all the URLs available to it, the status of the certificate is treated as unknown. Whether a private or public key with an unknown status is used is determined by the setting of **revocationunknown**.

While only one CRL for each CA is supported, there can be multiple copies of the same CRL in different locations, reflected in different URLs among a user's public key certificates. Messaging Server tries all the URL locations for a certificate until it gains access to the CRL.

You can manage multiple copies of a CRL for optimum access by periodically downloading the current CRL from the CA to a place where you want it. While you cannot change the URLs embedded in the certificates, you can redirect Messaging Server to use new CRL locations by mapping the URLs in a certificate to a new URL containing the CRL information. Create a list of one or more mapping definitions in the LDAP directory with this syntax:

```
msgCRLMappingRecord=url_in_certificate==
new_url[|url_login_DN|url_login_password]
```

url_in_certificate is the URL in the certificate containing the old information to locate the CRL. new_url is the new URL containing the new CRL information. **url_login_DN** and **url_login_password** are the DN and password of the entry allowed access to new_url. Both are optional, and if specified, will be used for the new URL access only.

If the DN and password fails, LDAP access is denied and no retry with other credentials is attempted. These login credentials are only valid for LDAP URLs. If you use **crlurllogindn** and **crlurlloginpw** in **smmime.conf**, then you don't need to specify the login DN and password in the mapping record. See "Accessing LDAP for Public Keys, CA certificates and CRLs Using Credentials".

Only one layer of mapping is allowed. Different URLs in the certificates can be mapped to the same new URL, but you cannot assign a certificate URL to multiple new URLs. For example, the following mapping list is not valid:

```
msgCRLMappingRecord=URL12==URL45
msgCRLMappingRecord=URL12==URL66
msgCRLMappingRecord=URL12==URL88
msgCRLMappingRecord=URL20==URL90
msgCRLMappingRecord=URL20==URL93
```

The next example is a correct mapping list:

```
msgCRLMappingRecord=URL12==URL45
msgCRLMappingRecord=URL14==URL66
msgCRLMappingRecord=URL88==URL66
msgCRLMappingRecord=URL201==URL90
msgCRLMappingRecord=URL202==URL93
```

Once you have created the mapping definitions in your LDAP directory, use **crlmappingurl** in the **smime.conf** file to specify the directory information to locate them. See the discussion about the **smime.conf** file and its parameters in the Messaging Server documentation for more information.

Proxy Server and CRL Checking

If your system uses a proxy server between client applications and the Messaging Server, CRL checking can be blocked despite the fact that you correctly configured the S/MIME applet to perform CRL checking. When this problem occurs, users of Convergence Mail receive error messages alerting them to revoked or unknown status for valid key certificates.

The following conditions cause the problem:

- CRL checking is requested with these configuration values:

  - **crlenable** parameter in the **smime.conf** file is set to **1**
  - **local.webmail.cert.enable** option of Messaging Server is set to **1**
- The communications link between the S/MIME applet and the proxy server is not secured with SSL, but the S/MIME applet is expecting a secured link because the **checkoverssl** parameter in the **smime.conf** file is set to **1**

To solve this problem, you can:

- Set up the communications link between the client machines and proxy server as a secured link with SSL and leave all the configuration values as they are. Or,
- Leave the communications link unsecured and set **checkoverssl** to **0**.

For more information see "Securing Internet Links With SSL".

Using a Stale CRL

Checking a certificate against a CRL is done by the Messaging Server after the S/MIME applet sends it a request to do so. Rather than download a CRL to memory each time a certificate is checked, Messaging Server downloads a copy of the CRL to disk and uses that copy for certificate checking. Every CRL has a next-update field which specifies the date after which a newer CRL version should be used. The next-update date can be viewed as an expiration date or time limit for using the CRL. A CRL that is past it's next-update date is considered old or stale and triggers Messaging Server to download the latest version of the CRL the next time a certificate is checked.

Every time the S/MIME applet requests that a certificate be checked against a CRL, the Messaging Server does the following:

1. Compares the current date to the next-update date of the CRL.
2. If the CRL is stale, the Messaging Server downloads the latest version of the CRL to replace the stale CRL on disk and checking proceeds. However, if a newer CRL cannot be found or cannot be downloaded, the value of **crlusepastnextupdate** in the **smime.conf** file is used to determine what to do.
3. If **crlusepastnextupdate** is set to **0**, the stale CRL is not used and the certificate in question has an ambiguous status. The S/MIME applet uses the value of **revocationunknown** in **smime.conf** to determine what to do next:
   a. If **revocationunknown** is set to **ok**, the certificate is treated as valid and the private or public key is used to sign or encrypt a message.
   b. If **revocationunknown** is set to **revoked**, the certificate is treated as invalid, the private or public key is not used to sign or encrypt a message, and a pop-up error message alerts the mail user that the key cannot be used.

      If **crlusepastnextupdate** is set to **1**, the S/MIME applet continues to use the stale CRL which causes no interruption of processing within Convergence Mail, however a message is written to the Messaging Server log file to alert you to the situation.

This sequence of events continues to occur as certificates are checked against the CRL. As long as the Messaging Server can download a newer version of the CRL in a timely manner, and depending on the settings in the **smime.conf** file, mail processing proceeds without interruption. Check the Messaging Server log periodically for repeated messages that indicate a stale CRL is in use. If a newer CRL cannot be downloaded, you need to investigate why it is inaccessible.

Determining Which Message Time to Use

The **timestampdelta** parameter is used primarily for these purposes:

1. To handle the situation of a message that takes a long time to arrive at its destination. For this case, the sender's key might be treated as an invalid key despite the fact that the key was valid when the message was sent.
2. To limit the trust in a message's sent time because sent times can be faked.

There are two times associated with every message:

- The time when the message was sent, as found in the Date line of the message header detail
- The time when the message arrives at its destination, as found in the last Received line of the message header detail

> **Note:** View the message header detail by clicking the triangle icon at the right hand side of a message's From field.

A certificate that was valid when a message was sent can be revoked or expired by the time the message reaches its destination. When this happens, which time should be used when checking the validity of the certificate, the sent time or the received time? Using the sent time would verify that the certificate was valid when the message was sent. But always using the sent time does not take into account the fact that it might take a long time for a message to arrive at its destination, in which case it would be better to use the received time.

You can influence which time to use for CRL checking by using the **timestampdelta** parameter in the **smime.conf** file. Set this parameter to a positive integer, representing seconds. If the received time minus the value of **timestampdelta** is a time before the sent time, the sent time is used. Otherwise, the received time is used. The smaller the value of **timestampdelta**, the more often the received time is used. When **timestampdelta** is not set, the received time is always used. In the Messaging Server documentation, see the discussion about the **timestampdelta** parameter in the **smime.conf** file.

Trouble Accessing a CRL

For a variety of reasons, such as network or server problems, a CRL might be unavailable when Messaging Server attempts to check a certificate against it. Rather than let the Messaging Server spend its time constantly trying to gain access to the CRL, you can use the **crlaccessfail** parameter in the **smime.conf** file to manage how often it attempts to access the CRL, freeing up the Messaging Server for other tasks.

Define the following with **crlaccessfail**:

- How many failed attempts are counted (an error message is written to the Messaging Server log after each failed attempt)
- Over what period of time the failed attempts are counted
- How long to wait before attempting a new cycle of accessing the CRL

In the Messaging Server documentation, see the discussion about the **crlaccessfail** parameter in the **smime.conf** file.

When a Certificate is Revoked

When a public key's certificate does not match any entry on the CRL, the private or public key is used to sign or encrypt an outgoing message. When a certificate matches an entry on the CRL or the certificate's status is unknown, a private or public key is considered revoked. By default Convergence Mail does not use a key with a revoked certificate to sign or encrypt an outgoing message. If the private key of a signed message is revoked by the time the recipient reads the message, the recipient receives a warning message indicating that the signature should not be trusted.

If desired, you can change the various default policies for all revoked certificates with the following parameters in the **smime.conf** file:

- Set **sendsigncertrevoked** to **allow** to sign an outgoing message with a private key that is considered revoked because its public key's certificate is revoked
- Set **sendencryptcertrevoked** to **allow** to encrypt an outgoing message with a public key that has a revoked certificate
- Set **revocationunknown** to **ok** to treat a certificate as valid whose status is unknown; the private or public key is used to sign or encrypt an outgoing message

Granting Permission to Use S/MIME Features

Permission to use the various mail services available through Convergence can be given or denied with LDAP filters. A filter is defined with the **mailAllowedServiceAccess** or **mailDomainAllowedServiceAccess** LDAP attributes. Generally speaking, a filter works in one of three ways:

- Permission to given to all users for all services when no filter is used
- Permission is explicitly given to a list of users for specified service names (a plus sign (+) precedes the service name list)
- Permission is explicitly denied to a list of users for specified service names (a minus sign (-) precedes the service name list)

The required mail service names for S/MIME are *http*, *smime*, and *smtp*. If you need to restrict the use of S/MIME among Convergence users, use the appropriate LDAP attribute syntax and service names to create a filter. The attributes are created or modified with LDAP commands.

S/MIME Permission Examples

The following examples block access to the S/MIME features for one Convergence user:

```
mailAllowedServiceAccess
mailAllowedServiceAccess: -smime:*$+imap,pop,http,smtp:*
```

or

```
mailAllowedServiceAccess: +imap,pop,http,smtp:*
```

The following examples block access to the S/MIME features for all Convergence users in a domain:

```
mailDomainAllowedServiceAccess: -smime:*$+imap:*$+pop:*$+smtp:*$+http:*
```

or

```
mailDomainAllowedServiceAccess: +imap:*$+pop:*$+smtp:*$+http:*
```

See the Messaging Server documentation for more information about filters and their syntax.

Managing Certificates for S/MIME

Most of the following examples use the **ldapsearch** and **ldapmodify** commands to search an LDAP directory for user keys and certificates. These commands are provided with Directory Server. See your Oracle Directory Server Enterprise Edition documentation for more information about the commands.

CA Certificates in an LDAP Directory

This example adds a certificate for a certificate authority to an LDAP directory. The directory structure for these certificates already exists. The certificate and the LDAP entries where it belongs are entered into the **add-root-CA-cert.ldif** file. All text is entered into the file in ASCII text except for the certificate information, which must be entered as Base64 encoded text:

```
dn: cn=SMIME Admin,ou=people,o=demo.siroe.com,o=demo
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: certificationAuthority
cn: RootCACerts
sn: CA
authorityRevocationList: novalue
certificateRevocationList: novalue
cacertificate;binary:: MFU01JTUUEjAQBgNVBAsTCU1zZ1NlcnZlcjcMBoGA1UEAxMTydG
QGEwJVUzEOMAwGA1UEMFUJTUUxEjAQBgNVBAsTCU1zZ1NlcnZlcjEMBoGA1UEAxMTQ2VydG
aFw0wNjAxMwODAwMDBaM267hgbX9FExCzAJByrjgNVBAk9STklBMQwCgYDVQQVHR8EgaQwg
YTAlVMRMQYDVQQIEwpDQUxJRk9STklBMQwwCgYDVQQKEwww3ltgYz11lzAdBgNVBpYSE9Vc
5yZWQaddWlm899XBsYW5ldC5jb20wgZ8wDQYJoGBAK1mUTy8vvnOFg4mlHjkghytQUR1k8l
5mvWRf77ntm5mGXRD3XMU4OciUq6zUfIg3ngvxlLyERTIqjUS8HQU4R5pvj+rrVgsAGjggE
+FNAJmtOV2A3wMyghqkVPNDP3Aqq2fkcn4va3C5nRNAYxNNVE84JJ0H3jyPDXhMBlQU6vQn
weMBAAjggEXMIIBEzARBglghkgBhCAQEEBApqlSai4mfuvjh02SQkoPMNDAgTwMB8GA1UdI
QYMBaAEd38IK05AHreiU9OYc6vNMOwZMIGsBgNVHR8EgaQwgaEwb6BtoGuGaWxkYXA6Lyht
bmcucmVkLmlbGFuZXQuY29tL1VJD1DXJ0aWZpY2F0ZSBNYW5hZ2VyLE9VPVBlb3BsZSxPPW
aWxxYT9jZXJ0aZpY2jdu2medXRllkghytQURYFNrkuoCygKoYoaHR0cDovL3Bla2kghytQU
Zy5yZWQuaXBsYW5lC5jb20vcGVranLmNybDAeBgNVHREEFzAVgRNwb3J0aWEuc2hhb0BzdW
4uY29tMA0GCxLm78freCxS3Pp078jyTaDci1AudBL8+RrRUQvxsMJfZeFED+Uuf10Ilt6kw
Tc6W5UekbirfEZGAVQIzlt6DQJfgpifGLvtQ60Kw==
```

The CA's certificate is added to the LDAP directory with an **ldapmodify** command:

```
ldapmodify -a -h demo.siroe.com -D "cn=Directory Manager" -w mypasswd -v
 -f add-root-CA-cert.ldif
```

The value of the **trustedurl** parameter in **smime.conf** specifies the location of the CA certificates in the LDAP directory. In the following example, **trustedurl** is set to:

```
trustedurl==ldap://demo.siroe.com:389/cn=SMIME Admin, ou=people,
o=demo.siroe.com,o=demo?cacertificate;binary?sub?
(objectclass=certificationAuthority)
```

Public Keys and Certificates in an LDAP Directory

This example demonstrates adding a mail user's public key and certificate to the LDAP directory. It assumes the mail user already exists in the LDAP directory. The key and certificate, and the LDAP entries where it belongs, are entered into the

add-public-cert.ldif file. All text is entered into the file as ASCII text except for the key and certificate information, which must be entered as Base64 encoded text.

```
dn: uid=JohnDoe,ou=People, o=demo.siroe.com,o=demo
changetype: modify
replace: usercertificate
usercertificate;binary:: MFU01JTUUxEjAQBgNVBAsT1zZ1NlcnZlcjMBoGA1UEAxMTydG
QGEwJVUzEAwGA1hMFU01JTUUxEjAQBgNVBAsTCU1zZ1NlcnZlcjEcMBoGA1UEAxMTQ2VydG
aFw0wNjAxMTODAwaM267hgbX9FExCzAJBgwyrjgNVBAk9STklBMQwwCgYDVQQVHR8EgaQwg
AlVzMRMwEQYDVQQIDQUxJRk9STklBMQwwCgYDVQQKEwww3ltgoOYz11lzAdBgNVBpYSE9Vc
5yZWaddiiWlm899XBsYW5ldb20wgZ8wDQYJoGBAK1mUTy8vvO2nOFg4mlHjkghytQUR1k8l
5mvgcWL77ntm5mGXRD3XMU4OcizUfIg3ngvxlLKLyERTIqjUS8HQU4R5pvj+rrVgsAGjggE
+FG9NAqtOV2A3wMyghqkVPNDP3Aqq2BYfkcn4va3RNAYxNNVE84JJ0H3jyPDXhMBlQU6vQn
1NAgMBGjggEXMIIBEzARBglghkgBhvhCAQEEBApqlSai4mfuvjh02SQMNDAgTwMB8GA1UdI
QYMBaEd38IK05AHreiU9OYc6v+ENMOwZMIGsBgNVHR8EgaQwgaEwb6BuGaWxkYXA6Lyht74
tpbmcmVkLmlwbGFuZXQuY29tL1VJRD1DZXJ0aWZpY2F0ZSBNYW5hZ2V9VPVBlb3BsZSxPPW
1haWxT9jZXJ0aWZpY2jdu2medXRllHjkghytQURYFNrkuoCygKoYoaHDovL3Bla2kghytQU
luZy5WQuaXBsYW5ldC5jb20vcGVraW5nLmNybDAeBgNVHREEFzAVgRNw0aWEuc2hhb0BzdW
4uY29A0GCxLm78UfreCxS3Pp078jyTaDv2ci1AudBL8+RrRUQvxsMJfZD+Uuf10Ilt6kwhm
Tc6W5UekbirfEZGAVQIzlt6DQJfgpifGLvtQ60Kw==
```

The **ldapmodify** command is used to add the public key and certificate to the LDAP directory:

```
ldapmodify -a -h demo.siroe.com -D "cn=Directory Manager" -w mypasswd -v
 -f add-public-cert.ldif
```

The value of the **certurl** parameter in **smime.conf** specifies the location of the public keys and their certificates in the LDAP directory. In the following example, **certurl** is set to:

```
certurl==ldap://demo.siroe.com:389/ou=people, o=demo.siroe.com,
o=demo?userCertificate;binary?sub?
```

Verifying That Keys and Certificates Exist in the LDAP Directory

The following examples demonstrate searching an LDAP directory for CA certificates and public keys and their certificates.

Searching for One CA Certificate

In the following example, the base DN defined by the **-b** option, **cn=SMIME admin, ou=people,o=demo.siroe.com,o=demo objectclass=***, describes one CA certificate in the LDAP directory. If found in the directory, **ldapsearch** returns information about the certificate to the **ca-cert.lidf** file.

```
ldapsearch -L -h demo.siroe.com -D "cn=Directory Manager" -w mypasswd -b
"cn=SMIME admin, ou=people,o=demo.siroe.com,o=demo" "objectclass=*"
> ca-cert.ldif
```

The following example shows the search results in the **ca-cert.ldif** file. The format of the file's contents is a result of using the *-L* option of **ldapsearch**.

```
more ca-cert.ldif
dn: cn=SMIME admin,ou=people,o=demo.siroe.com,o=demo
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: certificationAuthority
cn: RootCACerts
cn: SMIME admin
```

```
sn: CA
authorityRevocationList: novalue
certificateRevocationList: novalue
cacertificate;binary:: MFU01JTUUxEjAQBgNVBAsTCU1zZNlcnZlcjcMBoGA1UEAxMTydG
QGEwJVEOMAwGA1UEChMFU0UUxEjAQBgNVBAsTCU1zZ1NlcnZlcjEcMBoGA1UEAxMTQ2VydG
aFw0jAxMTIwODAwMDBaM267X9FExCzAJBgwyrjgNVBAk9STklBMQwwCgYDVQQVHR8EgaQwg
YlVzMRMwEQYDVQQIEwpDQUx9STklBMQwwCgYDVQQKEwww3ltgoOYz11lzAdBgNVBpYSE9Vc
5yQuaddiiWlm899XBsYW5ljb20wgZ8wDQYJoGBAK1mUTy8vvO2nOFg4mlHjkghytQUR1k8l
5mcWRfL77ntm5mGXRD3XMciUq6zUfIg3ngvxlLKLyERTIqjUS8HQU4R5pvj+rrVgsAGjggE
+FNAJmqtOV2A3wMyghqkDP3Aqq2BYfkcn4va3C5nRNAYxNNVE84JJ0H3jyPDXhMBlQU6vQn
1NABAAGjggEXMIIBEzglghkgBhvhCAQEEBApqlSai4mfuvjh02SQkoPMNDAgTwMB8GA1UdI
QYMAFEd38IK05AHreOYc6v+ENMOwZMIGsBgNVHR8EgaQwgaEwb6BtoGuGaWxkYXA6Lyht74
tpbucmVkLmlwbGFuZY29tL1VJRD1DZXJ0aWZpY2F0ZSBNYW5hZ2VyLE9VPVBlb3BsZSxPPW
1haWYT9jZXJ0aWZpdu2medXRllHjkghytQURYFNrkuoCygKoYoaHR0cDovL3Bla2kghytQU
luZyZWQuaXBsYW5ldb20vcGVraW5nLmNybDAeBgNVHREEFzAVgRNwb3J0aWEuc2hhb0BzdW
4uYtMA0GCxLm78Ufre3Pp078jyTaDv2ci1AudBL8+RrRUQvxsMJfZeFED+Uuf10Ilt6kwhm
Tc6W5UekbirfEZGAVQIzlt6DQJfgpifGLvtQ60Kw==
```

Searching for a Several Public Keys

In the following example, the base DN defined by the **-b** option, **o=demo.siroe.com,o=demo objectclass=***, is such that all public keys and certificates found at and below the base DN in the LDAP directory are returned to the file **usergroup.ldif**:

```
ldapsearch -L -h demo.siroe.com -D "cn=Directory Manager" -w mypasswd
-b "o=demo.siroe.com,o=demo" "objectclass=*" > usergroup.ldif
```

Searching for One Public Key

In the following example, the base DN defined by the **-b** option, **uid=JohnDoe, ou=people,o=demo.siroe.com,o=demo objectclass=***, describes one public key and its certificate in the LDAP directory:

```
ldapsearch -L -h demo.siroe.com -D "cn=Directory Manager" -w mypasswd -b
"uid=JohnDoe, ou=people,o=demo.siroe.com,o=demo" "objectclass=*" > public-key.ldif
```

The following example shows the search results in the **public-key.ldif** file. The format of the file's contents is the result of using the **-L** option of **ldapsearch**.

```
more public-key.ldif
dn: uid=sdemo1, ou=people, o=demo.siroe.com, o=demo
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: siroe-am-managed-person
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: userPresenceProfile
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: icsCalendarUser
objectClass: sunUCPreferences
mail: JohnDoe@demo.siroe.com
mailHost: demo.siroe.com
.
.
uid: JohnDoe
.
.
mailUserStatus: active
```

```
inetUserStatus: active
.
.
usercertificate;binary:: MFU01JTUUxEjAQBgNBAsTCU1zZ1NlcnZjcMBoGA1UEAxMTydG
QGEwJEOwGA1UEChMFU01JTUUxEjAQBgNVBAsTCU1zZ1NlcnZlcjEcMBoGA1UEAxMTQ2VydG
aFw0MTIwODAwMDBaM267hgbX9FExCzAJBgwyrjgNVBAk9STklBMQwwCgYDVQQVHR8EgaQwg
YTAlVEQYDVQQIEwpDQUxJRk9STklBMQwwCgYDVQQKEwww3ltgoOYz11lzAdBgNVBpYSE9Vc
5yZWQdWlm899XBsYW5ldC5jb20wgZ8wDQYJoGBAK1mUTy8vvO2nOFg4mlHjkghytQUR1k8l
5mvgc7ntm5mGXRD3XMU4OciUq6zUfIg3ngvxlLKLyERTIqjUS8HQU4R5pvj+rrVgsAGjggE
+FG9NmV2A3wMyghqkVPNDP3Aqq2BYfkcn4va3C5nRNAYxNNVE84JJ0H3jyPDXhMBlQU6vQn
1NAgMAgEXMIIBEzARBglghkgBhvhCAQEEBApqlSai4mfuvjh02SQkoPMNDAgTwMB8GA1UdI
QYMBaEdK05AHreiU9OYc6v+ENMOwZMIGsBgNVHR8EgaQwgaEwb6BtoGuGaWxkYXA6Lyht74
tpbucmVkwbGFuZXQuY29tL1VJRD1DZXJ0aWZpY2F0ZSBNYW5hZ2VyLE9VPVBlb3BsZSxPPW
1haxYT9jZaWZpY2jdu2medXRllHjkghytQURYFNrkuoCygKoYoaHR0cDovL3Bla2kghytQU
luZyZWQuaYW5ldC5jb20vcGVraW5nLmNybDAeBgNVHREEFzAVgRNwb3J0aWEuc2hhb0BzdW
4u9tMA0GC78UfreCxS3Pp078jyTaDv2ci1AudBL8+RrRUQvxsMJfZeFED+Uuf10Ilt6kwhm
Tc6W5UekbirfEZGAVQIzlt6DQJfgpifGLvtQ60Kw==
.
.
```

Network Security Services Certificates

Various certificates used for Network Security Services (NSS) are stored in their own database, which is not an LDAP database. Two utilities, **certutil** and **crlutil**, are provided with Messaging Server to store the certificates and associated CRLs in the database. You can also use these utilities to search the database.

See the Directory Server documentation for more information about **certutil**. Use the help text that comes with **crlutil** for more information about that utility (view the online help of both utilities by executing them without arguments).

Configuring Messaging Server to Use S/MIME in Convergence

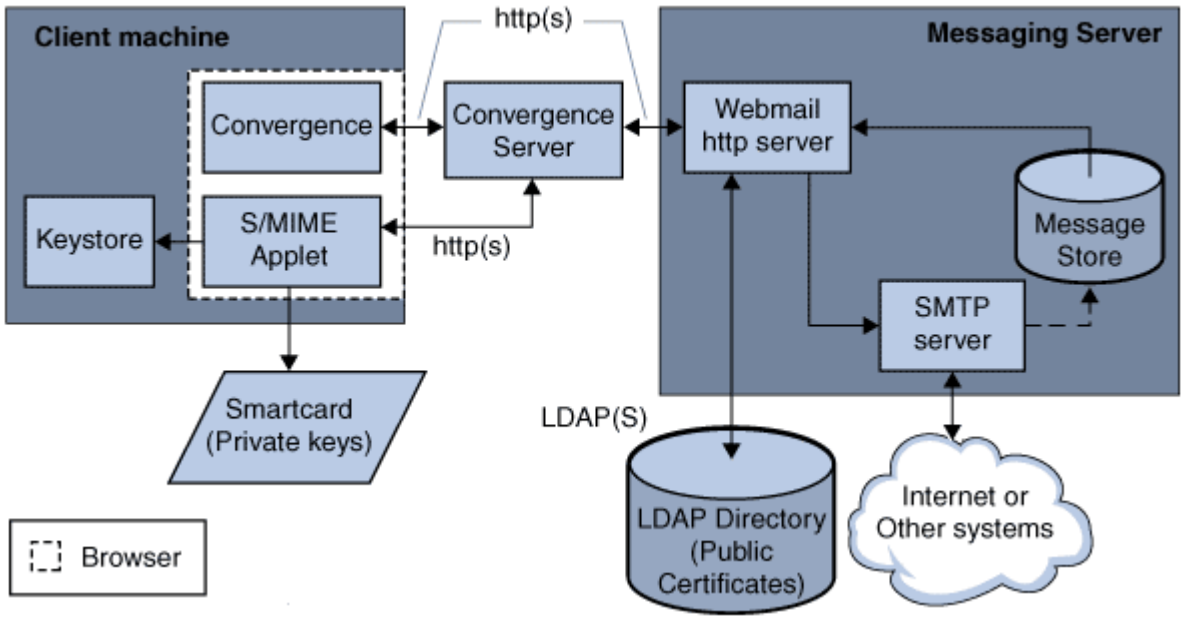
This section explains what the S/MIME applet is and provides a basic configuration procedure to set up S/MIME for Convergence. The configuration process involves setting parameters for the S/MIME applet and options for Messaging Server.

Overview of the S/MIME Applet

The process of signing a message, encrypting a message, or decrypting a message, along with the various procedures to verify private and public keys, are handled by a special applet, referred to as the S/MIME applet. The configuration of the S/MIME features is done with parameters in the **smime.conf** file and options of Messaging Server.

Figure 5–2 shows the S/MIME Applet in relation to other system components.

Figure 5–2 S/MIME Applet Location



The S/MIME applet is downloaded each time a user logs in to Convergence unless caching is enabled for the JRE on the user's machine. When caching is enabled, a copy of the S/MIME applet is saved on the user's machine after the initial download which prevents downloading the applet every time the user logs in.

Caching can improve performance so you might direct your users to enable caching, by following the instructions with the JRE.

Configuring S/MIME

The configuration file for S/MIME, **smime.conf**, contains descriptive comments and an example of each S/MIME parameter. The **smime.conf** file is included with Messaging Server, located in the directory *Messaging_Server_Home***/config/**, where *Messaging_Server_Home* is the directory where Messaging Server is installed.

To configure S/MIME:

1. Verify that the basic features of Convergence are working after you install Messaging Server.
2. If you haven't already, create or obtain private-public key pairs, with certificates in standard X.509 v3 format, for all your mail users who have permission to use the S/MIME features.
3. If smart cards are used for keys and certificates:
   a. Distribute the smart cards to your mail users.
   b. Ensure that the smart card reading devices and software are properly installed on each client machine where Convergence is accessed.
4. If local key stores of the browsers are used to store keys and certificates, instruct your mail users how to download their key pairs and certificate to the local key store.
5. Ensure that the correct libraries are on the client machines to support smart cards or local key stores. See "Key Access Libraries for the Client Machines" for more information.

6. Set up your LDAP directory to support S/MIME:

   a. Store all certificates for the CAs in the LDAP directory, accessible by Directory Server, under the distinguished name for certificate authorities. The LDAP attribute for these certificates is **cacertificate;binary**. Write down the directory information where you store them. You'll need this information for a later step.

      See **trustedurl** in Table 5–3 for an example of specifying LDAP directory information.

   b. Store the public keys and certificates in the LDAP directory accessible by Directory Server. The LDAP attribute for public keys and certificates is **usercertificate;binary**. Write down the directory information where you store them. You'll need this information for a later step.

      See **certurl** in Table 5–3 for an example of specifying LDAP directory information.

   c. Ensure that all users who send or receive S/MIME messages are given permission to use S/MIME with an LDAP filter in their user entries. A filter is defined with the **mailAllowedServiceAccess** or **mailDomainAllowedServiceAccess** LDAP attributes. By default, if you do not use **mailAllowedServiceAccess** or **mailDomainAllowedServiceAccess**, all services including S/MIME, are allowed. If you explicitly specify services with these attributes, then the services HTTP and SMTP, and S/MIME, must be specified to give mail users permission to use the S/MIME features. See "Granting Permission to Use S/MIME Features" for more information.

7. Edit the **smime.conf** file with any available text editor. See comments at the beginning of the file for parameter syntax. All text and example parameters in **smime.conf** are preceded with a comment character (#). You can add the parameters you need to **smime.conf** or copy a parameter example to another part of the file and change its value. If you copy and edit an example, be sure to remove the # character at the beginning of its line. Add these parameters to the file, each on its own line:

   a. **trustedurl** (see Table 5–3): set to the LDAP directory information to locate the certificates of the CAs.

   b. **certurl** (see Table 5–3): set to the LDAP directory information to locate the public keys and certificates.

   c. **usersertfilter** (see Table 5–3): set to the value of the example in the **smime.conf** file. The example value is almost always the filter you want. Copy the example and delete the # character at the beginning of the line. This parameter specifies a filter definition for the primary, alternate, and equivalent email addresses of a Convergence user to ensure that all of a user's private-public key pairs are found when the key pairs are assigned to different mail addresses.

   d. **sslrootcacertsurl** (see Table 5–3): if you are using SSL for the communications link between the S/MIME applet and Messaging Server, set **sslrootcacertsurl** with the LDAP directory information to locate the certificates of CAs that are used to verify the Messaging Server's SSL certificates. See "Securing Internet Links With SSL" for more information.

   e. **checkoverssl** (see Table 5–3): set to **0** if you are not using SSL for the communications link between the S/MIME applet and Messaging Server.

   f. **crlenable** (see Table 5–3): set to **0** to disable CRL checking for now because doing CRL checking might require adding other parameters to the **smime.conf** file.

g. **logindn** and **loginpw** (see Table 5–3): if the LDAP directory that contains the public keys and CA certificates requires authentication to access it, set these parameters to the distinguished name and password of the LDAP entry that has read permission.

   The values of **logindn** and **loginpw** are used whenever the LDAP directory is accessed with the LDAP information specified by the **crlmappingurl**, **sslrootcacertsurl**, or **trustedurl** parameters. See the discussion about the **smime.conf** file. Do not set **logindn** and **loginpw** if authentication is not required to access the LDAP directory.

8. Set the Messaging Server options with the **configutil** command

   a. **local.webmail.smime.enable** - set to **1**.

   b. **local.webmail.cert.enable** - set to **1** if you want to verify certificates against a CRL. See "Messaging Server configutil Options for S/MIME" for more information.

9. Enable S/MIME in the Convergence server using the **iwcadmin** command:

```
iwcadmin -o smime.enable -v true
```

10. Restart GlassFish Server.

11. Convergence is now configured for the S/MIME features. Verify that the S/MIME features are working with the following steps:

    a. Restart the Messaging server.

    b. Check the Messaging server log file: *Messaging_Server_Home***/log/http**, for diagnostic messages relating to S/MIME.

    c. If any problems were detected for S/MIME, the diagnostic messages help you determine how to correct the problem with the configuration parameters.

    d. Correct the necessary configuration parameters.

    e. Repeat Steps a. through d. until there are no more diagnostic messages for S/MIME in the Messaging Server's log file.

    f. Check that the S/MIME features are working with the following steps:

    g. Log in to Messaging server from a client machine. Answer the special prompts for the S/MIME applet with Yes or Always. See "Managing Certificates for S/MIME".

    h. Compose a short message, addressed to yourself.

    i. Encrypt your message by checking the **Encrypt** check box at the bottom of the Compose window if it is not already checked.

    j. Click **Send** to send the encrypted message to yourself. This should exercise most of the mechanisms for keys and certificates.

    k. If you find problems with the encrypted message, the most likely causes are the values you used for LDAP directory information in the **smime.conf** file and/or the way keys and certificates are stored in the LDAP directory. Check the Messaging Server log for more diagnostic messages. Table 5–3 lists many options you might want to use to further configure your S/MIME environment. See the discussion about **smime.conf** file in the Messaging Server documentation for more information.

Table 5–3 Messaging Server S/MIME Parameters for Convergence

Required Parameters for S/MIME	Parameters for Smart Cards and Local Key Stores	Parameters for CRL Checking	Parameters for Initial Settings and Secured Links
certurl*	platformwin	checkoverssl	alwaysencrypt
logindn	NA	crlaccessfail	alwayssign
loginpw	NA	crldir	sslrootcacertsurl
trustedurl*	NA	crlenable	NA
usercertfilter*	NA	crlmappingurl	NA
NA	NA	crlurllogindn	NA
NA	NA	crlurlloginpw	NA
NA	NA	crlusepastnextupdate	NA
NA	NA	readsigncert	NA
NA	NA	revocationunknown	NA
NA	NA	sendencryptcert	NA
NA	NA	sendencryptcertrevoked	NA
NA	NA	readsigncert	NA
NA	NA	sendsigncertrevoked	NA
NA	NA	timestampdelta	NA

You must specify a value for the **certurl**, **trustedurl**, and **usercertfilter** parameters because they have no default value.

