HARRIERWEB-3.3

Harrier Web Server Administration Guide

Isode

Table of Contents

1 Software version

This guide is published in support of Isode Harrier Web R3.3. 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 configure Harrier Web, a server application which provides a web-browser interface for clients wishing to use Military Messaging or Internet Mail.

3 Related publications

A separate guide is published for Users of the Harrier client in order to send and receive messages. See the Harrier Web Server User's Guide.

Related topics are discussed in the volumes of the Isode documentation set for other servers which are listed below.

Volume	Title
SWADM	*M-Switch Administration Guide*
SWAAG	*M-Switch Advanced Administration Guide*
SWOPG	*M-Switch Operators Guide*
VAUADM	*M-Vault Administration Guide*
MBOXADM	*M-Box Administration Guide*

4 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	*isoentities*
Program and macro names	mkpasswd
Input to the system	`cd newdir`
Cross references	see Section 5, “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.

5 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
(BINDIR)	Programs run by users.	*C:\Program Files\Isode \Harrier\bin*	*/opt/isode/bin*
(CACHEDIR)	Cache files.	*C:\Isode\Harrier\cache*	*/var/cache/isode/harrier*
(CLIENTDIR)	Default Harrier client files	*C:\Program Files\Isode \Harrier\share\webapps \harrier*	*/opt/isode/share/webapps/ harrier*
(DATADIR)	Storing local data.	*C:\Isode\Harrier*	*/var/isode/harrier*
(ETCDIR)	System-specific configuration files.	*C:\Isode\Harrier\etc*	*/etc/isode/harrier*
(LIBDIR)	Libraries.	*C:\Program Files\Isode \Harrier\bin*	*/opt/isode/harrier/lib*
(LOGDIR)	Log files.	*C:\Isode\Harrier\log*	*/var/log/isode/harrier*
(SBINDIR)	Programs run by the system administrators.	*C:\Program Files\Isode \Harrier\bin*	*/opt/isode/sbin*
(SHAREDIR)	Configuration files that may be shared between systems. Common data and documentation files.	*C:\Program Files\Isode \Harrier\share*	*/opt/isode/share*
(TMPDIR)	Temporary files.	*C:\Isode\Harrier\tmp*	*/var/tmp/isode/harrier*

6 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/subscribe.html

7 Export controls

Many Isode products use TLS (Transport Layer Security) to encrypt data in transit. This means that these products are subject to UK Export Controls.

For some countries (at the time of shipping this release, these comprise all EU countries, United States of America, Canada, Australia, New Zealand, Switzerland, Norway, Japan), these Export Controls can be handled by administrative process as part of evaluation or purchase. For other countries, a special Export License is required. This can be applied for only in context of a purchase order for those Isode products.

You must ensure that you comply with these Export Controls where applicable, i.e. if you are licensing or re-selling Isode products.

The TLS feature of Isode products is enabled by a TLS Product Activation feature. This feature may be turned off, and Isode products without this TLS feature are not export controlled. This can be helpful to support evaluation of Isode products in countries that need a special export license.

Isode products are used to administer sensitive data and so Isode strongly recommends that all operational deployments of Isode products use the export-controlled TLS feature.

