

WEBAPPS-17.0

Web Applications Administration Guide

Isode

Table of Contents

Chapter 1	Introduction.....	1
	A brief introduction to Isode web applications and their architecture.	
Chapter 2	Basic installation and configuration.....	4
	This chapter shows how to install and configure Apache Tomcat to support Isode's web applications.	
Chapter 3	Applications.....	8
	This chapter describes the existing applications such as the Phone Book, Directory Browser, Internet Messaging Administrator, Audit Database, etc.	
Chapter 4	Servlets.....	19
	This chapter describes all provided Isode servlets (both universal and specialized).	
Chapter 5	DSIForm and vCard configuration.....	39
	This chapter describes DSIForm and vCard configuration.	
Chapter 6	Customization and branding.....	49
	This chapter describes how to customize and brand the existing applications.	

Isode and Isode are trade and service marks of Isode Limited.

All products and services mentioned in this document are identified by the trademarks or service marks of their respective companies or organizations, and Isode Limited disclaims any responsibility for specifying which marks are owned by which companies or organizations.

Isode software is © copyright Isode Limited 2002-2018, all rights reserved.

Isode software is a compilation of software of which Isode Limited is either the copyright holder or licensee.

Acquisition and use of this software and related materials for any purpose requires a written licence agreement from Isode Limited, or a written licence from an organization licensed by Isode Limited to grant such a licence.

This manual is © copyright Isode Limited 2018.

1 Software version

This guide is published in support of Isode M-Vault R17.0, M-Switch R17.0, and M-Box R17.0. It may also be pertinent to later releases. Please consult the release notes for further details.

2 Readership

This guide is intended for administrators who plan to deploy, configure and customize Isode Web applications as well as create their own personalized Directory Services Interface applications.

It doesn't contain information on how to build web applications in general or how to create the HTML or CSS code required when branding an application or instructions on configuring Apache Tomcat. (Only basic Tomcat configuration required to run provided applications is described).

3 Typographical conventions

The text of this manual uses different typefaces to identify different types of objects, such as file names and input to the system. The typeface conventions are shown in the table below.

Object	Example
File and directory names	<i>isoentities</i>
Program and macro names	mkpasswd
Input to the system	cd newdir
Cross references	see Section 4, "File system place holders"
Additional information to note, or a warning that the system could be damaged by certain actions.	Notes are additional information; cautions are warnings.

4 File system place holders

Where directory names are given in the text, they are often place holders for the names of actual directories where particular files are stored. The actual directory names used depend on how the software is built and installed. All of these directories can be changed by configuration.

Certain configuration files are searched for first in (*ETCDIR*) and then (*SHAREDIR*), so local copies can override shared information.

The actual directories vary, depending on whether the platform is Windows or UNIX.

Name	Place holder for the directory used to store...	Windows (default)	UNIX
(<i>ETCDIR</i>)	System-specific configuration files.	<i>C:\Isode\etc</i>	<i>/etc/isode</i>
(<i>SHAREDIR</i>)	Configuration files that may be shared between systems.	<i>C:\Program Files\Isode\share</i>	<i>/opt/isode/share</i>
(<i>BINDIR</i>)	Programs run by users.	<i>C:\Program Files\Isode\bin</i>	<i>/opt/isode/bin</i>
(<i>SBINDIR</i>)	Programs run by the system administrators.	<i>C:\Program Files\Isode\bin</i>	<i>/opt/isode/sbin</i>
(<i>EXECDIR</i>)	Programs run by other programs; for example, M-Switch channel programs.	<i>C:\Program Files\Isode\bin</i>	<i>/opt/isode/libexec</i>
(<i>LIBDIR</i>)	Libraries.	<i>C:\Program Files\Isode\bin</i>	<i>/opt/isode/lib</i>
(<i>DATADIR</i>)	Storing local data.	<i>C:\Isode</i>	<i>/var/isode</i>
(<i>LOGDIR</i>)	Log files.	<i>C:\Isode\log</i>	<i>/var/isode/log</i>
(<i>CONFPDUSPOOLDIR</i>)	Large PDUs on disk.	<i>C:\Isode\tmp</i>	<i>/var/isode/tmp</i>
(<i>QUEDIR</i>)	The M-Switch queue.	<i>C:\Isode\switch</i>	<i>/var/isode/switch</i>
(<i>DSADIR</i>)	The Directory Server's configuration.	<i>C:\Isode\d3-db</i>	<i>/var/isode/d3-db</i>

5 Support queries and bug reporting

A number of email addresses are available for contacting Isode. Please use the address relevant to the content of your message.

- For all account-related inquiries and issues: customer-service@isode.com. If customers are unsure of which list to use then they should send to this list. The list is monitored daily, and all messages will be responded to.
- For all licensing related issues: license@isode.com.
- For all technical inquiries and problem reports, including documentation issues from customers with support contracts: support@isode.com. Customers should include relevant contact details in initial calls to speed processing. Messages which are continuations of an existing call should include the call ID in the subject line. Customers without support contracts should not use this address.
- For all sales inquiries and similar communication: sales@isode.com.

Bug reports on software releases are welcomed. These may be sent by any means, but electronic mail to the support address listed above is preferred. Please send proposed fixes with the reports if possible. Any reports will be acknowledged, but further action is not guaranteed. Any changes resulting from bug reports may be included in future releases.

Isode sends release announcements and other information to the Isode News email list, which can be subscribed to from the address: <http://www.isode.com/company/contact.php>

6 Export controls

Many Isode products use protocols and algorithms to encrypt data on connections. If you license the higher grade encryption (HGE) Isode products they are subject to UK Export controls.

You must ensure that you comply with these controls where applicable, i.e. if you are licensing or re-selling Isode products outside the Community with the HGE option selected.

All Isode Software is subject to a license agreement and your attention is also called to the export terms of your Isode license.

Chapter 1 Introduction

A brief introduction to Isode web applications and their architecture.

Isode provides a powerful and flexible set of applications that interact with information in M-Vault, M-Switch and M-Box.

This document describes also included web application framework and ready to use components which allows customers to build custom applications.

1.1 Terms

A brief explanation of terms used in this document.

Web server

Hardware and software used to deliver on request web content (HTML pages, images, files etc) through the internet.

Servlet container

(Also known as a Web container) is a software part of a web server which is responsible for processing requests (HTTP request). The response on request can be simply the content of a static file (like HTML, JPG, PDF etc.) or it can be generated dynamically using built-in functions/methods/procedures.

All Isode applications use Apache Tomcat servlet “container”.

Servlet

Java programming language class used to extend web server functionality (usually used to generate dynamically HTML pages).

Most of Isode servlets allows to generate a few HTML pages required to provide complete solution (for example User manager servlet provides search page allowing to display list of users, editing page allowing to create/edit user details and deleting page allowing to delete user information after confirmation).

Servlets are compiled Java class files, which cannot be modified.

All Isode servlets are collected in Isode shared libraries so can be re-used by different applications, including custom applications.

JSP

JavaServer Pages (JSP) allows static web pages definitions (HTML, XML) to be extended dynamic content by using embedded Java scripts or non-standard tags. JSPs are translated into servlets at runtime; each JSP's servlet is cached and re-used until the original JSP is modified.

JSP files are used to personalize/brand Isode applications.

Web application

A web application is defined as a specific set of web-pages (servlets) with some common appearance (branding) and local configuration.

Web applications (which are provided as files with *.war extensions) contain only specific page templates with all required static elements (such as CSS file, images etc.) and configuration which defines mapping between URLs and servlets (defines which servlets are to be used).

Application framework

Usually different customers require a different set of web-pages but generally all of these applications include common or similar pages such as login pages, index/welcome pages etc.

Isode web application framework provides utility classes and ready to use servlets (common and specialized) allowing to build various applications.

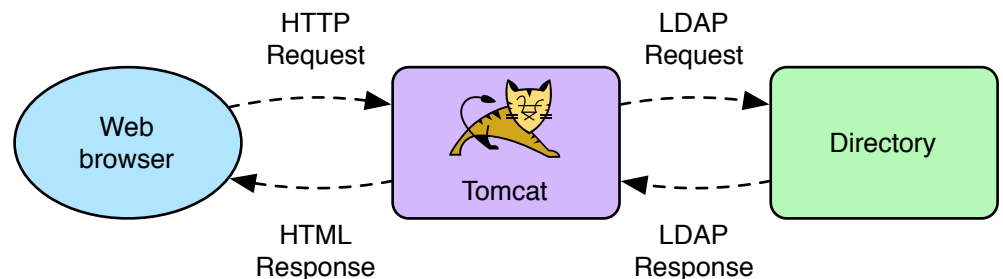
1.2 Architecture

The core of all the Isode web applications is the DSI (Directory Services Interface).

The DSI-based application (executed within a Tomcat servlet container) receives incoming HTTP (HyperText Transport Protocol) requests on a specified server port. The servlet responsible for processing request can generate another requests to an LDAP (Lightweight Directory Access Protocol) directory such as M-Vault. The results from LDAP requests are processed by servlet and returned to the browser in HTML (HyperText Markup Language).

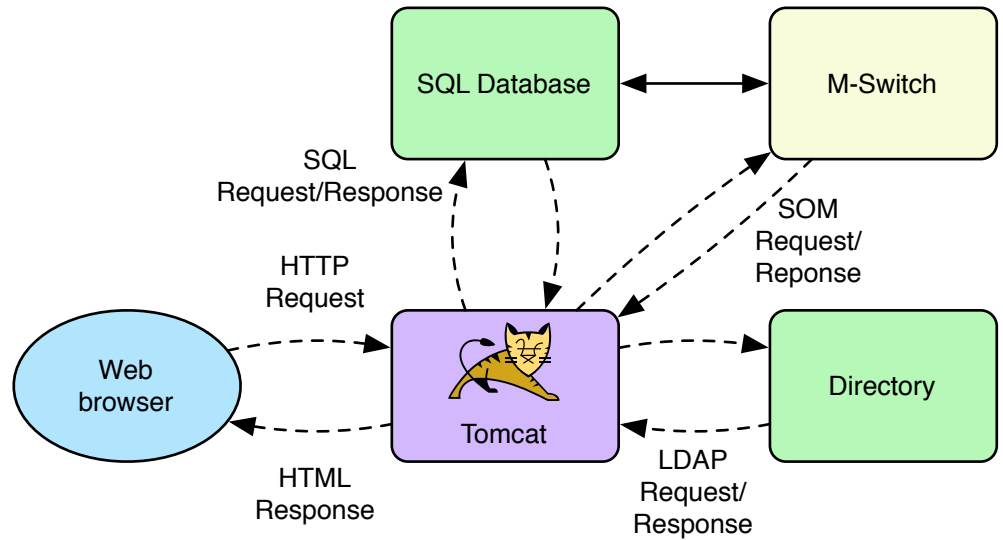
All Isode web applications use directory authentication. To support this one of the common DSI servlets generates typical Login page and tries to bind to directory using provided credentials. Successful bind is being kept associated with a web browser session and allows to perform all following directory operations.

Figure DSInterface request-response flow. DSInterface request-response flow



DSInterface is provided with three fully functional sample applications: Phone book, Directory Browser and Personal Information Administrator.

Other Isode web libraries contain specialized servlets for Isode server management. They provide access to the M-Switch/M-Box configuration stores within directory, M-Switch Audit Database using a standard JDBC interface and M-Switch server directly using Isode-specific protocols such as SOM (Switch Operations Management).

Figure IMAAdmin request-response flow. IMAAdmin request-response flow

These specialized applications are provided as part of the Isode server products such as M-Switch to allow the operation and administration of these systems using a web browser. Some of these functions are also available as native GUIs. These are fully documented in the Administration Guide for the relevant server product e.g. M-Switch.