Accessing LDAP for Public Keys, CA certificates and CRLs Using Credentials

Public keys, CA certificates, and CRLs required for S/MIME may be stored in an LDAP directory. The keys, certificates, and CRLs may be accessible from a single URL or multiple URLs in LDAP. For example, CRLs may be stored in one URL and public keys and certificates may be stored in another. Messaging Server allows you to specify which URL contains the desired CRL or certificate information, as well as the DN and password of the entry that has access to these URLs. These DN/password credentials are optional; if none are specified, LDAP access first tries the HTTP server credentials, and if that fails, it tries accessing it as *anonymous*.

Two pairs of **smime**.conf credential parameters may be set to access the desired URLs: **logindn** and **loginpw**, and **crlurllogindn** and **crlurlloginpw**.

logindn and **loginpw** are the credentials used for all URLs in **smime.conf**. They specify the DN and password of the LDAP entry that has read permission for the public keys, their certificates, and the CA certificates as specified by the **certurl** and **trustedurl** parameters.

crlurllogindn and **crlurlloginpw** specifies the DN and password of the LDAP entry that has read permission for the resulting URL from the mapping table (see "Accessing a CRL" for more information). If these credentials are NOT accepted, LDAP access is denied and no retry with other credentials is attempted. Either both parameters must be specified, or both must be empty. These parameters do not apply to the URLs that come directly from the certificate.

Setting Passwords for Specific URLs

Messaging Server allows you to specifically define the DN/ password pairs for accessing the following **smime.conf** URLs: **certUrl**, **trustedUrl**, **crlmappingUrl**, **sslrootcacertsUrl**.

The syntax is as follows:

url_typeURL [CommSuite:*URL_DN* | *URL_password*]

Example:

```
trustedurl==ldap://mail.siroe.com:389/cn=Directory Manager, ou=people,
o=siroe.com,o=ugroot?cacertificate?sub?(objectclass=certificationauthority) |
cn=Directory manager | boomshakalaka
```

Summary of Using LDAP credentials

This section summarizes the use of LDAP credentials.

- All LDAP credentials are optional; if none are specified, LDAP access first tries the HTTP server credentials, and if that fails, tries *anonymous*.

Two pairs of **smime.conf** parameters are used as credentials for the two sets of URLs that may be specified:

- **logindn** and **loginpw** - all URLs in **smime.conf**
- **crlurllogindn** and **crlurlloginpw** - all URLs from mapping table

These are known as the default LDAP credential pair.

- Any URL specified in **smime.conf** or via mapping CRL URLs can have an optional local LDAP credential pair specified.
- Credentials are checked in order in which each is specified:
  1. Local LDAP credential pair - if specified, only one tried
  2. Default LDAP Credential Pair - if specified, and no Local LDAP credential pair, only one tried
  3. Server - if neither Local LDAP credential pair nor default LDAP credential pair specified, first tried
  4. *anonymous* - last tried only if server fails or none specified
- If a URL has a Local LDAP credential pair specified, it is used first; if the access fails, access is denied.
- If a URL has no Local LDAP credential pair specified, the corresponding default LDAP credential pair is used; if access fails, then access is denied.

Messaging Server configutil Options for S/MIME

You can use the Messaging Server configuration utility to configure S/MIME options on Messaging Server.

To set S/MIME options using the Messaging Server configutil:

1. Log in to Messaging Server as root.
2. Change to the *Messaging_Server_Home***/sbin** directory.

   where *Messaging_Server_Home* is the directory where Messaging Server is installed.

3. Set the Messaging Server options from Table 5–4 using the configutil as desired for your system. Unless stated otherwise, an option is not required to be set.

Table 5–4 Messaging Server configutil Options for S/MIME

Parameter	Description
local.webmail.cert.enable	Controls whether the process that handles CRL checking should do CRL checking.*0* - The process does not check a certificate against a CRL. This is the default.*1* - The process checks a certificate against a CRL. When set to *1*, ensure that the *crlenable* parameter in the **smime.conf** file is set to *1*.
local.webmail.cert.port	Specifies a port number on the machine where the Messaging Server runs to use for CRL communication. This port is used locally for that machine only. The value must be greater than 1024. The default is *55443*. This is a required option if the default port number is already in use.
local.webmail.smime.enable	Controls whether the S/MIME features are available to Convergence Mail users. Choose one of these values:*0* - the S/MIME features are unavailable for Convergence Mail users even though the system is configured with the correct software and hardware components. This is the default.*1* - the S/MIME features are available to Convergence Mail users who have permission to use them. Example: `configutil -o local.webmail.smime.enable -v 1`

Messaging Server smime.conf Parameters

The **smime.conf** file is included with the Messaging Server. The file is located in the directory *Messaging_Server_Home***/config/**, where *Messaging_Server_Home* is the directory where Messaging Server is installed. All text and parameter examples in the file are preceded with a comment character (#).

You can add parameters with your values to the **smime.conf** file or you can edit the parameter examples. If using an example, copy the example to another part of the file, edit the parameter's value, and remove the # character at the beginning of the line.

Edit **smime.conf** with any available text editor after you install Messaging Server. The parameters, listed in Table 5–5, are not case sensitive and unless otherwise stated, are not required to be set.

Table 5–5 Messaging Server smime.conf Parameters

Parameter	Description
alwaysencrypt	Controls the initial setting for whether all outgoing messages are automatically encrypted for all Convergence users with permission to use S/MIME. Each Convergence user can override this parameter's value for their messages by using the check boxes described in "Signature and Encryption Settings". Choose one of these values:*0* - do not encrypt messages. The encryption check boxes within Convergence are displayed as unchecked. This is the default.*1* - always encrypt messages. The encryption check boxes within Convergence are displayed as checked. Example: *alwaysencrypt==1*
alwayssign	Controls the initial setting for whether all outgoing messages are automatically signed for all Convergence users with permission to use S/MIME. Each Convergence user can override this parameter's value for their messages by using the check boxes described in "Signature and Encryption Settings". Choose one of these values:*0* - do not sign messages. The signature check boxes within Convergence are displayed as unchecked. This is the default.*1* - always sign messages. The signature check boxes within Convergence are displayed as checked. Example: *alwaysensign==1*
certurl	Specifies the LDAP directory information to locate the public keys and certificates of Convergence users (the LDAP attribute for public keys is *usercertificate;binary*). See "Managing Certificates for S/MIME" for more information about certificates. This parameter must point to the highest node in the user/group of the LDAP directory information tree (DIT) that includes all users that are being served by the Messaging Server. This is particularly important for sites with more than one domain; the distinguished name must be the root distinguished name of the user/group tree instead of the subtree that contains users for a single domain. This is a required parameter that you must set. Example: *certurl==ldap://mail.siroe.com:389/ou=people,o=siroe.com,o=ugroot*
checkoverssl	Controls whether an SSL communications link is used when checking a key's certificate against a CRL. See "Securing Internet Links With SSL" for more information. Choose one of these values:*0* - do not use an SSL communications link.*1* - use an SSL communications link. This is the default. A problem can occur when a proxy server is used with CRL checking in effect. See "Proxy Server and CRL Checking".
crlaccessfail	Specifies how long to wait before the Messaging Server attempts to access a CRL after it has failed to do so after multiple attempts. This parameter has no default values. Syntax: *crlaccessfail==number_of_failures:time_period_for_failures:wait_time_before_retry* where:*number_of_failures* is the number of times that the Messaging Server can fail to access a CRL during the time interval specified by *time_period_for_failures.* The value must be greater than zero.*time_period_for_failures* is the number of seconds over which the Messaging Server counts the failed attempts to access a CRL. The value must be greater than zero.*wait_time_before_retry* is the number of seconds that the Messaging Server waits, once it detects the limit on failed attempts over the specified time interval, before trying to access the CRL again. The value must be greater than zero. Example: *crlaccessfail==10:60:300* In this example, Messaging Server fails 10 times within a minute to access the CRL. It then waits 5 minutes before attempting to access the CRL again. See "Trouble Accessing a CRL".
crldir	Specifies the directory information where the Messaging Server downloads a CRL to disk. The default is *Messaging_Server_Home***/data/store/mboxlist**, where *Messaging_Server_Home* is the directory where Messaging Server is installed. See "Using a Stale CRL" for more information.
crlenable	Controls whether a certificate is checked against a CRL. If there is a match, the certificate is considered revoked. The values of the *send*revoked* parameters in the **smime.conf** file determine whether a key with a revoked certificate is rejected or used by Convergence. See "Verifying Private and Public Keys" for more information. Choose one of these values:*0*- each certificate is not checked against a CRL.*1*- each certificate is checked against a CRL. This is the default. Ensure that the*local.webmail.cert.enable* option of the Messaging Server is set to *1*, otherwise CRL checking is not done even if *crlenable* is set to *1*.

Table 5–5 (Cont.) Messaging Server smime.conf Parameters

Parameter	Description
crlmappingurl	Specifies the LDAP directory information to locate the CRL mapping definitions. This parameter is only required when you have mapping definitions. See "Accessing a CRL" optionally add the DN and password that has access to the URL. Syntax: *crlmappingurlURL* [|*URL_DN* |*URL_password*] Example: `crlmappingurl==ldap://mail.siroe.com:389/ cn=XYZ Messaging, ou=people, o=mail.siroe.com,o=isp?msgCRLMappingRecord?sub? ( objectclass=msgCRLMappingTable) | cn=Directory Manager | pAsSwOrD`
crlurllogindn	Specifies the distinguished name of the LDAP entry that has read permission for the CRL mapping definitions (not if the entry is directly from the certificate, see "Accessing a CRL"). If values for *crllogindn* and *crlloginpw* are not specified, the Messaging Server uses the log in values for the HTTP server to gain entry to the LDAP directory. If that fails, Messaging Server attempts to access the LDAP directory anonymously. Example: *crllogindn==cn=Directory Manager*
crlurlloginpw	Specifies the password, in ASCII text, for the distinguished name of the*crllogindn* parameter. If values for *crllogindn* and *crlloginpw* are not specified, Messaging Server uses the log in values for the HTTP server to gain entry to the LDAP directory. If that fails, Messaging Server attempts to access the LDAP directory anonymously. The value may be obfuscated with base64 by using $== instead of == as the delimiter (this feature was introduced in Messaging Server 7 Update 1). Example: *crlloginpw==zippy* or*crlloginpw$==emlwcHk=*
crlusepastnextupdate	Controls whether a CRL is used when the current date is past the date specified in the CRL's next-update field. See "Using a Stale CRL" for more information. Choose one of these values:*0* - do not use the stale CRL.*1* - use the stale CRL. This is the default.
logindn	Specifies the distinguished name of the LDAP entry that has read permission for the public keys and their certificates, and the CA certificates located in the LDAP directory specified by the *certurl* and *trustedurl* parameters. If values for *logindn* and *loginpw* are not specified, the Messaging Server uses the log in values for the HTTP server to gain entry to the LDAP directory. If that fails, Messaging Server attempts to access the LDAP directory anonymously. Example: *logindn==cn=Directory Manager*
loginpw	Specifies the password, in ASCII text, for the distinguished name of the *logindn* parameter. If values for *logindn* and *loginpw* are not specified, Messaging Server uses the log in values for the HTTP server to gain entry to the LDAP directory. If that fails, Messaging Server attempts to access the LDAP directory anonymously. The value may be obfuscated with base64 by using $== instead of == as the delimiter (this feature was introduced in Messaging Server 7 Update 1). Example: *loginpw==SkyKing* or*loginpw$==U2t5S2luZw==*
platformwin	Specifies one or more library names that are necessary when using smart cards or a local key store on a Windows platform. Change this parameter only if the default value does not work for your client machines. The default is:*platformwin==CAPI:library=capibridge.dll;* See "Key Access Libraries for the Client Machines" for more information.
readsigncert	Controls whether a public key's certificate is checked against a CRL to verify an S/MIME digital signature when the message is read. (A private key is used to create a digital signature for a message but it cannot be checked against a CRL, so the certificate of the public key associated with the private key is checked against the CRL.) See "Verifying Private and Public Keys". Choose one of these values:*0* - do not check the certificate against a CRL.*1* - check the certificate against a CRL. This is the default.
revocationunknown	Determines the action to take when an ambiguous status is returned when checking a certificate against a CRL. In this case, it is not certain whether the certificate is valid or has a revoked status. See "Verifying Private and Public Keys" for more information. Choose one of these values: *ok* - treat the certificate as valid.*revoked* - treat the certificate as revoked. This is the default.

Table 5–5 (Cont.) Messaging Server smime.conf Parameters

Parameter	Description
sendencryptcert	Controls whether the certificate of a public key that is used to encrypt an outgoing message is checked against a CRL before using it. See "Verifying Private and Public Keys". Choose one of these values:*0* - do not check the certificate against a CRL.*1* - check the certificate against a CRL. This is the default.
sendencryptcertrevoked	Determines the action to take if the certificate of a public key that is used to encrypt an outgoing message is revoked. See "Verifying Private and Public Keys" for more information. Choose one of these values: *allow* - use the public key.*disallow* - do not use the public key. This is the default.
sendsigncert	Controls whether a public key's certificate is checked against a CRL to determine if a private key can be used to create a digital signature for an outgoing message. (A private key is used for a digital signature but it cannot be checked against a CRL, so the certificate of the public key associated with the private key is checked against the CRL.) See "Verifying Private and Public Keys" for more information. Choose one of these values:*0* - do not check the certificate against a CRL.*1* - check the certificate against a CRL. This is the default.
sendsigncertrevoked	Determines the action to take when it is determined that a private key has a revoked status. (A private key is used to create a digital signature for a message but it cannot be checked against a CRL, so the certificate of the public key associated with the private key is checked against the CRL. If the public key certificate is revoked, then its corresponding private key is also revoked.) See "Verifying Private and Public Keys" for more information. Choose one of these values: *allow* - use the private key with a revoked status.*disallow* - do not use the private key with a revoked status. This is the default.
sslrootcacertsurl	Specifies the distinguished name and the LDAP directory information to locate the certificates of valid CAs which are used to verify the Messaging Server's SSL certificates. This is a required parameter when SSL is enabled in the Messaging Server. See "Securing Internet Links With SSL" for more information. If you have SSL certificates for a proxy server that receives all requests from client application, the CA certificates for those SSL certificates must also be located in the LDAP directory pointed to by this parameter. You can also optionally add the DN and password that has access to the URL. Syntax: *crlmappingurlURL* [|*URL_DN* |*URL_password*] Example: `sslrootcacertsurl==ldap://mail.siroe.com:389/cn=SSL Root CA Certs,ou=people,o=siroe.com,o=isp? cacertificate; binary?base? (objectclass=certificationauthority)|cn=Directory Manager | pAsSwOrD`
timestampdelta	Specifies a time interval, in seconds, that is used to determine whether a message's sent time or received time is used when checking a public key's certificate against a CRL. The parameter's default value of zero directs Convergence to always use the received time. See "Determining Which Message Time to Use" for more information. Example: *timestampdelta==360*
trustedurl	Specifies the distinguished name and LDAP directory information to locate the certificates of valid CAs. This is a required parameter. You can also optionally add the DN and password that has access to the URL. Syntax: *crlmappingurlURL* [|*URL_DN* |*URL_password*] Example: `trustedurl==ldap://mail.siroe.com:389/cn=Directory Manager, ou=people, o=siroe.com,o=ugroot?cacertificate?sub? (objectclass=certificationauthority)| cn=Directory Manager | pAsSwOrD`
usercertfilter	Specifies a filter definition for the primary, alternate, and equivalent email addresses of a Convergence user to ensure that all of a user's private-public key pairs are found when they are assigned to different mail addresses. This parameter is required and has no default values.

Managing Attachment Previewing

By default, Convergence can preview only JPG, GIF, and TXT email attachments. In a desktop environment, native applications must be installed to view email attachments such as Office documents, or browser plug-ins must be installed in the browser to enable Convergence to preview PDF attachments.

If Convergence is integrated with Oracle Outside In Transformation Server, Convergence is capable of previewing many different file types regardless of the web browser, including DOC and XLS type email attachments.

See *Convergence Installation and Configuration Guide* for information about installing Outside In Transformation Server and configuring it for Convergence.

About Outside In Transformation Server and the Outside In Proxy

Each time a user previews an attachment, Convergence attempts to open it in the browser. If Convergence is not able to open the attachment by default, it sends the attachment to Outside In Transformation Server. The transformation server transforms the attachment into HTML, which Convergence can render in the browser.

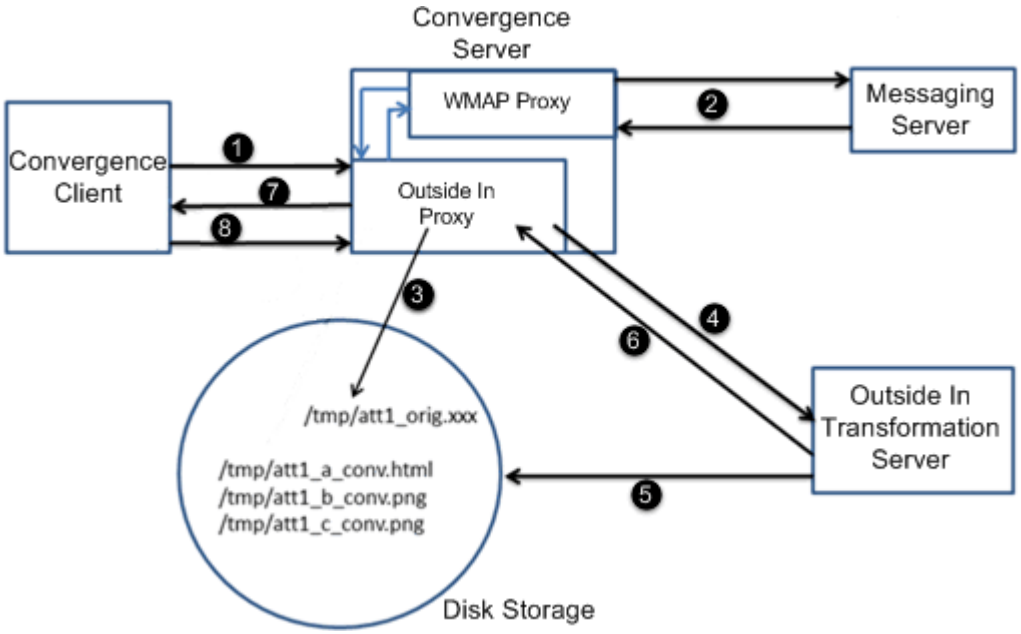
The transformation server can handle a large number of simultaneous requests by placing attachment requests in a queue.

The Outside In proxy creates a temporary directory for each user requesting to view attachments. For each transformation, the Outside In proxy creates a temporary subdirectory under the user directory. The Outside In proxy passes the input directory containing the transformed attachment and the output directory of the transformed attachment to the Transformation Server. The Outside In proxy deletes the subdirectory after a configurable time-out period has passed.

The Convergence server manages file management to the transformation server. Convergence uses a session cookie and a server-generated URL token for each attachment request. For security, Convergence masks the URL token.

Figure 5–3 shows the attachment preview workflow.

Figure 5–3 Convergence Attachment Request Workflow



The following list explains the attachment preview workflow from Figure 5–3.

1. The Convergence client sends the request to the Convergence server.

   If the request is for an attachment type that can be rendered natively in the browser, the request is sent to the WMAP proxy.

   If the request is for an attachment type that cannot be rendered natively in the browser, the request is first passed to the Outside In proxy, and then to the WAMP proxy.

2. The request is sent to the Messaging server. The Messaging server sends the attachment back to the Convergence server.

3. For attachment types that cannot be rendered natively in the browser, the Outside In proxy sends the response from the Messaging server to the disk storage.

4. For attachment types that cannot be rendered natively in the browser, the Outside In proxy communicates where it saved the attachment to the Outside In Transformation server, and also informs the transformation server where to save the attachment after it has been converted.

5. For attachment types that cannot be rendered natively in the browser, the transformation server converts the attachment into a format that can be natively rendered in the browser and saves it to the directory provided by the Outside In proxy.

6. For attachment types that cannot be rendered natively in the browser, the transformation server informs the Outside In proxy that it has completed transforming the file.

7. If the original request was for an attachment type that could be rendered native in the browser, the Convergence server sends the attachment to the browser.

   If the original request was for an attachment type that could not be rendered native in the browser, the Outside In proxy provides the browser with a redirection URL to the transformed attachment on the disk storage.

8. If the original request was for an attachment type that could not be rendered native in the browser, the Convergence client accesses the transformed attachment using the URL provided by the Outside In proxy and renders it in the browser.

Configuring File Directory Access

Convergence and Outside In Transformation Server have to be configured so that they can both read and write attachments in the storage disk. Convergence must have full permissions to the storage disk to read, write, and delete files.

The Convergence Server and the Outside In Transformation Server can run on the same machine or on different machines. Configure the transformation server as a network file system (NFS).

- If the transformation server is running on Solaris:
  - Share the **/export/tsdir/** directory.

    ```
    chmod 700 /export/tsdir
    ```

  - Edit **/etc/dfs/dfstab** and add the following line:

    ```
    share -F nfs -d [-o root=host_name] "tsdir" /export/tsdir
    ```

    Include the **-o** parameter when the Convergence server and the transformation server are running as local root, where *host_name* is the host name of the

Convergence server. Omit the **-o** parameter when the Convergence server and the transformation server are running as the same user.

- Create a soft link or mount to the NFS directory. For example:

```
-s //net/host_name/export/tsdir /export/tsdir
```

- If the transformation server is running on Linux:
  - Share the **/export/tsdir/** directory:

```
chmod 700 /export/tsdir
```

  - If the Convergence server and the transformation server are running as the same user, edit **/etc/exports** and add the following line:

```
/export/tsdir
```

    If the Convergence server and the transformation server are running as local root, edit **/etc/exports** and add the following line:

```
/export/tsdir host_name(rw,no_root_squash)
```

  - Create a soft link or mount to the NFS directory. For example:

```
-s //net/host_name/export/tsdir /export/tsdir
```

The Outside In proxy generates a unique URL for each attachment and provides it to the Convergence client.

The following example shows the sample configuration settings for Outside In proxy in the Convergence **configuration.xml** file:

```
<OINService>
     <ServiceName v="SUN_OIN_SERVICE"/>
     <BackendServiceDetails>
          <Enable v="true"/>
          <HostName v="oin server name"/>
          <PortNumber v="60611"/>
     </BackendServiceDetails>
     <TsdirPath v="/export/tsdir"/>
     <AutoPruneInterval v="5"/>
</OINService>
```

You can use the **iwcadmin** command to configure the parameters for the Outside In proxy.

Managing Attachment Life Cycles

The Outside In proxy manages the life cycle of attachments, including temporary directories, file creation, deletion, and purging, and the number of directories and disk space per user.

By default, the Outside In proxy automatically deletes an attachment from the storage disk after five minutes.

Use the **iwcadmin** command to configure the duration after which attachments are deleted from the storage disk. For example, to configure the proxy to delete attachments from the disk after three minutes:

```
iwcadmin -o oin.autopruneinterval -v 3
```

Supporting Extended Character Locales

Oracle Outside In Transformation Server supports many typical font sets and some extended font sets. However, depending on the locales being used in your deployment, you may need to install and configure additional font sets to support the rendering of attachments.

By default, when the transformation server cannot render characters because the font is missing, it replaces the character with an asterisk. For example, if a user is using Convergence with the Japanese locale, but the transformation server does not have access to Japanese font sets, the transformation server will render attachments with asterisks.

Install all required fonts on the host machine where the transformation is installed and export the GDFONTPATH environment variable.

See the Oracle Outside In Technology documentation for more information.

Customizing Transformation Blacklist

The Outside In Transformation Server blacklist enumerates the types of files that are prevented from being sent to the transformation server, such as ZIP files or EXE files.

You can customize the blacklist to add or remove file types. See the discussion about customizing the attachment blacklist in *Convergence Customization Guide* for more information.

Enabling Anti-Spam

You can configure Convergence to take action against spam messages in the following ways:

- By setting the anti-spam related parameters in Convergence
- By integrating a spam filter in Messaging Server in addition to setting the anti-spam related parameters in Convergence

Configuring Convergence to Combat Spam

Set the following parameters in Convergence:

- **mail.spam.enableaction**: Set this parameter to **true** to enable the anti-spam functionality. Setting this parameter will enable users to take action against spam messages.

  ```
  iwcadmin -o mail.spam.enableaction  -v true
  ```

- **mail.spam.folder**: Set this parameter to the folder name into which spam messages should be moved.

  ```
  iwcadmin -o mail.spam.folder -v SpamFolder
  ```

You must restart the GlassFish server starting with after making the configuration changes.

When you set the parameters, the following spam related functionality is enabled in the Convergence client:

- A system folder is made available as the designated spam folder. This is based on the value set for the **mail.spam.folder** parameter assigned by the administrator.

- Users will be able to mark messages as spam or not spam. Messages marked as spam are moved into the designated spam folder and messages that are marked as not spam are moved into the Inbox.

Configuring Messaging Server to Combat Spam

A more effective way to counter spam messages is to deploy a spam filer at the back-end Messaging Server in addition to enabling the anti-spam functionality in Convergence. For information on how to integrate a spam filter with the Messaging Server, see the Instant Messaging documentation.

After integrating the spam filter, set the value of the **service.feedback.spam** parameter in Messaging Server to the email address at which spam reports are accepted.

```
configutil -o  service.feedback.spam -v email_address
```

When you set this parameter, the following spam related functionality will be available to the Convergence client.

- Users will be able to mark messages as spam. When users mark a message as spam, the message is flagged in the message store, and forwarded to the email address set for the **service.feedback.spam** configuration utility option. The spam messages are marked in the message list and displayed with a warning in the message viewer.
- Users will be able to mark messages incorrectly identified as spam, as not spam. When the user marks incorrectly identified spam messages as not spam, the flag is removed from the message in the message store.

If Messaging Server is configured with a spam filter that accepts reports of messages that are incorrectly identified as spam, set the value of the parameter **service.feedback.notspam** to the email address at which Convergence will forward the messages marked as not a spam.

```
configutil -o service.feedback.notspam -v email_address
```

Note: You must restart Messaging Server after making these configuration changes.

Removing Rich Text Formatting for Email Composition

Convergence enables you to remove the Rich Text Formatting option for composing messages. To do so, set the **client.enablertfcompose** configuration property to **false**. By default, this parameter is set to **true**. For example:

```
iwcadmin -o client.enablertfcompose -v false
```

6

Address Book Service Administration

This chapter explains how to administer the address book service in Oracle Communications Convergence provided by Convergence Server.

See "Enabling Core Services for Convergence" for information about enabling services.

The address book service can be also provided by Oracle Communications Contacts Server. See *Contacts Server System Administrator's Guide* for information about administering Contacts Server.

See *Convergence Installation and Configuration Guide* for information about configuring Convergence with Contacts Server.

Configuring Horizontal Scalability for the Personal Address Book

Convergence server enables you to scale and support large number of users. Convergence server stores the information of a user's personal address book in the User/Group LDAP. This attribute is denoted by the psRoot attribute.

The psRoot is an attribute in the user's LDAP that specifies the host of the LDAP server, the port it is listening to port, and the DN where the Address Book entries for the user is stored. The psRoot attribute is in the form *`ldap://ldap_host:ldap_port/DN`*. The value of psRoot attribute determines the DB type and DB location.

For example of how a psRoot attribute looks in a user's LDAP entry:

```
ldap://siroe.com:389/piPStoreOwner=jsmith,o=siroe.com,o=PiServerDb
```

Where:

- *siroe.com:389* is the host name and port number of the LDAP server. In this example, the LDAP server listens to port 389.
- *piPStoreOwner=jsmith,o=siroe.com,o=PiServerDb* specifies the DB of the Personal Store.

> **Note:** The address book server does not provide any utility to distribute psRoot values for users, according to any scalability policy. Administrators need to set a specific policy suited best for the organization and use custom scripts to set the psRoot value for that policy.

Horizontal Scalability Architecture

The following are the key components of the Address Book Horizontal Scalability architecture:

- Personal Store
- DBMap
- DB

A Personal Store stores the address book information of a user. It contains the definition of all the address books that a user has created, along with all the entries in those address books. Personal Stores are represented as URLs, which describe the directory instance in which they are located and the DN within that particular directory instance.

A DBMap is a collection of DBs of the same type.

A DB contains a collection of Personal Stores. The address book can access any number of DBs. Every DB is defined by an identifier in configuration file that defines the connection parameters for that DB. A DB of different type points to different DB locations.

The psRoot attribute can be turned on or off using the **iwcadmin** command-line interface by setting the **ab.useuserpsroot** to **false**. If set to false, Convergence uses the Default Server value that is set in the Convergence configuration.

Set **ab.useuserpsroot** to **true** to use the user's psRoot value. At runtime, the value of psRoot attribute is resolved to a directory instance using *ldaphost* and *ldapport*. Based on *ldaphost* and *ldapport*, the Identifier to the database will be resolved. Here *Identifier* is an arbitrary string that distinguishes one instance from the other.

Setting the psRoot Value Automatically

When a new user logs in, default values are set for the psRoot attribute in the user's entry. For new users, a psRoot value is constructed by using the psRoot pattern and *DefaultServer* defined in the default configuration. For example, when you use the default psRoot pattern, the default psRoot value is in the format:

```
ldap://default_server_host:port/piPStoreOwner=%U,o=%D,o=PiServerDb
```

where:

- *%U* is the login ID of the user. For example, jsmith.
- *%D* is the domain of the user. For example *siroe.com*.

The following example shows how to configure horizontal scalability of address book in a deployment where there are two directory servers: *ds1.siroe.com*.

Use following commands to enable horizontal scalability:

To configure personal address book to use directory server *ds1.siroe.com*:

```
iwcadmin -o ab.pstore.[psidentifier1].ldaphost -v ds1.siroe.com
iwcadmin -o ab.pstore.[psidentifier1].ldapport -v 389
iwcadmin -o ab.pstore.[psidentifier1].ldapbinddn -v "cn=Directory Manager"
iwcadmin -o ab.pstore.[psidentifier1].ldapbindcred -v abbbbc
```

To configure personal address book to use directory server *ds2.siroe.com*:

```
iwcadmin -o ab.pstore.[psidentifier2].ldaphost -v ds2.siroe.com
iwcadmin -o ab.pstore.[psidentifier2].ldapport -v 389
iwcadmin -o ab.pstore.[psidentifier2].ldapbinddn -v "cn=Directory Manager"
iwcadmin -o ab.pstore.[psidentifier2].ldapbindcred -v aaaaabbbb
```

To enable horizontal scalability, you must set the **ab.useuserpsroot** configuration parameter to **true**:

```
iwcadmin -o ab.useuserpsroot -v true
```

To set the default server, you must set the **ab.pstore.defaultserver** configuration parameter to the personal store identifier:

```
iwcadmin -o ab.pstore.defaultserver -v psidentifier2
```

Where *psidentifier2* is default server. If psRoot attribute is not present, *ds2.siroe.com* will be used for personal address book. When a new user logs in, default values are set for the psRoot attribute in the user's entry.

Setting Up Address Book JMQ Notification

Convergence provides a notification module that enables administrators to plug-in a JMS based notification service. The notification module publishes messages to the configured JMS brokers.

Convergence provides a notification service for the Personal Address Book (PAB). The notification module publishes notification messages to a JMS broker when certain state changes occur in a user's PAB. The notification messages are published on a JMS topic or a queue that can be consumed by an appropriate consumer.

This technical note provides an overview of Convergence's address book notification service and provides information about how to configure the notification service.

Prerequisites for Setting up the Notification Service

This section provides information on the prerequisites for the working with this feature. The administrator must have working knowledge of the following products and technologies:

- Administration knowledge of GlassFish Server - The administrator must create the JMS-based connection factories and destination resources.
- Convergence Administration - The administrator must have working knowledge of administering Convergence.

Configuring Convergence

To configure Convergence for address book notification service, you must perform the following high-level steps:

1. Set up the message queue notification service configuration parameters.
2. Choose a notification strategy.
3. Set the Convergence configuration parameters to set notifications.

Message Queue Notification Service Configuration

To make use of the notification service in Convergence, you must first enable Convergence to use the notification service. To do this, enable the following Convergence parameters.

- **notify.service.enable** - Set this parameter to **true** to enable the notification service.

  ```
  iwcadmin -o notify.service.enable -v true
  ```

The address book notification service publishes notifications to multiple destinations. The destination can be a topic or a queue. Each destination is uniquely identified by a service name. The service name is then resolved Convergence and the notifications are

sent to the destination based on the destination type, destination name, and the connection attributes of the service name.

The service name is any unique string. The service name acts as an identifier for a particular destination. For each service name, the various attributes such as the destination type, destination name, and the connection attributes must be set.