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

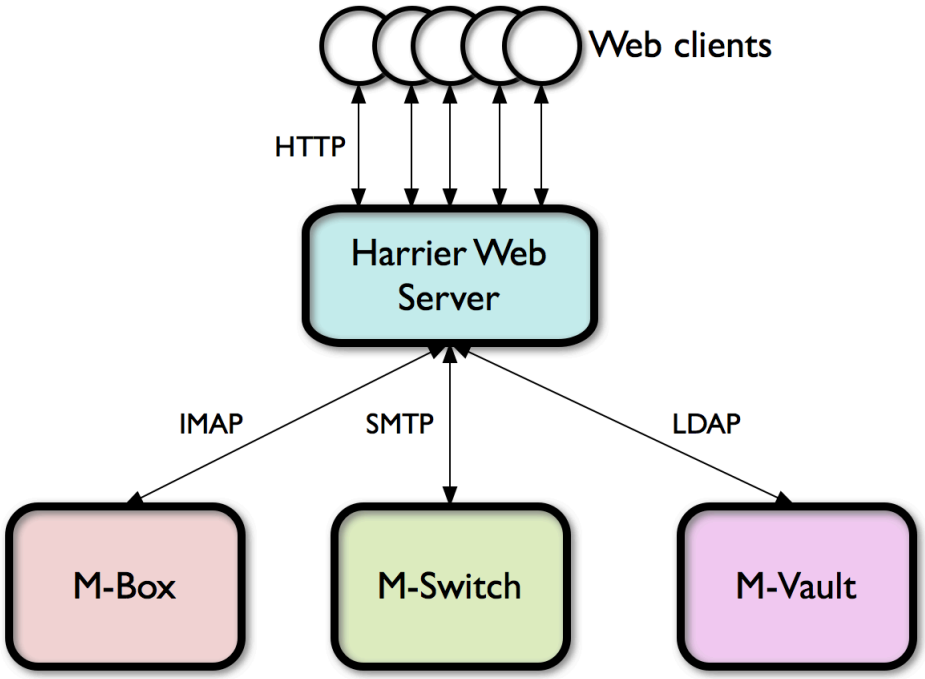
Chapter 1 Introduction to Harrier Web

This section introduces Harrier Web server and talks about how its configuration is stored.

1.1 Overview

The Isode Web Email server, Harrier, provides a zero-footprint web mail client that allows users to access email. Harrier Web Server uses standards-based technologies including HTTP, IMAP, SMTP and LDAP

Figure 1.1. Web clients accessing mail via Harrier Web Server



The Harrier Web Server establishes connections to IMAP, SMTP and LDAP servers on behalf of individual users, who need only supply a single set of login credentials in order to be able to send and read email, and to access an address book.

Chapter 2 Harrier Server Configuration

This section provides a detailed description of Harrier server configuration options.

Harrier Web Server reads its configuration from an XML file located in *(ETCDIR)/harrier_web_conf.xml*. This chapter describes the options that are available in the configuration file and how to make changes to them so that the Harrier Web Server will use them.

2.1 Initial configuration

This section describes initial configuration of Harrier Web Server. Please read this section to find out about all configuration steps required before starting Harrier Web Server for the first time.

This section also provides basic information about services included in Harrier Web Server and describes how to start/stop Harrier Web Server services once the initial configuration is complete.

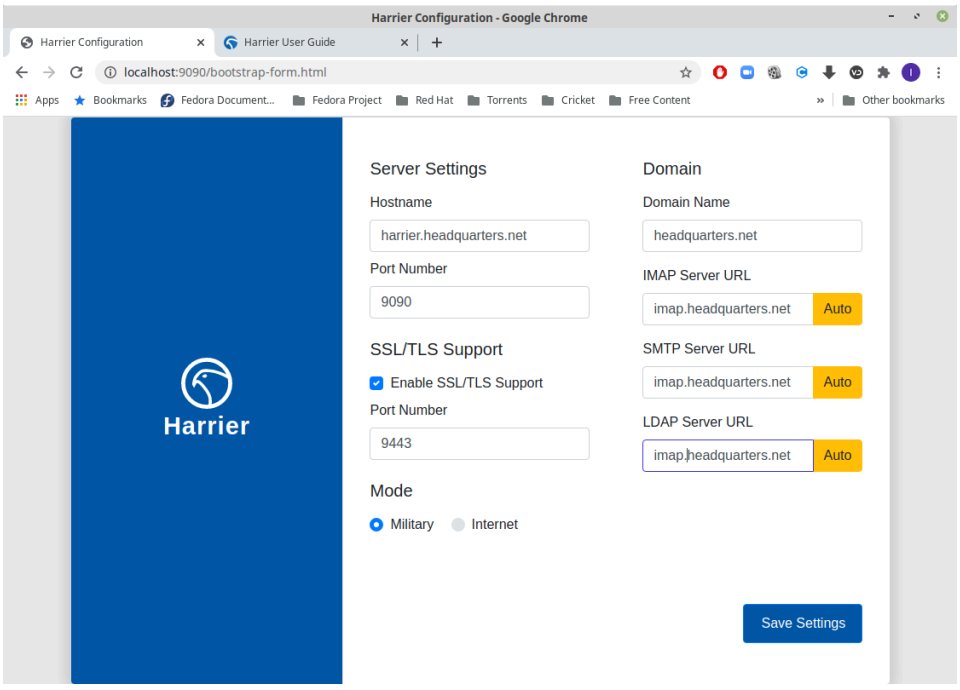
2.1.1 Harrier Web Server configuration file