There are three kind of Isode management web application users:

- system administrator
- delegated administrator
- regular user

For each user type there are separate applications with a different set of available features.

Chapter 2 Basic installation and configuration

This chapter shows how to install and configure Apache Tomcat to support Isode's web applications.

The Isode web applications require Apache Tomcat and a Java JRE. An "embedded" copy of Tomcat which is recommended for evaluation and testing is provided in the Isode packages. For full deployments (requiring branding) you should use either an OS vendor-supplied copy of Tomcat, or a separately downloaded copy of Tomcat. This is termed an "external" Tomcat in the following sections.

Both embedded and external Apache Tomcat configurations are described in the following chapters.

2.1 Installing the Java Runtime Environment (JRE)

Before installing and configuring Tomcat you must ensure that you have the Java Standard Edition Runtime Environment (JRE) 7.0 installed. This can be downloaded if necessary from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Check the Isode release notes for the latest information on dependencies such as the Java JRE.

The Java SE Development Kit (JDK) can also be used as it includes the Java Runtime Environment.

The parent directory where the packages are installed is needed later on, and is referred to as *JAVA_HOME*. For example the value might be */usr/java/jre1.7.0_21* on Unix.

2.2 Installing and configuring the Apache Tomcat server

All Isode web applications are used with Tomcat 7.0 servlet container.

If you want to use web applications for evaluation/testing only or system level administration (where branding is not required) you can use provided Tomcat installation embedded within Isode software (see [Section 2.2.1, "Embedded Apache Tomcat server"](#)). If you need branded/non-standard version see [Section 2.2.2, "External Apache Tomcat server"](#).

The steps below assume that Tomcat is installed in the *CATALINA_HOME* directory and the JDK is installed in the *JAVA_HOME* directory. To find the correct values for (*LIBDIR*) etc, please see [Section 4, "File system place holders"](#).

2.2.1 Embedded Apache Tomcat server

Isode Web Applications Setup command-line tool provided with embedded Tomcat allows to create Tomcat configuration files and deploy provided web applications:

2.2.1.1 UNIX

```
# cd /opt/isode/sbin/isode-tomcat-setup
```

2.2.1.2 WINDOWS

Launch "Isode 16.0" -> "Web Applications Setup Utility" from menu Start.

Warning: The "Windows Security Alert" window may pop-up during setup on Windows blocking network access for Java Platform - Grant the access by checking "Private networks..." and/or "Public networks" (as needed) and pressing "Allow access" button.

2.2.1.3 ALL

Respond to all questions (press the Enter button to confirm suggested values or type your own values) to specify Tomcat operational ports and select which applications to install.

2.2.2 External Apache Tomcat server

Separate copy of Tomcat 7 can be downloaded from <http://tomcat.apache.org>.

2.2.2.1 UNIX

The following configuration steps must be performed on Unix:

1. Install Tomcat by untarring `apache-tomcat-version.tar.gz` while in the `/opt` subdirectory, e.g.:

```
# cd /opt
# tar xvfz /tmp/apache-tomcat-7.0.40.tar.gz
```

2. Add the Isode Java classes from `(LIBDIR)/java/classes` to Tomcat's shared classes. In the file `(CATALINA_HOME)/conf/catalina.properties` set the `shared.loader` option. If a value is not already set, use:

```
shared.loader = (LIBDIR)/java/classes/*.jar
```

If a value is already set, append the new classes instead:

```
shared.loader = existing-information,(LIBDIR)/java/classes/*.jar
```

3. Edit the file `(CATALINA_HOME)/bin/setenv.sh` (if it doesn't exist create one) and add the following lines depending on whether you are using 32-bit or 64-bit Isode packages:

```
JAVA_HOME=the directory containing the JRE/JDK
# On 32-bit systems do this:
JAVA_OPTS="-Djava.library.path=/opt/isode/lib"
# On 64-bit systems do this instead:
JAVA_OPTS="-Djava.library.path=/opt/isode/lib -d64"
# On all systems:
CATALINA_HOME=/opt/apache-tomcat-version
LD_LIBRARY_PATH=/opt/isode/lib
export JAVA_HOME JAVA_OPTS CATALINA_HOME LD_LIBRARY_PATH
```

2.2.2.2 Windows

1. Run the Tomcat installer, e.g. *Apache-Tomcat-7.0.40.exe*
2. After accepting the license make sure that the **Service** component is selected which is located below the **Tomcat** component. This is needed so that Tomcat is started automatically when Windows starts.
3. Select the location of a previously installed JRE on the **Java Virtual Machine** page, for example *C:\Program Files\Java\jre1.7.0*.
4. Don't start Tomcat at this point.
5. Add the Isode Java classes from *(LIBDIR)\java\classes* to Tomcat's shared classes. In the file *(CATALINA_HOME)\conf\catalina.properties* set the `shared.loader` option. If a value is already set, use:

```
shared.loader = (LIBDIR)/java/classes/*.jar
```

If a value is already set, append the new classes instead:

```
shared.loader = existing-information,(LIBDIR)/java/classes/*.jar
```

6. Set the `PATH` system variable so that Tomcat can find the Isode native libraries installed in the Isode *(LIBDIR)*, e.g. *libjavadsapi.dll*. You can do that as follows: Open the Windows Control Panel. Open the **System Properties** tab, go to the **Advanced** page and click on the **Environment Variables** button. Select the `PATH` variable in the **System variables** panel and append the directory containing the Isode DLLs (e.g. *C:\Program Files\Isode\bin*) to the current value. Setting this variable will require a Windows restart.

2.2.2.3 All

To enable secure connections (using SSL and HTTPS) follow the Tomcat documentation for the version of Tomcat you have installed. For Tomcat 7 you can find this documentation at <http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>

2.3 Starting the Apache Tomcat server

2.3.1 Embedded version on Unix

For embedded installation Isode provides `init.d` script allowing to manage tomcat daemon:

```
# service isode-tomcat start
```

2.3.2 External version on Unix

On Unix full Tomcat installation contains own launching script *(CATALINA_HOME)/bin/startup.sh* script, for example:

```
# cd /opt/apache-tomcat-7.0.40/bin
# ./startup.sh
```

2.3.3 Embedded version on Windows

From menu Start run as an Administrator "Isode 16.0" -> "Isode Service Configuration".

If "isode.tomcat" service is not visible select menu "Actions" -> "Install Isode Services". The "Isode Service Wizard" will appear where you should check "Isode Embedded Tomcat Server" and press "Finish" button.

Select "isode.tomcat" service and press "Start" button.

2.3.4 External version on Windows

On Windows start Tomcat by using the Windows **Service Control Manager** or the Tomcat system tray tool, if one was provided by the Tomcat installation.

2.3.5 All

After successful start the default Tomcat start-up page will be accessible on port 8080 on the machine running Tomcat <http://localhost:8080/>

To access the page (and later Isode web applications) from other machines you will have to use full server name, for example if the machine running Tomcat is called mail.example.com, It can be accessed at the following URL <http://mail.example.com:8080/>

2.4 Stopping the Apache Tomcat server

2.4.1 Embedded version on Unix

Used provided init.d script allowing to manage tomcat daemon:

```
# service isode-tomcat stop
```

2.4.2 External version on Unix

On Unix stop Tomcat by running the (*CATALINA_HOME*)/bin/shutdown.sh script, for example:

```
# cd /opt/apache-tomcat-7.0.40/bin
# ./shutdown.sh
```

2.4.3 Embedded version on Windows

From menu Start run as an Administrator "Isode 16.0" -> "Isode Service Configuration".

Select "isode.tomcat" service and press "Stop" button.

2.4.4 External version on Windows

On Windows stop Tomcat by using the Windows **Service Control Manager** or Tomcat system tray tool, if one was provided by the Tomcat installation.

Chapter 3 Applications

This chapter describes the existing applications such as the Phone Book, Directory Browser, Internet Messaging Administrator, Audit Database, etc.

Isode web applications include servlets which are responsible for generating web pages, provided in shared libraries (*.jar files).

Web Applications provided as *.war files contain application configuration specifying the base URL, the list of used servlets, the mapping between servlets and URLs and general page template (branding).

3.1 Directory Services Interface

Directory Services Interface (DSI) is a framework to build directory oriented applications. It includes some general-purpose applications and sample applications based on DSI Forms.

A set of sample DSInterface applications are provided as part of M-Vault, which can be used as provided, or can be used as a base for customized applications.

3.1.1 Phone Book

dsinterface-phonebook.war -> <http://localhost:8080/dsinterface-phonebook>

A simple application which contains two pages: the first displays a list of users with basic information and phone numbers (see [DSIForm - List](#)) and the second (called from a link in the list for each person) displays selected personal details (see [DSIForm - Show](#)).

The personal information returned is filtered so that only entries which contain at least one phone number are displayed.

Both anonymous (without login) and authenticated (after login) access are available what may have influence on displayed users depending on directory configuration (which users can be displayed for anonymous users and which for specific user bind).

Setting-up:

1. [Section 3.4, "Installing Isode Web Applications"](#) (*dsinterface-phonebook.war*)
2. [Section 3.5.2, "Configuring directory access"](#)
3. [Section 3.5.6, "Configuring password self-reset"](#) (optional)

3.1.2 Directory Browser

dsinterface-directory.war -> <http://localhost:8080/dsinterface-directory>

This is an extension of the Phone Book application. It contains a sample Index page with welcome text and a link to the contacts browser page - the same as in the phone book application but without any filtering.

Both anonymous (without login) and authenticated (after login) access are available what may have influence on displayed users depending on directory configuration (which users can be displayed for anonymous users and which for specific user bind).

Setting-up:

1. [Section 3.4, “Installing Isode Web Applications”](#) (*dsinterface-phonebook.war*)
2. [Section 3.5.2, “Configuring directory access”](#)
3. [Section 3.5.6, “Configuring password self-reset”](#) (optional)

3.1.3**Personal Information Administrator**

dsinterface-pia.war -> <http://localhost:8080/dsinterface-pia>

A more comprehensive sample application is the Personal Information Administrator. It contains [Login/Logout](#) pages (logging in is required to access other pages), an index page, a simple white pages editor (DSIForm based), [Password Setup](#) page, groups browser (DSIForm based) and a contacts browser (which is the same as in the Directory browser).

Setting-up:

1. [Section 3.4, “Installing Isode Web Applications”](#) (*dsinterface-phonebook.war*)
2. [Section 3.5.2, “Configuring directory access”](#)
3. [Section 3.5.6, “Configuring password self-reset”](#) (optional)

3.1.4**Password Manager**

dsinterface-pwdmgr.war -> <http://localhost:8080/dsinterface-pwdmgr>

Specialised application (not based on DSIForm) allowing to manage users' password. Configuration allows to specify group entry of users allowed to access this application. Manageable user entries limitations are also configurable.

Setting-up:

1. [Section 3.4, “Installing Isode Web Applications”](#) (*dsinterface-phonebook.war*)
2. [Section 3.5.2, “Configuring directory access”](#)
3. [Section 3.5.7, “Configuring password managers”](#)
4. [Section 3.5.6, “Configuring password self-reset”](#) (optional)

3.2**Audit Database applications**

The first set of specialised applications created for Isode servers management. The M-Switch audit database allows to perform various tasks which are divided depending on targeted user. This division resulted in creation of three separate web applications: AuditDB-System, AuditDB-Delegated and AuditDB-Personal.

3.2.1**AuditDB-System**

auditdb-system.war -> <http://localhost:8080/auditdb-system>

Pages for system administrator

Message Tracking, Quarantine Manager, Statistics

Setting-up:

1. [Section 3.4, “Installing Isode Web Applications”](#) (*dsinterface-phonebook.war*)
2. [Section 3.5.2, “Configuring directory access”](#)

3. [Section 3.5.4, “Configuring system administrators”](#)
4. [Section 3.5.10, “Configuring audit database access”](#)
5. [Section 3.5.6, “Configuring password self-reset”](#) (optional)

3.2.2 AuditDB-Delegated

auditdb-delegated.war -> <http://localhost:8080/auditdb-delegated>

Currently there are no AuditDB pages for delegated administrator so this application is empty (this is for future use).