- **notify.mq.[serviceName].enable**: Set this parameter to **true** to enable the notification service for a destination. For example:

  ```
  iwcadmin -o notify.mq.[serviceName1].enable -v true
  ```

- **notify.mq.[serviceName].destinationtype**: For each destination, the destination type must be set. The valid values are: **TOPIC** or **QUEUE**. For example:

  ```
  iwcadmin -o notify.mq.[serviceName1].destinationtype -v TOPIC
  ```

- **notify.mq.[serviceName].destinationname**: The destination name. This name must match the corresponding JMS connection in the GlassFish Server. For example:

  ```
  iwcadmin -o notify.mq.[serviceName1].destinationname -v destinationName1
  ```

- **notify.mq.[serviceName].connection**: Connection attribute.

  ```
  iwcadmin -o notify.mq.[serviceName1].connection -v JMS_connection_factory
  ```

- **notify.mq.[serviceName1].resourcetype**: Specifies the *resourcetype* and needs to be set to **producer** for address book notifications to work

  ```
  iwcadmin -o notify.mq.[serviceName1].resourcetype -v producer
  ```

  If you do not set this parameter, address book notifications do not work as expected.

  The values for the *destinationtype*, *destinationname*, and *connection* must be the same as the settings for the JMS resources: Connection Factory and JMS resources when configuring GlassFish Server.

Notification Strategies

Convergence provides various notification strategies. You can employ a notification strategy based on how you want to publish and broadcast the notifications. You can set up the following types of notification strategies:

- **User Specific Notification** Use the user specific notification strategy to trigger notifications to be published based on the state changes of particular contacts in your address books. To enable notification for a per-user, you must set the **abEventNotificationDestination** attribute in the user's LDAP entry to the name of the destination to which the notifications must be published.

  The user must have the **SunUCPreferences** object class available in the LDAP.

- **Notification for All Users** To enable notification for all users, you must set the following parameters:

  – **ab.pstore.notification.destination**

  – **ab.pstore.notification.notifyall**

- **Domain Based Configuration** If you want to trigger notifications to be published based on the domains, you must set the appropriate domain level attributes. For example:

```
iwcadmin -o ab.{siroe.com}.psrootpattern  -v ldap:///
piPStoreOwner=%U,o=%D,o=PiServerDb
iwcadmin -o ab.{siroe.com}.pstore.defaultserver -v myldap
iwcadmin -o ab.{siroe.com}.pstore.[myldap].ldaphost -v newLdap.siroe.com
iwcadmin -o ab.{siroe.com}.pstore.[myldap].ldapport -v 389
iwcadmin -o ab.{siroe.com}.pstore.[myldap].ldapbinddn -v 'cn=Directory Manager'
iwcadmin -o ab.{siroe.com}.pstore.[myldap].ldapbindcred -v password
```

The following example shows how to set domain level notifications. In this example, the triggers have been set on the Create Contact, Create Contact Photo, Delete Contact, and Modify Contact actions. You can set the triggers based on your requirements.

```
iwcadmin -o ab.{siroe.com}.pstore.notification.destination  -v serviceName1
iwcadmin -o ab.{siroe.com}.pstore.notification.event.createcontact -v  true
iwcadmin -o ab.{siroe.com}.pstore.notification.event.createcontactphoto -v  true
iwcadmin -o ab.{siroe.com}.pstore.notification.event.deletecontact -v  true
iwcadmin -o ab.{siroe.com}.pstore.notification.event.modifycontact - v true
iwcadmin -o ab.{siroe.com}.pstore.notification.notifyall -v  true
```

Note: You must restart the GlassFish Server on which Convergence is deployed after making the configuration changes.

Setting Event Notification Triggers

Table 6–1 lists the types of notifications provided by Convergence:

Table 6–1 Notification Triggers

Notification Trigger	Configuration Parameter	Description
Create Contact	ab.pstore.notification.event .createcontact	This parameter, when set to *true*, triggers a notification when a new contact is created.
Delete Contact	ab.pstore.notification.event .deletecontact	This parameter, when set to *true* triggers a notification when a contact is deleted.
Modify Contact	ab.pstore.notification.event .modifycontact	This parameter, when set to *true*, triggers a notification when a contact is modified.
Create Contact Photo	ab.pstore.notification.event .createcontactphoto	This parameter, when set to *true* triggers a notification when a user adds a photo for a contact.

Configuring GlassFish Server

This section provides information about the various configuration steps that need to be performed in GlassFish for the notification service to work. The JMS connection factory and destination resources must be set.

The following examples show how to create the JMS connection factory and destination resources. You can create these either using the-asadmin-command-line-utility or by using the Administration Console.

1. Create the JMS Connection Factory.

```
GlassFish_Home/bin/asadmin -u admin -p 4848 -passwordfile /export/pass
create-jms-resource --restype javax.jms.TopicConnectionFactory --description
"example of creating a JMS connection factory" jms/ConnectionFactory
```

Note: To configure the JMS connection factory on a remote host, set the appropriate remote host options. Otherwise, the remote host will not receive JMS messages. For more information on setting remote host options, refer to the GlassFish Server help:

```
GlassFish_Home/bin/asadmin create-jms-resource - help
```

2. Create the JMS Destination Queue.

```
GlassFish_Home/bin/asadmin -u admin -p 4848 --passwordfile /export/pass
create-jms-resource --restype javax.jms.Queue jms/Queue
```

3. Create the JMS Destination Topic.

```
GlassFish_Home/bin/asadmin -u admin -p 4848 --passwordfile /export/pass
create-jms-resource --restype javax.jms.Topic jms/Topic
```

Troubleshooting the Notification Service

When configuring Convergence for notification, use the notification log levels to troubleshoot any problems that you encounter when working with this feature. You can set the log levels for the notification service. Using the **iwcadmin** command, you can set the logging levels for the *log.NOTIFY.level* parameter.

Data Format used for Notification Service

This section provides information about the format in which data is passed over as part of the notification message. The notification message must be used by the consumers of the notification service. The notification message contains the vCard of the user along with the following details:

- *message*
- *domain*
- *bookid*
- *timestamp*
- *uid*
- *operation*
- *entryid*

Message Format: Create Contact

The following is the data format of the notification sent when a new contact is created:

```
Message:BEGIN:VCARD
VERSION:3.0
PROFILE:VCARD
PRODID:Sun Address Book
UID:
FN:user1
N:d;user1;;;
NICKNAME:
ORG:siroe;
TITLE:Mr
ANNIVERSARYDATE:--
BDAY:--
```

```
TEL;TYPE=WORK,PREF:
TEL;TYPE=HOME:
TEL;TYPE=CELL:
TEL;TYPE=PAGER:
TEL;TYPE=FAX:
EMAIL;TYPE=INTERNET;TYPE=WORK;TYPE=PREF:user1@siroe.com
EMAIL;TYPE=INTERNET;TYPE=HOME:
EMAIL;TYPE=INTERNET;TYPE=OTHER:
ADR;TYPE=HOME:;;;;;;
ADR;TYPE=WORK:;;;;;;
ADR;TYPE=OTHER:;;;;;;
NOTE:
URL;TYPE=HOME:
URL;TYPE=WORK:
X-IMADDR1:
X-IMSERVICE1:SunIM
X-IMADDR2:
X-IMSERVICE2:AIM
CALURI:
FBURL:
END:VCARD

domain  siroe.com
bookid  e11dbf2a4c610
timestamp 20090524T205226Z
uid ngc5
operation CreateContact
entryid e12174653ce40
```

Message Format: Modify Contact

The following is the data format of the notification sent when a contact information is modified:

```
Modify Contact:
===========
Message: BEGIN:VCARD
VERSION:3.0
PROFILE:VCARD
PRODID:Sun Address Book
UID:e1218771a5d22
FN:user1 d
N:d;user1;;;
NICKNAME:
ORG:sun;
TITLE:mts
ANNIVERSARYDATE:--
BDAY:--
TEL;TYPE=WORK,PREF:
TEL;TYPE=HOME:
TEL;TYPE=CELL:
TEL;TYPE=PAGER:
TEL;TYPE=FAX:
EMAIL;TYPE=INTERNET;TYPE=WORK;TYPE=PREF:user2@siroe.com
EMAIL;TYPE=INTERNET;TYPE=HOME:user2@siroe.com
EMAIL;TYPE=INTERNET;TYPE=OTHER:
ADR;TYPE=HOME:;;;;;;
ADR;TYPE=WORK:;;;;;;
ADR;TYPE=OTHER:;;;;;;
NOTE:
URL;TYPE=HOME:
```

```
URL;TYPE=WORK:
X-IMADDR1:
X-IMSERVICE1:SunIM
X-IMADDR2:
X-IMSERVICE2:AIM
CALURI:
FBURL:
END:VCARD

domain  siroe.com
bookid  e11dbf2a4c610
timestamp       20090524T205309Z
uid     ngc5
operation       ModifyContact
entryid e12174653ce40
```

Message Format: Delete Contact

The following is the data format of the notification sent when a contact is deleted:

```
Delete Contact:
===========
Message: null
domain  siroe.com
bookid  e11dbf2a4c610
timestamp       20090524T205425Z
uid     ngc5
operation       DeleteContact
entryid e121746661f21
```

Message Format: Create Contact Photo

The following is the data format of the notification sent when a photo is assigned to a contact:

```
Message: R0lGODlhMgArAHAAACH/C05FVFNDQVBFMi4wAwEAAAAh/
glnaWY0ajEyMTYAIfkEBSAABwAsAAAAADIAKwCCAAAAAAD/AP8AhAAA/wAA//8AAAAAAAAA/
54utz+MMoJgb2W6n2L/
961jY0FnmdGcgDqfuoqtW8dyhFt2wDZP7qdyfWjAATFxVDYQcVmSEeQ13RqjsnDNKXD1LKM3hEZ28Ka3hd
YgUW6o9ovumoFtgX4n/
m8XKoreIFFeyF0OwVrbIJhVH6HiWxPcVSHKTMqaZWaiJBaGIaEX1ySYRiTMJuoIJ8cTqlbYiJXrpWwSqQV
To5cRLcjUwAEwru7MJG4uSHCBMHMnzxKeb80zc3Mmpimv8rL19iRiz7c3rXg4eKcoZaKY3ArxXKlbZ2X2G
usOMdqyPkVn58DZPUzwk/gwBz8jh28RM/gQnkF/
y38FOBCRSABKmbMmCdwQruNIC+WCLmRjIyPIc8pCkTS3a9AMAVJwhBT5bR/
HT3hfMizQQIAIfkEBSAABwAsAAAAADIAKwCCAAAAAAD/AP8AhAAA/wAA//8AAAAAAAAA/
54utz+MMqpgL2W6n2L/
961jY0FnmdGcgDqfuoqtW8dyhFt2wDu6CkgD9LTmFCYXajIAAiYs1dSWYBWnhQhaFrVbkvYiXd55CIXFwH
W2qxNzanrc+5kH8YwHTx+V6vrdngue19yfnZ3VHlHUk2HOYp5O1YiRIKRQSqVJR1bmCGZGIFwl4Nxog8Yi
SmYWoicTK6KriMxXoyDuDdGlXi6XQTBQq9gYatSosLBBHq1qnelIcsAwku1h0XR0svMuxx+T9mR1NVVK0
5/xtpdNMREdDHQre6Wm03a8T4zv2T6LAADRAn0x+uVPYKp8v1QiBANw3v0ZAg0+
KyhhQAYAwDAGCxI40VUOCwIyEhS44+SGR9JHFkSHCU/KFWSQAfuz8E0NV1KnEiP58O
GQBkkAAA7
domain  siroe.com
bookid  e1223b29da380
timestamp  20090702T111156Z
uid     ngc2
operation  SetContactPhoto
entryid e1223b29daad1
```

Configuring Address Book to Use Different Directory Server from the User Group Server

To configure Personal Address Book to use directory server other than user group directory server, set the following configuration parameters:

- **ab.pstore.[***identifier***].ldaphost** - Set this parameter to the host name of the LDAP server.
- **ab.pstore.[***identifier***].ldapport** - Set this parameter to the port number on which the LDAP server listens.
- **ab.pstore.[***identifier***.ldapbinddn** - Set this parameter to the LDAP bind dn value of the LDAP server.
- **ab.pstore.[***identifier***].ldapbindcred** - Set this parameter to the Bind credentials of the LDAP server.

The following example shows the configuration parameter settings:

```
iwcadmin -o ab.pstore.[psidentifier1].ldaphost -v host.siroe.com
iwcadmin -o ab.pstore.[psidentifier1].ldapport -v 400
iwcadmin -o ab.pstore.[psidentifier1].ldapbinddn -v "cn=Directory Manager"
iwcadmin -o ab.pstore.[psidentifier1].ldapbindcred -v dmcredentials
```

Personal store can be configured with multiple directory servers. In this example **psidentifier1** is used to identify personal store configuration for **siroe.com**.

If the configured directory server needs to act as the personal store's default server, then set the **ab.pstore.defaultserver** configuration parameter. For example:

```
iwcadmin -o ab.pstore.defaultserver -v psidentifier1
```

Configuring the Corporate Directory

To configure corporate directory to use directory server other than user group directory server, set the following configuration parameters:

- **ab.corpdir.[***identifier***].ldaphost**
- **ab.corpdir.[***identifier***].ldapport**
- **ab.corpdir.[***identifier***].ldapbinddn**
- **ab.corpdir.[***identifier***].ldapbindcred**

The following example has the configuration parameters settings:

```
iwcadmin -o ab.corpdir.[identifier].ldaphost -v host.siroe.com
iwcadmin -o ab.corpdir.[identifier].ldapport -v 400
iwcadmin -o ab.corpdir.[identifier].ldapbinddn -v "cn=Directory Manager"
iwcadmin -o ab.corpdir.[identifier].ldapbindcred -v xyzxyz
```

Where *identifier* identifies the corporate directory configuration for **host.siroe.com**. For a single corporate directory configuration, you must use **default** as the identifier.

See "Setting Up Multiple Corporate Directories" for information about configuring and enabling multiple corporate directories.

Enabling Address Autocomplete for the Corporate Directory

To enable autocomplete of email address for Corporate Directory, you must set the **client.enablecorpabautocomplete** configuration parameter to **true**.

```
iwcadmin -o client.enablecorpabautocomplete -v true
```

Note: The search results will appear in the Convergence client, after the first three characters of the name or email address are typed.

Setting Up Domain-Based Configuration for Address Book

You can set up a domain based configuration for Personal Address Book and Corporate Directory.

To set up domain-based configuration for Personal Address Book, set the following parameters by using the **iwcadmin** command:

- **ab.{***identifier***}.psrootpattern**
- **ab.{***identifier***}.pstore.defaultserver**
- **ab.{***identifier***}.pstore.[domain].ldaphost**
- **ab.{***identifier***}.pstore.[***domain***].ldapport**
- **ab.{***identifier***}.pstore.[***domain***].ldapbinddn**
- **ab.{***identifier***}.pstore.[***domain***].ldapbindcred**

The following example shows the configuration parameter settings:

```
iwcadmin -o ab.{domain.com}.psrootpattern -v ldap:///
piPStoreOwner=%U,o=%D,o=PiServerDb
iwcadmin -o ab.{domain.com}.pstore.defaultserver -v domainid1
iwcadmin -o ab.{domain.com}.pstore.[domainid1].ldaphost -v host.xyz.com
iwcadmin -o ab.{domain.com}.pstore.[domainid1].ldapport -v 400
iwcadmin -o ab.{domain.com}.pstore.[domainid1].ldapbinddn -v "cn=Directory
Manager"
iwcadmin -o ab.{domain.com}.pstore.[domainid1].ldapbindcred -v xyzcred
```

Where *domain.com* is the domain (within curly braces).

All the configuration data for the domain **domain.com** is grouped in to one logical set identified by using the identifier **domainid1**.

The example shows the minimum set of configuration parameters that you need to set for the domain based configuration for Personal Address Book. However, you can set other configuration parameters.

To set the **lookthrulimit** to **2000** for Personal Address Book in domain **domain.com**, type the following command:

```
iwcadmin -o ab.{domain.com}.pstore.lookthrulimit -v 2000.
```

To set up domain-based configuration for Corporate Directory:

1. Set the following configuration parameters:
   - **ab.{***identifier***}.corpdir.[***domain***].urlmatch**
   - **ab.{***identifier***}.corpdir.[***domain***].searchattr**
   - **ab.{***identifier***}.corpdir.[***domain***].lookthrulimit**
   - **ab.{***identifier***}.corpdir.[***domain***].ldaphost**
   - **ab.{***identifier***}.corpdir.[***domain***].ldapport**

- **ab.{***identifier***}.corpdir.[***domain***].ldapbinddn**
- **ab.{***identifier***}.corpdir.[***domain***].ldapbindcred**

For example:

```
iwcadmin -o ab.{domain.com}.corpdir.[corpdomainid1].urlmatch
-v ldap://corp-directory1
iwcadmin -o ab.{domain.com}.corpdir.[corpdomainid1].searchattr
-v entry/displayname,@uid
iwcadmin -o ab.{domain.com}.corpdir.[corpdomainid1].lookthrulimit
-v 3000
iwcadmin -o ab.{domain.com}.corpdir.[corpdomainid1].ldaphost
-v host.abc.com
iwcadmin -o ab.{domain.com}.corpdir.[corpdomainid1].ldapport
-v 389
iwcadmin -o ab.{domain.com}.corpdir.[corpdomainid1].ldapbinddn
-v "cn=Directory Manager"
iwcadmin -o ab.{domain.com}.corpdir.[corpdomainid1].ldapbindcred
-v abcabc
```

Where *domain.com* specifies the domain. All the configuration data for the domain **domain.com** is grouped in to one logical set identified by using identifier **corpdomainid1**.

Note: The value for the **urlmatch** configuration parameter must be unique. Format for **urlmatch** is ldap://*unique_value* or ldap://*host*:*port*/DN e.g. ldap://corp-directory1 ,ldap://corporatedirectory2, ldap://somehost:390/ou=people,o=ab.org etc.

First time when user does address book operation (apart from login.wabp), corporate directory entry (under piPStoreOwner=*user*, o=*domain*, o=PiServerDb) with piRemotePiURL attribute value as urlmatch gets created. After this if urlmatch is changed, either delete such entries so that this entry gets created when first AB command is issued or update corporate directory entry for all users with new urlmatch value.

---

2. Copy **dictionary-***locale***.xml** (for example: **dictionary-en.xml**) from *Convergence_Home*/**config/templates/ab/domain/defaultps** to *Convergence_Home*/**config/templates/ab/domain/***domain-directory*. The **dictionary-***locale***.xml** file can be updated in order to change or to customize display name and description.

Disabling the Corporate Directory in Specific Domains

In some cases, you might want to disable your corporate directory in certain domains. To do so, follow these steps:

1. Set both personal address book and Corporate Directory settings as described in "Setting Up Domain-Based Configuration for Address Book".
2. Disable the Corporate Directory for the specific domain:

   ```
   iwcadmin -o ab.{domain.com}.corpdir.[default].enable" -v false
   ```

3. Restart GlassFish Server.

Note: You can ignore errors or exceptions in the log files.

Changing the Default Corporate Directory Search Filter in Address Book

To change the default corporate directory search filter, set the **ab.corpdir.[***identifier***].searchfilter** configuration parameter with the search criteria you want to base your corporate directory searches on.

The following example shows the usage of search customization:

```
iwcadmin -o ab.corpdir.[default].searchattr -v entry/displayname,@uid,person/
surname
iwcadmin -o ab.corpdir.[default].searchfilter -v
'(&(&([filter])(|(objectClass=GROUPOFUNIQUENAMES)(objectClass=GROUPOFURLS) \\
(objectClass=ICSCALENDARRESOURCE)(objectClass=INETORGPERSON)))(objectClass=*))'
```

Where **[filter]** is replaced with the search generated by the **ab.corpdir.[***identifier***].searchattr** configuration option.

The example produced the following LDAP output in the corporate LDAP directory access logs when an end-user searched for **"bob"**:

```
[13/Oct/2008:11:51:54 +1100] conn=686404 op=30 msgId=576 - SRCH
base="o=sun.com,o=isp" scope=2
filter="(&(&(|(|(cn=bob*)(uid=bob*))(sn=bob*))(|(objectClass=GROUPOFUNIQUENAMES)(o
bjectClass=GROUPOFURLS)
(objectClass=ICSCALENDARRESOURCE)(objectClass=INETORGPERSON)))(objectClass=*))"
attrs="objectClass createTimestamp cn uid description mail multiLineDescription
modifyTimestamp"
```

Configuring Virtual List View for Convergence Corporate Directory

Follow these steps to configure Convergence to make use of virtual list view (VLV):

1. Configure Directory Server with VLV. For more information on creating and managing browsing indexes in Directory Server:
   - Configuring VLV Browsing Indexes for Directory Server
   - See *Directory Server Administration Guide*.
2. Set the VLV filter and scope in the corporate directory.

   ```
   iwcadmin -o ab.corpdir.[default].vlvfilter -v "(&(mail=*)(cn=*))"
   iwcadmin -o ab.corpdir.[default].vlvscope -v 2
   ```

3. Enable the **ab.corpdir.[default].vlvpaging** configuration parameter to **true**.

   ```
   iwcadmin -o ab.corpdir.[default].vlvpaging -v true
   ```

About Supported vCard Standards

Convergence supports the following vCard standards:

- vCard 2.1
- vCard 3.0

Convergence supports the following encoding formats for importing and exporting vCard:

- UTF-8

- ISO-8859-1
- BIG5
- EUC-CN
- EUC-JP
- EUC-KR
- SHIFT_JIS

Changing the Locale Character Set for Importing or Exporting vCard Entries

Convergence supports the following locales by default:

- English
- Japanese
- French
- German
- Spanish
- Korean
- Traditional Chinese
- Simplified Chinese

For each locale, configuration parameters for import and export exist in the Convergence server. By default, these configuration parameters are assigned a character encoding when you install Convergence.

Table 6–2 shows the default encoding formats for locales when Convergence is installed. The table also lists the configuration parameters that are assigned for storing the import and export preference for the locale.

Table 6–2 Supported Default vCard Locales

Locale	Encoding	Import Parameter	Export Parameter
English	UTF-8	ab.import.vcard.misc.en	ab.export.vcard.misc.en
Japanese	UTF_8	ab.import.vcard.misc.ja	ab.export.vcard.misc.ja
French	UTF-8	ab.import.vcard.misc.fr	ab.export.vcard.misc.fr
German	UTF-8	ab.import.vcard.misc.de	ab.export.vcard.misc.de
Korean	UTF-8	ab.import.vcard.misc.ko	ab.export.vcard.misc.ko
Traditional Chinese	UTF-8	ab.import.vcard.misc.zh-tw	ab.export.vcard.misc.zh-tw
Simplified Chinese	UTF-8	ab.import.vcard.misc.zh-cn	ab.export.vcard.misc.zh-cn

In the previous table, the character encoding for English is set to UTF-8. This setting means that when you import or export vCard contacts to or from the Convergence client, the vCard entries are imported or exported in the UTF-8 format character set. In this case, UTF-8 is the default setting for English users.

To enable the Convergence client to import or export vCard entries to other character sets, set the address book vCard configuration parameter in the Convergence server.

Type the **iwcadmin** command to set the import and export character set preferences for the configuration parameters of the locale. This command enables you to change the character set encoding for importing or exporting vCard entries.

To change the character encoding for the Japanese user vCard from UTF-8 to Shift_JIS for example, set the corresponding configuration parameters for import and export.

To set the character encoding to import vCard entries for the Japanese locale, type the following command:

```
iwcadmin -o ab.import.vcard.misc.ja -v Shift_JIS
```

To set the character encoding to export vCard entries for the Japanese locale, type the following command:

```
iwcadmin -o ab.export.vcard.misc.ja -v Shift_JIS
```

The vCard entries are imported or exported in the Shift_JIS encoding character set.

Note: You must set the same character set encoding for both import and export for a locale.

Enabling Contact Export and Import with Photo in vCard

vCard 3.0 enables users to include photos in their contacts. By default, Convergence does not import or export photos of your contacts. If you want photos to be imported or exported, you must enable the **ab.exportphoto** and **ab.importphoto** configuration parameters.

To enable exporting of contacts with photo in Vcard 3.0 format, type the following command:

```
iwcadmin -o ab.exportphoto -v true
```

To import contacts with photo in Vcard 3.0 format, type the following command:

```
iwcadmin -o ab.importphoto -v true
```

Hiding Administrator Accounts in the Default Domain Corporate Directory

When looking in the Corporate Directory of the default domain all the administrative accounts are being displayed. These can be hidden by using psIncludeInGAB attribute in the ldap server. The default value of this attribute is true.

If you want to hide users in the Corporate Directory, set in a first step the psIncludeInGAB attribute to false for these users. Next, the corporate directory search filter needs to exclude these users with their psIncludeInGAB attribute set to false. For example:

```
iwcadmin -o ab.corpdir.[default].searchfilter  -v
"(&(&(&([filter])(|(objectClass=GROUPOFUNIQUENAMES)(objectClass=GROUPOFURLS) \\
(objectClass=ICSCALENDARRESOURCE)
(objectClass=INETORGPERSON)))(objectClass=*))(!(psIncludeInGAB=false)))"
```

About Personal Address Book Contacts Deleted by the End User

If a contact has been deleted by the end user, Convergence determines what do to with that information based on how you set the **ab.pstore.deleteperm** configuration parameter. If you set the parameter to **true**, the contact is deleted from the user's

personal address book entries on Directory Server. If, however, you set **ab.ps.deleteperm** to **false**, the following attribute/value pair is added to the deleted contact in Directory Server:

```
delete: true
```

The contact no longer appears in Convergence as if it were permanently deleted from the Directory Server.

This task can be particularly useful when you are synchronizing deleted contact entries in Microsoft Outlook and Convergence when using Connector for Microsoft Outlook.

Enhancing Corporate Directory Search Using VLV Indexing

Virtual List View (VLV) index, also known as browsing index, is similar to indexes or views in a database. Create the VLV indexes to reduce the time taken to search the LDAP entries. If a Directory Server deployment contains several LDAP entries, then searching the entries takes considerably more time. Directory Server enables you to create indexes that reduce the search time.

Creating the VLV Index in the Directory Server

You must set the following parameters in the LDIF file to enable VLV indexes in the directory server.

- *search_base*
- *vlv_search_filter*
- *vlv_sort_attribute*
- *vlv_scope*

If multiple back-end user/group Directory Servers are configured for a system, you will need to create indexes for each user/group Directory Server instance.

You require the Directory Server settings information before setting the VLV browsing indexes. Directory Server settings are present in the **dse.ldif** file in the *DS_Home*/**config** directory (where *DS_Home* is the directory in which the directory server software is installed). Note the value of the *cn* attribute.

The following is a code sample of the **dse.ldif** file:

```
dn: cn=isp,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: extensibleObject
objectClass: nsBackendInstance
cn: isp
creatorsName: cn=directory manager
modifiersName: cn=directory manager
entrydn: cn=isp,cn=ldbm database,cn=plugins,cn=config
numSubordinates: 4
nsslapd-suffix: o=isp
nsslapd-cachesize: -1
nsslapd-cachememsize: 10485760
nsslapd-readonly: off
nsslapd-require-index: off
nsslapd-directory: /var/opt/SUNWdsee/dsins1/db/isp
```

Create two files in a temporary location after you identify the required Directory Server setting entries. For example, **tmp1.vlv** and **tmp2.vlv**. These files must contain information about various indexes and search options that you can create on Directory Server.

The **temp1.vlv** file should have the following parameters.

temp1.vlv

```
dn: cn=browsing_index,cn=database_name,cn=ldbm database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvSearch
cn: browsing_index
vlvbase: o=isp
vlvscope: 2
vlvfilter: (&(mail=*)(cn=*))
aci: (targetattr="*")(version 3.0; acl "VLV for Anonymous";
allow (read,search,compare) userdn="ldap:///anyone";)
```

Table 6–3 lists the parameters for **temp1.vlv**.

Table 6–3 Configuration Parameters for temp1.vlv File

Parameters	Description
browsing-index	Any name
database-name	Same as the existing database name that is given when directory server is configured by running directory preparation tool (**comm_dssetup.pl**). This database name should not be changed.
vlvbase	Value from which you want the search to proceed. For example, instead of **dc=example,dc=siroe,dc=com** you can provide a subtree **o=example.siroe.com,dc=example,dc=siroe,dc=com**.
vlvscope	Similar to the LDAP protocol scoping number: **0** -Indicates searching only the base level entry **1** - Indicates searching only the entries at one level below the search base. If you set *vlvScope* to *1*, you must create a *vlvSearch* or *vlvIndex* for each organization unit (*ou*) where you want a VLV index. **2** - Indicates searching of the entries at all levels and all its descendants.
vlvfilter	Filter that is used to match and filter results. When you perform a search, only those LDAP entries that have both *mail* and *cn* attributes defined are returned. Some of the entries might not have *mail* attribute. If they do not have the *mail* attribute, modify the *vlvfilter* as *(|(mail=*)(cn=*))*. The OR operator is used instead of the AND operator. Only include contacts (for example: objectclass=inetorgperson) while creating the VLV index so that the address book search excludes groups, resources and any other entries that are not contacts. This is mandatory for Convergence 2. Leave an extra blank line after the last line in both the **temp1.vlv** and **temp2.vlv** files to make sure that all the entries in the files are read when the LDAP is modified.

The **temp2.vlv** file should have the following parameters:

temp2.vlv

```
dn: cn=Sort by cn,cn=browsing_index,cn=database_name,cn=ldbm
database,cn=plugins,cn=config
changetype: add
```

```
objectClass: top
objectClass: vlvIndex
cn: Sort by cn
vlvSort: cn
```

The **temp1.vlv** and **temp2.vlv** files specify what to index and the parameter by which results must be sorted. You need to create multiple indexes by creating the following files: **temp3.vlv**, **temp4.vlv**, **temp5.vlv, temp6.vlv**, **temp7.vlv**, **temp8.vlv**, and **temp9.vlv**.

The **temp3.vlv** file should have the following parameters:

```
dn: cn=Reverse Sort by cn,cn=browsing_index,cn=database_name,cn=ldbm
database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvIndex
cn: Reverse Sort by cn
vlvSort: -cn
```

The **temp4.vlv** file should have the following parameters:

```
dn: cn=Sort by sn,cn=browsing_index,cn=database_name,cn=ldbm
database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvIndex
cn: Sort by sn
vlvSort: sn
```

The **temp5.vlv** file should have the following parameters:

```
dn: cn=Reverse Sort by sn,cn=browsing_index,cn=database_name,cn=ldbm
database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvIndex
cn: Reverse Sort by sn
vlvSort: -sn
```

The **temp6.vlv** file should have the following parameters:

```
dn: cn=Sort by mail,cn=browsing_index,cn=database_name,cn=ldbm
database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvIndex
cn: Sort by mail
vlvSort: mail
```

The **temp7.vlv** file should have the following parameters:

```
dn: cn=Reverse Sort by mail,cn=browsing_index,cn=database_name,cn=ldbm
database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvIndex
cn: Reverse Sort by mail
vlvSort: -mail
```

The **temp8.vlv** file should have the following parameters:

```
dn: cn=Sort by givenname,cn=browsing_index,cn=database_name,cn=ldbm
```

```
database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvIndex
cn: Sort by givenname
vlvSort: givenname
```

The **temp9.vlv** file should have the following parameters:

```
dn: cn=Reverse Sort by givenname,cn=browsing_index,cn=database_name,cn=ldbm
database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvIndex
cn: Reverse Sort by givenname
vlvSort: -givenname
```

To modify the LDAP using **tmp1.vlv** and **tmp2.vlv** enter the following command:

```
ldapmodify -h directory_server_fully_qualified_host_name -p directory_server_port
-D "cn=Directory Manager" -f path/temp1.vlv
ldapmodify -h directory_server_fully_qualified_host_name -p directory_server_port
-D "cn=Directory Manager" -f path/temp2.vlv
```

Similarly, you can modify the LDAP by using all other VLV files:

Generating Indexes

Generate the indexes for the settings to take effect. Perform the following steps during a scheduled change window to restart Directory Server.

Perform the following steps to generate the indexes:

1. Change the directory to the Directory Server installation.

   ```
   cd /opt/SUNWdsee/ds6/bin
   ```

2. Stop the Directory Server instance.

   ```
   dsadm stop /var/opt/SUNWdsee/dsins1/
   ```

3. Populate the index entries by using the **dsadm reindex** command. The **reindex** option requires you to provide the *vlv_sort_attribute*, the path to the Directory Server instance, and the value of the user group base.

   ```
   dsadm reindex -l -t "Sort by cn" /var/opt/SUNWdsee/dsins1/ "o=isp"
   ```

4. Start the Directory Server instance.

   ```
   dsadm start /var/opt/SUNWdsee/dsins1/
   ```

If you require multiple sort attributes for **tmp3.vlv** and **tmp4.vlv**, generate indexes for each of *cn*, *sn*, and *mail*.

Configuring Convergence

You need to configure Convergence to use the indexes after generating the indexes for Directory Server. Using the **iwcadmin** command, set the following Convergence parameters:

- **ab.corpdir.[default].vlvfilter**
- **ab.corpdir.[default].vlvscope**

- **ab.corpdir.[default].vlvpaging**
- **ab.corpdir.[default].vlvsortby**
- **ab.corpdir.[default].vlvsearchbase**
- **ab.corpdir.[default].vlvsortby**

For example:

```
iwcadmin -u admin_user_id -o ab.corpdir.[default].vlvfilter -v "(&(mail=*)(cn=*))"
iwcadmin -o ab.corpdir.[default].vlvscope -v 2
iwcadmin -o ab.corpdir.[default].vlvpaging -v true
iwcadmin -o ab.corpdir.[default].vlvsortby -v "entry/displayname,person/
surname,email,person/givenname"
iwcadmin -o ab.corpdir.[default].vlvsearchbase -v "o=isp"
```

Note: The value for **ab.corpdir.[default].vlvfilter** is *(&(mail=*)(cn=*))*. This value should exactly match with the value provided in the Directory Server settings and the match should be a string match. It cannot even be *(&(cn=*)(mail=*))* because interchanging the *mail* and *cn* attributes causes a mismatch with the settings in the Directory Server.