Harrier Web Server reads its configuration from *(ETCDIR)/harrier_web_conf.xml*. A sample configuration file is provided in *(ETCDIR)/harrier_web_conf.xml.sample*. It may be useful to use the contents of this file as a starting point, by copying it to *(ETCDIR)/harrier_web_conf.xml* and then editing it before starting the Harrier Web Server.

2.1.2 Harrier Web Server Bootstrap

When Harrier Web Server starts up and it is unable to read its configuration from *(ETCDIR) /harrier_web_conf.xml* it enters bootstrap mode. Opening web browser for `http://localhost:9090` presents simple wizard allowing to create minimal configuration sufficient for simple evaluation or just a test of proper installation.

Figure 2.1. Harrier Web Server Bootstrap Page



Enabling SSL/TLS support allows to encrypt configuration file and use self-signed certificate generated on server start with new configuration file (backup of preexisting certificates is supported).

The **Auto** buttons next to URL fields allows to populate them with sample value based on domain name.

Pressing **Save Settings** tests provided values and if everything is OK configuration file will be created.

2.1.2.1 Product Activation

> **Note:** The previous licensing system used by earlier versions of Harrier has been superseded by Product Activation which works in a different way.

You will need a product activation key in order to run Harrier Web Server.

To activate Harrier Web Server please contact support@isode.com for the Messaging Activation Server. When you connect to the Messaging Activation Server, you will be taken through the steps to generate a product activation request key, which should be sent to Isode (support@isode.com). Other Isode products may require separate activation.

2.1.3 Harrier Web Server runtime user

On UNIX systems, you should create a runtime user (for example using **useradd** on Linux) to be used for the Harrier Web Server. The name of this user is then set in the configuration file (see Section 2.4.3.4, “Runtime User”).

2.2 Installing an Isode activation file

In order to create or start a Harrier instance on a host system, the activation file for the respective product is required.

Isode MAS web application manages activation keys on a host system. Each system hosting a product component will have its own set of activation files.

Questions regarding licensing should be directed to licensing@isode.com.

2.3 Starting and stopping Harrier

2.3.1 Harrier process

Harrier includes a single process isode.harrierwebserver. Subsequent references will use a shortened form "harrierwebserver". This section summarizes installation of Harrier service and how to start it on different platforms.

2.3.2 Starting/stopping Harrier on Linux

On Linux Harrier Web Server is managed as any regular daemon using systemd commands

- To start Harrier run: **sudo systemctl start harrier**

- To stop it run: **sudo systemctl stop harrier**
- To check if Harrier server is running: **sudo systemctl status harrier**

2.3.3 Starting/stopping Harrier on Windows

On Windows Harrier server process is installed as a Windows service. This process runs under the Local System account.

By default Harrier service is set for **Automatic start** at system startup.

In order to stop or start Harrier manually use standard Windows Services manager (You can also use Isode Service Manager if installed separately with other Isode packages).

2.4 Server configuration options

The Harrier Web Server configuration file *(ETCDIR)/harrier_web_conf.xml* contains an XML document. The top level XML element is `harrier`. Each configuration option is represented as an XML element.

2.4.1 Configuration file structure

The configuration file has general options located in elements under the root element. As well as general options, other tags contain groups of options:

- The `<domains>` element contains information about the domains which are supported by the Harrier Web Server.

2.4.2 Configuration file variables

Wherever file paths are specified inside the configuration file, you can use one of several pre-defined placeholders rather than entering an absolute file path. These all have the form `$(`*name*`)`. You can use these placeholders wherever a file path is expected in the configuration file.