Setting-up:

1. [Section 3.4, “Installing Isode Web Applications”](#) (*dsinterface-phonebook.war*)
2. [Section 3.5.2, “Configuring directory access”](#)
3. [Section 3.5.4, “Configuring system administrators”](#)
4. [Section 3.5.5, “Configuring delegated administration”](#)
5. [Section 3.5.10, “Configuring audit database access”](#)
6. [Section 3.5.6, “Configuring password self-reset”](#) (optional)

3.2.3 AuditDB-Personal

auditdb-personal.war -> <http://localhost:8080/auditdb-personal>

Currently this only provides access to personal [Quarantine Manager](#).

Setting-up:

1. [Section 3.4, “Installing Isode Web Applications”](#) (*dsinterface-phonebook.war*)
2. [Section 3.5.2, “Configuring directory access”](#)
3. [Section 3.5.10, “Configuring audit database access”](#)
4. [Section 3.5.6, “Configuring password self-reset”](#) (optional)

3.3 Internet Messaging Administrator

The Internet Messaging Administrator contains applications targeted for three different user groups: System administrators (*imadmin-system*), Delegated administrators (*imadmin-delegated*) and end-users (*imadmin-personal*). All of them provide the most complete set of functionality for directory and messaging configuration and management.

3.3.1 IMAdmin-System

imadmin-system.war -> <http://localhost:8080/imadmin-system>

System administrator pages allows administrators to create/manage system-wide settings.

General Pages

[Directory Configuration Browser, User Manager](#)

M-Box Administration Pages

[Message Store Configuration, Group Manager, Shared Folders Root Manager](#)

Message Operator Pages (M-Switch Audit Database)

[Message Tracking, Quarantine Manager, Statistics](#)

IMAdmin Administration Pages

[System Administrators Group Manager](#), [Delegate Administration Realms Manager](#)

Setting-up:

1. [Section 3.4, “Installing Isode Web Applications”](#) (*imadmin-system.war*)
2. [Section 3.5.2, “Configuring directory access”](#)
3. [Section 3.5.4, “Configuring system administrators”](#)
4. [Section 3.5.5, “Configuring delegated administration”](#) (optional)
5. [Section 3.5.6, “Configuring password self-reset”](#) (optional)

3.3.2 IMAdmin-Delegated

imadmin-delegated.war -> <http://localhost:8080/imadmin-delegated>

Delegated administrator pages allows administrators to perform limited administration tasks within a specific administration realm (usually associated with a domain).

Administration Realm pages

[User Manager](#), [Delegate Administration Realms Manager](#), [Directory Browser](#)

Setting-up:

1. [Section 3.4, “Installing Isode Web Applications”](#) (*imadmin-delegated.war*)
2. [Section 3.5.2, “Configuring directory access”](#)
3. [Section 3.5.4, “Configuring system administrators”](#) (to allow manage delegated administrators)
4. [Section 3.5.5, “Configuring delegated administration”](#)
5. [Section 3.5.6, “Configuring password self-reset”](#) (optional)

3.3.3 IMAdmin-Personal

imadmin-personal.war -> <http://localhost:8080/imadmin-personal>

Personal management pages allow individual users to configure personal anti-spam configuration, such as managing message quarantine etc.

Information

[Directory Browser](#)

Management

[Quarantine Manager](#)

Settings

[General \(white pages\)](#), [Password Setup](#), [Delivery Rules](#), [White List Editor](#), [Anti-Spam Settings](#)

Setting-up:

1. [Section 3.4, “Installing Isode Web Applications”](#) (*imadmin-personal.war*)
2. [Section 3.5.2, “Configuring directory access”](#)
3. [Section 3.5.6, “Configuring password self-reset”](#) (optional)

3.4 Installing Isode Web Applications

Each Isode Web application specific data (configuration, personalisation and references to shared servlets) is being provided as a *.war file (full names are specified in chapters describing particular applications).

To install a web application in the Tomcat Servlet container it's *.war file must be simply copied into appropriate directory (Tomcat takes care about the rest of the process like unpacking, internal configuration etc.).

3.4.1 Embedded Tomcat

Isode Tomcat Setup Tool is responsible for both configuring Tomcat and installing/upgrading web applications - You must simply launch it and confirm which *.war files to install (see [Section 2.2.1, "Embedded Apache Tomcat server"](#)).

3.4.2 External Tomcat

Copy selected web application files *.war to the (*CATALINA_HOME*)/webapps, e.g.:

UNIX:

```
# cp /opt/isode/lib/java/webapps/dsinterface-*.war \
/opt/apache-tomcat-7.0.40/webapps/
```

WINDOWS:

```
C:\> copy "C:\Program Files\Isode\bin\java\webapps\dsinterface-*.war" \
"C:\Program Files\Apache Software Foundation\Tomcat 7.0.40\webapps\"
```

3.5 Configuring Isode Web Applications

Following sub-chapters describes various configuration tasks referenced from all the documentation and are not supposed to be executed one by one.

3.5.1 Creating dsinterface.xml file

The (*ETCDIR*)/dsinterface.xml is the core configuration file for all directory based applications. It is responsible for keeping directory access information as well as specific applications/servlets options.

You can create it using provided sample file:

UNIX:

```
# cp /etc/isode/dsinterface.xml.sample /etc/isode/dsinterface.xml
```

WINDOWS:

```
C:\> copy "C:\Isode\etc\dsinterface.xml.sample" \
"C:\Isode\etc\dsinterface.xml"
```

3.5.2 Configuring directory access

Ensure *dsinterface.xml* file exists (see [Section 3.5.1, "Creating dsinterface.xml file"](#)).

Edit this file to ensure all settings are correct (all options considered in this chapter are below `<ldap>` element):

`<url>` - URL specifying address of the directory server. You may need to replace localhost with the domain of your LDAP server. And the port number to a value other than 389 (the standard LDAP port number). M-Vault servers may be set to run on other ports. 19389 is sometimes used, especially for Unix servers so that they do not need to run as root (required for low number ports).

Note: Users logging to web application use SASL ID as their login. To support SASL based authentication DSI requires information about current directory SASL configuration - to allow this special user must be created within directory with appropriate privileges and indicated in *dsinterface.xml* (it is called SASL Reader later in this documentation):

`<sasl_reader_dn>` - The DN of the entry for a user who has permission to read SASL configuration. Both this setting and `<sasl_reader_pwd>` must be specified, once the user has been created.

Tip: Most Isode templates used to create DSAs contain "DSA Operators" group with privileges required by SASL Reader. You can then create "Person" object somewhere in your directory, setup the password in this entry, add it's DN to "DSA Operators" group members and put DN to `<sasl_reader_dn>` and password to `<sasl_reader_pwd>` to fulfil requirements

`<sasl_config_dn>` - The DN of the entry in the directory that contains SASL configuration. If no value is specified (the default), then DSI will locate the entry by performing a search for `(objectclass=isodeDSAConfiguration)`. Since in most cases there will only be one such entry, it is only necessary to set this value if extra configuration entries exist.

```
<dsinterface>
  <ldap>
    <url>ldap://localhost:19389</url>

    <sasl_reader_dn>cn=SASL Reader,cn=Users,c=xx</sasl_reader_dn>
    <sasl_reader_pwd>secret</sasl_reader_pwd>

    <sasl_config_dn>$(default)</sasl_config_dn>

    (...)

  </ldap>
</dsinterface>
```

3.5.3 Configuring login options

Ensure *dsinterface.xml* file exists (see [Section 3.5.1, “Creating dsinterface.xml file”](#)) and is basically configured (see [Section 3.5.2, “Configuring directory access”](#)).

Note: All login options must be in *dsinterface.xml* under <ldap> element.

3.5.3.1 SASL ID login

By default, web based applications authenticate to the directory using a SASL ID and user password (for example, "webuser@mysystem.net"/"secret")

The SASL ID will be mapped to a DN in the directory using whatever mapping schema has been configured in the directory.

By default SASL configuration is stored in M-Vault within Isode DSA Config entry (cn=core,cn=config) (You can easily configure it using M-Vault Console's SASL Configuration). If you want to use alternative SASL config entry or select which to use in case of multiple Isode DSA Config entries available within the same directory you need to indicate its dn in <sasl_config_dn> option.

Within the directory, you must configure a user who has read access to the SASL configuration. This user's DN and password should be specified with <sasl_reader_dn> and <sasl_reader_pwd>.

Warning: If the SASL reader is not correctly specified, this may lead to unpredictable behaviour, since in the absence of available SASL configuration information, a default set of rules will be used which may lead to authentication problems for other users.

SASL configuration in case of search based mappings specifies only attributes used to distinguish user, and domain entries. In some deployments it may be not sufficient so *dsinterface.xml* allows to specify additionally <sasl_user_filter> and <sasl_domain_filter> LDAP filters.

3.5.3.2 Alternative login

Next to SASL ID based login it is allowed to specify user defined login search filters <alt_login_search_filter>. There can be multiple such filters and each as a separate XML element and responsible for matching login provided by user in login page with specific and unique entries.

```
<alt_login_search_filter>(&!(mailLocalAddress=%LOGIN%)(objectClass=person))</alt_login_search_filter>
```

The login dialog value is passed as a %LOGIN% variable. The '&' character used in LDAP filters has special meaning in XML so it must be encoded as '&' (it applies as well to other XML special characters used there).

Tip: You can disable SASL ID based login and use alternative's only by setting <saslid_login> to "off".

3.5.4 Configuring system administrators

Ensure *dsinterface.xml* file exists (see [Section 3.5.1, “Creating dsinterface.xml file”](#)) and is basically configured (see [Section 3.5.2, “Configuring directory access”](#)).

System administrators within Isode web applications means users allowed to login (having SASL ID and password specified) and allowed to perform directory operations requiring extensive privileges.

DSInterface applications depends on directory access control where users are granted privileges through being members of privileged groups. The only option which needs to be configured is `<system_admin_group_dn>` which should contain DN of the group associated with privileges to create/edit/delete entries and attributes in data areas of the DIT (do not need user password read privilege but requires modification privilege) and privilege to manipulate ACI.

Tip: Most Isode templates used to create DSAs contain "DSA Managers" group which can be used here.

Note: System administrators group members are the only allowed to login into web application designed for such users (like IMAdmin-System).

```
<dsinterface>
  <ldap>
    (...)

    <system_admin_group_dn>cn=DSA Operators,cn=Groups,c=xx
    </system_admin_group_dn>

    (...)
  </ldap>
</dsinterface>
```

3.5.5 Configuring delegated administration

Ensure *dsinterface.xml* file exists (see [Section 3.5.1, "Creating dsinterface.xml file"](#)) and is basically configured (see [Section 3.5.2, "Configuring directory access"](#)).

Delegated administration (Currently used in IMAdmin-Delegated) allows selected users to perform limited administration operations.

Delegated administration is always associated with Realm defining administrators, various limits and some settings for regular users associated with given realm. Administration realms can be managed easily with IMAdmin-System application which contains [Delegate Administration Realms Manager](#).

The most important (usually the only) thing which requires configuration is `delegated_admin_realms_base_dn` which should indicate DN of the "Container" entry within your directory used as a parent for all Realm entries.

Other options define ACI objects names which are perfectly fine to be used with default values [Delegate Administration Realms Manager](#) will create them automatically if required.

```
<dsinterface>
  <ldap>
    (...)

    <delegated_admin_realms_base_dn>cn=Realms,c=xx
    </delegated_admin_realms_base_dn>

    (...)
  </ldap>
</dsinterface>
```

3.5.6 Configuring password self-reset

Ensure *dsinterface.xml* file exists (see [Section 3.5.1, "Creating dsinterface.xml file"](#)) and is basically configured (see [Section 3.5.2, "Configuring directory access"](#)).