The default corporate directory is used in the previous commands. The same set of commands apply to the nondefault corporate address book **ab.corpdir.[identifier].vlvscope** or the domain based corporate address book **ab.{identifier}.corpdir.[domain].vlvscope**.

The purpose of the parameter **vlvsortby** is that in case the server does not receive any **sortby** attribute from the client, the search results are sorted by the value set for this parameter. This applies only when VLV is setup.

You must restart the application after making any configuration changes in Convergence.

When you search a Corporate Address Book, you will see a drop down list in the Convergence client interface with the following search attributes:

- Display name
- Email
- First name
- Last name

You must have VLV indexes set up for these attributes to work. If VLV is not set, the default search is done by Display name.

Verifying the VLV Settings

To verify VLV settings:

1. For the VLV search to be active when you search the corporate directory, the following four entities sent by the Convergence server should match with the values in Directory Server:
   - Search base
   - Search scope

- VLV filter
- Sort attribute

Convergence only supports *cn*.

2. Log in to Convergence and type a search command in the corporate directory to check the Directory Server access log files. The two cases A and B with corresponding access log files of Directory Server are shown:

```
ldapsearch -D "cn=Directory Manager" -w password -b dc=example,dc=com -x -S cn
-G "0:3:name1" "(|(mail=*)(cn=*))" sn cn
```

Directory Server Access Log file A

```
[02/Dec/2008:12:46:52 +0100] conn=53 op=1 msgId=2 - SRCH
base="dc=example,dc=com" scope=2 filter="(|(mail=*)(cn=*))" attrs="sn cn"
[02/Dec/2008:12:46:52 +0100] conn=53 op=1 msgId=2 -  SORT cn
[02/Dec/2008:12:46:52 +0100] conn=53 op=1 msgId=2 -  VLV 0:3:rao 128:156 (0)
[02/Dec/2008:12:46:52 +0100] conn=53 op=1 msgId=2 - RESULT err=0 tag=101
nentries=4 etime=0
```

```
ldapsearch -D "cn=Directory Manager" -w password -b dc=example,dc=com -x -S sn
-G "0:3:name1" "(|(sn=*)(cn=*))" sn cn
```

Directory Server Access Log file B

```
[02/Dec/2008:12:45:34 +0100] conn=52 op=1 msgId=2 - SRCH
base="dc=example,dc=com" scope=2 filter="(|(sn=*)(cn=*))" attrs="sn cn"
[02/Dec/2008:12:45:34 +0100] conn=52 op=1 msgId=2 -  SORT sn (156)
[02/Dec/2008:12:45:34 +0100] conn=52 op=1 msgId=2 -  VLV 0:3:name1 97:156 (0)
[02/Dec/2008:12:45:34 +0100] conn=52 op=1 msgId=2 - RESULT err=0 tag=101
nentries=4 etime=0 notes=U
```

Searches in A and B might vary based on the *-S* sort attribute. In this case, VLV is setup with *cn* as the sort attribute.

VLV index is used only if **vlvSort**, **vlvbase**, **vlvscope**, and **vlvfilter** are matched with the given attributes. In case A all the attributes are matched. Hence the VLV index is used. In case B the VLV Index is not used as the sort attribute passed is *sn* whereas the setup has *cn*. See the *notes=U* in the Log file B displays that the search was unindexed. You can still continue to search with the *-S* server sort option. It will always be unindexed if no VLV Index is present that matches the specific search. Also notice the line "VLV 0:3:rao" which means that a VLV search was performed and from the point where a match was found 3 other entries were returned apart from the match. The zero before 3 signifies that no entries above the match in the sort order are returned.

For example, assume that the Directory Server has a VLV in the following sorted order:

```
person1 age1 address1 email1@siroe.com
person2 age2 address2 email2@siroe.com
person3 age3 address3 email3@siroe.com
person4 age4 address4 email4@siroe.com
person5 age5 address5 email5@siroe.com
person6 age6 address6 email6@siroe.com
person7 age7 address7 email7@siroe.com
person8 age8 address8 email8@siroe.com
```

Search for *cn=person4* with the range as *1:3:person4*.

```
person1 age1 address1 email1@siroe.com
                      person2 age2 address2 email2@siroe.com
                      person3 age3 address3 email3@siroe.com <-----
```

```
   Search match --->     person4 age4 address4 email4@siroe.com          |
                         person5 age5 address5 email5@siroe.com          | Range
of results returned
                         person6 age6 address6 email6@siroe.com          |
                         person7 age7 address7 email7@siroe.com <-----
                         person8 age8 address8 email8@siroe.com
```

Since you have searched for 1:3:person4, the results returned are one entry before the match, three entries after the match, and the match entry itself. To verify, type the following command on the Directory Server to view the results:

```
ldapsearch -h directory_server_fully_qualified_host_name -p directory_server_port
-TD "cn=directory manager" -w directory_manager_password -b search_base -s
search_scope -x -S sort_attribute -G
"number_of_results_before_match:number_of_results_after_match:search_string"
vlv_filter
```

For example:

```
ldapsearch -h siroe.com -TD "cn=directory manager" -w password -b
"dc=siroe,dc=siroe,dc=com" -s sub -x -S cn -G "1:3:person4" "(&(mail=*)(cn=*))"
```

The results displayed should match the results in Convergence.

The parameters **ab.corpdir.[CommSuite:default].searchattr** and **ab.corpdir.[CommSuite:default].searchfilter** are not involved in the VLV search.

Same as the existing database name that is given when directory server is configured by running directory preparation tool (comm_dssetup.pl). This database name should not be changed.

7

Calendar Service Administration

This chapter explains how to administer the calendar service in Oracle Communications Convergence.

See "Enabling Core Services for Convergence" for information about enabling services.

Enabling CalDAV Service

To configure CalDAV Service with Convergence, you must have the CalDAV server installed and configured.

To enable Convergence to work with CalDAV, perform the following steps:

1. Enable the following CalDAV related parameters in Convergence:
   - **caldav.enable** - Set this parameter to **true** to enable the search service.

     ```
     iwcadmin -o caldav.enable -v true
     ```

   - **caldav.host** - Set this parameter to the host name on which the CalDAV server installed.

     ```
     iwcadmin -o caldav.host -v siroe.com
     ```

   - **caldav.port** - Set this parameter to the web component port number on which CalDAV is deployed. This should be same as the port number specified for **Server Instance HTTP Port** in the GlassFish Server Configuration Details panel during the Calendar Server Initial Configuration.

     ```
     iwcadmin -o caldav.port -v port_number
     ```

   - **caldav.proxyadminid** - Set this parameter to the proxy admin id on which CalDAV is deployed. This should be same as the Administrator User Id specified during Calendar Server 7 Initial Configuration.

     ```
     iwcadmin -o caldav.proxyadminid -v proxy_admin_id
     ```

   - **caldav.proxyadminpwd** - Set this parameter to the proxy admin password on which CalDAV is deployed. This should be same as the Administrator password specified during Calendar Server Initial Configuration.

     ```
     iwcadmin -o caldav.proxyadminpwd -v proxy_admin_pwd
     ```

   - **caldav.serviceuri** - Set this parameter to the serviceuri on which CalDAV is deployed. This should be same as the URI Path where the Calendar Server is deployed and should be suffixed with /wcap. For example, if the URI path where Calendar Server is deployed is /caldav, then this parameter should be set to /caldav/wcap.

```
iwcadmin -o caldav.serviceuri -v service_uri
```

Note: Convergence can be configured to enable calendar service using both CS 6.x and CalDAV back-end servers and it is called co-existence mode. In this mode of configuration some users may be using CS 6.x server and others might have been migrated to CalDAV server. You need to set the **caldav.davuserattr** parameter to an LDAP attribute used in the user entry to indicate that the user has been migrated to CalDAV. The default value of this attribute is **davStore** (defined as part of **davEntity** ObjectClass). If this attribute is not present in user LDAP entry then it indicates that you are a CS 6.x user and not a CalDAV user.

```
iwcadmin -o caldav.davuserattr -v user_attribute
```

2. Restart the GlassFish server.

Enabling SMS Calendar Notifications in Convergence

You can configure Convergence to send you SMS notifications of your calendar events.

When a user sets up calendar event reminder SMS notifications in the Convergence UI, Calendar Server generates the notifications as specially formatted messages with SMS addresses as recipients; they are then submitted to Messaging Server for processing. Within Messaging Server, the SMS channel processes the notification messages. The messages are submitted to an SMSC provider (through an SMPP protocol) to be delivered as SMS messages to SMS addresses.

SMS addresses, which are added by end users to the Convergence UI, should be of the form **+***subscriber_number***@sms.***your_domain*, where subscriber_number is usually a phone number where a user expects to receive SMS notifications. The format of each subscriber_number is specific to the SMSC provider. For example, some providers might require an international format with the country code. The portion **@sms.***your_domain* represents a site-specific domain name and is the same name that is used in the Messaging Server's SMS channel configuration.

To enable SMS calendar notifications in Convergence:

1. Set up and configure an SMS channel in Messaging Server. See *Messaging Server System Administrator's Guide* for more information.
2. Enable SMS notifications for calendar events in Convergence server:

   ```
   iwcadmin -o user.cal.enablesmsnotify -v true
   ```

Convergence users can now enable SMS calendar notifications in Convergence in the Convergence **Options** tab.

8

Instant Messaging Service Administration

This chapter explains how to administer the instant messaging service in Oracle Communications Convergence.

See "Enabling Core Services for Convergence" for information about enabling services.

Configuring Multiple Domains for Instant Messaging

After creating a new non-default domain, you need to use the **imadmin** command-line utility to enable Instant Messaging for users in a new domain:

In this example the user or group base is **dc=example,dc=com**. The new domain is called *Hosted Domain* and it has a DNS domain name of **other.hosteddomain.com**.

1. Enter the following command:

   ```
   imadmin assign_services
   ```

2. You are prompted to provide the base domain name. Enter **o=Hosted Domain,dc=aus,dc=example,dc=com**.
3. Edit the Convergence **httpbind.conf** file to include both default domain and hosted domains to the **default.domains** attribute, for example:

   ```
   default.domains=example.com, other.hosteddomain.com
   ```

You should then be able to log in to Convergence as **user@hosteddomain**. The default domain user can log in with just the UID.

For more information on hosted domain support in Instant Messaging, see *Instant Messaging Server Installation and Configuration Guide*.

Configuring Convergence to Display Presence Information in Email

To configure Convergence to display the instant messaging presence status of users in email messages with Instant Messaging Server 8:

1. Go to the *Instant_Messaging_Server_Home*/**config** directory and edit the **iim.conf** file.
2. Add the following lines in the **iim.conf** file.

   ```
   iim_server.roster.extra = "true"
   iim_server.roster.extra.attributes.mail = "mailalternateaddress, mail"
   iim_ldap.user.attributes = "mailalternateaddress, mail"
   ```

3. Restart the Instant Messaging server with the **imadmin** command-line utility.

```
imadmin stop
imadmin start
```

To configure Convergence to display the instant messaging presence status of users in email messages with Instant Messaging Server 9:

1. Run the **imconfutil** command-line utility to set the following properties in the **iim.conf.xml** file.

   ```
   imconfutil set-prop -u -c /opt/sun/comms/im/config/iim.conf.xml
   iim_server.roster.extra=true iim_ldap.user.attributes=mail
   ```

2. Restart the Instant Messaging server.

Configuring Instant Messenger Status to Update Based on Calendar Availability

You can configure Convergence to automatically update the user's instant messaging status based on the user's calendar availability. If, in his calendar, the user is in a meeting, then the user's status automatically updates to a busy status.

Convergence must be integrated with Instant Messaging Server 9.0.1 and Calendar Server 7.

To configure the instant messaging status to automatically update based on calendar availability, see the discussion on configuring the Instant Messaging Server calendar agent in *Instant Messaging Server System Administrator's Guide*

9

Configuring Convergence to Use Proxy Authentication

This chapter describes how to enable Proxy Authentication in Convergence. The proxy authentication mechanism uses various components that Convergence depends on. You must have thorough knowledge of the following products and technologies:

- Convergence administration
- Directory Server administration
- Knowledge of Communications Suite Schemas

Proxy authentication is performed by using the credentials of a more privileged user on behalf of a normal user. The user name and password of the privileged user requesting the authentication is sent with the user name of the user requesting the authentication.

The parameters include:

- **username** - The user name of the privileged user.
- **password** - The password of the privileged user.
- **proxyauth** - The user name of the user for whom authentication is requested.

The protocol request must pass these parameters for performing authentication.

Configuring Convergence for Proxy Authentication

For proxy authentication to work, the privileged user (the Proxy Admin user) must be provisioned for the domain. A user is considered a proxy administration user if the LDAP entry has **isMemberOf** operational attribute, whose value is set to the DN of **Service Administrators**. The administration user must be a member of the **Service Administrators** group in the DC tree.

For example:

```
cn=Service Administrators, ou=Groups, DC_Root
```

The **Service Administrators** group and the administration user are provisioned when the administrators for Oracle Communications Messaging Server (**admin**) and Oracle Communications Calendar Server (**calmaster**) are configured. This user can also be used for Convergence proxy authentication.

To configure proxy authentication in Convergence, enable proxy authentication by setting the **auth.ldap.enableproxyauth** configuration parameter.

For example:

```
iwcadmin -u admin -o auth.ldap.enableproxyauth -v true
```

Note: Convergence does not provision an administrator user.

Proxy Authentication Request

Convergence requires the following parameters for performing proxy authentication based on a specific format that is applicable to the login.iwc or login.wabp commands.

For example:

```
http://hostname:port/iwc/
login.iwc?username=username_privileged_user&password=password_privileged_user&prox
yauth=username&fmt-out=text/json
```

Where the values for:

- *username_privileged_user* is the user name of the privileged user.
- *password_privileged_user* is the password of the privileged user.
- *username* is the user name of the user for whom authentication is requested.
- **fmt-out=text/json** specifies the JSON output. XML output is no longer valid.

10

Convergence Properties Reference

This chapter lists all the configuration parameters that are available in Oracle Communications Convergence. Each parameter is described with its name and a description of its purpose. Use the **iwcadmin** command-line utility to update the configuration properties for your deployment. See "Using the Convergence Administration Utility" for more information.

Global Convergence Configuration Properties

Whenever you make changes to the configuration files, you must stop and restart the client software because the configuration files are only read at startup. The client restart is required so that the changes you have made to take effect.

When you configure Convergence using the configuration utility, most of the parameters are assigned default values. You can change the default values depending on the changing business needs for your site. You can use the **iwcadmin** command to get the values that are assigned to any of the parameters.

```
iwcadmin -o parameter_name
```

In the following configuration properties tables, the command-line option name found in the left column is the parameter you use after *-o* option in the **iwcadmin** command. The property name shown in the right column is how the property is represented in the configuration file. Do not use the property name from the right column for the *-o* option. In addition, the right column is a definition for the option, containing the following details: the name of the property found in the configuration file, the data type for the expected value, the default value if any, whether or not this property is mandatory for proper configuration, and whether or not this property was set by the initial configuration program.

Unless specified, these parameters have a PUBLIC access type. Any RESTRICTED access types are for properties that perform special bulk updates. Use properties with RESTRICTED access types cautiously.

The following tables list the Convergence Server global configuration properties:

Table 10–1 Deployment-Level Global Configuration Properties

Option Name	Description
base.defaultdomain	Default domain to use for user resolution ■ Allowed Pattern/Values: [A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)+ ■ Data Type: String
base.loginseparator	Character to be used as login separator (between user ID and domain). It should match any one of the character defined in service.loginseparator of mail and calendar back end service ■ Allowed Pattern/Values: a character ■ Data Type: String ■ Default value: @
base.defaultlocale	Default locale to be used ■ Default value: en_us ■ Data Type: String
base.passivatesession	Enabling this option will allow web container to passivate all active sessions else all active session will be terminated upon session activation event. While typically run in a cluster, this parameter can also be enabled in a non-cluster environment. ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
base.enablehosteddomain	Whether hosted domains is enabled ■ Data Type: boolean ■ Allowed Pattern/Values: true or false ■ Default value: true
base.port	Port number at which the application listens ■ Default value: 8080 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer

Table 10–1 (Cont.) Deployment-Level Global Configuration Properties

Option Name	Description
base.sslport	SSL Port number at which the application listens ■ Default value: 8181 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer
base.enableauthonlyssl	SSL can be used only for authentication and the subsequent access via non-ssl ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
base.ipaccessurl	The access URL for this application. The URL must use IP address instead of host name. ■ Default value: null ■ Allowed Pattern/Value: scheme://*IP_address*:*port* (example: [http://123.456.789.12:8080]) ■ Data Type: String
base.ipsecurity.enable	IP address along with the token is used for authorization if set to true ■ Allowed Pattern/Values: true or false ■ Default value: false ■ Data Type: boolean
base.ignoreurldomain	Prevents the use of the URL domain. ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Value: true or false
base.authcookiepath	Cookie path for authorization cookie. ■ Default value: null ■ Data Type: String
base.hstsmaxage	The number of seconds, after receiving a request with STS header, that the host is considered as a Known HSTS Host. A value of 0 indicates that HSTS is not enforced. ■ Default value: 0 ■ Allowed Pattern/Values: 0 or higher integer ■ Data Type: Integer

Table 10–2 LDAP User and Group Configuration Properties

Option Name	Description
ugldap.schemaversion	Schema level used by the deployment ■ Default value: 1 ■ Allowed Pattern/Values: 1 or 2 ■ Data Type: Integer
ugldap.dcroot	Domain component root suffix ■ Default value: Not Applicable ■ Allowed Pattern/Values: (.=.([,;+].))? ■ Data Type: String
ugldap.basedn	Base DN to start the user search from ■ Default value: Not Applicable ■ Allowed Pattern/Values: (.=.([,;+].))? ■ Data Type: String
ugldap.ugfilter	User/group filter to apply while user lookup ■ Default value: (uid=%U%V) ■ Data Type: String
ugldap.domainfilter	Domain filter to apply while domain lookup ■ Default value: (&(objectClass=sunManagedOrganization)(|(sunPreferredDomain=%V)(associatedDomain=%V))) ■ Data Type: String
ugldap.srchopattrs	Comma-separated list of retrievable LDAP operational attributes ■ Default value: *,isMemberOf ■ Data Type: String
ugldap.host	Host name of the LDAP service ■ Default value: Not Applicable ■ Allowed Pattern/Values: [A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)*(:[1-9][0-9]*)?)+(\,([A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)*(:[1-9][0-9]*)?))* ■ Data Type: String
ugldap.port	Port number at which LDAP service listens ■ Default value: 389 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer
ugldap.enablessl	Whether LDAP is SSL enabled ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ugldap.minpool	Minimum number of connections in LDAP Pool ■ Default value: 1 ■ Allowed Pattern/Values: Greater than 0 and less than the max pool ■ Data Type: Integer

Table 10–2 (Cont.) LDAP User and Group Configuration Properties

Option Name	Description
ugldap.maxpool	Maximum number of connections in LDAP Pool ■ Default value: 30 ■ Allowed Pattern/Values: Greater than 0 and greater than the min pool ■ Data Type: Integer
ugldap.timeout	LDAP operation time out in seconds ■ Default value: 30 ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Data Type: Integer
ugldap.refreshinterval	Time interval (in minutes) after which, connections in LDAP pool will be re-created. 0 means no refresh is required ■ Default value: 30 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ugldap.monitoringinterval	Monitoring interval (in seconds) for LDAP pool, when the LDAP server is down ■ Default value: 60 ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Data Type: Integer
ugldap.binddn	The admin DN used for creating LDAP connection pool ■ Default value: Not Applicable ■ Allowed Pattern/Values: (.=.([,;+].))? ■ Data Type: String
ugldap.bindpwd	The admin DN password ■ Default Value: Not Applicable ■ Data Type: String

Table 10–3 Authentication Configuration Properties

Option Name	Description
auth.cert.enable	Enables and disables X509 Certificate-based authentication. ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
auth.cert.enablefallback	Enables and disables fallback to form-based login. This option should be set in conjunction with auth.cert.enable. ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
auth.ldap.enable	This creates default configuration parameters required to enable LDAP authentication mechanism. Specific parameters can further be modified/created using parameter-specific CLI option. ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false

Table 10–3 (Cont.) Authentication Configuration Properties

Option Name	Description
auth.ldap.loginimpl	An implementation of LoginModule interface (JAAS technology in Java). This property refers to a pluggable custom authentication module ■ Default value: Not applicable ■ Data Type: String
auth.ldap.callbackhandler	An implementation of HttpCallbackHandler class, which extends CallBackHandler (JAAS technology in Java). This property refers to a pluggable custom authentication module ■ Default value: com.sun.comms.client.security.auth.AppCallbackHandler ■ Data Type: String
auth.ldap.schemaversion	The value of this should be same as ugldap. ■ Default value: 2 ■ Allowed Pattern/Values: 1 or 2 ■ Data Type: Integer
auth.ldap.dcroot	The value of this should be same as ugldap.dcroot ■ Default value: Not Applicable ■ Allowed Pattern/Values: (.=.([,;+].))? ■ Data Type: String
auth.ldap.basedn	The value of this should be same as ugldap.basedn ■ Default value: Not Applicable ■ Allowed Pattern/Values: (.=.([,;+].))? ■ Data Type: String
auth.ldap.ugfilter	The value of this should be same as ugldap.ugfilter ■ Default value: (uid=%U%V) ■ Data Type: String
auth.ldap.domainfilter	The value of this should be same as ugldap.domainfilter ■ Default value: (&(objectClass=sunManagedOrganization)(|(sunPreferredDomain=%V)(associatedDomain=%V))) ■ Data Type: String
auth.ldap.host	Host name of the auth LDAP service ■ Default value: Not Applicable ■ Allowed Pattern/Values: [A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)*(:[1-9][0-9]*)?)+(\,([A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)*(:[1-9][0-9]*)?))* ■ Data Type: String
auth.ldap.port	Port number at which auth LDAP service listens ■ Default value: 389 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer
auth.ldap.enablessl	Whether auth LDAP is SSL enabled ■ Default value: true ■ Allowed Pattern/Values: true or false ■ Data Type: boolean

Table 10–3 (Cont.) Authentication Configuration Properties

Option Name	Description
auth.ldap.minpool	Minimum number of connections in LDAP Pool ■ Default value: 1 ■ Allowed Pattern/Values: Greater than 0 and less than max pool ■ Data Type: Integer
auth.ldap.maxpool	Maximum number of connections in LDAP Pool ■ Default value: 30 ■ Allowed Pattern/Values: Greater than 0 and greater than min pool ■ Data Type: Integer
auth.ldap.timeout	LDAP operation time out in seconds ■ Default value: 30 ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Data Type: Integer
auth.ldap.refreshinterval	Time interval (in minutes) after which, connections in LDAP pool will be re-created. 0 means no refresh is required ■ Default value: 30 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
auth.ldap.monitoringinterval	Monitoring interval (in seconds) for LDAP pool, when the LDAP server is down ■ Default value: 60 ■ Allowed Pattern/Values: Greater than or equal 1 ■ Data Type: Integer
auth.ldap.binddn	The admin DN used for creating LDAP connection pool ■ Default value: Not applicable ■ Allowed Pattern/Values: (.*=.*([,;\+].*)*)? ■ Data Type: String
auth.ldap.bindpwd	The admin DN password ■ Default value: Not Applicable ■ Data Type: String
auth.ldap.enableproxyauth	Enables proxy authentication of the user ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
auth.custom.servicename	Name of service for custom authentication module ■ Default value: Not Applicable ■ Data Type: String
auth.custom.loginimpl	An implementation of LoginModule interface (JAAS technology in Java). This property refers to a pluggable custom authentication module ■ Default value: Not Applicable ■ Data Type: String

Table 10–3 (Cont.) Authentication Configuration Properties

Option Name	Description
auth.custom.callbackhandler	An implementation of HttpCallbackHandler class, which extends CallBackHandler (JAAS technology in Java). This property refers to a pluggable custom authentication module ■ Default value: Not Applicable ■ Data Type: String
auth.misc	Placeholder for custom auth provider configuration ■ Allowed Pattern/Values: user-defined-attribute ■ Data Type: String
auth.adminuserlogin.enable	Whether proxy admins are allowed to login through web client ■ Default value: true ■ Data Type: boolean

Table 10–4 Mail Service Configuration Properties

Option Name	Description
mail.enable	Whether mail service is enabled or not ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
mail.host	Host name of the back-end mail service ■ Default value: Not Applicable ■ Allowed Pattern/Values: [A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)* ■ Data Type: String
mail.port	Port number at which back-end mail service listens ■ Default value: 8990 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer
mail.enablemsgpreview	Turns on/off the mail preview pane ■ Default: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false If mail.enablemsgpreview is true, the user's preference (LDAP attribute: nswmExtendedUserPrefs:mePreviewEnabled=true/false) is checked and returned accordingly. In other words, the user can disable mail preview pane, even though it is site-enabled. However, if mail.enablemsgpreview is false, the mail preview pane is disabled, irrespective of user preference.
mail.enablessl	Whether mail service is SSL enabled ■ Default value: true ■ Allowed Pattern/Values: true or false ■ Data Type: boolean

Table 10–4 (Cont.) Mail Service Configuration Properties

Option Name	Description
mail.requesttimeout	Time out value in seconds to use if Mail server does not respond within this time. Zero means never time out ■ Default value: 180 ■ Data Type: Integer ■ Allowed Pattern/Values: Greater than or equal to 0
mail.restrictanyone	Mirror option of store.privatesharedfolders.restrictanyone on Oracle Communications Messaging Server ■ Default value: false ■ Allowed Pattern/Values: true or false ■ Data Type: boolean
mail.cookiename	Cookie name used by mail service as session identifier ■ Default value: webmailsid ■ Data Type: String
mail.proxyadminid	Back-end mail service's proxy admin UID. Used for proxy-auth to mail service. This should be of form: uid@domain if hosted domains setup is used ■ Default value: Not Applicable ■ Data Type: String
mail.proxyadminpwd	Back-end mail service's proxy admin password. Used for proxy-auth to mail service ■ Default value: Not Applicable ■ Data Type: String
mail.uwcsievecompatible	Specifies whether the sieve should be compatible with Communications Express ■ Default value: true ■ Allowed Pattern/Values: true or false ■ Data Type: boolean
mail.spam.folder	Spam folder used to move messages marked as spam by the user ■ Default value: spam ■ Data Type: String
mail.spam.enableaction	Specifies whether Spam Action (ability to mark/unmark messages as spam) should be enabled ■ Default value: false ■ Allowed Pattern/Values: true or false ■ Data Type: boolean
mail.pop.refreshinterval	Time interval (in sec) for the client to check the external mail server for new messages ■ Default value: 600 ■ Allowed Pattern/Values: 0-3600 seconds ■ Data Type: Integer

Table 10–4 (Cont.) Mail Service Configuration Properties

Option Name	Description
mail.pop.requesttimeout	Time interval (in sec) to wait for the response for POP requests. Zero means never time out ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Default value: 600 ■ Data Type: Integer
mail.maxpool	Maximum number of connections per route in a pool; this setting can be used when setting up a connection manager. ■ Default: 100 ■ Data Type: Integer
mail.pooltimeout	Maximum amount of time (in sec) to wait while retrieving a connection from the pool; this setting can be used when setting up a connection manager. ■ Default: 240 ■ Data Type: Integer

Table 10–5 Logging Configuration Properties

Option Name	Description
log.enableusertrace	Specifies whether user IP address and session ID should be included in the logs. Log pattern must include %X{ipaddress} and %X{sessionid}. ■ Allowed Pattern/Values: true or false ■ Default value: true ■ Data Type: boolean
log.locationtype	Definition for specifying Log Location Type. Currently supported location type: FILE, CONSOLE (aka STDOUT). ■ Default value: CONSOLE ■ Allowed Pattern/Values: FILE or CONSOLE ■ Data Type: String
log.location	The Location value is the location of Log file (and hence is applicable only for FILE type) ■ Default value: /data/logs/iwc.log ■ Data Type: String
log.adminloglocationtype	Log location type for admin log file ■ Default value: FILE ■ Allowed Pattern/Values: FILE or CONSOLE ■ Data Type: String
log.adminloglocation	The location of admin log file (and hence is applicable only for FILE type) ■ Default value: /data/logs/iwc_admin.log ■ Data Type: String
log.sizetriggerval	Set the maximum size in KB, that the log file is allowed to reach before being rolled over to backup files ■ Default value: 2048 ■ Allowed Pattern/Values: Greater than 0 KB ■ Data Type: Integer

Table 10–5 (Cont.) Logging Configuration Properties

Option Name	Description
log.timetriggerval	The rolling schedule is specified by this pattern. Set the Date pattern at which the log file will be rolled over to backup files ■ Default value: null ■ Allowed Pattern/Values: This pattern should follow the SimpleDateFormat conventions. For examples and more details, refer to DailyRollingFileAppender documentation in Apache Log4j project. ■ Data Type: String
log.maxbackupindex	This option determines how many backup files are kept before the oldest is erased. This option takes a positive integer. If set to zero, there will be no backup files and the log file will be truncated when it reaches the size trigger value. The max backup index option is considered only if size trigger is set and is ignored for time trigger. ■ Default value: 1 ■ Data Type: Integer
log.pattern	The log record pattern used by the loggers ■ Default value: %c: %p from %C : Thread %t at time %d{HH:mm:ss,SSS} --- %m %n ■ Allowed Pattern/Values: The pattern is closely related to the conversion pattern of the print function in C. For detailed patterns, refer to Pattern Layout documentation in Apache Log4j project ■ Data Type: String
log.DEFAULT.level	Level of Logging ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.CONFIG.level	Level of Logging for Config module ■ Default value: WARN ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.AUTH.level	Level of Logging for Auth module ■ Default value: DEBUG ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.PROXY_MAIL.level	Level of Logging for Proxy Mail module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.ADDRESS_BOOK.level	Level of Logging for Address Book module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.PROXY_CAL.level	Level of Logging for Proxy Cal module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String

Table 10–5 (Cont.) Logging Configuration Properties

Option Name	Description
log.PROXY_NAB.level	Level of Logging for Contacts Server proxy module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.PROTOCOL.level	Level of Logging for Protocol module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.SIEVE.level	Level of Logging for Sieve module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.NOTIFY.level	Level of logging for notification module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.ADMIN.level	Level of Logging for Admin module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.PROXY_ISS.level	Level of Logging for ISS (MISO) proxy module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.ENS.level	Level of logging for ENS module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String
log.PROXY_OIN.level	Level of Logging for Proxy OIN module ■ Default value: INFO ■ Allowed Pattern/Values: OFF | ERROR | WARN | INFO | DEBUG ■ Data Type: String

Table 10–6 Calendar Service Configuration Properties for Calendar Server 6.3

Option Name	Description
cal.autoprovision	Determines if calendar auto-provision on the back-end Oracle Communications Calendar Server is enabled. This option should be set in conjunction with local.autoprovision in Calendar Server 6.x. ■ Allowed Pattern/Value: true or false ■ Default Value: false ■ Data Type: boolean true (local.autoprovision in Calendar Server 6.x = yes) Calendar Service is enabled to allow for auto-provisioning by Calendar Server (even if user is provisioned by DA as a mail-only user). cal.autoprovision=false (local.autoprovision in Calendar Server 6.x = no) Checks for the presence of icsCalendarUser object class and disallows the service if not present. For deployments that provision mail-only users, this option can be used to disable deployment-wide calendar service.
cal.enable	Whether Calendar service is enabled or not ■ Default value: false ■ Allowed Pattern/Values: true or false ■ Data Type: boolean
cal.host	Host name of the back-end Calendar service ■ Default value: Not Applicable ■ Allowed Pattern/Values: [A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)* ■ Data Type: String
cal.port	Port number at which back-end Calendar service listens ■ Default value: 80 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer
cal.enablessl	Whether SSL is enabled for calendar service ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
cal.requesttimeout	Time out value in seconds to use if Calendar server does not respond within this time. Zero means never time out ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Default value: 180 ■ Data Type: Integer
cal.proxyadminid	Back end Calendar service's proxy admin UID. Used for proxy-auth to cal service. This should be of form: uid@domain if hosted domains setup is used ■ Default value: Not Applicable ■ Data Type: String

Table 10–6 (Cont.) Calendar Service Configuration Properties for Calendar Server 6.3

Option Name	Description
cal.proxyadminpwd	Back end calendar service's proxy admin password. Used for proxy-auth to calendar service ■ Default value: Not Applicable ■ Data Type: String
cal.maxpool	Maximum number of connections per route in a pool ■ Default: 100 ■ Data Type: Integer
cal.pooltimeout	Defines the time out (seconds) used when retrieving a connection from the pool. ■ Default: 240 ■ Data Type: Integer ■ Allowed Pattern/Values: Greater than or equal to 1