Name	Description
`$(harrier.dir.cache)`	(CACHEDIR) file system placeholder
`$(harrier.dir.client)`	(CLIENTDIR) file system placeholder
`$(harrier.dir.etc)`	(ETCDIR) file system placeholder
`$(harrier.dir.lib)`	(LIBDIR) file system placeholder
`$(harrier.dir.log)`	(LOGDIR) file system placeholder
`$(harrier.dir.share)`	(SHAREDIR) file system placeholder
`$(harrier.dir.tmp)`	(TMPDIR) file system placeholder
`$(harrier.dir.var)`	(DATADIR) file system placeholder
`$(harrier.dir.etc_or_share)`	First checks (ETCDIR) then (SHAREDIR)
`$(harrier.dir.app)`	Directory where Harrier Web Server executable is located
`$(harrier.dir.cfg)`	Directory where current configuration file is located

2.4.3 General Options

This section describes configuration options that can be configured directly under the root level element harrier

2.4.3.1 servpass:info

Description: the service name to be used to identify the *servpass* key that is used to obfuscate sensitive data in this file. This option is empty by default, which means that sensitive data will not be obfuscated.

The value in this option will be used by the `spasscrypt` utility to obfuscate fields in the file which have been identified as sensitive. See Section 2.8, “Storing passwords”.

Default value: -None-

XML element name: `servpass:info`

Parent XML element: top level `<harrier>`

Example: `<servpass:info service="isode.harrier"/>`

2.4.3.2 Name

Description: Server name used in various places (user agent name, HTTP responses etc...).

Default value: Isode Harrier Web Server

XML element name: `name`

Parent XML element: top level `<harrier>`

Example: `<name>Isode Harrier Web Server</name>`

2.4.3.3 Host

Description: Specifies text to be used when generating a unique string used for the `message-id` header for messages that have been composed within Harrier. If this is not specified, then the a value is derived from the hostname in the URL that the user supplied to connect to Harrier. Setting this option avoids exposing this hostname in the message header.

A value for `host` is required when a self-signed certificate is to be generated for the Harrier Web Server (see Section 2.4.6.4, “Key”).

Default value: -None-

XML element name: `host`

Parent XML element: top level `<harrier>`

Example: `<host>example.com</host>`

2.4.3.4 Runtime User

Description: Specifies the OS user that Harrier Web Server will run as. This option only applies on Linux systems. To use this option the server will need to be started as the root user: it will bind to any ports (which may include privileged ports) before dropping privileges to run as the specified runtime user. Harrier Web Server is also able to bind to a privileged port before dropping user privileges.

Default value: -None-

XML element name: `runtime_user`

Parent XML element: top level `<harrier>`

Example: `<runtime_user>harrier</runtime_user>`

2.4.3.5 Shutdown watchdog timeout

Description: The shutdown watchdog allows to enforce server shutdown after specific timeout (period of graceful shutdown attempt).

0 - means watchdog disabled.

Notice: If set it should be longer than `ws_shutdown_timeout` and `http_shutdown_timeout`.

Default value: 5s

XML element name: `shutdown_watchdog_timeout`

Parent XML element: top level `<harrier>`

Example: `<shutdown_watchdog_timeout>5s</shutdown_watchdog_timeout>`

2.4.3.6 Default Domain

Description: Specifies default domain name which allows to login with user name only (The default domain is being added to name to determine full user login).

Default value: -None-

XML element name: `default_domain`

Parent XML element: top level `<harrier>`

Example: `<default_domain>example.com</default_domain>`

2.4.4 HTTP server

This section describes configuration options used to control/override the behaviour of Harrier Web Server when acting as an HTTP server. Specifically, the `<http_map>` may be used to alter the path used to locate resources that may be used by the client, as described below.

2.4.4.1 Listen Addresses

Description: The `port` option can be used to specify that the server should listen for incoming HTTP/WS requests using the specified port on all available network addresses. The `listen` option provides more fine-grained control over which network addresses/ ports should be used.

The HTTPS/WSS acceptance is defined separately (See Section 2.4.6.2, “Listen Addresses”).

If only address is specified it uses port number from `port` option..