User forgetting login password is quite common situation. DSInterface supports self-reset password mechanism allowing to solve the problem without engaging any operator.

All described options are configured in *dsinterface.xml* and under `<password_reset>` element.

By default the mechanism support is disabled (as not configured would lead to unwanted configuration errors reports). To enable it change `<enabled>` to "yes". Next you must indicate DN and password of the user which will be responsible for changing the password (you must create such user within your directory - it doesn't need to have SASL ID but must have privileges to modify other users password)

Tip: Most Isode templates used to create DSAs contain "Password Managers" group which can be used to grant The "password reset manager" user privileges (just add the user to members of this group).

Password reset operation must be authenticated in other way than password therefore confirmation email is being used here. `<smtp>` options must be set to allow sending such emails.

```
<dsinterface>
  (...)

  <!-- Password reset servlet configuration -->
  <password_reset>

    <!-- Access to password reset from login page
    - default "no" -->
    <enabled>yes</enabled>

    <!-- Distinguished Name and password of the Password Manager,
    whose bind will be used to perform reset operations. This user
    must have access rights that allow changing other user entries,
    including password attributes.
    The default values for these fields are empty, and so must be
    specified before the password reset function will work. -->
    <manager_dn>cn=Password Reset Manager,cn=Users,c=xx
      </manager_dn>
    <manager_pwd>secret</manager_pwd>

    <!-- SMTP server used to send verification emails.
    If your SMTP server requires authentication (RFC 2554 (4))
    before accepting mail you have to setup username and password
    here -->
    <smtp host="localhost" port="587"
      username="pwdr@yourcompany.com">
      <password servpass:encrypt="true">secret</password>
    </smtp>

    <!-- Email subject. The default is "Password Reset Request" -->
    <subject>$(default)</subject>

    <!-- Verification email template file name.
    The default is
    "$(isode.dir.etc)password-reset-notification-template.html" -->
    <template>$(default)</template>
```

```

    <!-- Site name (may be as a HTML link) used in verification
    email. The default is "YOUR SITE" -->
    <site>$(default)</site>

</password_reset>

(...)
</dsinterface>

```

3.5.7 Configuring password managers

Ensure *dsinterface.xml* file exists (see [Section 3.5.1, "Creating dsinterface.xml file"](#)) and is basically configured (see [Section 3.5.2, "Configuring directory access"](#)).

Tip: Most Isode templates used to create DSAs contain "Password Managers" group which can be used here.

```

<dsinterface>
  (...)

  <!-- Password manager options. Any of the options can be set
  to "$(default)"; The default values are described below. -->
  <password_manager>
    <!-- The DN of a "groupOfNames" entry in the directory,
    whose members are allowed to use the password manager
    web page. By default, no password manager group DN is
    defined, which means that no-one is authorized to use
    the password manager. -->
    <group_dn>$(default)</group_dn>

    <!-- Search filter used to find entries in the directory
    whose password can be changed by the password manager
    web page. The default search filter is
    "(objectclass=inetorgperson)" -->
    <search_user_query>$(default)</search_user_query>

    <!-- The DN used as the base DN when searching for entries
    in the directory whose password can be changed.
    The default base DN is the root DN, which means that the
    whole directory will be searched. -->
    <search_base_dn>$(default)</search_base_dn>

    <!-- Whether password-policy information should be shown
    to the password manager ("Yes") or not ("No"). Note that
    if a Directory Server is not implementing password policy,
    then this option has no effect.
    But for a server that implements password policy, the
    option controls whether any policy related information
    that the server returns (such as "password is too short")
    is displayed to the user.
    The default for this value is "Yes". -->
    <using_pp>$(default)</using_pp>

  </password_manager>
  (...)
</dsinterface>

```

3.5.8 Creating imadmin.xml file

Create a IMAdmin configuration file using the provided sample file.

Unix:

```
# cp /etc/isode/imadmin.xml.sample /etc/isode/imadmin.xml
```

Windows:

```
C:\> copy "C:\Isode\etc\imadmin.xml.sample" "C:\Isode\etc\imadmin.xml"
```

3.5.9 Creating auditdb.xml file

Web applications accessing M-Switch audit database re-use access configuration from `auditdb.xml` so it may be already created if Tomcat is co-located on the same host as `auditdb` daemons/services

To create an audit database configuration file use provided sample file:

Unix

```
# cp /etc/isode/auditdb.xml.sample /etc/isode/auditdb.xml
```

Windows

```
C:\> copy "C:\Isode\etc\auditdb.xml.sample" "C:\Isode\etc\auditdb.xml"
```

3.5.10 Configuring audit database access

Ensure `auditdb.xml` file exists (see [Section 3.5.9, "Creating auditdb.xml file"](#)).

For documentation of audit database access configuration see "M-Switch Admin Guide" chapter "Message Audit Database".

3.5.11 Creating ms.conf file

Create a M-Box configuration file using the provided sample file.

UNIX:

```
# cp /etc/isode/ms.conf.ldap /etc/isode/ms.conf
```

WINDOWS:

```
C:\> copy "C:\Isode\etc\ms.conf.ldap" "C:\Isode\etc\ms.conf"
```

3.6 Updating Isode Web Applications

To update web applications, copy the updated `*.war` files as described in the installation section of this manual and restart Tomcat.

Chapter 4 Servlets

This chapter describes all provided Isode servlets (both universal and specialized).

Isode servlets are task oriented and each of them can generate one or more web pages associated with specific responsibility area (functional and privilege based). For example the Personal Quarantine Manager servlet has an index/search page which allows you to find quarantine messages, pages with message details and pages which allow messages to be released (covering complete functionality) and it works only for messages associated with logged in user (privilege).

Each servlet is designed for a specific kind of user (System administrator, Delegated administrator, Personal (end user)). Sometimes the same servlet can be shared, but for a user with lower rights some options are hidden. For example a User Editor is shared between System and Delegated administrators, but the Delegated version is limited to a subset of the System's option).

4.1 Common servlet parameters

All servlets can be called with the appropriate set of parameters using both HTTP `GET` and `POST` methods. Handling `GET` and `POST` requests is unified and transparent for all servlets, therefore all parameters which are normally sent by form using the `POST` method can be initialised by URL parameters. Significant parameters are described within the sections for the specific servlets.

Common parameters include:

`form_id`

All forms contain a hidden field with the form ID which is used to distinguish forms if there are more than one on the page. The existence of this parameter (in a URL for example) is interpreted as a form submission, which allows you to recognize which form has been submitted. The default value of this field is `form` and providing this parameter allows the simulation of form submission.

`returl`

Use this parameter to specify the destination URL when all forms lead to the same page after submission.

4.2 Basic servlets

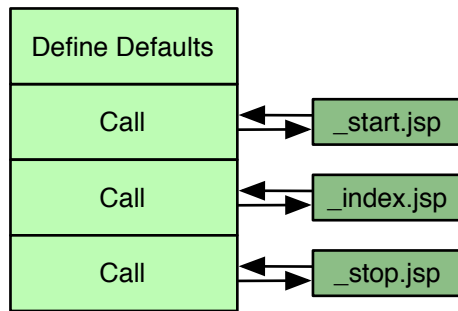
This chapter describes basic web-application framework servlets which have no dependency on external services.

4.2.1 IndexServlet

`com.isode.web.servlets.IndexServlet`

Almost every web application requires an index/home/welcome page. It can be created directly in HTML, as a JSP file or even as a custom servlet. `DSInterface` provides a special

index servlet which doesn't generate any content but calls another external JSP file called *_index.jsp* to provide the content.



Index servlet doesn't use any HTTP parameters but still supports generation multiple pages - each *_index.jsp* must be in separate folder visible as part of URL (must map directly folder name to part of URL). For example `http://localhost:8080/application/index` expects *_index.jsp* in main application folder, `http://localhost:8080/application/subfolder/index` expects in *subfolder*.

Note: We strongly recommend using this servlet as it means we can share page templates prepared for other servlets which allows for common exception handling.

4.2.2 ErrorPageServlet

```
com.isode.web.servlets.ErrorPageServlet
```

This servlet is similar to index servlet as it is wrapping external **.jsp* file but it is supposed to handle standard HTTP error pages (allowing customization).

URL parameters:

code

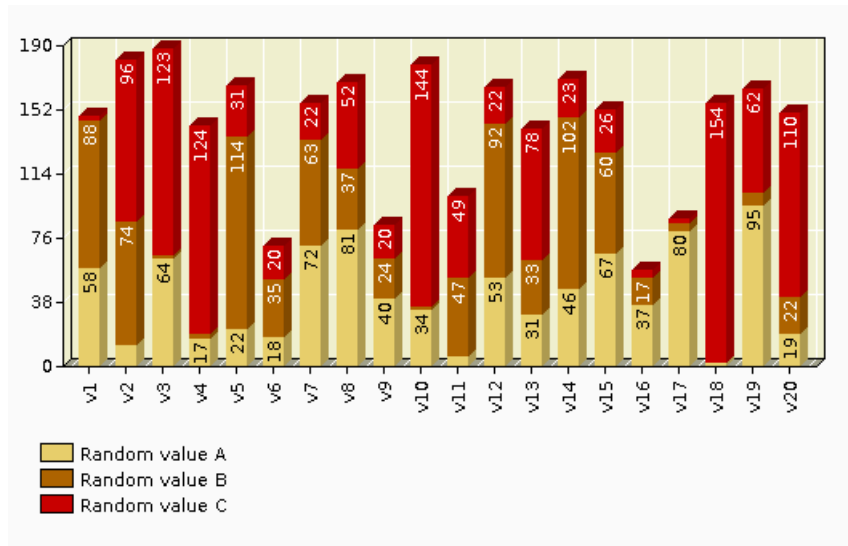
An optional integer code of HTTP error (default is 404) and determines error file name (like *_404.jsp*).

Note: Error JSP files must be located in common folder.

4.2.3 ChartServlet

```
com.isode.web.servlets.ChartServlet
```

Very flexible and configurable servlet generating chart image (PNG).



URL parameters:

style

Chart style: columns (default), stackcolumns, bars, stackbars, lines.

scale

Values scale: linear, logarithmic

var

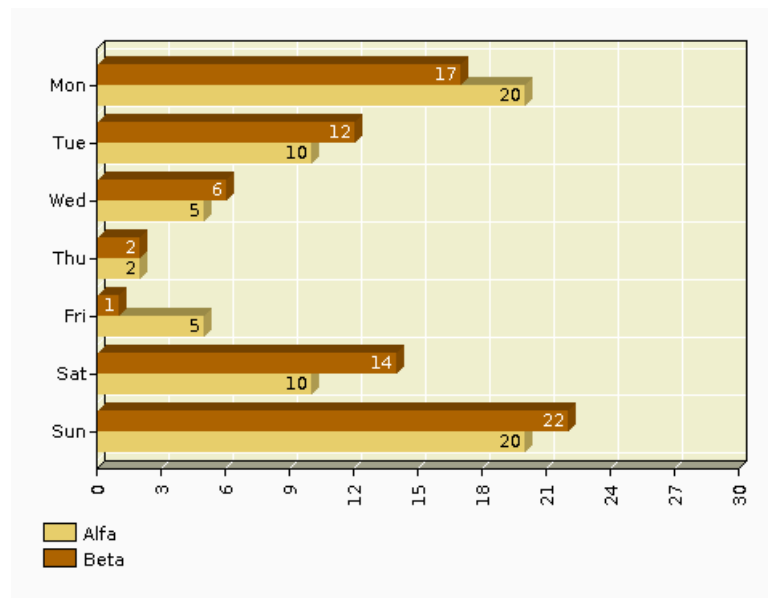
Comma separated list of variables. Names listed here are displayed in the chart legend. The number of variables determines interpretation of values list (see next parameter).

val

Comma separated sequence of values (value name displayed in the chart followed by values for all variables).