Table 10–7 Calendar Service Configuration Properties for Calendar Server 7

Option Name	Description
caldav.enable	Whether CalDAV Calendar service is enabled or not ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
caldav.host	Host name of the back end CalDAV service ■ Default value: Not Applicable ■ Allowed Pattern/Values: [A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)* ■ Data Type: String
caldav.port	Port number at which back end CalDAV service listens ■ Default value: 8080 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer
caldav.enablessl	Whether SSL should be used against back end CalDAV service ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
caldav.requesttimeout	Time out value in seconds to use if CalDAV server does not respond within this time. Zero means never time out ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Default value: 180 ■ Data Type: Integer
caldav.serviceuri	Context URI at which the WCAP interface in CalDAV service is accessible ■ Default value: /wcap ■ Data Type: String
caldav.proxyadminid	Back end CalDAV service's proxy admin UID. Used for proxy-auth to cal service. This should be of form: uid@domain if hosted domains setup is used ■ Default value: Not Applicable ■ Data Type: String

Table 10–7 (Cont.) Calendar Service Configuration Properties for Calendar Server 7

Option Name	Description
caldav.proxyadminpwd	Back end CalDAV service's proxy admin password. Used for proxy-auth to calendar service ■ Default value: Not Applicable ■ Data Type: String
caldav.davuserattr	Attribute name in the user's LDAP entry indicating the user is a CalDAV user in a co-existence deployment ■ Default value: davstore ■ Data Type: String
caldav.groupobjectclass	Object class names of groups to be filtered while searching for Corp-Dir groups. The filter matches with any one of the configured object class names to retrieve the results ■ Default value: null ■ Data Type: String
caldav.autoprovision	Whether CalDAV auto-provision in the back end CalDAV Server is enabled or not. ■ Allowed Pattern/Values: true or false ■ Default value: false ■ Data Type: boolean
caldav.davuserobjectclass	Name of the LDAP object class which should be present for valid CalDAV users if auto-provisioning is disabled ■ Allowed Pattern/Values: Name of the LDAP object class ■ Default value: icsCalendarUser ■ Data Type: String
caldav.wcapversion	WCAP Version of the CalDAV Service ■ Default value: 7.0 ■ Data Type: String
caldav.maxpool	Maximum number of connections per route in a pool; this setting can be used when setting up a connection manager. ■ Default: 100 ■ Allowed Pattern/Values: Greater than 0. ■ Data Type: Integer
caldav.pooltimeout	Defines the time out (seconds) used when retrieving a connection from the pool. ■ Default: 240 ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Data Type: Integer

Table 10–8 Address Book Service Configuration Properties for Contacts Server

Option Name	Description
nab.enable	Whether the address book service provided by Contacts Server is enabled ■ Data Type: boolean ■ Allowed Pattern/Values: true or false ■ Default value: false
nab.host	Host name of the back-end address book service provided by Contacts Server ■ Allowed Pattern/Values: [A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)* ■ Data Type: String
nab.port	Port number at which back-end address book service provided by Contacts Server service listens ■ Default value: 8080 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer
nab.enablessl	Whether SSL is enabled to Contacts Server ■ Default value: true ■ Allowed Pattern/Values: true or false ■ Data Type: boolean
nab.requesttimeout	Time out value in seconds to use if address book service provided by Contacts Server does not respond within this time. Zero means never time out ■ Default value: 180 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
nab.proxyadminid	Contacts Server proxy admin UID. Used for proxy-auth to address book service. This should be of form: uid@domain if hosted domains setup is used ■ Data Type: String
nab.proxyadminpwd	Contacts Server proxy admin password. Used for proxy-auth to address book service ■ Data Type: String
nab.nabuserattr	Attribute name in the user's LDAP entry indicating whether the address book service is provided by Contacts Server or Convergence in a co-existence deployment ■ Default value: nabStore ■ Data Type: String
nab.maxpool	Maximum number of connections per-route ■ Default value: 100 ■ Allowed Pattern/Values: Greater than 0. ■ Data Type: Integer
nab.pooltimeout	Defines the time out (seconds) used when retrieving a connection from the pool. ■ Default value: 240 ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Data Type: Integer
nab.serviceuri	Context URI at which the address book service provided by Contacts Server is accessible ■ Data Type: String

Table 10–9 Indexing Search and Service Configuration Properties

Option Name	Description
ISS.enable	Whether ISS service is enabled or not ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ISS.host	Host name of the back end ISS service ■ Default value: Not Applicable ■ Allowed Pattern/Values: [A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)* ■ Data Type: String
ISS.port	Port number at which back end ISS service listens ■ Default value: 8080 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer
ISS.enablessl	Whether SSL is enabled for ISS service ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ISS.requesttimeout	Time out value in seconds to use if ISS server does not respond within this time. Zero means never time out ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Default value: 180 ■ Data Type: Integer
ISS.proxyadminid	Back end ISS service's proxy admin UID. Used for proxy-auth to ISS service. This should be of form: uid@domain if hosted domains setup is used ■ Default value: Not Applicable ■ Data Type: String
ISS.proxyadminpwd	Back end ISS service's proxy admin password. Used for proxy-auth to ISS service ■ Default value: Not Applicable ■ Data Type: String
ISS.maxpool	Maximum number of connections per route in a pool; this setting can be used when setting up a connection manager. ■ Default: 100 ■ Data Type: Integer
ISS.pooltimeout	Maximum amount of time (in sec) to wait while retrieving a connection from the pool; this setting can be used when setting up a connection manager. ■ Default: 240 ■ Data Type: Integer

Table 10–10 Address Book Service Configuration Properties for Convergence WABP

Option Name	Description
ab.enable	Enable or disable WABP service ■ Default value: false ■ Allowed Pattern/Values: true or false ■ Data Type: boolean
ab.purgetype	Enables WABP purge, which permanently deletes entries marked for deletion. If **ab.purgetype** is **auto** then purging happens automatically upon login. If **ab.purgetype** is **manual** then purging can be done by invoking the **purge_entries.wabp** command. ■ Default value: auto ■ Allowed Pattern/Values: manual | auto ■ Data Type: String ■ Access Type: RESTRICTED
ab.expireperiod	WABP Purge, period (in days) after which the entries get deleted permanently. This is applicable only when **enableautopurge** is set to true ■ Default value: 30 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.purgeinterval	When **ab.purgetype** is set to **auto**, this parameter specifies the interval (in days) between purges of the database. ■ Default value: 0 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.maxpostlength	Defines the maximum content-length of a POST command. -1 means no limit. ■ Default value: -1 ■ Allowed Pattern/Values: -1, 0 or greater than 0 ■ Data Type: Integer
ab.mycontacttag	Tag name for my contact ■ Default value: My Contact ■ Data Type: String
ab.myfavoritestag	Tag name for my favorites ■ Default value: My Favorites ■ Data Type: String
ab.maxphotosize	Maximum allowed photo size in bytes ■ Default value: 102400 ■ Allowed Pattern/Values: Greater than 0 ■ Data Type: Integer
ab.maxphotowidth	Limit on dimension (width in pixels) of images being served ■ Default value: 2000 ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Data Type: Integer

Table 10–10 (Cont.) Address Book Service Configuration Properties for Convergence WABP

Option Name	Description
ab.maxphotoheight	Limit on dimension (height in pixels) of images being served ■ Default value: 2000 ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Data Type: Integer
ab.exportphoto	If this is enabled it exports contacts with photo data in vCard 3.0 format ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.importphoto	If this is enabled it imports contacts with photo data in vCard 3.0 format ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.import.vcard.misc	Specify encoding to be used during import corresponding to each locale ■ Default value: UTF-8 ■ Data Type: String
ab.export.vcard.misc	Specify encoding to be used during export corresponding to each locale ■ Default value: UTF-8 ■ Data Type: String
ab.maxpagedsearch	Max number of simultaneous paged search for an instance of PersonalStore ■ Default value: 10 ■ Allowed Pattern/Values: Greater than 1 ■ Data Type: Integer
ab.retries	Number of retries to fetch default address book when a new user logs in ■ Default value: 0 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.psrootpattern	Defines a default psRoot pattern for users that do not have the psRoot attribute. %U = uid of the user ("jsmith"), %D = domain of the user ("somedomain.com"), %O = most significant part of the domain ("somedomain") ■ Default value: ldap:/// piPStoreOwner=%U,o=%D,o=PiServerDb ■ Allowed Pattern/Values: Starts with ldap:// ■ Data Type: String

Table 10–10 (Cont.) Address Book Service Configuration Properties for Convergence WABP

Option Name	Description
ab.ldapdelay	Amount of delay in number of milliseconds to be introduced to compensate delays due to LDAP updates ■ Default value: 0 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.storecachecount	Enable cache entry count ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.storeentrieslimit	Total number of entries allowed in the user's address book. ■ Default value: 1000 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.storequotawarn	Indicate whether quota warning can be issued or not. A positive integer greater than zero indicates a warning else no warning. ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Default value: 100 ■ Data Type: Integer
ab.useuserpsroot	Whether the per User psRoot should be used or not ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.pstore.notification.notifyall	Enable address book notification for all users ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.pstore.notification.event.createcontact	Enable notification for contact creation ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.pstore.notification.event.modifycontact	Enable notification for contact modification ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.pstore.notification.event.deletecontact	Enable notification for contact deletion ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false

Table 10–10 (Cont.) Address Book Service Configuration Properties for Convergence WABP

Option Name	Description
ab.pstore.notification.event.createcontactphoto	Enable notification for adding contact photo ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.pstore.notification.destination	Comma separated list of destination. Used only when notify all users is enabled ■ Default value: null ■ Data Type: String
ab.pstore.[*identifier*].ldappoolmin	Minimum connections to the LDAP server ■ Default value: 1 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.pstore.[*identifier*].ldappoolmax	Maximum connections to the LDAP server ■ Default value: 4 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.pstore.[*identifier*].ldappooltimeout	Max time (in seconds) to wait for a connection to be freed up ■ Default value: 10 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.pstore.[*identifier*].ldappoolrefreshinterval	Time interval (in minutes) after which, connections in LDAP pool will be re-created. 0 means no refresh is required ■ Default value: 0 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.pstore.[*identifier*].ldappoolmonitoringinterval	Monitoring interval in seconds for LDAP pool, when the LDAP server is down ■ Default value: 60 ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Data Type: Integer
ab.pstore.[*identifier*].ldaphost	Host name of the LDAP service ■ Default value: Not Applicable ■ Allowed Pattern/Values: [A-Za-z0-9\\-]+(\.[A-Za-z0-9\-]+)* ■ Data Type: String
ab.pstore.[*identifier*].ldapport	Port number at which LDAP service listens ■ Default value: 389 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer

Table 10–10 (Cont.) Address Book Service Configuration Properties for Convergence WABP

Option Name	Description
ab.pstore.[*identifier*].ldapbinddn	The admin DN used for creating LDAP connection pool. This pool will be used for PStore lookup ■ Default value: Not Applicable ■ Allowed Pattern/Values: (.*=.*([,;\+].*)*)? ■ Data Type: String
ab.pstore.[*identifier*].ldapbindcred	The admin DN's password, used for creating LDAP connection pool. This pool will be used for PStore lookup. ■ Default value: Not Applicable ■ Data Type: String
ab.pstore.[*identifier*].enableldapssl	Enable LDAP SSL ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.pstore.urlmatch	Specifies the type of URL this instance of the plug-in is responsible for. This value should be unique and is case sensitive. ■ Default value: ldap:// ■ Allowed Pattern/Values: Starts with ldap:// ■ Data Type: String
ab.pstore.randompaging	Specifies if the plug-in support access to any page, or if each page must be accessed starting at page 1. If false, the coresrv will loop until it gets to the right page. ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.pstore.logintype	This can be: anon (anonymous), restricted (login as user who has rights to view/write DB), or proxy (login as user that can 'masquerade') ■ Default value: restricted ■ Allowed Pattern/Values: anon, restricted, or proxy ■ Data Type: String
ab.pstore.defaultserver	Default server (identifier) used for construction psRoot ■ Default value: null ■ Data Type: String
ab.pstore.displayname	Display Name for Personal book ■ Default value: Personal Address Book ■ Data Type: String
ab.pstore.description	Description for Personal book ■ Default value: This is your personal Address Book ■ Data Type: String

Table 10–10 (Cont.) Address Book Service Configuration Properties for Convergence WABP

Option Name	Description
ab.pstore.getalldbattr	This defines if all the database attributes should be passed in the LDAP search true or false. ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.pstore.lookthrulimit	This is the max number of entries to read in any one search. Should be set to max in directory or largest AB possible. ■ Default value: 0 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.pstore.deleteperm	Mark the contact/group as deleted instead of permanently deleting it by setting following parameter as false ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.pstore.allowdupentry	Parameter which, if set to true, allows personal address book entries/groups to have the same name ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.pstore.admingroupdn	DN of admin group. If a user belong to this group then he is eligible to purge all user's contacts which are marked for deletion ■ Default value: null ■ Allowed Pattern/Values: (.*=.*([,;\+].*)*)? ■ Data Type: String
ab.pstore.collationrule	Locale on whose basis collation rule should be applied for Personal Address Book ■ Default value: en-US ■ Data Type: String
ab.pstore.collationsearchfield	Search Fields for which collation rule should be applied. The fields provided here should be disambiguator formatted fields. For example, entry/displayname, person/givenname, and so on. ■ Default value: null ■ Data Type: String
ab.corpdir.[*identifier*].ldappoolmin	Minimum connections to the LDAP server ■ Default value: 1 ■ Data Type: Integer ■ Allowed Pattern/Values: Greater than or equal to 0
ab.corpdir.[*identifier*].ldappoolmax	Maximum connections to the LDAP server ■ Default value: 4 ■ Data Type: Integer ■ Allowed Pattern/Values: Greater than or equal to 0

Table 10–10 (Cont.) Address Book Service Configuration Properties for Convergence WABP

Option Name	Description
ab.corpdir.[*identifier*].ldappooltimeout	Max time (in seconds) to wait for a connection to be freed up ■ Default value: 10 ■ Data Type: Integer ■ Allowed Pattern/Values: Greater than or equal to 0
ab.corpdir.[*identifier*].ldappoolrefreshinterval	Time interval (in minutes) after which, connections in LDAP pool will be re-created. 0 means no refresh is required ■ Default value: 0 ■ Data Type: Integer ■ Allowed Pattern/Values: Greater than or equal to 0
ab.corpdir.[*identifier*].ldappoolmonitoringinterval	Monitoring interval (in seconds) for LDAP pool, when the LDAP server is down ■ Default value: 60 ■ Data Type: Integer ■ Allowed Pattern/Values: Greater than 0
ab.corpdir.[*identifier*].ldaphost	Host name of the LDAP service ■ Default value: Not Applicable ■ Allowed Pattern/Values: [A-Za-z0-9\\-]+(\.[A-Za-z0-9\-]+)* ■ Data Type: String
ab.corpdir.[*identifier*].ldapport	Port number at which LDAP service listens ■ Default value: 389 ■ Data Type: Integer ■ Allowed Pattern/Values: 0 to 65535
ab.corpdir.[*identifier*].ldapbinddn	The admin DN used for creating LDAP connection pool. This pool will be used for corpdir lookup ■ Default value: Not Applicable ■ Allowed Pattern/Values: (.*=.*([,;\+].*)*)? ■ Data Type: String
ab.corpdir.[*identifier*].ldapbindcred	The admin DN password, used for creating LDAP connection pool. This pool will be used for corpdir lookup. ■ Default value: Not Applicable ■ Data Type: String
ab.corpdir.[*identifier*].enableldapssl	Enable LDAP SSL ■ Default value: false ■ Allowed Pattern/Value: true or false ■ Data Type: boolean
ab.corpdir.[*identifier*].enable	Whether corporate directory is enabled or not ■ Default value: true ■ Allowed Pattern/Value: true or false ■ Data Type: boolean

Table 10–10 (Cont.) Address Book Service Configuration Properties for Convergence WABP

Option Name	Description
ab.corpdir.[*identifier*].urlmatch	Specifies the type of URL this instance of the plug-in is responsible for. This value should be unique and is case sensitive. ■ Default value: ldap:// ■ Allowed Pattern/Values: Starts with ldap:// ■ Data Type: String
ab.corpdir.[*identifier*].wildcardsearch	Specifies the minimum number of characters that need to be provided in a wildcard search. For example, 0 - entry/displayname=*, 1 - entry/displayname=a* ■ Default value: 0 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
ab.corpdir.[*identifier*].randompaging	Specifies if the plug-in support access to any page, or if each page must be accessed starting at page 1 If false, the coresrv will loop until it gets to the right page. ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.corpdir.[*identifier*].vlvpaging	Use VLV if DB has a VLV set for the default search type ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.corpdir.[*identifier*].logintype	This can be: anon (anonymous), restricted (login as user who has rights to view/write DB), or proxy (login as user that can 'masquerade') ■ Default value: restricted ■ Allowed Pattern/Values: anon, restricted, or proxy ■ Data Type: String If you are performing an anonymous search (specifically, **ab.corpdir.[***identifier***].logintype = anon**), you need to set the following additional parameters: **ab.corpdir.[***identifier***].ldaphost =** *ldap_host* and **ab.corpdir.[***identifier***].ldapport =** *ldap_port*.
ab.corpdir.[*identifier*].searchfilter	Search filter for corporate directory searches. Syntax: *(&(&(*[filter])*(|(objectClass=GROUPOFUNIQUENAMES)(objectClass=GROUPOFURLS)(objectClass=ICSCALENDARRESOURCE)(objectClass=INETORGPERSON)))(objectClass=*)))*, Where [filter] will be replaced with search criteria. Ex: If search criteria is cn=* then [filter] will be replaced with cn=* ■ Default value: null ■ Allowed Pattern/Values: Refer RFC 2254 ■ Data Type: String
ab.corpdir.[*identifier*].vlvfilter	VLV Search filter for corporate directory searches. ■ Default value: null ■ Allowed Pattern/Values: Refer RFC 2254 ■ Data Type: String

Table 10–10 (Cont.) Address Book Service Configuration Properties for Convergence WABP

Option Name	Description
ab.corpdir.[*identifier*].vlvsearchbase	VLV search base dn from where the corporate directory vlv searches are performed. ■ Default value: null ■ Allowed Pattern/Values: (.*=.*([,;\+].*)*)? ■ Data Type: String
ab.corpdir.[*identifier*].vlvsortby	VLV sort by fields for performing corporate directory searches. Multiple fields must be comma separated. For example, entry/displayname,person/surname. ■ Allowed Pattern/Values: XPath of sort by attributes. ■ Multiple fields must be comma separated. For example, XPath for cn is entry/displayname, sn is person/surname. ■ Data Type: String
ab.corpdir.[*identifier*].vlvscope	VLV Search scope used for corporate directory searches. ■ Default value: 2 ■ Allowed Pattern/Values: 0|1|2 ■ Data Type: Integer
ab.corpdir.[*identifier*].defaultserver	Default server (identifier) used for construction psRoot ■ Default value: null ■ Data Type: String
ab.corpdir.[*identifier*].displayname	Display Name for corp dir ■ Default Value: Corporate Directory ■ Data Type: String
ab.corpdir.[*identifier*].description	Description for corporate directory ■ Default Value: This is your Corporate Directory ■ Data Type: String
ab.corpdir.[*identifier*].searchattr	This defines the attributes to be used while obtaining an entry from DB. Provide the attributes as comma-separated. For example: entry/displayname,@uid. This is required especially for contacts and groups which can have different RDN's to identify them. ■ Default value: entry/displayname ■ Data Type: String Convergence can be configured to search corporate directory on required fields. For example, when the search string is "someone" and if you want to search this string only in the uid, set *ab.corpdir.[identifier].searchattr* to *@uid*. Contact is represented by XML element *<abperson uid="db:uid"> .....</abperson>*. The @ symbol is used to represent the attribute in the XML element. For example, the mapping could be something like the following: 1. uid @uid 2. displayname entry/displayname 3. givenname person/givenname 4. surname person/surname. To refer uid, use @uid. The symbol @ must be used because the uid is attribute of an element.
ab.corpdir.[*identifier*].groupoc	Comma separated list of object classes to identify group entries. ■ Default Value: (objectclass=groupOfUniqueNames) ■ Data Type: String

Table 10–10 (Cont.) Address Book Service Configuration Properties for Convergence WABP

Option Name	Description
ab.corpdir.[*identifier*].resourceoc	Comma separated list of object classes to identify resource entries. ■ Default value: (objectclass=ICSCALENDARRESOURCE) ■ Data Type: String
ab.corpdir.[*identifier*].getalldbattr	This defines if all the database attributes should be passed in the LDAP search. Valid values are true or false. ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
ab.corpdir.[*identifier*].lookthrulimit	This is the max number of entries to read in any one search. Should be set to max in directory or largest AB possible. ■ Default value: 0 ■ Allowed Pattern/Values: 0 or greater ■ Data Type: Integer
ab.corpdir.[*identifier*].collationrule	Locale on whose basis collation rule should be applied for Corporate Directory ■ Default Value: en-US ■ Data Type: String
ab.corpdir.[*identifier*].collationsearchfield	Search Fields for which collation rule should be applied. The fields provided here should be disambiguator formatted fields. For example, entry/displayname, person/givenname etc. ■ Default Value: null ■ Data Type: String

Table 10–11 Deployment or Domain Specific Configuration Properties

Option Name	Description
client.updateunreadcount	Whether to update unread count for all folders when 'Get Mail' is clicked. Default is false. ■ Allowed Pattern/Values: true or false ■ Default value: false ■ Data Type: boolean
client.mailcheckinterval	Time interval (in sec) for the client to check the mail server for new messages ■ Default value: 300 ■ Allowed Pattern/Values: 0-3600 seconds ■ Data Type: Integer
client.mailautosaveinterval	Time interval (in sec) to auto-save partially composed emails as a draft. This option is to prevent inadvertent loss of a partially composed message ■ Default value: 60 ■ Allowed Pattern/Values: 0-600 seconds ■ Data Type: Integer
client.corpabentriesperpage	Default number of entries per page used for corporate directory search. ■ Default value: 100 ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Data Type: Integer

Table 10–11 (Cont.) Deployment or Domain Specific Configuration Properties

Option Name	Description
client.dictlocale	Default dictionary used by the site for spell check ■ Default value: en-US ■ Data Type: String
client.antispamurl	Site specified service endpoint, which can permit each site to train their anti-spam service to recognize the message as spam in the future ■ Default value: /antispam ■ Data Type: String
client.autologouttime	Time out period (in min) to auto log off users (by client) after a predefined period of inactivity ■ Default value: 15 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
client.smarttznames	Site wide defined set of time zones ■ Default value: "" ■ Data Type: String
client.enablecustomization	Turn on or off customization service ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
client.enablertfcompose	Turn on/off RTF editing for entire deployment. If it set to false then user's preference to enable or disable RTF editing will be ignored by convergence client. The default value is true. ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
client.enablecorpabautocomplete	Turn on/off Auto completion of addresses from Corporate Address Book. ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
client.misc	This facilitates adding custom client preference. For example, misc.{custom-attribute}> ■ Allowed Pattern/Values: user-defined-attribute ■ Data Type: String
client.groupsearchuniqueid	Group search unique id field facilitates adding client specific custom field for unique Id. Multiple fields must be comma separated. Ex: uid,cn ■ Default value: uid ■ Data Type: string
client.groupsearchuniqueentry	Group search unique entry is XPath of group search unique id attributes. Ex: XPath for uid is entry/@entryID, cn is entry/displayname. Multiple fields must be comma separated. The order of the group search unique entry attribute should match exactly the order of group search unique id attributes. ■ Default value: entry/@entryID ■ Data Type: string

Table 10–11 (Cont.) Deployment or Domain Specific Configuration Properties

Option Name	Description
client.mainpage	Location of the static html main page ■ Default value: /iwc_static/layout/main.html ■ Data Type: String
client.loginpage	Location of the static html login page ■ Default value: /iwc_static/layout/login.html ■ Data Type: String
client.anoncalviewpage	Location of the static html Anonymous calendar view page ■ Default value: /iwc_static/layout/calendar.html ■ Data Type: String
client.uploadfilemethod	Enables or disables attachment progress indicator in HTML5 web browsers. Use [iframe | html5] method for uploading attachment file, the specified method also determines whether a progress bar can be shown. If 'iframe' method is chosen, no progress bar is shown. If 'html5' method is chosen, a progress bar is shown for HTML 5 browsers. However, non HTML 5 browsers, e.g IE 8 or 9 will revert back to iframe method ■ Default Value: html5 ■ Data Type: String ■ Allowed Pattern/Values: iframe (hide progress indicator) or html5 (display progress indicator)
client.screennameeditable	Turn on/off editing user's display name through mail's local identity option. ■ Allowed Pattern/Values: true or false ■ Default value: false ■ Default value: false ■ Data Type: boolean
client.changepasswordpage	The URL for changing the user's password after it expires ■ Data Type: String

Table 10–12 Administration Service Configuration Properties

Option Name	Description
admin.enablessl	Whether SSL is enabled for admin service ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false

Table 10–12 (Cont.) Administration Service Configuration Properties

Option Name	Description
admin.enablemonitoring	Whether monitoring is enabled ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
admin.adminpwd	Application's administrator password. This is used by the CLI/Monitoring mechanism to provide authorized access to application administration ■ Default value: Not Applicable ■ Data Type: String
admin.keystorepwd	Keystore password for SSL enabled admin server ■ Default value: Not Applicable ■ Data Type: String

Table 10–13 Single-Sign-On Configuration Properties

Option Name	Description
sso.oam.enable	This creates default configuration parameters required to enable OAM SSO mechanism. Specific parameters can further be modified/created using parameter-specific CLI option. This flag differs from sso.enable ■ Allowed Pattern/Values: true
sso.ms.enable	This creates default configuration parameters required to enable MS SSO mechanism. Specific parameters can further be modified/created using parameter-specific CLI option. This flag differs from sso.enable ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false ■ Access Type: RESTRICTED
sso.servicename	This specifies the enabled SSO service name ■ Data Type: String
sso.enable	This specifies whether SSO service is enabled or not ■ Default value: false ■ Allowed Pattern/Values: true or false ■ Data Type: boolean
sso.enablesignoff	Whether single sign off service is enabled or not ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
sso.ssoserviceimpl	SSO implementation provider name ■ Default value: Not Applicable ■ Data Type: String
sso.notifyserviceimpl	Notification service implementation ■ Default value: null ■ Data Type: String

Table 10–13 (Cont.) Single-Sign-On Configuration Properties

Option Name	Description
sso.enablerefreshsso	Whether SSO token refresh is enabled or not ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
sso.refreshinterval	After what percentage of convergence session time out interval, SSO token should be refreshed ■ Default value: 80 ■ Data Type: Integer
sso.misc	Placeholder for custom SSO provider configuration ■ Allowed Pattern/Values: user-defined-attribute ■ Data Type: String
sso.adminuid	Admin userid for SSO provider ■ Default value: Not Applicable ■ Data Type: String
sso.adminpwd	Admin password for SSO provider ■ Default value: Not Applicable ■ Data Type: String
sso.loginpage	Location of the login page to which the user is redirected to. ■ Default value: null ■ Data Type: String

Table 10–14 Instant Messaging Service Configuration Properties

Option Name	Description
im.enable	Enable or disable IM service ■ Data Type: boolean ■ Allowed Pattern/Value: true or false

Table 10–15 S/MIME Configuration Properties

Option Name	Description
smime.enable	Enable or disable S/MIME service ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Value: true or false

Table 10–16 User Preferences Configuration Properties

Option Name	Description
user.common.defaultapp	The default application to display to user upon login ■ Default value: mail ■ Allowed Pattern/Values: Name of the service. For example, mail, calendar ■ Data Type: String
user.common.theme	Specifies the name of default user interface theme used ■ Default value: theme_blue ■ Data Type: String
user.common.defaultmailhandler	Specifies the default mail handler for all mail links ■ Default value: uc ■ Data Type: String
user.common.dateformat	Specifies date display and input format ■ Default value: M/D/Y ■ Allowed Pattern/Values: This can be any of M/D/Y, D/M/Y, Y/M/D ■ Data Type: String
user.common.datedelimiter	Delimiter is the character that separates date, month and year in the date ■ Default value: / ■ Allowed Pattern/Values: This can be any of -, / or . ■ Data Type: String
user.common.timeformat	Specifies the time display format ■ Default value: 12 ■ Allowed Pattern/Values: This can be any of 12 or 24 ■ Data Type: Integer
user.common.timezone	Specifies the time zone used to normalize all time/date information in the client ■ Default value: America/Los_Angeles ■ Data Type: String
user.common.enablesmartTZ	Allows the end user to enable or disable the smart Time zone feature for the client ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Value: true or false
user.ab.name	Specifies the name of address book ■ Default value: Personal Address Book ■ Data Type: String
user.ab.description	Specifies the description of address book ■ Default value: This is the personal address book ■ Data Type: String
user.ab.entriesperpage	Specifies the number of entries to be displayed per page ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Default value: 100 ■ Data Type: Integer

Table 10–16 (Cont.) User Preferences Configuration Properties

Option Name	Description
user.cal.defaultview	Calendar view to be presented at log in ■ Default value: dayview ■ Allowed Pattern/Values: This can be any of dayview, weekview, monthview, next7view, agendaview ■ Data Type: String
user.cal.defaultcategory	Specifies the default category for a event or a task ■ Default value: Business ■ Allowed Pattern/Values: Default Category for an event or a task. Ex: Appointment, Breakfast, Business ■ Data Type: String
user.cal.daystart	Start time hour for displaying calendar information ■ Default value: 9 ■ Allowed Pattern/Values: Value of the hour in 24 hr format (0 - 23 hrs) ■ Data Type: Integer
user.cal.dayend	End time hour for displaying calendar information ■ Default value: 18 ■ Allowed Pattern/Values: Value of the hour in 24 hr format (0 - 23 hrs) ■ Data Type: Integer
user.cal.weekfirstday	First day of the week to be displayed on user's calendar ■ Default value: 1 ■ Allowed Pattern/Values: Valid values are 1 through 7. 1 - Sunday, 2 - Monday. etc. ■ Data Type: Integer
user.cal.weekenddays	Specifies the weekend days ■ Default value: 1,7 ■ Allowed Pattern/Values: Valid values are 1 through 7. 1 - Sunday, 2 - Monday. etc. ■ Data Type: String
user.cal.reminderinterval	Amount of time before the event that an alarm should be sent ■ Default value: -PT0H30M ■ Data Type: String
user.cal.enablenotify	Enables/disables email notifications being sent for the event reminder ■ Default value: 0 ■ Allowed Pattern/Values: 0 - disable, 1 - enable ■ Data Type: Integer
user.cal.enablesmsnotify	Enables/disables sms notifications being sent for the event reminder ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false

Table 10–16 (Cont.) User Preferences Configuration Properties

Option Name	Description
user.cal.enableinvitenotify	Enables/disables email notifications being sent when the calendar receives an invitation ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
user.cal.eventfilter	Specifies the type of events to be displayed ■ Default value: null ■ Data Type: String
user.mail.deleteonlogout	Specifies if mails marked as deleted has to be removed when user logs out of application ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
user.mail.autospellcheck	Specifies if auto spell check is enabled ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
user.mail.blockimages	Specifies if images in the incoming mail should be shown or blocked ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
user.mail.mailsperpage	Specifies the number of mails to display per page ■ Default value: 20 ■ Data Type: Integer
user.mail.sortorder	Specifies the sorting order ■ Default value: R ■ Data Type: String
user.mail.sortbycol	Specifies which column to be used to sort the mails ■ Default value: 6 ■ Data Type: Integer
user.mail.enablertfcompose	Specifies if compose window should use RTF ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
user.mail.displaycol	Specifies which columns to display in mail view ■ Default value: 2,1,4,3,5,6,0,7 ■ Data Type: String
user.im.defaultgroup	Default group to which the new contacts are added ■ Default value: Friends ■ Data Type: String

Table 10–16 (Cont.) User Preferences Configuration Properties

Option Name	Description
user.im.enableidlewait	Change my status to idle when I am inactive ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
user.im.idlewaittime	Change my status to idle when I am inactive for this many minutes ■ Default value: 10 ■ Data Type: Integer
user.im.enableawaywait	Change my status to away when I am inactive ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
user.im.awaywaittime	Change my status to away when I am inactive for this many minutes ■ Default value: 10 ■ Data Type: Integer
user.im.chatfont	Default text font in chat window ■ Default value: Arial ■ Data Type: String
user.im.chattypeface	Default font typeface in chat window ■ Default value: Italic ■ Data Type: String
user.im.fontsize	Default font size in chat window ■ Default value: 10 ■ Data Type: Integer
user.im.fontcolor	Default font color in chat window ■ Default value: #000000 ■ Data Type: String
user.im.bgcolor	Default background color in chat window ■ Default value: #ffffff ■ Data Type: String
user.im.showpane	Stores user preference of state of IM pane ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
user.im.lastpresencemsg	Last presence message to persist upon logout ■ Default value: "" ■ Data Type: String
user.im.lastpresencestatus	Status string indicating presence ■ Default value: online ■ Data Type: String ■ Allowed Pattern/Values: online, offline, away, invisible

Table 10–16 (Cont.) User Preferences Configuration Properties

Option Name	Description
user.im.persistpresence	Determines whether to retain user's presence upon logout ■ Default value: true ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
user.smime.sendsigned	Default option whether to digitally sign an outgoing message ■ Default value: no ■ Allowed Pattern/Values: Value can be yes or no ■ Data Type: String
user.smime.sendencrypted	Default option whether to encrypt an outgoing message ■ Default value: no ■ Allowed Pattern/Values: Value can be yes or no ■ Data Type: String
user.smime.enablepreview	Default option whether to preview an outgoing message ■ Default value: no ■ Allowed Pattern/Values: Value can be yes or no ■ Data Type: String

Table 10–17 Event Notification System Configuration Properties

Option Name	Description
ens.service.enable	Enable or disable event notification system ■ Default value: false ■ Allowed Pattern/Values: true or false ■ Data Type: boolean
ens.[*Service_Name*].enable	Enable or disable notification service associated with this service name ■ Allowed Pattern/Values: true or false ■ Data Type: boolean
ens.[*Service_Name*].servicename	The name used to identify this service in ENS. Setting this to blank deletes this service. ■ Data Type: String
ens.[*Service_Name*].datasource	The name used to identify data source for this service. ■ Data Type: String
ens.[*Service_Name*].threadpoolsize	The number of threads to be created to process incoming messages ■ Default value: 10 ■ Data Type: Integer

Table 10–18 Address Book Service JMQ Notification Configuration Properties

Option Name	Description
notify.service.enable	Enable or disable notification service ■ Default value: false ■ Data Type: boolean ■ Allowed Pattern/Values: true or false
notify.service.mq.threadpoolsize	The number of threads to be created in the publisher/subscriber service. This parameter is optional. ■ Default value: 3 ■ Allowed Pattern/Values: Greater than or equal to 1 ■ Data Type: Integer
notify.mq.[%serviceName%].servicename	The name used to identify this service. Setting this to blank deletes this service. ■ Data Type: String
notify.mq.[%serviceName%].enable	Enable or disable notification service associated with this service name ■ Data Type: boolean ■ Allowed Pattern/Value: true or false
notify.mq.[%serviceName%].destinationtype	The destination-type (Topic or Queue) of the destination associated with this service ■ Allowed Pattern/Values: TOPIC or QUEUE ■ Data Type: String

Table 10–19 Outside In Proxy Configuration Properties

Option Name	Description
oin.enable	Whether OIN service is enabled or not ■ Default value: false ■ Allowed Pattern/Values: true or false ■ Data Type: boolean
oin.host	Host name of the back-end OIN service ■ Allowed Pattern/Values: [A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)* ■ Data Type: String
oin.port	Port number at which back-end OIN service listens ■ Default value: 60572 ■ Allowed Pattern/Values: 0 to 65535 ■ Data Type: Integer

Table 10–19 (Cont.) Outside In Proxy Configuration Properties

Option Name	Description
oin.requesttimeout	Time out value in seconds to use if OIN server does not respond within this time. Zero means never time out ■ Default value: 180 ■ Allowed Pattern/Values: Greater than or equal to 0 ■ Data Type: Integer
oin.tsdirpath	Directory path for the OIN Transformation Server. Default path is /export/tsdir. Administrator needs to ensure this directory is setup with proper permissions for Convergence and Transformation Server to access. ■ Default value: /export/tsdir/ ■ Data Type: String
oin.autopruneinterval	Time interval (in minutes) to delete the transformed files in the TsdirPath ■ Allowed Pattern/Values: Greater than 0 ■ Default value: 5 ■ Data Type: Integer

11

Monitoring Convergence

This chapter describes how to collect data and monitor Oracle Communications Convergence activity.