Default value: "::" which means all IPv6 and IPv4 addresses

XML element name: `listen`

Parent XML element: top level `<harrier>`

Example:

```
<!-- listen on 127.0.0.1 interface and port -
     as defined in port option -->
<listen>127.0.0.1</listen>

<!-- listen on 192.168.0.100 interface and port 9999 -->
<listen>192.168.0.100:9999</listen>

<!-- listen on IPv6 localhost interface and port -
```

```
     as defined in port option -->
<listen>::1</listen>

<!-- listen on IPv6 fe80::9ada:8bdb:9871:6d1b interface -
     and port 9999 -->
<listen>[fe80::9ada:8bdb:9871:6d1b]:9999</listen>
```

2.4.4.2 Port

Description: Specifies the default port that Harrier Web Server will listen on for incoming HTTP/WS connections (used by `listen` option)

Default value: 9090

XML element name: `port`

Parent XML element: top level `<harrier>`

Example: `<port>9009</port>`

2.4.4.3 URL mapping

Description: Specifies the location in the local filesystem which can be used to resolve URLs from the client. This option takes following parameters:

- `url` specifies a URL (or the start of a URL) in the client's request
- `pattern` used instead of `url` in more complex cases and specifies a URL regular expression pattern in the client's request
- `path` specifies the path on the local filesystem to be used to resolve the specified URL.

The default configuration contains a definition for `url="/"` which should not be modified. Other urls that may be modified are:

- `/data/mmhs-types.xml` is used to locate the catalog of MMHS types which are presented to the user when composing a message. The default location for this file is `$(harrier.dir.share)/webapps/harrier/data/mmhs-types.xml`

XML element name: `http_map`

Parent XML element: top level `<harrier>`

Example: `<http_map url="/data/mmhs-types.xml" path="/usr/local/my-mmhs-types.xml"/>`

2.4.4.4 HTTP shutdown timeout

Description: Timeout for graceful HTTP connection shutdown. (0 means disabled)

Notice: If set it should be shorter than shutdown_watchdog_timeout.

Default value: 3s

XML element name: `http_shutdown_timeout`

Parent XML element: top level `<harrier>`

Example: `<http_shutdown_timeout>3s</http_shutdown_timeout>`

2.4.4.5 Control for which origins can call JavaScript APIs on the website

Description: Default value of the Access-Control-Allow-Origin header field. Should be one of `*`, `null` or <origin> (e.g. "https://example.org"). Special value `$1` can be used as

an <origin> constructed from the Host request header field. (The last option is useful if the Web Server serves multiple web domains.)

Default value: $1

XML element name: `access_control_allow_origin`

Parent XML element: top level `<harrier>`

Example: `<access_control_allow_origin>$1</access_control_allow_origin>`

2.4.5 Websocket

This section describes configuration options specific to websocket connections.

2.4.5.1 Websocket buffer size

Description: Specifies the buffer size to be used for websocket connections. This setting is reserved for use by Isode.

Default value: 64K

XML element name: `ws_buffer_size`

Parent XML element: top level `<harrier>`

Example: `<ws_buffer_size>64K</ws_buffer_size>`

2.4.5.2 Websocket permessage-deflate extension

Description: Websocket permessage-deflate extension (enables compressed transmission).

Default value: on

XML element name: `ws_deflate`

Parent XML element: top level `<harrier>`

Example: `<ws_deflate>on</ws_deflate>`

2.4.5.3 Websocket handshake timeout

Description: Websocket handshake timeout (0 means disabled).

Default value: 5s

XML element name: `ws_handshake_timeout`

Parent XML element: top level `<harrier>`

Example: `<ws_handshake_timeout>5s</ws_handshake_timeout>`

2.4.5.4 Websocket inactivity timeout

Description: Websocket inactivity timeout - triggers ping to check if client connection is alive and if there is no response connection is being closed. (0 means disabled)

Default value: 30s

XML element name: `ws_inactivity_timeout`

Parent XML element: top level `<harrier>`

Example: `<ws_inactivity_timeout>30s</ws_inactivity_timeout>`

2.4.5.5 Websocket shutdown timeout

Description: Timeout for graceful Websocket connection shutdown. (0 means disabled)

Notice: If set it should be shorter than shutdown_watchdog_timeout.

Default value: 3s

XML element name: `ws_shutdown_timeout`

Parent XML element: top level `<harrier>`

Example: `<ws_shutdown_timeout>3s</ws_shutdown_timeout>`

2.4.6 SSL

This section describes configuration options relating to SSL configuration for incoming HTTPS connections. SSL configuration for outgoing IMAP, SMTP and LDAP connections is configured in the domains section (see Section 2.5, “Domains”).