Sample: `chart.png?style=bars; var=Alfa,Beta;`

`val=Mon,20,17,Tue,10,12,Wed,5,6,Thu,2,2,Fri,5,1,Sat,10,14,Sun,20,22`



val_format

Allows to determine how values are displayed/formatted in the chart (see java DecimalFormat parameters for documentation)

val_max

Suggest max value which is used to force chart axis length/division.

`var_space`
Space (in pixels) between variable representations (bars/columns etc.).

`val_space`
Space (in pixels) between value representations (bars/columns etc.).

`depth`
The 3D effect parameter determining depth of elements (default 5 pixels).

`var_width`
The width (in pixels) of variable representations (bars/columns etc.).

`width`
The chart canvas width (in pixels). (It is usually calculated from other parameters)

`height`
The chart canvas height (in pixels). (It is usually calculated from other parameters).

`img_color_bg`
Whole image background color (hexadecimal RGB value like default fffffff = white).

`img_color_fg`
Basic image foreground color (hexadecimal RGB value like default 000000 = black).

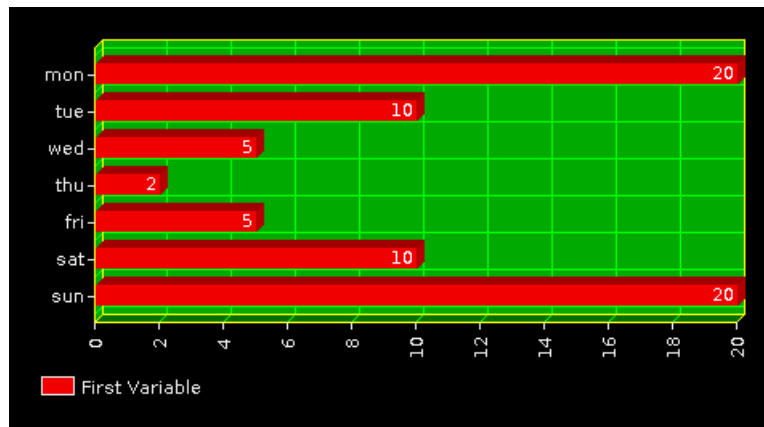
`canvas_color_bg`
Chart canvas background color (hexadecimal RGB value like default efefce).

`canvas_color_fg`
Chart canvas foreground color (hexadecimal RGB value like default 000000 = black).

`canvas_color_lines`
Chart canvas scale lines color (hexadecimal RGB value like default fffff = white).

`var_colors`
Comma separated list of colors used by specific variables (hexadecimal RGB value like ff0000 = red). The default is
e7ce6b,ad6300,c80000,00c800,c8c800,0000c8,c800c8,00c8c8,404040,808080

Sample of modified colors: `chart.png?style=bars; var=First Variable; val=mon,20,tue,10,wed,5,thu,2,fri,5,sat,10,sun,20; img_color_bg=0; img_color_fg=eeeeee; canvas_color_bg=00aa00; canvas_color_fg=ffff00; canvas_color_lines=00ff00; var_colors=ff00000`

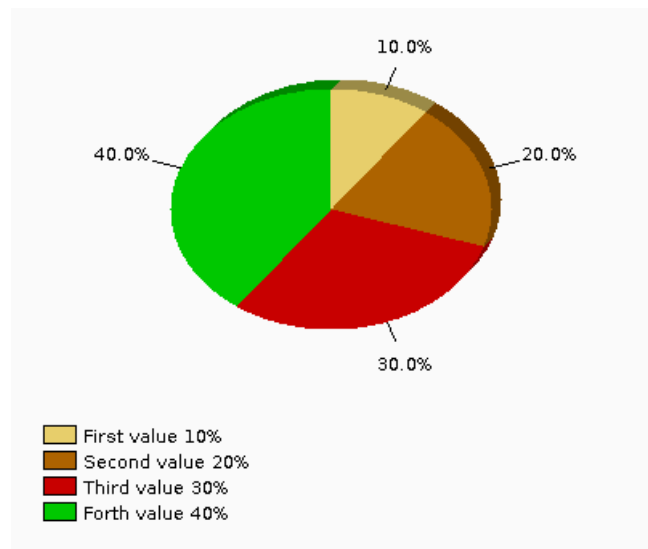


4.2.4 PieChartServlet

`com.isode.web.servlets.PieChartServlet`

Very flexible and configurable servlet generating pie-chart image (PNG).

Sample: `piechart.png?val=First value 10%25,10,Second value 20%25,20,Third value 30%25,30,Forth value 40%25,40`



URL parameters:

`val`

Comma separated sequence of values (value name and value pairs).

`depth`

The 3D effect parameter determining depth of elements (default 5 pixels).

`val_format`

Allows to determine how values are displayed/formatted in the chart (see java `DecimalFormat` parameters for documentation)

`img_color_bg`

Whole image background color (hexadecimal RGB value like default `ffffff` = white).

`img_color_fg`

Basic image foreground color (hexadecimal RGB value like default `000000` = black).

`var_colors`

Comma separated list of colors used by specific variables (hexadecimal RGB value like `ff0000` = red). The default is `e7ce6b,ad6300,c80000,00c800,c8c800,0000c8,c800c8,00c8c8,404040,808080`

4.3 DSInterface servlets

This chapter describes the DSInterface servlets which can be used in your own applications.

Directory Services Interface servlets allow to access directory data and to use directory as an authentication and access control core. DSI provides also all commonly used functionality like password management and universal servlets allowing to build own pages based on templates (DSIForm).

4.3.1 LoginServlet

`com.isode.dsinterface.LoginServlet`

The Login servlet allows for the authentication of a user for the duration of the current web application session.

All users (privileged and regular) authenticate with a SASL ID and a password. A full SASL ID including domain must be used, e.g. `username@company.com`.

Note: A typical use scenario assumes that this page is not called directly.

When (other) pages which require user authentication are requested – automatic redirection is executed to the Login page with the `returnl` parameter set to the requested page URL.

URL parameters:

`username`
User ID.

`password`
User password.

`returnl`
Return URL.

Tip: Providing all parameters including `form_id="form"` allows auto-login operation.

4.3.2 LogoutServlet

`com.isode.dsinterface.LogoutServlet`

Simple form-less page which can be used to logout a user (e.g. remove authentication from current session)

URL parameters:

`returnl`
Return URL.

4.3.3 PasswordServlet

`com.isode.dsinterface.PasswordServlet`

PasswordServlet generates simple page with a form allowing logged-in users to change their own password.

URL parameters:

`op2`
Old password for confirmation.

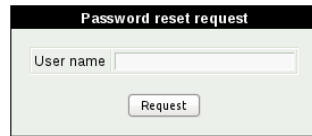
- np1 New password.
- np2 Repeated new password.

4.3.4 PasswordResetServlet

`com.isode.dsinterface.PasswordResetServlet`

PasswordResetServlet after configuration extends Login page by providing mechanism for self-reset password in case of forgetting (see: [Section 3.5.6, “Configuring password self-reset”](#)).

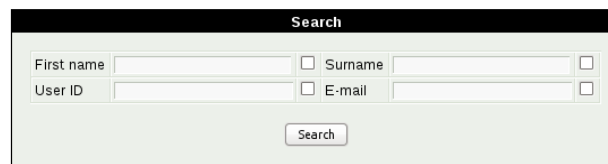
First "Password reset request" web page which submission results in generation of an email containing verification code. The email contains also a link leading back to password reset page where given code gives access to password change page.



4.3.5 PasswordManagerServlet

`com.isode.dsinterface.PasswordManagerServlet`

PasswordManagerServlet allows after configuration privileged users to change other users password (see: [Section 3.5.7, “Configuring password managers”](#)).



2 users found

PAGE [<< < 1 > >>] ROWS / PAGE [10 20 50 100]					
▲	First name	Surname	User ID	E-mail	Change
1	Albert	Einstein		ae@isode.com	
2	Thomas	Atkins		ta@isode.com	

PAGE [<< | < | **1** | > | >>] ROWS / PAGE [**10** | 20 | 50 | 100]

Tip: There is dedicated ready-to-use application with this servlet see: [Section 3.1.4, “Password Manager”](#).

4.3.6 ImageServlet

`com.isode.dsinterface.ImageServlet`

Servlet allowing for the display of images stored in the directory server.

URL parameters:

- dn Mandatory, the DN of the entry where image is stored.
- at Optional attribute type - if not specified then **jpegPhoto** is used.
- nr Optional image number if the attribute is multivalued – default is 0.

`src`

This servlet can also provide image which is not stored withing DSA but as a regular file within application file structure. This option indicates the file location (`dn`, `at` and `nr` parameters are ignored then).

`size`

Optional image size, if defined this causes the image to be proportionately scaled with the longer side sized to the value of this parameter.

Note: File sourced image is being used to display default image (`/common/images/noimage.gif`) in case of not available/specified image in DSA.

4.3.7

VCardServlet

`com.isode.dsinterface.VCardServlet`

A servlet generating a vCard from data collected in the directory.

The content of the generated vCard can be configured using the DSIFORM configuration (described in [Chapter 5, DSIFORM and vCard configuration](#)).

URL parameters:

`id`

Mandatory if `dn` is not defined - points to the SASL User ID which should be used to generate the vCard.

`dn`

Mandatory if `id` is not defined - points to the user entry DN which should be used to generate the vCard.

`cfg`

Specifies the DSIFORM configuration ID.

`form`

Optional switch to change the declared content type, which is useful for debugging. The default (if not set) is `text/x-vcard` which usually launches the action associated with vCard data. The only other possible value here is `text` which will change the content type to `text/plain` which allows the generated vCard to be displayed in the web browser instead.

4.3.8

DSIFORMServlet

`com.isode.dsinterface.DSIFORMServlet`

The “heart” of the DSInterface – DSIFORMServlet – allows for the generation of multiple types of web pages (see [Chapter 5, DSIFORM and vCard configuration](#)).

Instead of struggling with building directory interfacing pages from scratch and dealing with problems of how to search/display/edit directory entries – you can just use the predefined DSIFORM types and using simple configuration files define only the required features, i.e. those you want to search/display/edit.

To generate the appropriate DSIFORM the servlet requires some specific configuration. Usually this is done by pointing the `cfg` URL parameter at the configuration filename (without extension). Alternatively this can be hidden from the URL by specifying it instead in the in `web.xml` (application configuration file) and associated with a specific URL.

URL parameters:

`cfg`

Points to the DSIFORM configuration.

dn

Mandatory for single entry DSIForm types - DN specification.

web.xml parameters:

dsiforms-cfg

Points to the DSIForm configuration.

4.3.8.1 List

The most complex DSIForm page type is the “List” form. It allows the directory to be searched using configured criteria and displays results in a table.

The screenshot displays the 'isode Directory Services Interface' with a search form and a table of contacts. The search form includes fields for First Name, Surname, Email, and Car, along with a checkbox for 'Display entries with at least one phone number defined only' and a 'Search' button. Below the search form, it states 'Found 16 Contacts:' and shows a table with 5 rows of contact information.

Photo	Name	Phone (Work)	Home Phone	Mobile Phone	Car	V-Card	Person Details
	Sally Brown	070 4166 2923					
	Eric Perez	070 4166 2970	079 8082 1189				
	Lisa Warner	070 4166 2973	070 4166 5213				
	Vaughn Whiting						
	System Admin						

Each result row corresponds to a single entry. Columns (and fields in these columns) refer to “items” defined in the configuration (usually specified as single entry attributes). The “items” can be displayed in different ways:

1. simple text,
2. email,
3. URL,
4. image
5. link to another DSIForms
6. complex generation from multiple attributes (e.g. name+surname as a link to email).