Overview of Monitoring Convergence

Monitoring is the process of gathering, exposing, and computing run-time data to assess the performance of your Convergence deployment.

You use all the following tools to monitor Convergence:

- You use a Java management extensions (JMX) client, such as Jconsole, to gather and view JMX metrics.

  For more information about Jconsole, see the Jconsole documentation at:

  `http://docs.oracle.com/javase/7/docs/technotes/guides/management/jconsole.html`

- You use the **iwcmetrics** command-line utility to gather and view non-JMX metrics.

> **Note:** The **iwcmetrics** command cannot collect JMX-based metrics, and the JMX client cannot collect non-JMX metrics. You must use all methods to fully and properly monitor Convergence.

Before you can monitor Convergence, you must:

- Enable monitoring in Convergence

  See "Enabling Convergence Monitoring".

- Set up JMX-based server monitoring

  See "Configuring Convergence for JMX Monitoring".

See "Using Jconsole for Convergence Monitoring" for information about using Jconsole to monitor Convergence. See "About Convergence JMX Metrics" for information about the metrics collected by the JMX client.

See "Using the iwcmetrics Command for Convergence Monitoring" for information about using the **iwcmetrics** command to monitor Convergence. See "About Convergence Non-JMX Metrics" for information about the metrics collected by the **iwcmetrics** command.

Enabling Convergence Monitoring

Use the **iwcadmin** command-line utility to enable Convergence monitoring and data collection. Set the **admin.enablemonitoring** parameter to **true** and restart the GlassFish server:

```
iwcadmin -o admin.enablemonitoring -v true
```

Configuring Convergence for JMX Monitoring

To use a JMX-compliant GUI tool, such as or Jconsole, you must configure JMX-based server monitoring, the JVM, and the JAAS. For more information on JMX and JAAS settings and configuration files, see the JMX documentation at:

http://docs.oracle.com/javase/7/docs/technotes/guides/management/agent.html

Using Jconsole for Convergence Monitoring

Jconsole is a JMX client which you can use to collect and view Convergence JMX metrics. See "About Convergence JMX Metrics" for more information about the metrics you can collect and view with Jconsole.

To use Jconsole for Convergence monitoring:

1. Start Jconsole with the following command:

   ```
   $JAVA_HOME/bin/jconsole
   ```

   The Jconsole Connection Agent dialog box appears.
2. Click the **Advanced** tab.
3. In the **JMX URL** field enter

   service:jmx:rmi://*hostname*:*port*/**jndi/rmi://***hostname*:*port*/**jmxrmi.**

   > **Tip:** You can obtain this URL from the **iwc.log** file. The JMX console URL is written to the log file when Convergence server starts the admin server. For example:
   >
   > ```
   > CONFIG: INFO from com.sun.comms.client.admin.web.JMXAgent  Thread
   > pool-1-thread-7 \\
   > at 2009-02-23 21:55:31,981 - RMI connector server in non-SSL mode
   > started successfully.
   > CONFIG: INFO from com.sun.comms.client.admin.web.JMXAgent  Thread
   > pool-1-thread-7 \\
   > at 2009-02-23 21:55:31,983 - Service URL is: \\
   > [ service:jmx:rmi://siroe.com:50005/jndi/rmi://siroe.com:50005/
   > jmxrmi ]
   > ```

4. Enter the administrator user name and password.
5. Click **Connect**.
6. Expand the **Monitoring** node.

On the right hand side of the screen you will see the various components of JVM available in tabs. The leaves under the Monitoring node on the left hand side shows the various Instruments that can be used to monitor the JVM.

See "About Convergence JMX Metrics" for a list of the metrics available.

About Convergence JMX Metrics

A JMX client can collect and view the following Convergence metrics:

- Authentication LDAP
  - Host name of the directory server from which the connections are being served
  - Number of free connections in the pool
  - Number of used connections in the pool
- Calendar Service Connection
  - Total number of active sessions
  - Details of each active session. Including user ID, IP address, domain name, and the duration of this connection
  - Number of sessions since the start of the server
- Mail Service Connection
  - Total number of active sessions
  - Details of each active session. Including user ID, IP address, domain name, and the duration of this connection
  - Number of sessions since the start of the server
- Instant Messaging Service Connection
  - Number of sessions since the start of the server
- Session
  - Total number of active sessions
  - Details of each active session
  - Number of sessions since the start of the server
  - Number of failed attempts
- User and Group LDAP
  - Host name of the directory server from which the connections are being served
  - Number of free connections in the pool
  - Number of used connections in the pool
- Server
  - Active server duration

Note: The JMX client cannot collect non-JMX metrics. See "Overview of Monitoring Convergence" for information about collecting non-JMX metrics.

Using the iwcmetrics Command for Convergence Monitoring

The **iwcmetrics** command-line utility is a script in the *Convergence_Home*/**sbin** directory which you can use to collect and view Convergence non-JMX metrics. See "About Convergence Non-JMX Metrics" for information about Convergence non-JMX

metrics.

The following example shows the syntax of the **iwcmetrics** command:

```
iwcmetrics -U Convergence_URL -u user_name [-W password_file] -m
Metric1,Metric2,MetricN
```

Table 11–1 describes the valid parameters for the **iwcmetrics** command.

Table 11–1 Parameters for iwcmetrics Command

Parameter	Description
-U	Specifies the complete Convergence URL: http(s)://*hostname.domain*:*port*/*URI*. For example: `https://Convergence.MyDomain.com:8181/iwc`
-u	Specifies the user name. The **iwcmetrics** command can only collect metrics for the services which the user is privileged to use. To collect metrics for all services, specify a user name that has access to all Convergence services.
-W	Specifies the location of the encrypted password file. If you omit the `-W` parameter, the command-line utility asks you to provide your password. For this reason, the `-W` parameter is omitted from all examples in this guide.
-m	Specifies the metrics to collect. This parameter can specify a single metric, a comma-separated list of metrics, or one or more entire groups of metrics. Metrics are grouped together by service. The `-m` parameter supports the following groups: iwc (Convergence), mail (email), caldav (calendar), im (instant messaging), nab (Contacts Server address book), iss (indexing and search service). For example: `iwcmetrics -U Convergence_URL -u user_name -m metric1,metric2,group1,group2` Omit the `-m` parameter to collect all metrics. For example: `iwcmetrics -U Convergence_URL -u user_name` See "About Convergence Non-JMX Metrics" for more information about Convergence non-JMX metrics and the groups to which they belong.
-l	Lists all available metrics. You do not need to specify a user name or the Convergence URL. For example: `iwcmetrics -l`
-h	Displays information and help for the **iwcmetrics** command. You do not need to specify a user name or the Convergence URL. For example: `iwcmetrics -h`

The following list gives examples of using the **iwcmetrics** command:

- To display a list of all available metrics:

  ```
  iwcmetrics -l
  ```

- To display the help for the **iwcmetrics** command:

  ```
  iwcmetrics -h
  ```

- To collect all metrics:

  ```
  iwcmetrics -U Convergence_URL -u user_name
  ```

- To collect all metrics pertaining to the mail and address book services:

  ```
  iwcmetrics -U Convergence_URL -u user_name -m mail,nab
  ```

- To collect two metrics from different groups:

```
iwcmetrics -U Convergence_URL -u user_name -m im.responsetime,caldav.status
```

About Convergence Non-JMX Metrics

Table 11–2 lists the Convergence metrics that can be collected and viewed using the **iwcmetrics** command.

Table 11–2 Parameters for iwcmetrics Command

Parameter Name	Description
iwc.loginresponsetime	A measure of the time taken to log into Convergence. This metric is part of the iwc group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m iwc.loginresponsetime`
mail.status	Indicates the status of Oracle Communications Messaging Server. A value of **0** indicates that it is working. This metric is part of the mail group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m mail.status`
mail.responsetime	A measure of the response time between Convergence and Messaging Server. This metric is part of the mail group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m mail.responsetime`
nab.status	Indicates the status of Oracle Communications Contacts Server. A value of **0** indicates that it is working. This metric is part of the nab group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m nab.status`
nab.responsetime	A measure of the response time between Convergence and Contacts Server. This metric is part of the nab group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m nab.responsetime`
caldav.status	Indicates the status of Oracle Communications Calendar Server. A value of **0** indicates that it is working. This metric is part of the caldav group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m caldav.status`
caldav.responsetime	A measure of the response time between Convergence and Calendar Server. This metric is part of the caldav group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m caldav.responsetime`
im.status	Indicates the status of Oracle Communications Instant Messaging Server. A value of **0** indicates that it is working. This metric is part of the im group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m im.status`

Table 11–2 (Cont.) Parameters for iwcmetrics Command

Parameter Name	Description
im.responsetime	A measure of the response time between Convergence and Instant Messaging Server. This metric is part of the im group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m im.responsetime`
iss.status	Indicates the status of Oracle Communications Indexing and Search Service. A value of **0** indicates that it is working. This metric is part of the iss group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m iss.status`
iss.responsetime	A measure of the response time between Convergence and Indexing and Search Service. This metric is part of the iss group. Example: ■ `iwcmetrics -U Convergence_URL -u user_name -m iss.responsetime`

Note: The **iwcmetrics** command cannot collect JMX metrics. See "Overview of Monitoring Convergence" for information about collecting JMX metrics.

12

Troubleshooting Convergence

This chapter describes how to resolve problems you encounter in Oracle Communications Convergence.

Configuring Log Levels to Gather Information

This section covers how to configure log levels for the Convergence server. Log levels can be set by using the **iwcadmin** command.

For more information on the **iwcadmin** command, see "Using the Convergence Administration Utility".

The following are the log configuration parameters:

- LogLocation: Path to the directory where the log file is stored.
- LogPattern: Declares the information and format to specify what to log and in what format. For more information about how to specify the LogPattern, see the Log4J specification on the Apache web site:

  `http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html`

- LogRotation: Log rotation specifies the policy for rolling over logs to a new location. This release includes the following policies:
  - SizeTrigger policy: SizeTrigger is defined as the number of bytes of log information to accumulate before rolling the log over to a new location.
  - TimeTrigger policy: TimeTrigger is defined as the time of day to roll over the log to a new log location. The value is expressed as a *SimpleDatePattern*.
- Logger: The initial system Logger value is DEFAULT, that takes the default LogLevel. However, each module in Convergence can control the logging level of its own logs. For example, the authentication module might name its logger AUTH and set the log level to WARN. To know more about the various logging levels, see "About Log Levels".

Logging levels (*LogLevel*) are set using a predefined default set of log levels. For example:

- DEBUG
- INFO
- WARN
- ERROR
- OFF

The DEBUG level is the most verbose level. Do not to use this for everyday logging as it negatively impacts the server's performance. However, you should use this level when you need to trap as much information about a recurring problem. After capturing the required log data, you should return the log level to a lesser level of log setting.

13

Overview of Add-on Services in Convergence

This chapter describes the add-on framework, the add-on configuration files, and instructions for adding or removing these third-party services in the Convergence UI.

You can use the add-on framework to add third-party services to Convergence. For example:

- advertising
- click-to-call service
- multinetwork instant messaging
- SMS (both one-way and two-way)
- social media applications (Facebook, Twitter, and Flickr)
- video and voice calling capability with WIT
- video and voice calling capability and screen sharing with WebRTC integration

About the Add-on Framework

The add-on framework provides access from Convergence to the third-party service through the use of an ID. The ID is provided by the service. The method of getting the ID differs from service to service. See the individual services sections for information on how to obtain the ID.

The add-on services are configured through Convergence configuration files. The available add-on services are listed in Table 13–1. The names of the add-on services, as used in the configuration files, are given in parentheses. For example, the properties file for configuring the multinetwork instant messaging add-on is **imgateways.properties**:

Table 13–1 Add-On Services

Service Name	Description
Advertising (advertising)	Displays banner ads, text ads, and contextual ads in the Convergence UI. For more information, see "Configuring the Advertising Add-On Service in Convergence".
Click-to-Call (video)	Allows for video and audio/voice calling by clicking contact information in Instant Messaging or Address Book services in Convergence. For more information on WIT integration, see "Configuring WebRTC in Convergence with WIT Software". For more information on WebRTC integration, see "Configuring WebRTC in Convergence with WebRTC Session Controller". With WebRTC, you can also do screen sharing and multiparty video.

Table 13–1 (Cont.) Add-On Services

Service Name	Description
Video and audio calling (video)	Provides video and audio/voice calling through Convergence. For more information on WIT integration, see "Configuring WebRTC in Convergence with WIT Software". For more information on WebRTC integration, see "Configuring WebRTC in Convergence with WebRTC Session Controller". With WebRTC, you can also do screen sharing and multiparty video.
Multinetwork Instant Messaging (imgateways)	Provides access to all of a user's instant messaging accounts in a single place. You can currently chat with Facebook buddies when you set up Facebook with the social add-on service. For more information, see "Configuring Multinetwork Instant Messaging Add-On Services".
SMS (sms)	Provides one- and two-way SMS through Convergence. For more information, see "Configuring Convergence for SMS".
Social Media (social)	Provides access to social media, such as Facebook, Flickr, and Twitter, through a *Social* tab in the Convergence UI. Social media applications can only be added to the Social tab. For more information, see "Configuring Social Add-On Services in Convergence". Facebook integration with Convergence is not currently working due to changes Facebook made to their APIs, interfaces, and application registration approval. Oracle is exploring potential solutions and will provide an update when a solution is available.

About the Add-On Configuration Files

The configuration files for supported add-ons are installed in the **/var/opt/sun/comms/iwc/config/** directory. There are three types of configuration files:

- **add-ons.properties**
- **addon_name.json**
- **addon_name.properties**

add-ons.properties

The **add-ons.properties** file specifies the add-ons that are to be enabled. The following file lists the add-on services that are available in the **add-ons.properties** file that comes with the installation.

1. Add on configuration.
2. A sample entry look like this:

```
# addons=social, sms, advertising
addons=social, sms, advertising, video
```

Verify that the service you want to add to your Convergence deployment is in the **add-ons.properties** file before going onto configuring that service.

addon_name.json

Each add-on has its own **addon_name.json** file, containing **client** configuration parameters for the add-on. You must enable the add-on service in **addon_name.json** in order for the service to be active in your Convergence deployment.

In the following **sms.json** file, the SMS service is enabled. The comments describe each parameter:

```
{
    enabled: true,
    twowaysmsenabled:true, // Whether two way SMS is enabled or not
    channel: "sms-handle", // Messaging server SMS channel name
    folder:"SMS",          // Name of the SMS folder, where messaging server keeps
all
                                                    // the SMS messages
    NDNFolder:'INBOX'      // Where, Non Delivery Notification messages for SMS
will be
}
```

To disable an add-on service, set *enabled:false* in the **addon_name.json** file.

addon_name.properties

The **addon_name.properties** file contains **server** access parameters for add-on services. Social add-on services and some video add-on services use an **addon_name.properties** file: **social.properties**. The following example shows the authentication configuration for Twitter in the **social.properties** file:

```
...
serviceid = twitter

# OAuth related configuration
twitter.oauth.consumer.key=consumer_key
twitter.oauth.consumer.secret=consumer_secret
twitter.oauth.authorize.url=https://api.twitter.com/oauth/authorize
twitter.oauth.request.token.verb=POST
twitter.oauth.request.token.url=https://api.twitter.com/oauth/request_token
...
```

Configuring WebRTC Services in Convergence

Convergence supports many Web real-time communication (WebRTC) features, including peer-to-peer voice and video chat and screen sharing.

WebRTC is a standards-based API definition that does not require the use of web browser plug-ins or downloads.

You can integrate Convergence with either Oracle Communications WebRTC Session Controller or WIT Communications Server to deliver WebRTC services.

See one of the following sections for more information:

- Configuring WebRTC in Convergence with WebRTC Session Controller
- Configuring WebRTC in Convergence with WIT Software

To use the WebRTC services in Convergence, users must be using a supported browser and have a microphone and a webcam.

Configuring WebRTC in Convergence with WebRTC Session Controller

You can integrate Convergence with Oracle Communications WebRTC Session Controller to deliver WebRTC services in Convergence.

Install WebRTC Session Controller according to its documentation.

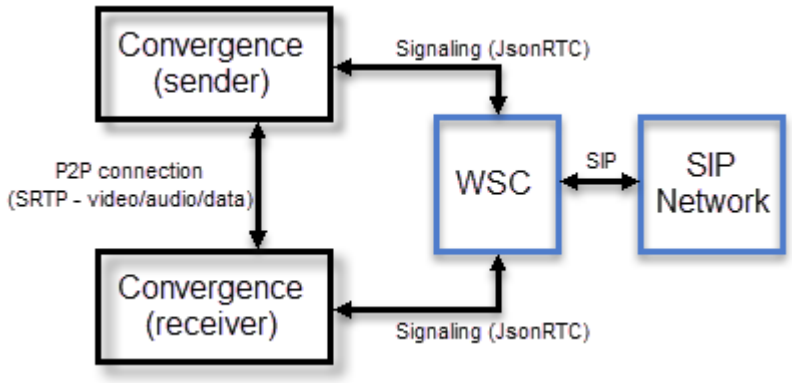
Convergence uses WebRTC Session Controller as a signaling channel to coordinate the communication between peers and to send control messages to start and end communication between peers and to report errors. WebRTC Session Controller also exchanges network configuration information and media capability with other communication channels. Signaling between peers must be successful before streaming can begin.

WebRTC Session Controller uses the JsonRTC protocol to establish a connection between the clients and to send messages between the Convergence sender and receiver.

Once Convergence and WebRTC Session Controller are integrated, you can enable peer-to-peer (P2P) communication between Convergence clients as well as Session Initiation Protocol (SIP) clients.

Figure 13–1 shows the high-level architecture of WSC and Convergence:

Figure 13–1 WSC and Convergence Architecture



WebRTC Session Controller uses a browser module, RTCPeerConnection, to communicate real time data between the Convergence sender and receiver. The data is sent through the Secure Real-time Transport Protocol (SRTP), which provides enhanced security features for real time transmission of multimedia data.

WebRTC Session Controller also integrates with SIP and the Public Switched Telephone Network (PSTN). For more information, see the WebRTC Session Controller documentation.

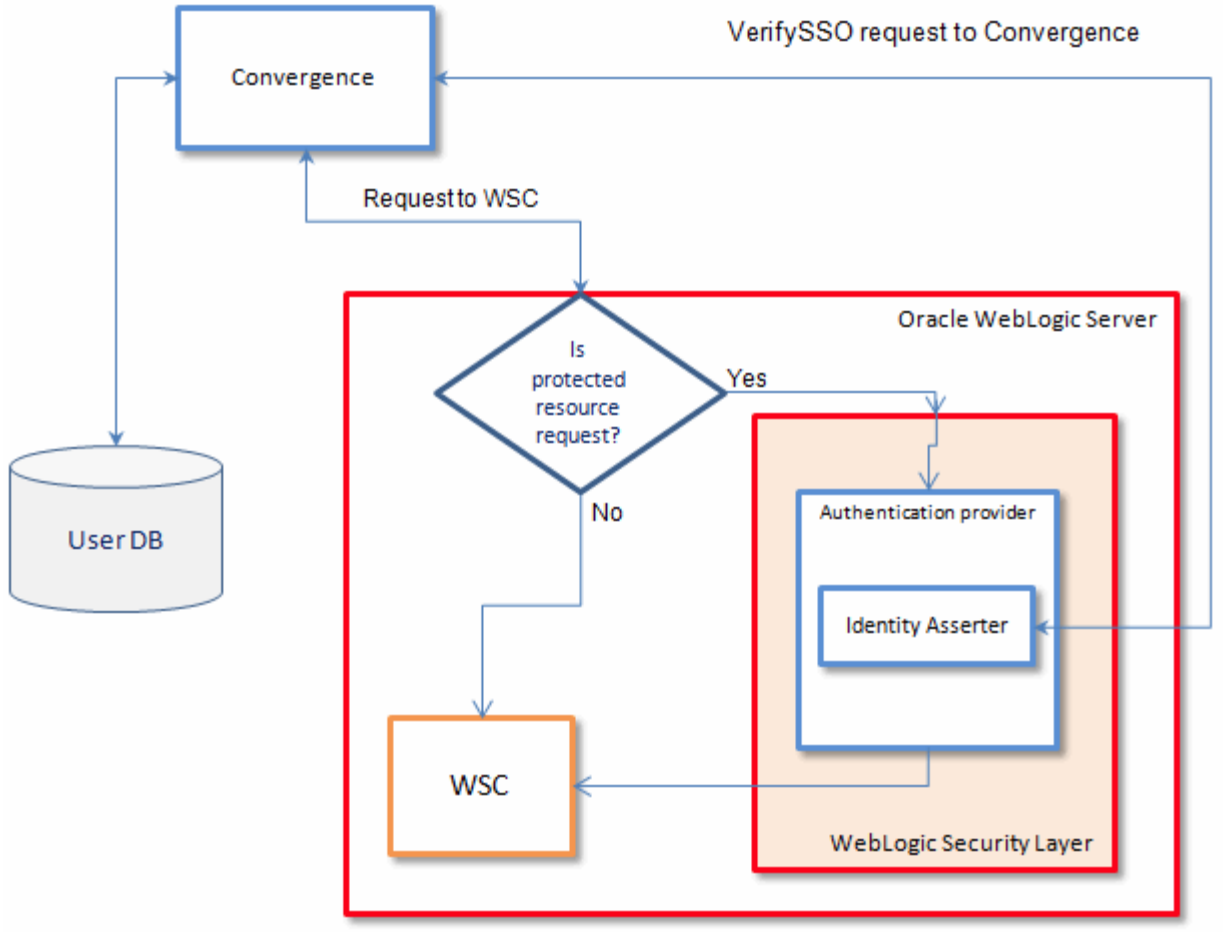
Convergence uses the WebRTC Session Controller JavaScript API to obtain access to a user's media (camera/microphone). For more information on the WSC JS API, see *WebRTC Session Controller Web Application Developer's Guide*.

About Authentication with WebRTC Session Controller and Convergence

Convergence uses Trusted Circle SSO for authentication. First, Convergence sends a request to access a protected resource in WebRTC Session Controller. If the resource request is not protected, Convergence is connected to the WebRTC Session Controller. If WebLogic Server detects that the requested resource is protected, it passes requests to an authentication provider which retrieves userID and token/sessionID from requests. Next, the authentication provider sends a request to the Convergence VerifySSO module. Convergence verifies if the user in the request has a valid session. If authentication succeeds, the request is trusted by WebLogic server the request will be passed to WSC server.

Figure 13–2 shows the trusted circle SSO authentication flow for Convergence with WSC.

Figure 13–2 Trusted Circle SSO Authentication Flow for Convergence with WebRTC Session Controller



Mapping Between Web Identity and IP Multimedia Subsystem Identity

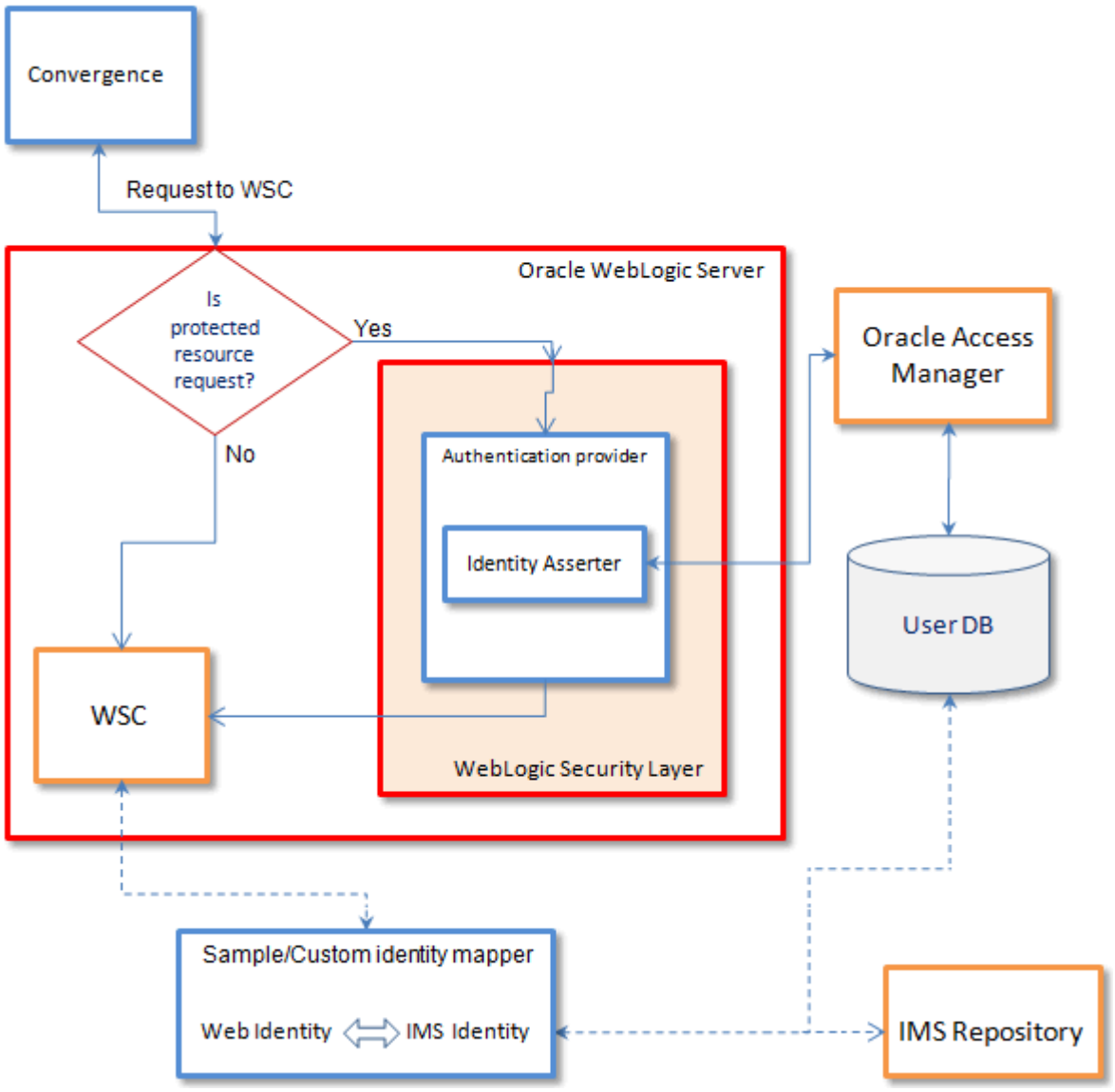
You can customize and implement your WebLogic authentication provider to map web identity with IP Multimedia Subsystem (IMS) identity.

Because customers might have different IMS user repositories, you can use a sample module to fetch mapping identity from your LDAP server. You write your own plug-in to interact with your IP Multimedia Subsystem (IMS) identity to fetch the mapping identity. If you already have an identity assertion system in place, then, you can add a custom authentication provider to the chain so that it looks at the principal returned by the identity asserter. In addition, you can find the SIP credentials for that principal and add the same credentials to the subject. You can follow this same approach if you are using an existing authentication provider.

Once mapping identity information is fetched, you can construct javax.security.auth.Subject and pass it to WebRTC Session Controller.

Figure 13–3 shows the WebRTC Session Controller security architecture for Convergence using the identity mapper.

Figure 13–3 WebRTC Session Controller Security Architecture for Convergence Using Identity Mapper



Enabling WebRTC Services in Convergence

To enable the WebRTC services in Convergence:

1. Add **video** to the **add-ons.properties** file.

   The **add-ons.properties** file is located in the **/var/opt/sun/comms/iwc/config/** directory. To enable the **video** add-on, add it as a value of the **addons** parameter, as in the following example:

   ```
   addons=sms, imgateways, advertising, social, video
   ```

   See "add-ons.properties" for information on the **add-ons.properties** file.

2. Edit the **video.json** file in the **/var/opt/sun/comms/iwc/config** directory for your WebRTC Session Controller configuration:

   a. Set **enabled** parameter to **true.**

   b. Set **provider** parameter to **wsc.**

c. Update the following parameters to point to the WebRTC Session Controller server. URLs to the WebRTC Session Controller server must use the server host name, not the server IP address.

- **wsurl**: The WebSocket URI to which Convergence connects. The context (for example, **/ws/webrtc/convergence**) should be one of application RequestURI in wsc-console.
- **logouturl**: The URL used to logout WebRTC Session Controller, the parameter "wsc_app_uri" is mandatory and the value should be the WebSocket context to which Convergence connects
- **isurl**: The WebRTC Session Controller JavaScript API URL
- **isextnurl**: URL points to WebRTC Session Controller extension Javascript API in Convergence

d. Update **pstnDomain** according to PSTN gateway domain if it is configured.

Note: The PSTN gateway is required to initiate calls with PSTN numbers. The PSTN gateway domain information can be obtained from the PSTN gateway administrator.

e. Add *provider* information. For example:

```
{
    "enabled": true,
    "provider": "wsc",
    "wsc": {
        "enabled": true,
        "jsPackageName" : "iwc.packages.Wsc",
        "jsClassName" : "iwc.service.addon.WSC",
        "logLevel": 2,
        "serviceDetails": {
            "wsurl": "wss://mercury-vm43.example.com:7002/ws/webrtc/
convergence",
            "logouturl": "https://mercury-vm43.example.com:7002/
logout?wsc_app_uri=/ws/webrtc/convergence",
            "jsurl": "https://mercury-vm43.example.com:7002/api/wsc.js",
            "jsextnurl": "../js/iwc/service/addon/wsc/lib/
wsc-extension.js",
            "pstnDomain": "anydomain.com",
        },
    }
```

3. Configure Trusted Circle SSO for Convergence Server.

- Set the SSO parameters using the **iwcadmin** command.

```
iwcadmin -f /space/enable-mssso
```

where */space/enable-mssso* and must have the following SSO parameters:

```
cat /space/enable-mssso
sso.ms.enable = true
sso.misc.IPSecurity = false
sso.misc.CookieAppPrefix = WSCAuthnToken
sso.misc.CookieDomain = .example.com  \\ Convergence and WebRTC Session
Controller must be in the same domain, or authentication fails and video
service is not possible.
sso.misc.SessionCookie =JSESSIONID
```

```
sso.misc.IWC-AppID =iwc
sso.misc.iwc-VerifyURL = http://convergenceHost:Port/iwc_context/VerifySSO?
```

Where *iwc_context* is the Convergence context path.

Note: If the value for **sso.misc.iwc-VerifyURL** is an HTTPS URL, and if SSO with WebRTC Session Controller does not work, make sure that you have imported the Convergence GlassFish's CA certificate into the WebRTC Session Controller WebLogic Server's trust store.