2.4.6.1 Port

Description: Specifies the default port that Harrier Web Server will listen on for incoming HTTPS/WSS connections (used by `ssl_listen` option)

Default value: 9443

XML element name: `ssl_port`

Parent XML element: top level `<harrier>`

Example: `<ssl_port>9003</ssl_port>`

2.4.6.2 Listen Addresses

Description: The `ssl_port` option can be used to specify that the server should listen for incoming HTTPS/WSS requests using the specified port on all available network addresses. The `ssl_listen` option provides more fine-grained control over which network addresses/ports should be used.

The HTTP/WS acceptance is defined separately (See Section 2.4.4.1, “Listen Addresses”).

If only address is specified it uses port number from `ssl_port` option..

Default value: "::" which means all IPv6 and IPv4 addresses

XML element name: `ssl_listen`

Parent XML element: top level `<harrier>`

Example:

```
<!-- listen on 127.0.0.1 interface and port as defined in ssl_port option -->
<ssl_listen>127.0.0.1</ssl_listen>
<!-- listen on 192.168.0.100 interface and port 9443 -->
<ssl_listen>192.168.0.100:9443</ssl_listen>
<!-- listen on IPv6 localhost interface and port as defined in ssl_port option -->
<ssl_listen>::1</ssl_listen>
<!-- listen on IPv6 fe80::9ada:8bdb:9871:6d1b interface and port 9443 -->
<ssl_listen>[fe80::9ada:8bdb:9871:6d1b]:9443</ssl_listen>
```

2.4.6.3 Certificate

Description: Specifies the full path to a file containing the Harrier Web server's own TLS certificate. This certificate will be sent by the server to any client connecting using HTTPS, and may be used by the client browser to confirm the server's identity when negotiating secure communication.

The certificate format must be in PEM or PKCS#12 format. The format is determined from the file extension, which is "pem" and "p12" respectively.

The value "$(harrier.tlscert)" can be used to reference the default certificate. It expands to "$(harrier.dir.etc)harriercert.pem". This option is ignored if the value of the `ssl_key` XML option is a PKCS#12 file, in which case the PKCS#12 file is assumed to contain both the private key and the corresponding certificate.

Default value: -None-

XML element name: `ssl_certificate`

Parent XML element: top level `<harrier>`

Example: `<ssl_certificate>$(harrier.dir.etc)harrier-cert.pem</ssl_certificate>`

2.4.6.4 Key

Description: specifies the full path to a file containing the private key belonging to the server TLS certificate.

If this option is not set or empty the HTTPS support is disabled and other related options are ignored.

The private key must be in PEM or PKCS#12 format. The format is determined from the file extension, which is "pem" and "p12" respectively.

The value "$(harrier.tlskey)" can be used to reference the default private key. It expands to "$(harrier.dir.etc)harrierkey.pem", if the file exists, otherwise it expands to "$(harrier.dir.etc)harrier.p12".

If `host` is defined, and `ssl_key` is set to $(harrier.tlskey) but the PCKS#12 file identified by $(harrier.tlskey) does not exist, then the Harrier Web Server will generate a self signed certificate and key, and store them in $(harrier.tlskey), and then update the configuration file with the passphrase for the generate PKCS#12 file. The passphrase will be obfuscated using servpass if that has been configured.

Default value: -None-

XML element name: `ssl_key`

Parent XML element: top level `<harrier>`

Example: `<ssl_key>$(harrier.dir.etc)harrier-key.pem</ssl_key>`

2.4.6.5 Password

Description: Specifies the password used to decrypt the server's private key. This option is empty by default, which means that the private key is not protected by any password.

The value in this option should be encrypted using the `servpass` utility (see Section 2.8, “Storing passwords”.

Default value: -None-

XML element name: `ssl_password`

Parent XML element: top level `<harrier>`

Example: `<ssl_password servpass:encrypt="true">secret</ssl_password>`

2.4.6.6 Certificate Chain

Description: You can specify zero or more `<ssl_chain_certificate>` options, each of which should contain the path to a file which contains the PEM representation of a single certificate. When the server performs a TLS handshake in response to an HTTPS connection, it sends a certificate chain consisting of its own `<ssl_certificate>` and all of the `<ssl_chain_certificate>` certificates which have been specified. The order of `<ssl_chain_certificate>` options is significant: the first should be the issuer of `<ssl_certificate>`, the next should be the issuer of that certificate, etc.