If there are some search items/view options enabled in a given configuration then a search form is generated allowing the user to specify/select searching/displaying details.

It is also possible to create basic/advanced mode view by configuring a separate list of displayed/searched items in the configuration.

Results can be sorted by selected column in both ascending as well as descending order if the field type supports this. Sorting by image for instance is not supported.

Results can be divided into pages (turned on as default) up to the search result limit configured in the directory. If the number of results exceeds the limit, an indication is displayed stating that not all possible results are displayed. In such a case sorting may be invalid.

4.3.8.2 Show

DSIFORM “Show” type allows the details of a single entry to be displayed in a simple table form where the first column contains item labels and the second column contains values.

The screenshot shows a web form titled "Contact Details". It features a "Photo" field with a small image of a man in a white shirt and tie. Below the photo is a table of contact information:

Job Title	Mr
Organization	
Department	
First Name	Eric
Surname	Perez
Display Name	Eric Perez
Email	eric.perez@acme.com
Phone (Work)	070 4166 2970
Home Phone	079 8082 1189
Mobile Phone	
Home Address	
Car	
X.509 Cert.	
V-Card	

At the bottom of the form is a "Return" button.

There is an item type allowing for the automatic generation of a link to DSIFORMs with all required details such as the DN of the requested entry and encoded return URL.

4.3.8.3 Edit

DSIFORM “Edit” type is similar to “Show” but instead of just displaying the information, it allows for the editing of the details of a single entry.

The screenshot shows a web form titled "Setup White Pages". It contains a table of contact information with input fields and green plus icons on the right side of each row, indicating editability:

Job Title	Mr	+
Department		+
Email	eric.perez@acme.com	+
Phone (Work)	070 4166 2970	+
Home Phone	079 8082 1189	+
Mobile Phone		+
Home Address		+
Car		+

At the bottom of the form are "Reset" and "Save" buttons.

4.3.9 DITBrowserServlet

`com.isode.dsinterface.DITBrowserServlet`

Universal DIT browser which displays DIT and selected entry content.

Warning: Applications using components calling pop-up DIT browser must define access to this servlet using `appname/dit` path mapping in `web.xml`.

URL parameters:

`dn`

The base DN for displayed DIT or Entry (depending on next parameter).

op

Requested operation: `show` - generates entry content (without page header/footer to be used in AHAH apps), `get` - generates single DIT node (for AHAH), `tree` - generates whole DIT (for AHAH), no `op` complete web page is generated as shown in screen-shot:

The screenshot displays the Isode Directory Services Interface - Test. The interface shows a directory tree on the left and details for the selected entry on the right. The tree structure is as follows:

- ROOT
 - c=xx
 - cn=Users
 - cn=Realms
 - cn=MailStore
 - cn=Groups
 - cn=User Managers
 - cn>Password Managers
 - cn=DSA Operators
 - cn=DSA Managers
 - cn>Data Managers
 - cn=ACI Managers

The details for the selected entry **cn=DSA Operators, cn=Groups, c=xx** are:

- hasSubordinates**
 - FALSE
- dseType**
 - { entry }
- subschemaSubentry**
 - cn=subschema-subentry,c=xx
- entryUUID**
 - 48751294-7418-4351-1034-484535253626
- modifiersName**
 - cn=Thomas Atkins,cn=Users,c=xx
- creatorsName**
 - cn=DSA Manager,cn=config
- modifyTimestamp**
 - 20130531125913.000000001Z
- createTimestamp**
 - 20130524095508.000000006Z
- member**
 - cn=Thomas Atkins,cn=Users,c=xx
 - cn=SASL Reader,cn=Users,c=xx
- description**
 - This group has permission to read the configuration of the directory server. It is suitable for operators who need this information, but do not have rights to make changes, and for automated processes monitoring the directory and requiring access to non-public configuration information.
- cn**
 - DSA Operators
- objectClass**
 - top
 - groupORNames

4.4 IMAdmin/AuditDB servlets

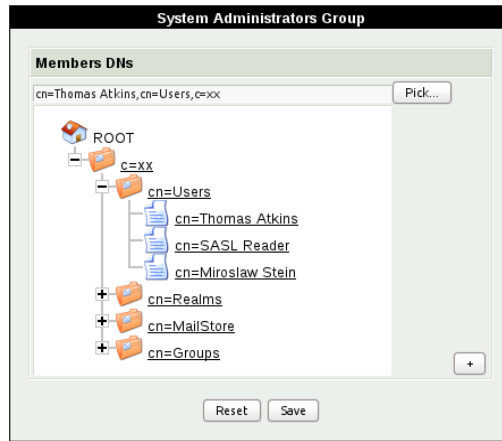
Isode IMAdmin and AuditDB web applications are used for Internet Messaging Administration. There are applications for specific user types (System, Delegated, Personal).

The AuditDB specific servlets works with data stored in audit database but use directory for it's authentication and access control. IMAdmin collects specialised directory viewing/editing servlets allowing to manage M-Switch and M-Box specific entries.

4.4.1 System Administrators Group Manager – System

```
com.isode.imadmin.system.AdminServlet
```

The System Administrators Group Manager allows to manage group of users privileged to access IMAdmin-system application.

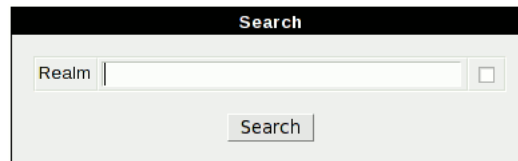


4.4.2 Administration Realm Manager – System/Delegated

`com.isode.imadmin.system.RealmsServlet`

`com.isode.imadmin.delegated.RealmsServlet`

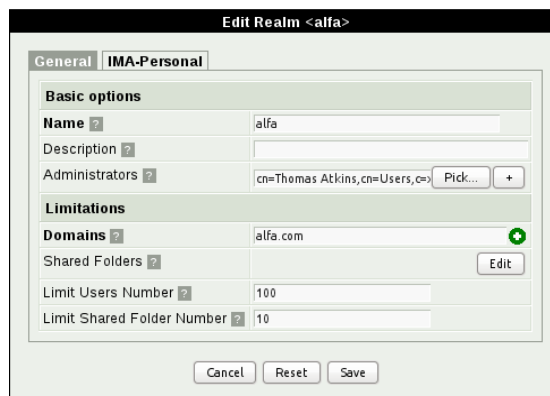
Users can be associated with an Administration Realm. A realm allows users from one or more domains to be organised into a group which can be managed by delegated administrators. Another way to use a realm is to limit IMAdmin users to see only the Contact list within their own realm.



Found 1 realm:

PAGE		<<	<	1	>	>>	ROWS / PAGE	10
		Name		Description				
1			mycompany					
Add new realm								

Both Delegated and Personal IMAdmins have some pages based on DSInterface which additionally allows for some personalization level per realm. WEB Applications Delegated Administrator/Personal allows the specification of a list of fields visible in white pages editor, contact list etc. Defaults and possible fields are defined in associated DSInterface configuration files (*imapersonlist.conf*, *imapersonshow.conf*, *imavcard.conf*).



Only the system administrator has access to all values visible in the Realm editor – some options are not visible to the Delegated administrator.

General (tab visible in System version only)**Name**

Realm name/identifier.

Description

Optional description

Administrators

Administrators of this realm (DNs)

Domains

Domains which can be used in this realm

Shared Folders

Shared folders managed by this realm.

Limit Users Number

Maximum number of users allowed in this realm. Empty means no limit.

Limit Shared Folder Number

Maximum number of shared folders allowed in this realm. Empty means no limit.

IMA-Personal setting (available both in System and Delegated version)

White Pages Fields

Comma separated list of fields editable in PIA White Pages where possible fields are: `title`, `department`, `phone`, `homephone`, `mobilephone`, `homeaddress`, `car`. An empty list means the default set.

Contacts Columns

Comma separated list of columns visible in contact search page where possible columns are: `photo`, `title`, `organization`, `department`, `firstname`, `surname`, `displayname`, `name`, `email`, `phone`, `homephone`, `mobilephone`, `car`, `certificate`, `vcard`, `details`. An empty list means the default set specified in the *imapersonlist.conf* file.

Contacts Search

Comma separated list of search fields visible in contact search page. Possible fields are the same as in “Contacts Columns”. An empty list means the default set specified in the *imapersonlist.conf* file.

Contacts Image Size

Default image size in contact search list in pixels.

Contact Details

Comma separated list of fields visible in contact details page. Possible fields are the same as in “Contacts Columns”. An empty list means the default set specified in the *imapersonshow.conf* file.

Contact Details Image Size

Image size displayed in person details page if available.

VCard Fields

Comma separated list of attributes included in generated vCards where the possible attributes are: `photo`, `title`, `organization`, `firstname`, `surname`, `email`, `phone`, `homephone`, `mobilephone`, `certificate`. An empty list means the default set specified in the *imavcard.conf* file.

4.5

M-Switch AuditDB servlets

The following web application servlets are available to browse M-Switch Audit DataBase information and manage associated services.

4.5.1 Message Tracking – System

com.isode.auditdb.web.system.MessageTrackingServlet

Search messages and display details of messages and their recipients:

More than 110 messages found

	Submit time	Queued time	Sender	Subject	Message Id	MTA	Priority	Size	State	Recipient
21	2009-04-23 00:02	2009-04-23 00:02	git-owner@vger.kernel.org	Re: [LIGIT PATCH 10/10] BROKEN TEST: ObjectLoader stays valid across repacks	<20090422230220.GQZ3R04@spearsce.org>	vger.kernel.org	Normal	2545	Delivered	brs@isode.com Original: bruce.stephens@isode.com
22	2009-04-23 00:01	2009-04-23 00:03	iesg-bounces@ietf.org	Re: Lars' Discuss on draft-ietf-idr-flow-spec	<D23A18FB42E3421CB0151FEF22573FDA6@your029b8cedce>	mailietf.org	Normal	10971	Delivered	mel@isode.com Original: alexey.meinikov@isode.com
23	2009-04-23 04:07	2009-04-23 04:09	angeredjk291@aussiedatingsearch.com	Viagra 50mg x 30 pills	<000401c0c3c09565e57086400a8c0@angeredjk291>	192.242.6.61	Normal	861	Quarantined	cjr@isode.com Original: claire.potter@isode.com
24	2009-01-12 10:47	2009-04-23 00:03	almiraschoeniger-antonuowog@gmail.com	Direkte Konferenz	<A49AD01CFAAE85F3@gmail.com>	c-98-208-154-53.hsd1.fl.comcast.net	Normal	1291	Quarantined	sek-spamtrap@isode.com Original: steve@isode.com
25	2009-04-23 13:46	2009-04-23 13:46	users-bounces+chris.ridd@isode.com@lists.opencsw.org	Re: [csw-users] Upgrade failure	<625385e30904230546n49083ae248fc6d12a0c28a88@mail.gmail.com>	mail.opencsw.org	Normal	5149	Delivered	cjr@isode.com Original: chris.ridd@isode.com
26	2009-04-23 02:03	2009-04-23 00:03	yvonneAshley_36@netzero.net	Order Controlled Meds Online	<49EFBE681004112@netzero.net>	62.47.51.94	Normal	4850	Quarantined	dbw-spamtrap@isode.com Original: bug-quipu@isode.com
									Quarantined	sek-spamtrap@isode.com Original: steve@isode.com

4.5.2 Quarantine Manager – System/Personal

com.isode.auditdb.web.system.QuarantineManagerServlet

com.isode.auditdb.web.personal.QuarantineManagerServlet

Search quarantined messages and release or delete them:

More than 110 messages found

	Queued time	Sender	Orig Sender	Subject	Recip	Orig Recip	State	State Time	Info	Score	Size
1	2009-04-29 02:09	bfjasoonalalir@gizmodo.com		More than a Free Lunch!	sjp@isode.com	Original: susan.patterson@isode.com	Quarantined	2009-04-29 02:09	INRBL score 17.62	17	4834
2	2009-04-29 02:10	jeupyaypo@macmail.com		More than a Free Lunch!	sjp@isode.com		Quarantined	2009-04-29 02:10	INRBL score 17.62	17	4393
3	2009-04-29 02:12	Nlberia@singaporeair.com		Treat her tonight to rock hard pumping	sjp@isode.com		Quarantined	2009-04-29 02:12	INRBL score 30.47	30	1106
4	2009-04-29 02:16	gcayugkbaug@bppd.com		Its action gives men desire	sjp@isode.com	Original: susan.patterson@isode.com	Quarantined	2009-04-29 02:16	INRBL score 32.29	32	1911
5	2009-04-29 02:26	bungalowqvz@musicloadz.com		Orgasms much more intense	sjp@isode.com	Original: susan.patterson@isode.com	Quarantined	2009-04-29 02:26	INRBL score 53.3	53	1860
6	2009-04-29 02:27	insistj@2cstudios.com		Are you sick because the size of your stick?	sjp@isode.com	Original: susan.patterson@isode.com	Quarantined	2009-04-29 02:27	INRBL score 44.68	44	1924
7	2009-04-29 02:27	remoteimg19@t-chris.com		Exhausted of instant headaches? Thanks to us you can forget about them.	sjp@isode.com		Quarantined	2009-04-29 02:27	INRBL score 35.86	35	20145
8	2009-04-29 02:36	ardellesiegmannebez@gmail.com		Dampfeinger perfekt www .dampf48.com	sjp@isode.com	Original: susan.patterson@isode.com	Quarantined	2009-04-29 02:36	INRBL score 5.37	5	1397
9	2009-04-29 02:37	tuproar@wikipedia.org		Feel 10 years younger in bed today	sjp@isode.com		Quarantined	2009-04-29 02:37	INRBL score -9.35	-9	1505
10	2009-04-29 02:45	migrantsgm89@trebol-producciones.com		Every woman can look like a supermodel and we can show that.	sjp@isode.com		Quarantined	2009-04-29 02:45	INRBL score 9.88	9	18862

The personal version of the Quarantine Manager is simplified and the user can only search/see/release their own messages:

Search

Received From: To:

Sender: Subject:

State:

More than 110 messages found

Received	Sender	Orig Sender	Subject	Recip	Orig Recip	Info	Score	Size	State
1	2009-05-13 18:15	mirosław.stein@isode.com Original: mst@isode.com	2 mst@isode.com	mst@isode.com	mst@isode.com	INRBL score 32.49	32	5192	Quarantined
2	2009-05-13 18:15	knighting7@north-of-boston.com	Show your partner that there still can be a lot of flame in your bed	mst@isode.com	mst@isode.com	INRBL score -13.06	-13	9083	Quarantined
3	2009-05-13 18:14	inundatedp443@merkle-korff.com	Smoking ruins! And if you prefer to remain healthy, ask us for our help.	mst@isode.com	mst@isode.com	INRBL score -15.66	-15	9092	Quarantined
4	2009-05-13 17:35	jesuitseg87@ultimatehvs.com	Acal Berry has saved lives , let it help yours get your free trial Now	mst@isode.com	mst@isode.com	INRBL score 19.5	19	1940	Quarantined
5	2009-05-13 17:33	spareibs76@studioarco.com	FW: Is your skills about to expired?	mst@isode.com	mst@isode.com	INRBL score 10.23	10	2432	Quarantined
6	2009-05-13 17:16	formlesscy61@madisonind.com	We sell the best alarm-clocks for your little fellow down there.	mst@isode.com	mst@isode.com	INRBL score -15.27	-15	9139	Quarantined
7	2009-05-13 16:57	sare00023996-mstein=esperanto.pl@z.erenewsletter.pl	Era Nowe Horyzonty - najlepsze kino niezalezne w empik.com >>	mst@isode.com	mst@isode.com	SPF_SOFTFAIL score -3.39	-3	17479	Quarantined
8	2009-05-13 16:45	roughneckj40@lou Hoover.com	Thanks to us your private life could be back on track.	mst@isode.com	mst@isode.com	INRBL score -16.92	-16	9066	Quarantined
9	2009-05-13 16:17	fdrdp0@thescotchcompany.com	Forget about sadness and be in a fantastic mood all the time.	mst@isode.com	mst@isode.com	INRBL score 1.16	1	7941	Quarantined
10	2009-05-13 16:14	pupso0@mostellarshreve.com	You have no strength any longer to satisfy your lover?	mst@isode.com	mst@isode.com	INRBL score -4.19	-4	9078	Quarantined

When user/administrator selects a message for release, the basic message details are displayed together with a query for confirmation.

Message Release

Are you sure you want to release this message?

Submitted	2009-04-29 02:40:47+01
Queued	2009-04-29 02:09:39+01
Sender	bfasoonalalir@gizmodo.com
Recipient	sjp@isode.com Original: susan.patterson@isode.com
subject	More than a Free Lunch!
size	4834

4.5.3 Statistics – System

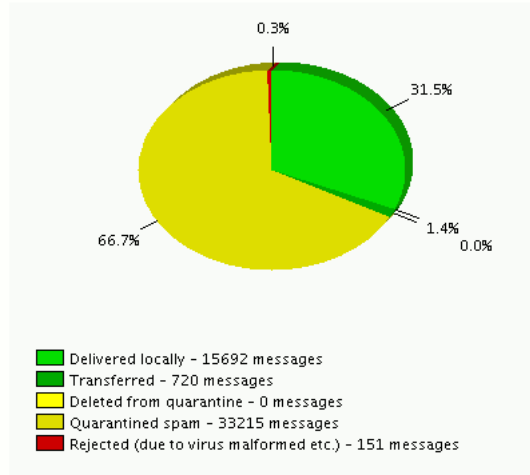
com.isode.auditdb.web.system.StatisticsServlet

Using information stored in the M-Switch Audit Database, it is possible to generate various statistical reports:

Generation Criteria

Date From	Monday	To	
Host	<input style="width: 90%;" type="text"/>		
Recip/Sender	<input style="width: 90%;" type="text"/>		

Summary chart



Summary list

State	Number of messages
Incoming Messages	26425
Outgoing Messages	49816
Processed Messages	38120
Delivered locally	15692
Transferred	720
Quarantined (spam etc.)	33215
Deleted from quarantine	0
Rejected (due to virus etc.)	151
Being processed	38

1. Summary of message delivery, rejection, queued message status
2. Summary of message counts, volumes (bytes) and message types: Period/Comparison Interval
3. Detailed information on messages. Including size and latency of: Incoming, Delivered locally, Transferred, Quarantined (spam) and Rejected messages
4. The most frequent viruses detected in messages
5. Spam score histogram
6. Messages size histogram
7. Compare Message Volume in hourly intervals over multiple days
8. Compare delivery/transfer times depending on message priority
9. Top 10 Message Originators
10. Top 10 destination MTAs
11. Top 10 transfer MTAs
12. Top 10 message recipients
13. Top 10 recipients of quarantined messages

4.6 M-Switch configuration servlets

The following web application servlets are available to manage an Isode messaging configuration.

4.6.1 White List – Personal

`com.isode.imadmin.personal.WhiteListServlet`

The white List page allows users to define their own list of email addresses which are not to be subjected to spam filtering and other content checking. Emails from such addresses will always be delivered.

4.6.2 Anti-Spam Settings – Personal

`com.isode.imadmin.personal.AntiSpamServlet`

More detailed anti-spam settings are provided with Anti-Spam Setting page:

Junk mail settings

Check Level

Alters the amount of checking performed on messages:

Default

Use the default checking.

None

Only perform anti-virus checking.

Mild

Do less checking than standard.

Standard

Do a standard amount of checking.

Aggressive

Do more checking.

Junk Action

Specifies what action should be taken for messages considered as junk:

Default

Use the default action.

Pass

Deliver the message unchanged.

Rewrite

Deliver the message, but annotated.

Redirect

Deliver to an alternative address.

Refuse

Non-deliver the message, informing the originator if required.

Discard

Discard the message silently.

Quarantine

Put the message in the quarantine store.

Junk Redirect Email

If redirect is configured above, this contains the address to redirect it to.

Quarantine specific attributes

Max Age

Maximum time to retain messages in quarantine in minutes.

Notification Email

Email address to which notification emails should be sent if different from the default.

Notification Schedule

Crontab style definitions (using time formats as given in the [Open Group Base Specifications Issue 6](http://www.opengroup.org/onlinepubs/000095399/utilities/crontab.html) [http://www.opengroup.org/onlinepubs/000095399/utilities/crontab.html]) to determine the schedule for notifications. If this is empty, then the default schedule will be used. The Isode-specific macro @never can be used to disable notifications.

4.7 M-Box configuration servlets

These servlets enable configuration of the M-Box server.

For detailed information about edited options see *M-Box Administration Guide*.

4.7.1 Directory Configuration Browser – System

`com.isode.imadmin.system.ConfigurationServlet`

Browse the current directory configuration.

4.7.2 Message Store Configuration – System

`com.isode.imadmin.system.SuiteServlet`

Create/Edit the message store configuration, including message size quota, delivery options.

4.7.3 Shared Folders Root Manager – System

`com.isode.imadmin.system.SharedServlet`

Create, manage and delete access permissions for shared email folders.

4.7.4 User Manager – System/Delegated

`com.isode.imadmin.system.UsersServlet`

`com.isode.imadmin.delegated.UsersServlet`

Create, manage and delete user email accounts.

4.7.5 Group Manager – System

`com.isode.imadmin.system.GroupsServlet`

Create, manage and delete user groups.

4.7.6 Delivery Rules – Personal

`com.isode.imadmin.personal.SieveScriptServlet`

This allows the configuration of vacation notices and filtering messages into folders.

Chapter 5 DSIForm and vCard configuration

This chapter describes DSIForm and vCard configuration.

DSIForm servlets and vCard servlets share the same configuration format. By default all DSIForm configurations are stored in a common directory as text files with a *.conf* extension. Configuration file names (without an extension) are used in the URL to specify which form definitions should be used.

5.1 General syntax

DSIForm configuration consists of a set of *key=value* lines specifying different aspects of the form.

5.1.1 Keys

Some keys are predefined and others can be defined by the user.

Such keys can be complex (they can comprise several parts separated by .)

```
[groupKey. ]+key=value
```

Example (items displayed in Contact List):

```
pageTitle=Contact Details
```

The order of the keys in the configuration is not significant. Key names are not case sensitive, and cannot contain white space. White space around the = is ignored.

The *value* can contain multiple values separated by , in which case the ordering of the values is significant.

Using prefix group keys it is possible to store multiple DSIForm configurations in a single file or directory entry attribute.

5.1.2 Values

Sometimes a definition contains a fixed number of parameters and the last parameter can specify a string or DN (which can include commas). The rest of the line is simply treated as part of the last parameter.

For example for one of search base DN definitions:

```
baseDN.baseDNID=name , DN
```

can be specified as:

```
baseDN.main=Main base DN,cn=Staff, o=My Organisation
```

so everything after first comma is treated as the second parameter *DN*.

5.1.3 Comments

Configuration files support comments. All lines beginning with # are ignored.

5.2 Extending / Modifying existing configuration

In simple cases all DSIForm details can be defined in a single file.

Form configurations can be created as an extension or modification of an existing form.

Often multiple forms have common values, for example the “Users List” can display some user information from a set displayed in the “User Details” page. To avoid duplication, you can create multiple configuration files and merge them together in the DSIForm configuration.