Configuring WebRTC Session Controller for Convergence

To make the WebRTC Session Controller work with Convergence, you need to run the WLST script configuration and do additional manual configuration:

1. Copy the **wlst** directory from **/opt/sun/comms/iwc/resources/uc/wsc/scripts/** to either your WebRTC Session Controller or WebLogic Server Home directory.
2. In **wlst/README.txt**, for the **allowedDomains** parameter, provide the complete host name as a value. The domain name is not a valid value. An asterisk is a valid value, meaning for all host names.
3. Run the WLST script to add the extension in WebRTC Session Controller. Instructions are provided in **wlst/README.txt**.
4. Configure Media Engine by adding the Media Engine node through the WebRTC Session Controller Console. See the discussion about signaling properties and media nodes in *WebRTC Session Controller System Administrator's Guide* for more information.
5. If the Media Engine is enabled, set **DMA_ENABLED** to **true** and set **PROXY_SIP_URI** (default registrar and router address) in the Script Library through the WebRTC Session Controller console.
6. Configure the SIP Proxy Server and Registrar IP address to accept all the domains on which Convergence is running.

Configuring the SSO Provider on WebLogic Server

1. Configure the SSO provider on the WebLogic Server by putting the SSO provider JAR file in WebRTC Session Controller, available with the Convergence installation files.
   a. Copy the **WLSIWCIdentityAsserter.jar** from the **/opt/sun/comms/iwc/lib/jars/** directory to the WebLogic installation directory, **WL_HOME\server\lib\mbeantypes**.
   b. Restart WebLogic Server if it is already running
2. Configure the identity assertion provider in the WebLogic administration console for WebRTC Session Controller.
   a. Log into the WebLogic administration console.
   b. In the left pane, select **Security Realms** and click the name of the realm to be configured, such as *myrealm*.
   c. Select **Authentication** from **Select Providers** and click **New**.
   d. Enter a name for the provider to be configured such as WLSIWCIdentityAsserter.

e. From the **Type** menu, select WLSIWCIdentityAsserter.

f. On the **Common** tab of the **Configuration** page for this authentication provider, make sure that **Base64DecodingRequired** is set to **false** and active types has **WSCAuthnToken-iwc** token type in the chosen list.

g. On the Provider-specific tab of the Configuration page, set the value of the 'User Group' attribute to the value of **Security Group** of the WebRTC Session Controller application. The value of WebRTC Session Controller application's Security Group can be found in the WebRTC Session Controller console. The **Application** tab in the WebRTC Session Controller console shows the value of Security Group, Request URI, allowed domains, and so forth.

Note: If the value of the provider's 'User Group' attribute differs from the Security Group configured in the WebRTC Session Controller application, the user authentication fails and video service is not available in Convergence. For example, if you have configured the Security Group as 'convergence' in the WebRTC Session Controller application, the SSO provider should also be configured with 'convergence' as the user's group name.

---

h. Make sure that the authentication provider's list does not contain the guest login, WSCServletAuthenticator.

i. On the **Common** tab of the **Configuration** page for the default authenticator, change the 'Control Flag' to **OPTIONAL**.

j. Restart WebLogic Server.

Configuring WebRTC in Convergence with WIT Software

You can integrate Convergence with WIT Communications Server to deliver WebRTC services in Convergence.

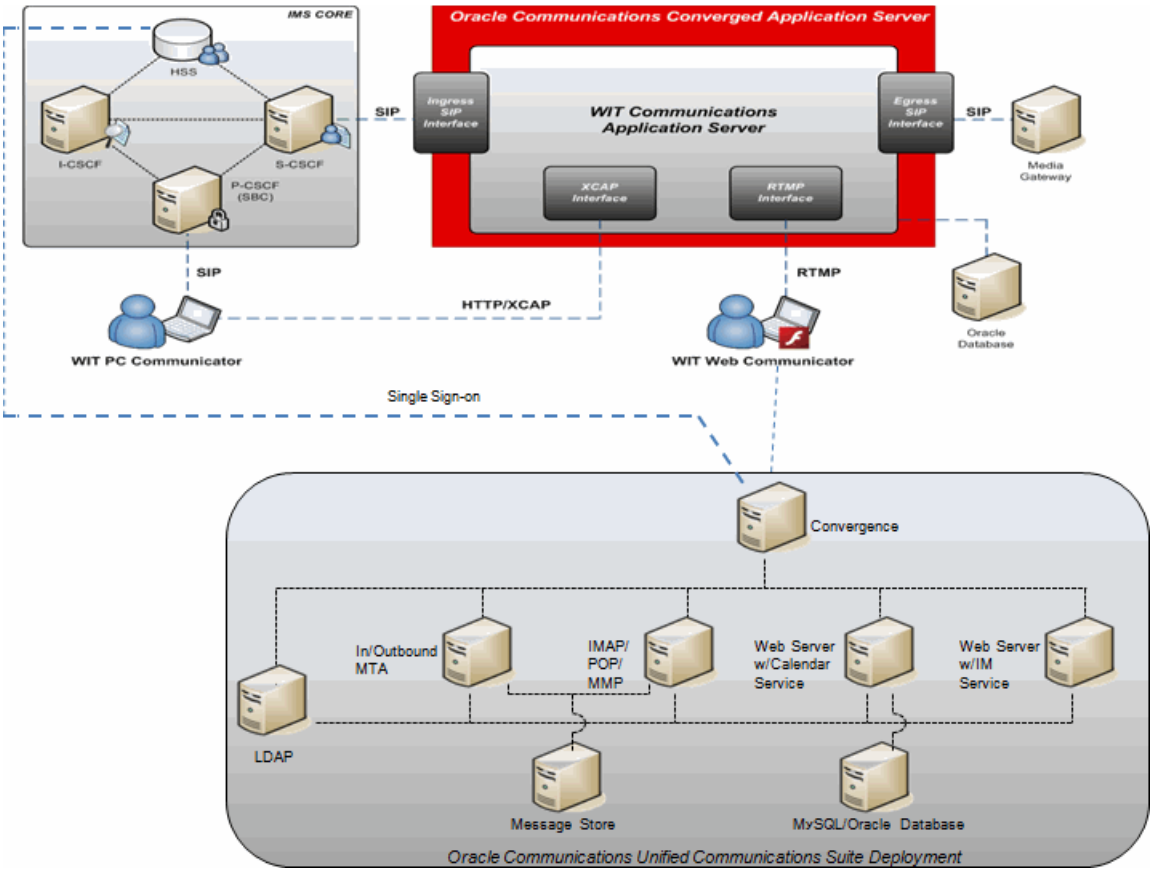
Install WIT Communications Server according to its documentation.

The WIT Software provides the following WebRTC features:

- Browser to Browser Audio Call - caller calls contact in his address book in the Convergence UI; the recipient must have browser to browser audio call enabled. Recipient answers call through browser.
- Browser to Phone Audio Call - caller calls contact in his address book in the Convergence UI. Recipient answers call on phone.
- Click-to-Call - caller clicks recipient's phone number stored in his address book the Convergence UI.
- Skype Call - caller selects option to call recipient by Skype.
- Video Call - caller does a live video call with recipient through address book or IM in the Convergence UI. Both caller and recipient must have video call enabled.

Convergence integrates with WIT Software's WIT Communications Server to provide back-end support for video and audio call. The WIT Communications Server works with Wowza Media Server to provide streaming. Figure 13–4 shows the WIT integration architecture with Convergence.

Figure 13–4 WIT Integration Architecture



You can integrate WebRTC capabilities to enable voice and video calls and screen sharing through Address Book or IM. For more information, see "Configuring WebRTC in Convergence with WebRTC Session Controller".

Adding WebRTC Services to Convergence with WIT Software

1. Add **video** to the **add-ons.properties** file.

   The **add-ons.properties** file is located in the **/var/opt/sun/comms/iwc/config/** directory. To enable the **video** add-on, add it as a value of the **addons** parameter, as in the following example:

   ```
   addons=sms, imgateways, advertising, social, video
   ```

   See "add-ons.properties" for information on the **add-ons.properties** file.

2. Use the **video.json** file in the **/var/opt/sun/comms/iwc/config** directory for your WIT configuration.

   - Set the initial line with *"enabled"* to *"true"*.
   - Change the *"provider"* value from *"wsc"* to *"wit"*.
   - Include *serverHost*, *serverPort*, *proxyHost*, *proxyPort*, and *userDomain* details from your WIT configuration. For example:

     ```
     video.json
     {
             enabled: true,
             serviceDetails:{
     ```

```
        locale: 'EN',
                serverHost: '10.178.212.172',
                serverPort: '1935',
                proxyHost: '10.178.212.172',
                proxyPort: '2222',
                backupPort: '1935',
                singleSignOnVerificationUrl: 'http://convergenceHost:Port/
iwc/VerifySSO?',
                isSingleSignOn: true,
                ssoCookieName: 'kendo-sso-iwc',
                policyFilePort: '843',
                userDomain: 'wcas.wit-software.com'
        }
}
```

Note: The **video.json** file is used for all audio and video add-on services that use the WIT server: video call, audio call, and click-to-call. Do not use the **clicktocall.json** to enable click-to-call services.

3. Configure Trusted Circle SSO.
   - Set the SSO parameters using the **iwcadmin** command.

     ```
     iwcadmin -f /space/enable-mssso
     ```

     where */space/enable-mssso* and must have the following SSO parameters:

     ```
     cat /space/enable-mssso
     sso.ms.enable = true
     sso.misc.CookieAppPrefix = kendo-sso
     sso.misc.CookieDomain = .com
     sso.misc.IPSecurity = false
     sso.misc.iwc-VerifyURL = http://convergenceHost:Port/iwc_context/VerifySSO?
     ```

     Where *iwc_context* is the Convergence context path.

Implementing the Identity Mapper

Identity mappers are used for authenticating WebRTC with video services in Convergence. By default, Convergence implements a default Convergence identity mapper which does mapping using the user/group LDAP store. An identity mapper is a sample authentication provider which can be used to perform Web ID to SIP (IMS) ID mapping for non-LDAP directory configurations.

Convergence uses a plug-in architecture so that you can implement your own custom identity mapper to your unique IMS user repository. The Convergence UI requests mapping values only if mapping is enabled on the Convergence server.

A new protocol command and Java interface in the Convergence server can be used to plug in your own implementation. The Convergence client uses this protocol to get mapping IDs. The new protocol's command handler has its own configuration; it is used as part of the video service configuration. The configuration has the capability to plug in a custom implementation that should adhere to the new Java interface.

See "Creating a Custom Identity Mapper" for information about creating a custom identity mapper.

Enabling Sound Alerts in Firefox

A sound alert audibly notifies a user when there is an incoming or outgoing WebRTC voice or video call. This section describes how to get sound alerts to work in Firefox.

Because of an issue with the mp3 MIME type that is returned by GlassFish Server, sound alerts do not work by default with Firefox browsers.

Use the following workaround to turn on sound alerts in Firefox:

1. Modify the **default-web.xml** file in *Convergence_Domain*/**config**.
2. Search for the MP3 extension and replace the **x-mpeg** mime_type with **mpeg** so it looks like the following example:

```
<mime-mapping>
<extension>mp3</extension>
<mime-type>audio/mpeg</mime-type>
</mime-mapping>
```

3. Restart the GlassFish server.

Creating a Custom Identity Mapper

You can create you own identity mapper for Convergence to deliver WebRTC services.

By default, Convergence uses User/Group LDAP directory server (that contains user information such as display name, time zone, telephone number, and so on) for identity mapping for WebRTC video services. The connection is performed by a Convergence identity mapper that is enabled by default.

You can use a custom identity mapper to look up routeable IDs in RDBMS, flat file, or HSS databases. Custom identity mappers are not intended to be used with Oracle Communications User/Group LDAP directory servers. The Convergence server provides an interface that enables you to create a custom identity mapper for WebRTC video services. This custom identity mapper is not available for WIT services.

Before designing a solution, you need to plan the following:

- The Convergence identity mapper framework is the default framework that is designed to work with Oracle Communications User/Group LDAP Directory Server.
- A custom identity mapper must be implemented for use with a non-LDAP user/groups directory.
- The custom identity mapper uses the com.sun.comms.client.addon.video.IdentityMapper Java interface that needs to be followed by custom identity mapper Java class This interface defines set of methods. For example, public String getMappedValue(String sourceValue) throws Exception;.
- The custom identity mapper library JAR file should be in the class path that is accessible by Convergence.
- While implementing the custom identity mapper, iwc.jar should be available in the classpath of development environment. The iwc.jar defines the interface that needs to be implemented by custom identity mapper.

Developing Sample Custom Identity Mapper Data Files

This section describes the files that are created for the custom identity mapper to work. Use this information as a reference to create other custom identity mappers to suit

your needs. This sample identity mapper uses file based look-ups to perform the identity mapping.

The following file (**idmappinginfo.txt**) is a sample set of data that could be used to retrieve mapped values:

```
hari@idmappersample.com=123@idmappersamplesip.com
abhi@idmappersample.com=456@idmappersamplesip.com
alice@idmappersample.com=789@idmappersamplesip.com
```

In **idmappinginfo.txt**, attributes are separated by an equal (=) characters. For example, the first record in the file provides information about the user hari@idmappersample.com whose routeable ID is 123@idmappersamplesip.com.

The custom class should implement the interface com.sun.comms.client.addon.video.IdentityMapper. This interface has the following two methods that should be implemented by the implemented class:

- **public void init(Properties ConfigProps)**: Store configuration properties so that these configuration properties can be used by other methods.
- **public String getMappedValue(String sourceValue) throws Exception**: Returns the mapped value for the specified source value, or null if neither one is found.

FileBasedCustomIdentityMapper.java below describes the classes that are used to implement the custom identity mapper using a file-based mapping store. The following is the core class:

```
package com.sun.comms.client.services.sample.identitymapper;

import com.sun.comms.client.addon.video.IdentityMapper;
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Properties;

public class FileBasedCustomIdentityMapper implements IdentityMapper {

    private static String mappingInfoFile;
    private Properties configProps;

    @Override
    public void init(Properties configProps) {
        this.configProps = configProps;

        mappingInfoFile = (String) this.configProps.get("idmappinginfofile");
    }

    @Override
    public String getMappedValue(String sourceValue) throws Exception {
        String mappedValue;

        mappedValue = getFromFile(sourceValue);

        return mappedValue;
    }

    private String getFromFile(String source) {

        if (source == null) {
```

```
            return null;
        }

        FileInputStream fis = null;
        BufferedReader reader = null;

        try {
            fis = new FileInputStream(mappingInfoFile);
            reader = new BufferedReader(new InputStreamReader(fis));

            String line = reader.readLine();
            while (line != null) {
                String[] keyValue = line.split("=");
                if (source.equals(keyValue[0])) {
                    return keyValue[1];
                }
                line = reader.readLine();
            }

        } catch (FileNotFoundException ex) {
            System.out.println("Mapping information file '" + mappingInfoFile + "'
not found.");
        } catch (IOException ex) {
            System.out.println("Unable to read from '" + mappingInfoFile + "' due
to:" + ex.getMessage());

        } finally {
            try {
                if (reader != null) {
                    reader.close();
                }
                if (fis != null) {
                    fis.close();
                }
            } catch (IOException ex) {
                System.out.println("Exception during file operation. Error:" +
ex.getMessage());
            }
        }

        return null;

    }

    // Test the custom implementation
    public static void main(String[] args) throws Exception {
        FileBasedCustomIdentityMapper idMapper = new
FileBasedCustomIdentityMapper();

        Properties properties = new Properties();
        properties.put("idmappinginfofile", "/export/IdentityMapper/
idmappinginfo.txt");
        idMapper.init(properties);

        String sourceValue = "hari@idmappersample.com";
        String mappedValue = idMapper.getMappedValue(sourceValue);

        System.out.println("Source value: [" + sourceValue + "]");
        System.out.println("Mapped value: [" + mappedValue + "]");
    }
```

```
}
```

Compiling the Sample Custom Identity Mapper

The custom identity mapper must be on a system that can be accessed by the GlassFish server. Place the JAR archive in a location outside of the Convergence installation or deployed directories.

To compile the sample custom identity mapper:

> **Note:** The paths used in this section may differ for your installation.

1. Create a sample directory for the source code, such as *sample_dir***/src**.
2. In *sample_dir***/src**, create an Java file named **FileBasedCustomIdentityMapper.java**.
3. Copy the core class from **FileBasedCustomIdentityMapper.java** into **FileBasedCustomIdentityMapper.java**. See "Developing Sample Custom Identity Mapper Data Files" for more information about the core class.
4. Compile the Java class files.

   ```
   cd sample_dir/src
   javac -classpath /opt/sun/comms/iwc/web-src/server/WEB-INF/lib/iwc.jar -d build
   src/FileBasedCustomIdentityMapper.java
   ```

5. Create a JAR.

   ```
   cd /sample_dir/src
   cd build
   jar -cvf ../FileBasedCustomIdentityMapper.jar *
   cd ..
   ```

   > **Note:** If your custom authentication module requires any additional JAR files or classes, they must be bundled along with the JAR file.

6. Add the JAR file to the deployed Convergence libraries with the GlassFish Server asadmin command.

   ```
   asadmin set applications.application.iwc.libraries=path/file.jar
   ```

   > **Note:** Depending on your version of GlassFish Server, the asadmin command may fail. If this happens, you can copy the JAR file into the Convergence library folder: *Convergence_domain***/applications/Convergence/WEB-INF/lib**.

Configuring the Sample Custom Identity Mapper

To configure the custom identity mapper with Convergence:

1. In **video.json**, verify that the service provider is configured and enabled.
2. Edit **video.properties** and add the service provider name from **video.json** for the serviceid parameter. For example:

   ```
   serviceid = wsc
   ```

3. Add the following parameters to **video.properties**:

```
wsc.identitymapper.enabled = true
wsc.identitymapper.idmappinginfofile = /export/sample/identitymapper/
idmappinginfo.txt
wsc.identitymapper.handler =
com.sun.comms.client.services.sample.identitymapper.FileBasedCustomIdentityMapp
er
```

4. Restart the GlassFish server.

Verifying the Custom Identity Mapper

You can verify the custom identity mapper by checking that the identity mapper look-up is performed on the local file as expected and is the returning correct values:

1. Create a few entries in **idmappinginfo.txt** with a Convergence user's email ID as key and the value as a sample SIP ID.
2. Use any SIP client which is connected to same SIP Proxy Server and Registrar IP that is used by Convergence.
3. From SIP client, call the user by using mapped ID, for example 123@idmappersamplesip.com. The call should establish successfully with the user hari@idmappersample.com.

Configuring Multinetwork Instant Messaging Add-On Services

This section describes how to configure multinetwork instant messaging (IM) add-on services in Convergence.

About Multinetwork IM Add-on Services in Convergence

You can configure multinetwork IM capability within Convergence. Currently, Convergence can be configured with Facebook Chat. In addition, you can connect with federated XMPP IM networks, which maintain an open directory that allows other IM networks to communicate with one another. See `http://xmpp.net/directory.php` for a list of public domains that allow XMPP S2S federation.

To configure multinetwork instant messaging in Convergence you must have Oracle Communications Instant Messaging Server release 9.0.1.4 or later.

Enabling Facebook Chat in Convergence

Enabling Facebook Chat in the Convergence UI allows you to chat with your Facebook friends. A Facebook user logs into his account in Convergence. Buddies are added to the Convergence Buddy List automatically. Facebook credentials are not saved in Convergence.

To enable Facebook Chat in the Convergence UI, you need to do the following:

1. Ensure that the Facebook gateway is configured in the Instant Messaging server. See your Instant Messenger documentation for more information.
2. Enable the Facebook Social add-on service. For more information, see "Configuring Social Add-On Services in Convergence".
3. Enable the multinetwork IM add-on service. Enable the *imgateways* add-on service for Facebook in the **imgateways.json** file; the **imgateways.json** file is in the **/var/opt/sun/comms/iwc/config** directory. To enable the *imgateways* add-on, set the *enabled* parameter at the top of the file to *true*. The following example shows a Facebook configuration:

```
{
    enabled: true,
    gateways: [
        {
                {
                        type: "facebook",
                        category: 'gateway',
                        enabled: true //Should be enabled only when Facebook
gateway in IM server is enabled
                }
    ]
}
```

4. Add *imgateways* to **addons.properties** located in the **/var/opt/sun/comms/iwc/config/** directory as in the following example:

```
addons=sms, imgateways, advertising, social
```

5. Restart GlassFish Server for your configuration changes to take effect.

Enabling Federated XMPP IM Networks in the Convergence UI

Enabling federated XMPP IM networks in the Convergence UI allows you to chat with your buddies on other XMPP IM networks. To enable federated XMPP IM networks in the Convergence UI, you need to do the following:

1. Ensure that XMPP Server Federation is configured in the Instant Messaging server. See your Instant Messenger documentation for more information.
2. Enable the multinetwork IM add-on service. Enable the *imgateways* add-on service for federated XMPP IM networks in the **imgateways.json** file; the **imgateways.json** file is in the **/var/opt/sun/comms/iwc/config** directory. To enable the *imgateways* add-on, set the *enabled* parameter at the top of the file to *true*. The following example shows a federated XMPP IM networks configuration:

```
{
  enabled: true,
  gateways: [
    {
      type: "groupone",
      category: 'federated',
      name: "Group One",
      domain: "groupone.com",
      serverurl: "example.groupone.com:5222",
      enabled: true
    },
  ]
}
```

   Change the *type*, *name*, *domain*, and *server_url* to the XMPP IM network service details.

3. Add *imgateways* to **addons.properties** located in the **/var/opt/sun/comms/iwc/config/** directory as in the following example:

```
addons=sms, imgateways, advertising, social
```

4. Restart the GlassFish server.
5. To display the IM icons of the XMPP federated service in the Convergence Instant Messaging window, you need to customize the Convergence UI. See *Convergence Customization Guide* for more information.

Configuring Convergence for SMS

You can configure Convergence to support either one-way or two-way SMS.

Configuring One-Way SMS for Convergence

This section describes how to configure one-way SMS so that users can send SMS messages that are 160 characters or less through the Convergence UI. In one-way SMS, senders are unable to receive SMS messages.

Configuring Messaging Server for One-Way SMS

To communicate with Short Message Service Centers (SMSCs), Messaging Server implements an MTA SMS channel which serves as an short message peer-to-peer (SMPP) client.

The following instructions describe how to configure Messaging Server for SMS, using either Messaging Server legacy or unified configuration. The two approaches are treated separately.

Configuring the SMS Add-on Service in the Convergence UI

To configure the SMS Add-On Service in so it displays in Convergence, enable SMS in the Convergence add-on services framework.

1. Enable the SMS add-on service and set parameters for it in the **sms.json** file. The **sms.json** file is in the in the **/var/opt/sun/comms/iwc/config/** directory. See the comments in the file for information on each parameter. The contents of the file at installation are:

```
{
    enabled: true,
    twowaysmsenabled:false,
    channel: "sms-handle",
    folder:"SMS",
    NDNFolder:'INBOX',
    numberhintenabled: true
}
```

   - Set **twowaysmsenabled** to **false**, so it enables one-way SMS.
   - The **channel** parameter requires the name of the MTA channel defined for SMS as part of configuring Messaging Server for SMS.
   - Do not change the default settings of the **folder** and **NDNFolder** parameters.

Restarting GlassFish Server

Use the **asadmin** command-line utility to restart the GlassFish server on which Convergence is deployed, so that your configuration changes can take effect. By default, *Convergence_Domain* is **domain1**.

```
asadmin stop-domain Convergence_Domain
asadmin start-domain Convergence_Domain
```

Configuring Two-Way SMS for Convergence

This section describes how to configure two-way SMS, where users can use the Convergence UI to send and receive SMS messages that are 160 characters or less.

Configuring Messaging Server for Two-Way SMS

To communicate with SMSCs, Messaging Server uses an MTA SMS channel which serves as an SMPP client. For two-way SMS, you also need to configure the SMS gateway server. The SMS gateway server is installed with Messaging Server.

The following instructions describe how to configure Messaging Server for SMS, by using either Messaging Server legacy or unified configuration. The two approaches are treated separately.

Configuring Instant Messaging Server for Two-Way SMS

You configure Instant Messaging Server for two-way SMS so that you receive pop-up notifications of SMS messages.

To configure the Instant Messaging Server for SMS:

1. Configure the Event Notification Service (ENS) that sends SMS notifications to the Instant Messaging server, which then forwards notifications on to the appropriate end user. Use the **imconfigutil** command-line utility to configure ENS:

   ```
   imconfutil add-component --config /opt/sun/comms/im/config/iim.conf.xml id=ens
   jid=kendo-ensjid password="xxx"
   ```

2. Using the **imadmin** command-line utility, stop and then restart the Instant Messaging server:

   ```
   imadmin stop
   imadmin start
   ```

Configuring GlassFish Server for Two-Way SMS

Convergence is deployed on GlassFish Server; before Convergence can be used for two-way SMS, GlassFish must be configured for two-way SMS.

To configure GlassFish Server for two-way SMS:

1. Create a Java Message Service (JMS) connection factory, and set connection-factory properties (see Oracle GlassFish Server administrator documentation for more information). The following example uses the **asadmin** command to create the connection factory (**jms/ConnectionFactoryMS**) and set properties. You can use the GlassFish Administration Console for the same purpose. The settings for **Username** and **Password** must be the same as the JMQ notification settings for **jmqnotify.jmqUser** and **jmqnotify.jmqpwd**).

   ```
   asadmin -p 5858 -u admin --passwordfile /var/tmp/aspass create-jms-resource
   --restype javax.jms.ConnectionFactory --description "a JMS connection factory"
   --property
   "AddressList=yyy.india.example.com\:7777:Username=user1:Password=xxx" jms/
   ConnectionFactoryMS"
   ```

2. Specify a destination for connections by creating a JMS destination resource and setting properties for it, as in the example that follows.

   ```
   asadmin -p 5858 -u admin --passwordfile /var/tmp/aspass create-jms-resource
   --restype javax.jms.Queue --property "Name=ucsms1:Description=ucs ms desc"
   ucsms1"
   ```

   In this example, **javax.jms.Queue** was set, because the earlier example of setting JMQ notification parameters the JMQ notification destination type was **queue**. If the destination type had been **topic**, then **javax.jms.Topic** would have been set.

Configuring ENS Support for Convergence

To configure ENS support for Convergence:

1. Edit **/var/opt/sun/comms/iwc/config/httpbind.conf** and set the following ENS parameters:
   - ens.server_url: the ENS Server URL
   - ens.component_jid=kendo-ensjid: the ENS JID
   - ens.component_password: the encrypted ENS password

   The following are examples of the settings:

   ```
   ens.server_url=IM_HOST:5269
   ens.component_jid=kendo-ensjid
   ens.component_password=rE9ZIq6H0r49RgsQrKHXsw==
   ```

2. Set ENS and notification-related parameters by using the **iwcadmin** command:

   ```
   iwcadmin -o notify.service.enable -v true
   iwcadmin -o notify.mq.[serv1].enable -v true
   iwcadmin -o notify.mq.[serv1].connection -v jms/ConnectionFactoryMS
   iwcadmin -o notify.mq.[serv1].destinationname v ucsms1
   iwcadmin -o notify.mq.[serv1].destinationtype -v QUEUE
   iwcadmin -o notify.mq.[serv1].resourcetype -v consumer
   iwcadmin -o notify.mq.[serv1].filter -v "JMSType='NewMsg' AND msgflags LIKE
   'sms%'"
   iwcadmin -o ens.service.enable -v true
   iwcadmin -o ens.[mail].enable -v true
   iwcadmin -o ens.[mail].datasource -v ucsms1
   ```

   Alternatively, you can create a file that contains the notification-related parameters and run the following command:

   ```
   iwcadmin -f ens_iwc_settings_file
   ```

   Contents of the *ens_iwc_settings_file* can be in the following format:

   ```
   notify.service.enable = true
   notify.mq.[serv1].enable = true
   notify.mq.[serv1].connection = jms/ConnectionFactoryMS
   notify.mq.[serv1].destinationname = ucsms1
   notify.mq.[serv1].destinationtype = QUEUE
   notify.mq.[serv1].resourcetype = consumer
   notify.mq.[serv1].filter="JMSType='NewMsg' AND msgflags LIKE 'sms%'"
   ens.service.enable = true
   ens.[mail].enable = true
   ens.[mail].datasource = ucsms1
   ```

Configuring the SMS Add-on Service in the Convergence UI

To configure the SMS Add-On Service in so it displays in Convergence, enable SMS in the Convergence add-on services framework.

Edit **sms.json** in **/var/opt/sun/comms/iwc/config/** to enable the SMS add-on and set required parameters, as in the following example, where *channel* must be set to the SMS channel name assigned in Messaging Server configuration and folder must be set to SMS. See the comments in the file for information on each parameter. The contents of the file at installation are:

```
{
    enabled: true,
```

```
    twowaysmsenabled:true,
    channel: "sms-handle",
    folder:"SMS",
    NDNFolder:'INBOX',
    numberhintenabled: true
}
```

Note: The *folder* parameter can be changed, however, it has to match the folder value specified in the Messaging Server configuration, specifically the *fileinto* folder specified in the *sms.filter*.

Restarting GlassFish

Restart the GlassFish server on which Convergence is deployed, so that your configuration changes can take effect. You will need to specify the GlassFish domain in which Convergence is deployed. In the following example, the domain is *domain1*:

```
/root/glassfish3/bin/asadmin stop-domain domain1
/root/glassfish3/bin/asadmin start-domain domain1
```

Configuring Social Add-On Services in Convergence

This section describes how to configure the social add-on service.

Facebook integration with Convergence is not currently working due to changes Facebook made to their APIs, interfaces, and application registration approval. Oracle is exploring potential solutions and will provide an update when a solution is available.

Note: Third-party parameters described in this configuration documentation might change as new versions of Facebook, Twitter, and Flickr are released. Always refer to the third-party developer documentation for the most up-to-date information on third-party parameters.

About Social Add-on Services in Convergence

The *social* add-on service provides access to Facebook, Twitter, and access to Flickr's public photos through the **Social** tab in Convergence. To integrate the social add-on services in the **Social** tab, you enable and configure each service separately.

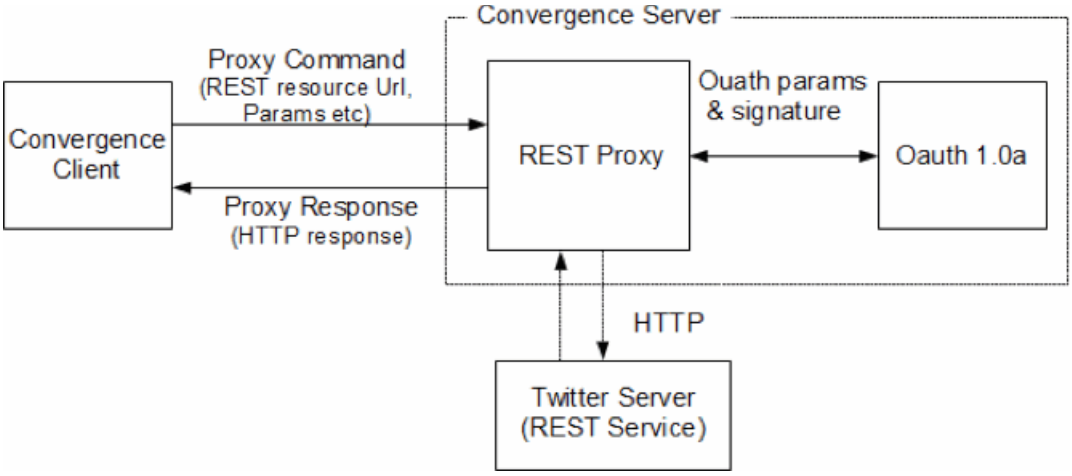
Twitter authentication and access is performed through the Twitter REST API. See the Twitter developer web site for more information about their API:

https://dev.twitter.com/docs/api

Because of cross-domain access restriction in web browsers, the REST calls are tunneled through Convergence Server.

Figure 13–5 shows the Twitter integration architecture.

Figure 13–5 Twitter Integration Architecture



For Facebook and Twitter, when a user performs a login, it is considered the session for access.