If you have a number of similar pages with minor differences between them, then you could create one basic page (DSIForm) configuration and then create additional configurations which contain references to the basic page and which include information about any differences.

5.2.1 BaseConfig option

The `BaseConfig` option allows the specification of a configuration to be used as the base for this configuration. The list order is significant as it influences the order and priority of modifications.

Syntax:

```
baseConfig=configID[,configID]+
```

For example:

```
baseConfig=common,modelForm
```

Initially, the `common` configuration will be used. New options from the `modelForm` configuration are added while options present in both will take the definitions from `modelForm`. The new configuration option will extend and modify the configuration.

5.3 Basic page options (DSIForm only)

5.3.1 Type option

DSIForm type - currently there are 4 possible form types which can be generated:

Syntax:

```
type=[list|show|edit|vcard]
```

Example for a “List” DSIForm:

```
type=list
```

If the configuration doesn’t specify a valid type it cannot be used directly to generate a DSIForm page. In this case it can only be used as a base for other configurations, for example to specify some common definitions for multiple forms.

The specification can also be inherited from one of the base configurations.

5.3.2 PageTitle option

The page title is mandatory for all DSIForm types except vCard.

Syntax:

```
pageTitle=page title
```

pageTitle

The following string is the page title.

Example:

```
pageTitle=Contact Details
```

5.3.3 EntryNames option

EntryNames option is useful only in the “List” DSIForm and allows for the specification of the singular and plural names of entries displayed in the results table (depending on number of search results).

```
entryNames=singular name, plural name
```

For example, in a Contacts List:

```
entryNames=Contact,Contacts
```

This is also used to display the summary of a search result (“Found 6 Contacts”).

5.4 Access limit

Access to the DSIForm page may be restricted.

5.4.1 AccessLimit.* definitions

Currently it is not possible to define custom access limitations. The provided set of predefined limitations are:

`accessLimit.noAccess`

Access turned off completely - useful for base configurations such as `defaultuser.conf` which we want to prevent from being displayed.

`accessLimit.ownEntryOnly`

Access by logged user to his/her own entry only.

`accessLimit.imaRealmSpecified`

Access to data associated with one of the IMA realms.

5.4.2 AccessLimits option

Limitations can be enabled by specifying them in the list:

```
accessLimits=accessLimitID[ ,accessLimitID]+
```

Example:

```
accessLimits=ownEntryOnly
```

5.5 Search base DN

Note: Search options can only be used in the “List” DSIForm.

Directory search operation is always performed relative to a base DN. DSIForm allows for the use of multiple base DNs. For security reasons it does not allow pointing to the base DN directly in the URL but all acceptable base DNs have to be defined in the configuration and assigned an ID. Then in URL base DN can be safely pointed to by its ID.

5.5.1 BaseDN.* definitions

Warning: At least one definition is required by the “List” DSIForm.

Base DN definition syntax:

```
baseDN.baseDNID=name ,DN
```

`baseDNID`

Unique ID of the base DN.

`name`

User friendly name for base DN which can be displayed in search form’s base DN selector (if required).

`DN`

Just a DN. It is also possible to use the following variables instead of a real DN specification.

Example:

```
baseDN.sasldn = SASL, $SaslBaseDN
baseDN.mydn = My Base DN, o=My Organisation,c=GB
```


5.5.2 BaseDNs option

Warning: This option is mandatory in the “List” DSIForm.

Not all defined base DN's have to be used in the DSIForm, those which are used need to be listed:

Syntax:

```
baseDNs=baseDNID[ ,baseDNID ]+
```

If there is only one `baseDNID` specified in the list it is just used. When multiple `baseDNID`s are specified an appropriate option in a search box appears which allows users to select one of them.

Example (including definitions):

```
baseDN.saslcn = SASL, $SaslBaseDN
baseDN.mydn = My Base DN, o=My Organisation,c=GB
baseDNs = saslcn, mydn
```

5.6 Search filtering

While the base DN allows you to specify where you want to search, filtering options refine the search further.

5.6.1 BaseFilter option

Warning: This option is mandatory in the “List” DSIForm. This option is deprecated. In the next release it will be replaced by the `Filter.*` definition.

`BaseFilter` stands for basic filtering in the “List” DSIForm. It is always used unconditionally.

Syntax:

```
baseFilter=LDAP filter
```

Example for a person list:

```
baseFilter=(objectclass=person)
```

5.6.2 Filter.* definition

Similar to `baseDN`, DSIForm doesn't accept filters provided directly (as a URL parameter for example). It is necessary to define LDAP filters and associate them with a filter ID which can be used later.

Predefined filters:

`filter.imaRealmMember`

Filter limiting searched entries to user entries which are associated with the IMAdmin realm of which the currently logged user is a member.

`filter.imaRealmAdmin`

Filter limiting searched entries to user entries which are associated with the IMAdmin realm where the currently logged user has delegated administration rights.

It is possible to define customised filters. These filters are used in the “List” DSIForm to automatically generate the option allowing the filter to be turned on/off.

Syntax:

```
filter.filterID= static:LDAP filter
```

Example:

```
filter.myFilter=static:(uid=abc)
```

5.6.3 Filters, filtersAdvanced options

To enable defined filters for use in basic and advanced mode there are `filters` and `filtersAdvanced` options.

Syntax:

```
filters=filterID[ ,filterID]+
filtersAdvanced=filterID[ ,filterID]+
```

Example:

```
filters=myFilter1,myFilter2
```

5.7 Items

DSIForm items stand for elements we would like to search, display or edit.

It is possible to define multiple types of items. Basic items refer to single entry attributes. We can also define displayable items which will be generated from multiple attributes, will generate links to other forms, etc.

The same item definitions can also be used to generate a search form allowing the modification of search filtering or corresponding editing field when it make sense for the specific item. For example it doesn't make sense to search by image.

The following item types descriptions will have information about usage scope:

1. Display – Item can be used in “List” and “Show” DSIForms to display specific values
2. Edit – Item can be used in “Edit” DSIForm to edit the associated directory's entry attribute
3. Search – Item allows to generate search option in “List” DSIForm.

5.7.1 **item.* definitions**

Basic syntax:

```
item.itemID=itemType:itemDetails
```

5.7.1.1 **Image**

Display image stored in entry attribute

Usage: display.

Syntax:

```
image:title[ ,attribute=jpegPhoto]
```

Displayable – displays image

5.7.1.2 **Text**

Attribute value treated as text.

Usage: display, edit, search.

Syntax:

```
text:title,attribute
```

5.7.1.3 **PersonNameEmail**

PersonNameEmail - person's name and surname (**givenName** and **sn** attributes) as an email link (**mail** attribute).

Usage: display.

Syntax:

```
personnameemail:title
```

5.7.1.4 **Email**

Attribute value treated as an email (displayed as an email link).

Usage: display, edit, search.

Syntax:

```
email:title,attribute
```

5.7.1.5 **Presence**

Presence item allows to recognize specific attribute value presence.

This item when displayed generates graphical indicator. When used to generate search option – generates selector for exist/not exist/all values.

Usage: display, search.

Syntax:

```
presence:title,attribute
```

5.7.1.6 vCard, show, edit

These items generate appropriate images and links to other DSIForms with given configurations using current entry DN

Usage: display.

Syntax:

```
vcard:title[,config=defaultVcard]
show:title,config
edit:title,config
```

5.7.1.7 DNList

Generates embedded table list view with entries specified by DN list stored in given attributes and columns specified by following items list

Usage: display.

Syntax:

```
dnlist:title,attribute,itemID[,itemID]+
```

5.7.2 items, itemsAdvanced options

In any given configuration many items can be defined; not all have to be used. To specify which items should be used in basic and advanced mode we create `items` and `itemsAdvanced` specifications as a comma separated list. The order of items in these specifications will be used later as an order for displaying items on the specified DSIForm. The items list is always mandatory for all page types.

In “List” and “Show” items are displayed, in “Edit” items are editable, in “vCard” items are included.

Syntax:

```
items=itemID[,itemID]+
itemsAdvanced=itemID[,itemID]+
```

5.7.3 vCard items

Warning: The vCard generator shares the DSIForm configuration format, however vCards can contain only specific values. It is not possible to define your own items and only predefined sets of items can be enabled.

Supported items are: surname, firstname, organization, phone, homephone, mobilephone, email, title, photo and certificate.

5.7.4 Search, searchAdvanced options

The “List” DSIForm generates a search form where selected items can be used to generate corresponding search fields/selectors.

Syntax:

```
search=itemID[ , itemID]+
      searchAdvanced=itemID[ , itemID]+
```

5.7.5 ObjectTypes option

In editing mode the `objectTypes` option should contain a list of all the object types used by enabled editable items (as some entries may require completing object types).

Syntax:

```
objectTypes=objectTypeName[ , objectTypeName ]+
```

5.8 Display options

Both “List” and “Show” DSIForms allow additional options to be specified which can modify some presentation details.

5.8.1 Option.* definitions

General syntax:

```
option.optionID=optionType:optionDetails
```

Predefined options:

`option.pagerOff`

Turn off displaying results divided into pages so that everything will be displayed in one long page.

5.8.1.1 ImageSizeDefault

Specify the default image size for given image type item.

Syntax:

```
imageSizeDefault:itemID, size
```

5.8.1.2 ImageSizeSelector

Generate an image size selector in the search box.

Syntax:

```
imageSizeSelector:itemID, label, size[ , size ]+
```

5.8.2 Options, optionsAdvanced selector

List of enabled options in basic and advanced mode.

Syntax:

```
options=optionID[,optionID]+  
optionsAdvanced=optionID[,optionID]+
```

5.8.3 defaultOrderBy option

Default result sorting order in List mode. Sorting is always enabled by the specified `itemID` (one of the column). If the `_` character is added directly after `itemID`, sorting is applied in descending order.

Syntax:

```
defaultOrderBy=itemID[_]
```

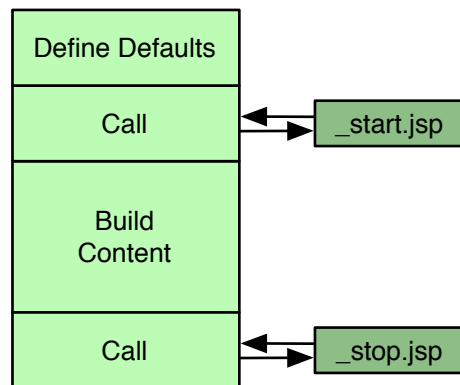
Chapter 6 Customization and branding

This chapter describes how to customize and brand the existing applications.

All the servlets provided by Isode are collected in shared libraries which cannot be modified.

6.1 Page template

To allow simple customization and branding, each servlet calls external **.jsp* files which contain a page template.



Initially when the servlet is called it prepares everything required to generate a web page including some variables with default values which can be used both by the servlet itself and the page template, such as the page title.

The page template is split into two parts: *_start.jsp* and *_stop.jsp*. Each part is called at the beginning and end of page content which allows for the separate generation of the page header with a title, menu etc. and the page footer.

Generally both files contain HTML for header and footer of the page, but because they are JSP files they can also contain scripting commands.

The appearance of the content generated by the servlets can be modified using a stylesheet referenced from the page template header.

Communication between the servlet and the template files is performed through HTTP request attributes. The most essential attribute is `page_title` which can be set by inserting the following in the HTML code:

```
<%= (String)request.getAttribute("page_title") %>
```

Isode does not provide documentation relating to the editing or configuration of Java Server Pages. Documentation and tutorials relating to this subject are widely available on the Internet.

Alternatively you can experiment by modifying a copy of the existing template – which contains all of the required elements.

6.2 Creating your own applications

To create your own web applications using the Isode libraries and interfaces, please contact Isode support at support@isode.com.