Enabling Add-On Services Through the Convergence Social Tab

To enable the *social* add-on service and make the individual services available through the **Social** tab, follow these steps:

1. Make sure that the social add-on is enabled in the **add-ons.properties** file; by default, the social add-on is enabled. See "add-ons.properties" for more information. The **add-ons.properties** file is located in the **/var/opt/sun/comms/iwc/config/** directory. To enable the *social* add-on, if it is not currently enabled, add it as a value of the *addons* parameter, as in the following example:

   ```
   addons=sms, imgateways, advertising, social
   ```

2. Enable the *social* add-on service in the **social.json** file; by default, the add-on is not enabled. The **social.json** file is in the **/var/opt/sun/comms/iwc/config** directory. To enable the *social* add-on, set the *enabled* parameter at the top of the file to *true*, as in the following example:

   ```
   {
       enabled:true,
       service:[
           {
                           ...
   ```

3. To access Twitter in Convergence through the **Social** tab:

   a. Create a Twitter app and specify a minimum of "read and write" access, so that the Convergence user is able to send a tweet through the Convergence UI. In addition, the *OAuth* access token should be created by clicking the "Create my access token" button in the app to allow logging into Twitter from Convergence. For information on creating the Twitter app ID and on obtaining the *OAuth*-related configuration parameters for your configuration, see the Twitter developer documentation.

   b. Set parameters for Twitter in **social.json**. The parameters that need to be set for Twitter, as listed in the installed **social.json** file, are:

      ```
      id:'twitter',
      enabled:true,
      ```

```
param: {
    tokens:{
        storetoken:'$storeToken-twitter',
        defaultStoreToken:true
```

c. The parameters that need to be set for Twitter, as listed in the installed **social.properties** file, are:

```
serviceid = twitter

# OAuth related configuration
twitter.oauth.consumer.key=consumer_key
twitter.oauth.consumer.secret=consumer_secret
twitter.oauth.authorize.url=https://api.twitter.com/oauth/authorize
twitter.oauth.request.token.verb=POST
twitter.oauth.request.token.url=https://api.twitter.com/oauth/request_token
twitter.oauth.access.token.verb=POST
twitter.oauth.access.token.url=https://api.twitter.com/oauth/access_token
twitter.oauth.callback.url=http://host_name:port/iwc/oauth/consumer/
callback/twitter
twitter.oauth.version=1.0a

# HTTP proxy related configuration
twitter.http.enablessl=true
twitter.http.host=api.twitter.com
twitter.http.port=443
twitter.http.socket-timeout=180
twitter.http.connection-manager-timeout=240
twitter.http.max-connections=100
```

The only *OAuth*-related configuration parameters you need to enter or change are:

- *twitter.oauth.consumer.key* Enter the consumer key.
- *twitter.oauth.consumer.secret* Enter the consumer secret.
- *twitter.oauth.callback.url* Enter a callback URL for your implementation.

d. Set HTTP-proxy related parameters as appropriate for your site:

```
/opt/glassfish3/bin/asadmin create-jvm-options
"-Dhttp.proxyHost=proxy_host" -p admin_port
/opt/glassfish3/bin/asadmin create-jvm-options
"-Dhttp.proxyPort=proxy_port" -p admin_port
```

4. To access Facebook through the **Social** tab, set the parameters for Facebook in **social.json**.

a. Create and add a Facebook app for your environment. Point the Facebook app's Canvas or website URL to the Convergence URL.

From the **apps** menu, select Status & Review and answer "yes" to "Do you want to make this app and all its live features available to the general public?" Enabling this setting allows Facebook to work within the Convergence UI. Refer to Facebook developer documentation for more information.

b. The parameters that you need to set for Facebook, as listed in the installed **social.json** file, are:

```
id:'facebook',
enabled:true,
param: {
```

```
key:'',
tokens:{
    accesstoken:'$accessToken-facebook',
    storetoken:'$storeToken-facebook',
    defaultStoreToken:true
```

For *key*, enter the AppID.

5. To access Flickr through the **Social** tab, set the parameters for Flickr in **social.json**:

   a. To access Flickr through the **Social** tab, obtain your Flickr API key. For more information, see the flickr web site:

   http://www.flickr.com/services/api/misc.api_keys.html.

   b. The parameters that you need to set for Flickr, as listed in the installed **social.json** file, are:

```
{
     id:'flickr',
     enabled:true,
     key:''
}
```

   Make sure Flickr is enabled (**enabled:true**) and that you insert your Flickr API key in the *key* field.

Configuring the Advertising Add-On Service in Convergence

This section describes how to configure the advertising add-on service in Convergence.

About the Advertising Add-On Service

The advertising add-on service makes it possible to display banner ads, text ads, and contextual ads in the Convergence UI. A system administrator can determine the events that trigger new ads and the location within the Convergence UI at which ads are displayed.

Ads can be displayed in:

- A *skyscraper* panel that appears on the right side of the Convergence UI. See "Displaying Ads in a Skyscraper Panel" for more information.
- An *ad* box, a box containing an add that is located within the email-message viewing area and can be positioned above or below email messages or to the right or left of email messages. See "Displaying Ads in an Ad Box" for more information.

Table 13–2 lists the advertising add-on and configuration files.

Table 13–2 Advertising Configuration and Add-on Files

File Name	Directory	Description
add-ons.properties	**var/opt/sun/comms/iwc/config/**	Add-ons are added to this file to enable specific services.
advertising.json	**var/opt/sun/comms/iwc/config/**	Provides file path to plug-in file and allows enabling of Skyscraper and Message Box ad placement, height of ads, and other characteristics in the display area
Plugin.js	*c11n_Home*/**allDomain/js/widget/advertising**	Sample configuration on how to create and configure ads. File can be renamed. Provides call back methods for each type of ad (Skyscraper and Ad Box). You can fill in each callback with code to retrieve ad images, assign them to the innerHTML of the supplied object and return.
Skyscraper.js and **Skyscraper.html**	*c11n_Home*/**allDomain/js/widget/advertising/** and *c11n_Home*/**allDomain/js/widget/advertising/templates/**	Sample configuration that's specific to Skyscraper ads. Provides examples on how to receive events from Convergence, what actions can be taken, controlling splitters, and mechanisms for displaying ads. These configuration files are specific to Skyscraper files and cannot be used in combination with Ad box ads.
Sample Ad Images	**iwc_static/layout/images/ads**	Sample ad images

Configuring Advertising for Convergence

See the **plugin.js**, **Skyscraper.js**, and **Skyscraper.html** samples to create and configure advertising for the Convergence UI.

Enabling the Advertising Add-On Service

To enable the advertising add-on:

1. Make sure that the advertising add-on is enabled in the **add-ons.properties** file; by default, the advertising add-on is enabled. See "add-ons.properties" for more information. The **add-ons.properties** file is located in the **/var/opt/sun/comms/iwc/config/** directory. To enable the advertising add-on, if it is not currently enabled, add it as a value of the *addons* parameter, as in the following example:

   ```
   addons=advertising
   ```

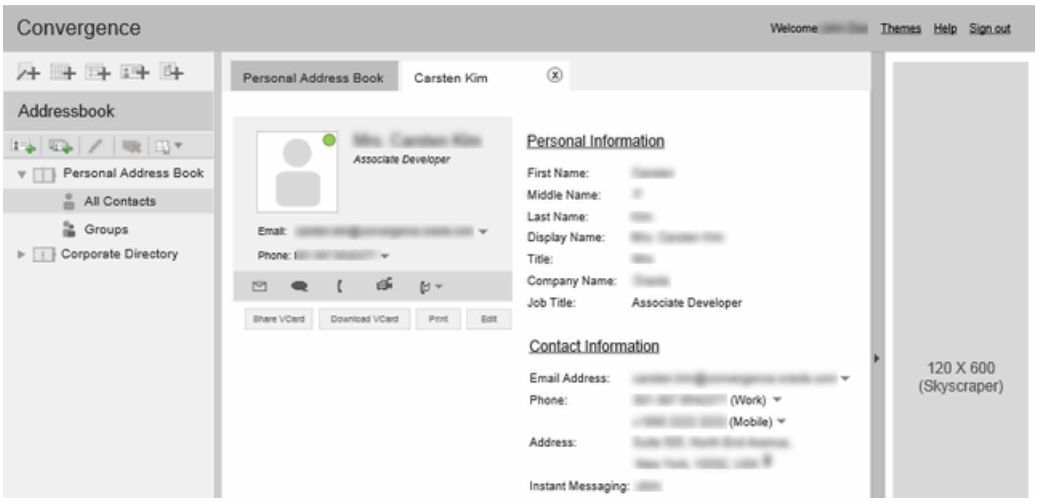
2. Enable the *advertising* add-on service in the **advertising.json** file; by default, the add-on is not enabled. The **advertising.json** file is in the **/var/opt/sun/comms/iwc/config** directory. To enable the *advertising* add-on, set the **enabled** parameter at the top of the file to **true**.
3. Verify that the *c11n_Home* directory exists. If it does not, create it by copying the *c11n_sample* directory. See *Convergence Customization Guide* for more information.
4. Enable the Convergence Server for customization. Use the **iwcadmin** command to set the **client.enablecustomization** parameter to **true**. For example:

   ```
   iwcadmin -o client.enablecustomization -v true
   ```

Displaying Ads in a Skyscraper Panel

Skyscraper panels are displayed on the right side of the Convergence UI, as in the following example:

Figure 13–6 Upper Portion of the Skyscraper Panel in Convergence



To configure an ad to display in a skyscraper panel, you edit the **plugin.js** file. To configure the characteristics of skyscraper panels, you set parameters in the **advertising.json** file.

Parameters for Configuring Skyscraper Panels in the advertising.json File

The **advertising.json** file contains the following parameters for configuring skyscraper panels:

- *enabled*: If set to **true** (the default), a skyscraper panel is added to the right side of the Convergence UI.
- *width*: The width, in pixels, of the skyscraper panel. By default, *width* is set to 160 pixels.
- *closeEnable*: If set to **true** (the default), the user can close the skyscraper panel by clicking a bar tab containing an arrow that appears. Once a user closes the panel, the panel is not displayed again until the user refreshes the Web page, opens Convergence in a new window, tab, or browser, or logs in again.
- *events*: Event parameters:
  - *enabled*: If set to **true**, ads can be displayed for specific events: adtime, mail, or calendar actions.
  - *adtime*: The duration of an ad, in seconds. The default is 30 seconds.
  - *mail*: An email action, such as opening a new mail tab or clicking through the email message grid can cause a refresh that replaces the current ad with a new ad. you cannot configure which mail actions trigger an ad refresh. You can only determine the frequency of ads, what ad to display, or which events to receive from Convergence (mail or calendar or both).
  - *calendar*: User actions involving the calendar can trigger an ad refresh. The calendar actions that can trigger a refresh are configured in the *skyscraper.js* file. Calendar events are similar to mail events in that you cannot configure which calendar actions trigger an ad refresh. You can only determine the

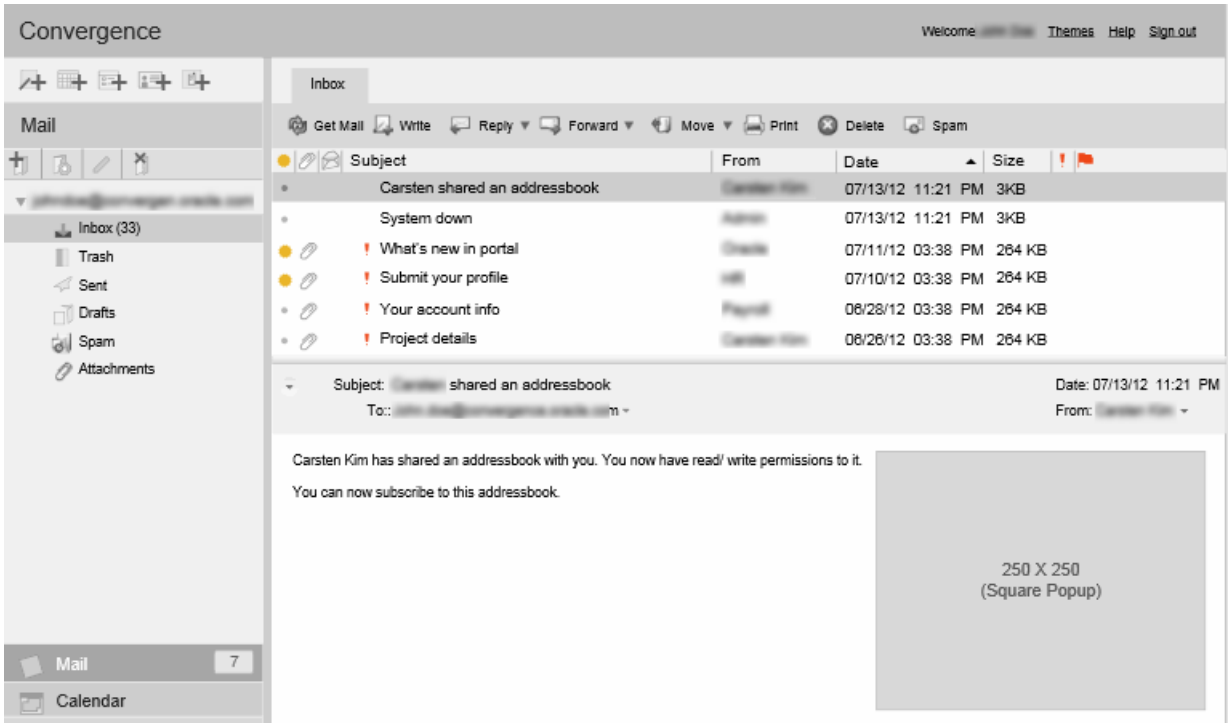
frequency of ads, what ad to display, or which events to receive from Convergence (mail or calendar or both).

– *all*: If set to **true**, any event within Convergence can be configured in **advertising.json** to trigger an ad refresh. By default, the *all* parameter is set to **false.**

Displaying Ads in an Ad Box

Boxes containing ads can be displayed above or below an email message, or to the left or right of an email message.

Figure 13–7 Message Ad Box in Convergence



To configure an ad to display in an ad box, you edit the **plugin.js** file. To configure the characteristics of ad boxes, you set parameters in the **advertising.json** file.

The **advertising.json** file contains the following parameters for configuring ad boxes:

- *enabled*: If set to **true** (the default), enables ads in ad boxes.
- *TopAd*: A banner ad displayed above an email message. Parameters:
  - *enabled*: If set to **true** (the default), *TopAd* banner ads are enabled.
  - *height*: The height of the banner ad, in pixels. The default is **60.**
- *RightAd*: An ad box displayed to the right of an email message. Parameters:
  - *enabled*: If set to **true** (the default), enables ad boxes to the right of the message area.
  - *width*: The width of the *RightAd* ad box, in pixels. The default is 250.
  - *height*: The height of the *RightAd* ad box, in pixels. The default is 250.
- *LeftAd*: An ad box displayed to the left of the message area. Parameters:

  - *enabled*: If set to **true**, enables ad boxes to the right of the message area. The default is **false.**
  - *width*: The width of the *LeftAd* ad box,in pixels. The default is 250.
  - *height*: The height of the *LeftAd* ad box,in pixels. The default is 250.
- *BottomAd*: A banner ad displayed below an email message. Parameters:
  - *enabled*: If set to **true** (the default), *BottomAd* banner ads are enabled.
  - *height*: The height of the banner ad, in pixels. The default is **60.**

14

Tuning GlassFish Server to Enhance Convergence Performance

Oracle Communications Convergence is a Java application bundled into a WAR file that runs inside an Oracle GlassFish Server web container. This chapter describes how to optimize the GlassFish server environment to allow Convergence to deliver the best possible performance.

For general GlassFish Server performance tuning, see *Oracle GlassFish Server Performance Tuning Guide*.

Convergence Performance Tuning Overview

Advances in storage, servers, and Java affect how one tunes web containers for middleware. There are systems with multi-threaded chips having 32 effective processors, operating systems with virtualized containers like Solaris zones, and file systems like ZFS that can spread files out over many disks. Java can automatically adjust itself based on dynamic conditions. The tuning options available are many, and you must choose what works for you.

The tuning guidance presented here offers options to examine and configure. However, these options do not address specific hardware configurations and are not guaranteed to improve performance for any particular hardware configuration, performance load, or type of load on your system.

Try out the options and tips that apply to your deployment, test their impact on performance, and tweak the option values as needed.

Use GlassFish Server's administration browser interface or command-line interface rather than directly editing the **domain.xml** file to make changes. The changes do not take effect until the domain instance has been restarted.

Make the modifications to the GlassFish server **domain/config/port** in which Convergence is running.

Tuning GlassFish Server Configuration Parameters

Table 14–1 lists the tested tuning parameters for GlassFish Server for Convergence. These tested parameters were set assuming 10,000 users with a deployment that included mail, calendar, and address book services. There are differences between tuning Convergence with IM enabled and with IM disabled.

Table 14–1 GlassFish Server Tuning Parameters

Parameter	Value (if IM Disabled)	Value (if IM Enabled)
server-config.network-config.protocols.protocol.http-listener-1.http.header-buffer-length-bytes	default	16384
server-config.network-config.protocols.protocol.http-listener-1.http.timeout-seconds	16	20
server-config.network-config.protocols.protocol.http-listener-1.http.file-cache.enabled	default	true
server-config.network-config.transports.transport.tcp.acceptor-threads	32 (64 core system)	-1
server-config.network-config.transports.transport.tcp.idle-key-timeout-seconds	default	300
server-config.thread-pools.thread-pool.http-thread-pool.min-thread-pool-size	32	8
server-config.thread-pools.thread-pool.http-thread-pool.max-thread-pool-size	256	512

Tip: For the request threads run HTTP requests, you want just enough: enough to keep the machine busy, but not so many that they compete for CPU resources – if they compete for CPU resources, then your throughput will suffer greatly. Too many request processing threads is often a big performance problem.

Determining how much is just enough depends – in a case where HTTP requests don't use any external resource and are hence CPU bound, you want only as many HTTP request processing threads as you have CPUs on the machine. But if the HTTP request makes a database call (even indirectly, like by using a JPA entity), the request will block while waiting for the database, and you could profitably run another thread. So this takes some trial and error, but start with the same number of threads as you have CPU and increase them until you no longer see an improvement in throughput.

Since Convergence communicates extensively with back-end messaging and calendar resources, request blocking could be an issue. You will need to monitor your own deployment and adjust the Thread Count accordingly.

Tuning Parameters for the HTTP Listener

To configure this setting, select Acceptor Threads from the **Configurations** menu (Configurations/server-config/Network Config/Transports/tcp/Acceptor Threads) in the GlassFish Server Administration Console. Using asadmin, change server-config.network-config.transports.transport.tcp.acceptor-thread parameter.

Increase the HTTP listener acceptor-threads. The default value is: acceptor-threads="1".

In the HTTP Service section of the GlassFish Server Administration Console, on the listener for the port for Convergence (such as 8080):

Start with a value of 2 and monitor the performance.

To configure this setting, select the Listener1* from the Configuration menu (Configuration/HTTP Services/HTTP Listeners/) in the GlassFish Server Administration Console:

(http-listener-1 is assumed to be in use for Convergence.)

Take these steps:

- Increase the acceptor-threads value to the number of CPUs on the system.
- If you have only one interface (NIC), change the default 0.0.0.0 IP address to your IP for the host.

Configuring GlassFish Server to Compress Client Files

You can improve server response times by reducing the size of the HTTP response. If you choose to implement this practice, realize that the server does more work to compress files which might impact the server's scalability under heavy loads.

To compress files sent to the client by using the GlassFish Server Administration Console:

1. Select the Compression Minimum Size from the (Configurations/server-config/ Network Config/Protocols/http-listener-1) menu in the GlassFish Server Administration Console.
2. Using asadmin, set the server-config.network-config.protocols.protocol.http-listener-1.http.compression-min-size-bytes parameter to 2000"

   (http-listener-1 is assumed to be in use for Convergence.)

Enhancing Browser Caching of Static Files

When this feature is implemented, GlassFish Server includes the Expires header in the HTTP response. The Expires header allows files cached in the browser to remain in cache for the time specified in the ExpiresFilter.class file.

To enable Expires headers:

1. At a command prompt, change directory to *Convergence_Domain*.
2. In Convergence_Domain/config/ edit the **default-web.xml** file.
3. Add the following filter rule directly below the existing Servlet Mappings rules:

```
<!-- Enable Expires Headers for Convergence files -->
  <filter>
    <filter-name>ExpiresFilter</filter-name>
    <filter-class>iwc.ExpiresFilter</filter-class>
  </filter>
  <filter-mapping>
      <filter-name>ExpiresFilter</filter-name>
      <url-pattern>/iwc_static/js/*</url-pattern>
      <url-pattern>/iwc_static/layout/*</url-pattern>
      <dispatcher>REQUEST</dispatcher>
      <dispatcher>FORWARD</dispatcher>
  </filter-mapping>
```

4. In *Convergence_Domain***/lib/classes**, create a new directory called **iwc**.
5. Copy the Oracle Communications Messaging Server class file **ExpiresFilter.class** into the **iwc** directory.

6. Restart the GlassFish server.

Tuning JVM Options

This section describes the JVM tuning options.

Activating the Garbage Collection Log

This log has negligible impact on server performance and provides valuable debugging and performance history data.

Add the following entry, using your own path. For example:

```
<jvm_options>-Xloggc:/opt/SUNWappserver/domains/domain1/logs/gclog</jvm-options>
```

This log is overwritten each time the server is restarted.

Invoking the Java HotSpot Server VM

Make sure that the JVM options in the **domain.xml** file for the GlassFish Server instance specify -server, not -client:

```
<jvm-options>-server</jvm-options>
```

Server Class machines are defined as having at least 2 CPUs and 2 GB of memory.

Remove the -client option if present and add the -server option. You can verify what mode the server actually started with by running:

```
grep 'HotSpot' server.log"*,
```

This will show either ...Client VM... or ...Server VM.....

To configure and activate a 64-bit JVM:

1. On Solaris, you can verify that the operating system kernel is running in 64-bit mode by running:

   /usr/bin/isainfo -kv

2. If needed, download and install the 64-bit jvm files on the JVM instance used by the GlassFish Server on the machine. Verify the 64-bit files are available by running:

   "/*server_java_dir*/java -d64 -version"

3. On the GlassFish Server, replace the JVM option, -server (or -client), with -d64

> **Note:** GlassFish Enterprise Server is not supported on 64-bit JVMs on Red Hat Linux. It is recommended that you run the latest version of the JDK with Convergence.

Tuning the JVM Heap Size

In the GlassFish Server Administration Console, under Configurations, select server-config >> JVM Settings Tab >> JVM Options Sub-Tab >> Add/Modify the options...

The min and max heap size options are: -XmsNNNNm and -XmxNNNNm.

Generally, set max heap as large as possible given the available memory on your machine. (Setting the min equal to the max improves JVM efficiency.) Total memory used is equal to the (JVM native heap space) + (Java Heap) + (Permanent Generation space). Leave room for the operating system and any other applications running on the machine too. Don't forget to reserve memory for the OS and avoid memory swapping at all costs.

For example, you could set the heap size options to

```
<jvm-options>-Xms2048m -Xmx2048m</jvm-options>
```

Setting Garbage Collection Algorithms

To increase the stability and predictability of the heap size and the ratios of its configuration, you can explicitly set the following parameters:

- -XX:+UseParallelGC: This parameter is used by default on a machine qualifying as Server Class. This default collector is sufficient.
- -XX:+UseParallelOldGC: This statement makes the tenured generation run GC in parallel, too. This is the default in JDK 6. In jdk1.5_u6 and greater you need to explicitly specify this option.
- -XX:-UseAdaptiveSizePolicy: Turn off GC ergonomics. Note the minus sign in this statement. Specify min and max values explicitly.
- -XX:NewRatio=1: Optimize the Young Generation Size. Using a ratio (as opposed to setting a numerical size with NewSize) allows for the maximum possible young generation size relative to the overall heap, no matter your MaxHeap size.

Tests show that most of the objects created for Convergence are short-lived, thus benefiting from a larger young generation size.

The NewRatio means {New:Old}. So, when NewRatio=1, then new:old = 1:1. Therefore, the young generation size = 1/2 of the total Java heap. The young generation size can never be larger than half the overall heap because - in the worst case - all the young generation space could be promoted to the old generation. Therefore, the old generation must be at least as large as the young generation size.

For more information about the NewRatio option, see the following Oracle web site:

http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html

Monitor your own heap usage with JConsole. See "Monitoring Convergence" for more information.

Setting the Permanent Generation Size

Be aware that MaxPermSize may need to be increased. JVM Efficiency is improved by setting PermSize equal to MaxPermSize. Start with the default, observe PermSpace usage and adjust accordingly:

<jvm-options>-XX:PermSize=192m -XX:MaxPermSize=192m</jvm-options>

Use a tool such as Jconsole or VisualVM to determine how best to optimize your own system.

Tuning the JVM RMI GC Interval Parameters

It is better if full Garbage Collections (GCs) on the Java heap do not occur frequently and are not called explicitly. It is best to let the JVM decide when to do full garbage collections.

Unfortunately, the GlassFish Server has a couple of JVM options for RMI applications that invoke full GCs often. If you are not running any applications using RMI, you should increase the rmi.dgc... values, or configure them never to occur.

You should also consider the ramifications of disabling explicit GCs. When another application is connecting to GlassFish Server with RMI, memory for objects in the Server heap will not be released and the calling application will not be able to release the reference to that object, thus possibly causing memory overflow on the other application.

These intervals are increased to 10 hours:

```
<jvm-options>-Dsun.rmi.dgc.server.gcInterval=36000000</jvm-options>
<jvm-options>-Dsun.rmi.dgc.client.gcInterval=36000000</jvm-options>
```

You can also use either of the following two options to prevent the full GCs invoked for RMI:

- Disable explicit GC by adding:

  ```
  <jvm-options>-XX:+DisableExplicitGC</jvm-options>
  ```

- Use JVM and set:

  ```
  -XX:+UseConcMarkSweepGC
  -XX:+ExplicitGCInvokesConcurrent
  ```

ExplicitGCInvokesConcurrent is available beginning with JVM 1.6.

Sample List of JVM Options

The following list is a sample section of the **domain.xml** file's JVM options:

```
<jvm-options>-server</jvm-options>
<jvm-options>-XX:+DisableExplicitGC</jvm-options>
<jvm-options>-XX:+UseParallelGC</jvm-options>
<jvm-options>-XX:+UseParallelOldGC</jvm-options>
<jvm-options>-XX:-UseAdaptiveSizePolicy</jvm-options>
<jvm-options>-Xms1024M -Xmx1024M</jvm-options>
<jvm-options>-XX:NewRatio=1</jvm-options>
<jvm-options>-XX:PermSize=192M</jvm-options>
<jvm-options>-XX:MaxPermSize=192M</jvm-options>
<jvm-options>-Xloggc:/opt/SUNWappserver/domains/domain1/logs/gclog</jvm-options>
```

Miscellaneous Performance Tuning Tips

- Class Data Sharing

  Class data sharing (CDS) is a new feature in J2SE 5.0. CDS applies only when the "Java HotSpot Client VM" is used. Since we recommend using the "Java HotSpot Server VM," this feature does not apply.

- Inspect Settings

  Inspect your settings with the following commands. To see all Java processes running on your machine:

```
jps -mlvV
```

To view your settings in effect for the JVM for the GlassFish Server:

```
jmap -heap java_process_id
```

- Monitoring the JVM

  JConsole is a built-in JVM monitoring tool. On the SUT, set the display variable to your local machine and run the following command: jconsole

  See the Jconsole documentation for more information.

- UseConcMarkSweepGC

  The intrepid system administrator may want to consider using **UseConcMarkSweepGC** instead of **UseParallelGC**. See the Java SE VM documentation at the following Oracle web site for more information:

  http://www.oracle.com/technetwork/java/javase/gc-tuning-6-140523.html

- GC 1 Algorithm

  See the discussion about Java garbage collection settings on the Oracle Technology Network:

  http://www.oracle.com/technetwork/java/javase/tech/g1-intro-jsp-135488.html

15

Setting Up Multiple Corporate Directories

You can configure Convergence to use multiple corporate directories, or configure Convergence to use a directory server other than the user group directory server.

Adding a Corporate Directory

To add a corporate directory or to use the directory server other than the user group directory server, set the following configuration parameters:

- **ab.corpdir.[***identifier***].ldaphost**
- **ab.corpdir.[***identifier***].ldapport**
- **ab.corpdir.[***identifier***].ldapbinddn**
- **ab.corpdir.[***identifier***].ldapbindcred**

The following example has the configuration parameters settings:

```
iwcadmin -o ab.corpdir.[default].ldaphost -v host.example.com
iwcadmin -o ab.corpdir.[default].ldapport -v 400
iwcadmin -o ab.corpdir.[default].ldapbinddn -v "cn=Directory Manager"
iwcadmin -o ab.corpdir.[default].ldapbindcred -v xyzxyz
```

The corporate directory can be configured with multiple directory servers. In this example *default* is used to identify corporate directory configuration for *host.example.com*. For a single corporate directory configuration, you must use *default* as the identifier.

Configuring Multiple Corporate Directories

1. To configure multiple corporate address books, set following parameters:

```
ab.corpdir.[identifier].ldaphost
ab.corpdir.[identifier].ldapport
ab.corpdir.[identifier].ldapbinddn
ab.corpdir.[identifier].ldapbindcred
ab.corpdir.[identifier].urlmatch
ab.corpdir.[identifier].searchattr
ab.corpdir.[identifier].displayname
```

Note: The value for the *urlmatch* configuration parameter must be unique.

- To search from Root: *ldap://corp-directory1*
- To search from *dn ou=people,o=ab.org*: *ldap://somehost:390/ou=people,o=ab.org*

Format for *urlmatch* is **ldap://***unique_value* or **ldap://***host***:***port***/DN**. For example:

```
-o ab.corpdir.[corpdir1].ldaphost -v budgie.india.example.com
-o ab.corpdir.[corpdir1].ldapport -v 389
-o ab.corpdir.[corpdir1].ldapbinddn -v "cn=Directory Manager"
-o ab.corpdir.[corpdir1].ldapbindcred -v netscape
-o ab.corpdir.[corpdir1].urlmatch -v ldap://corpdir1
-o ab.corpdir.[corpdir1].searchattr -v entry/displayname,@uid
-o ab.corpdir.[corpdir1].lookthrulimit -v 3000
-o ab.corpdir.[corpdir1].displayname -v "Second Corporate Book"
```

2. Restart the GlassFish server.

Note: In some cases, the corporate directories might not display. The workaround is to set the *urlmatch* configuration parameter, beginning with the default URL match value (*ldap://corpdirectory*). For example, for an organization adding multiple address books from three different entities: *CommerceDept*, *IntlTradeDiv*, and *DivofEmployment*, the *urlmatch* is set to the following:

```
ab.corpdir.[CommerceDept].urlmatch = ldap://corpdirectorycommerce
\\
/ou=People,ou=CommerceDepartment,o=cat.example.gov,dc=divemp,dc=gov
ab.corpdir.[IntlTradeDiv].urlmatch = ldap://corpdirectoryitd  \\
/
ou=People,ou=ITD,ou=CommerceDepartment,o=cat.example.gov,dc=divemp,
dc=gov
ab.corpdir.[DivofEmployment].urlmatch = ldap://corpdirectorydivemp
\\
/
ou=People,ou=DivofEmployment,ou=CommerceDept,o=cat.example.gov,dc=d
ivemp,dc=gov
```

Even though the Corporate Directories are properly set up and work as designed, they may display errors in the *iwc.log* or the Firebug log.

Disabling Corporate Directory (Newly Added or Default)

To disable a corporate directory, set the **ab.corpdir.[***identifier***].enable** parameter to **false**.

A

ExpiresFilter.java Reference

This appendix shows the contents of the **ExpiresFilter.java** file.

```
package iwc;

import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.TimeZone;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletResponse;

/**
 * The expires filter adds the expires HTTP header based on the deployment policy.
 * Many sites have a fixed deployment schedule where deployments take place
 * based on timed regular intervals. This filter adds the expires header of the
 * next possible deployment time, to support browser caching.
 * @author Chris Webster
 */
public class ExpiresFilter implements Filter {

        private FilterConfig filterConfig;
        private String expires;
        private long nextDeploymentTime;

        public ExpiresFilter() {
                expires = nextDeploymentTime();
        }

        private String nextDeploymentTime() {
                // assume next deployment is M-F at 09:45
                Calendar c = Calendar.getInstance();

                int dayOffset = 1;

                if (c.get(Calendar.DAY_OF_WEEK) == Calendar.FRIDAY) {
                        dayOffset+=2;
                }

                if (c.get(Calendar.DAY_OF_WEEK) == Calendar.SATURDAY) {
                        dayOffset++;
```

```
                }

                c.add(Calendar.DAY_OF_MONTH, dayOffset);
                c.set(c.get(Calendar.YEAR)+2, c.get(Calendar.MONTH),
                      c.get(Calendar.DAY_OF_MONTH), 9, 45);

                nextDeploymentTime = c.getTimeInMillis();

                String pattern = "EEE, dd MMM yyyy HH:mm:ss z";
                SimpleDateFormat sdf = new SimpleDateFormat(pattern);
                sdf.setTimeZone(TimeZone.getTimeZone("GMT"));
                return sdf.format(c.getTime());
        }

        private void addCacheHeaders(ServletRequest request, ServletResponse
response)
                throws IOException, ServletException {

                HttpServletResponse sr = (HttpServletResponse) response;
                sr.setHeader("Expires", expires);
                long now = (new Date()).getTime();

                long expireTime = nextDeploymentTime - now;
                expireTime /= 1000;
                sr.setHeader("Cache-Control", "max-age="+
                        Long.toString(expireTime)+";public;must-revalidate;");
        }

        /**
         *
         * @param request The servlet request we are processing
         * @param response The servlet response we are creating
         * @param chain The filter chain we are processing
         *
         * @exception IOException if an input/output error occurs
         * @exception ServletException if a servlet error occurs
         */
        public void doFilter(ServletRequest request, ServletResponse response,
                                                FilterChain chain)
                throws IOException, ServletException {

                addCacheHeaders(request, response);
                chain.doFilter(request, response);
        }

        /**
         * Return the filter configuration object for this filter.
         */
        private FilterConfig getFilterConfig() {
                return filterConfig;
        }

        /**
         * Set the filter configuration object for this filter.
         *
         * @param filterConfig The filter configuration object
         */
        private void setFilterConfig(FilterConfig filterConfig) {
                this.filterConfig = filterConfig;
        }
```

```
        /**
         * Destroy method for this filter
         *
         */
        public void destroy() {
        }

        /**
         * Init method for this filter
         *
         */
        public void init(FilterConfig filterConfig) {
                setFilterConfig(filterConfig);
        }

        /**
         * Return a String representation of this object.
         */
        @Override
        public String toString() {
                if (getFilterConfig() == null) {
                        return ("ExpiresFilter()");
                }
                StringBuffer sb = new StringBuffer("ExpiresFilter(");
                sb.append(getFilterConfig());
                sb.append(")");
                return (sb.toString());

        }


}
```