Default value: -None-

XML element name: `ssl_chain_certificate`

Parent XML element: top level `<harrier>`

Example: `<ssl_chain_certificate>$(harrier.dir.etc)cert-chain.pem</ssl_chain_certificate>`

2.4.6.7 Cipher List

Description: Specifies list of allowed ciphers used by HTTP/S and WebSocket/S server connections.

The list should be in format as described in: https://www.openssl.org/docs/man1.0.2/apps/ciphers.html

Default value: DEFAULT

XML element name: `ssl_cipher_list`

Parent XML element: top level `<harrier>`

Example: `<ssl_cipher_list>3DES</ssl_cipher_list>`

2.4.6.8 TLS Diffie-Hellman parameters

Description: Specifies the full path to a file containing the Harrier Web server's Diffie-Hellman parameters. The file is in PEM format.

In order to perform a DH key exchange the server must use a DH group (DH parameters) and generate a DH key. The server will always generate a new DH key during the negotiation.

As generating DH parameters is extremely time consuming, Harrier Web server doesn't generate the parameters on the fly but uses pregenerated parameters. (A pregenerated DH group is installed as $(harrier.dir.share)/harrier.dhp) DH parameters can be reused, as the actual key is newly generated during the negotiation. The risk in reusing DH parameters is that an attacker may specialize on a very often used DH group. So a particular installation should periodically regenerate their own DH parameters.

In order to regenerate DH parameters, one can run OpenSSL's executable, for example on Linux:

```
% openssl dhparam 2048 > -/etc/isode/harrier.dhp
```

Note that this is an advanced option and typically doesn't need to be changed. However system administrators should consider periodically regenerating DH parameters.

Default value: $(harrier.dir.etc_or_share)/harrier.dhp

XML element name: `ssl_dh_params_path`

Parent XML element: top level `<harrier>`

Example: `<ssl_dh_params_path>$(harrier.dir.etc_or_share)/harrier.dhp</ssl_dh_params_path>`

2.4.6.9 Redirect HTTP

Description: This option provides a means to prevent non HTTPS access, by forcing all HTTP accesses to redirect to the HTTPS port (using HTTP 301 response).

This option is ignored if SSL Key is not set (see: Section 2.4.6.4, “Key”)

Default value: false

XML element name: `ssl_redirect_http`

Parent XML element: top level `<harrier>`

Example: `<ssl_redirect_http>true</ssl_redirect_http>`

2.4.6.10 Key Size

Description: Specifies SSL key size (in bits) when SSL private key is automatically generated.

Default value: 0 which means RSA default (2048 bits)

XML element name: `ssl_key_size`

Parent XML element: top level `<harrier>`

Example: `<ssl_key_size>2048</ssl_key_size>`

2.4.7 Global S/MIME settings

Most of the S/MIME related settings are configured per domain (See Section 2.5.21, “S/MIME”) but few of them must be the same for all domains so they are configured under top level element and treated as global option shared by all domains

2.4.7.1 Trusted Anchors

Description: Specifies in separate elements one or more trusted certificates (Trusted Anchors) used for S/MIME signature verification.

Each element must contain path to a DER or PEM encoded certificate.

For verification to succeed, all trust anchors and intermediate certificates must be included.

XML element name: `smime_trusted_certificate`

Parent XML element: top level `<harrier>`

Example: `<smime_trusted_certificate>/path/to/file.der</smime_trusted_certificate>`

2.4.7.2 Intermediate Certificates

Description: Specifies in separate elements intermediate certificates used for S/MIME signature verification.

Each element must contain path to a DER or PEM encoded certificate.

For verification to succeed, all trust anchors and intermediate certificates must be included.

XML element name: `smime_intermediate_certificate`

Parent XML element: top level `<harrier>`

Example: `<smime_intermediate_certificate>/path/to/anotherfile.der</smime_intermediate_certificate>`

2.4.7.3 OCSP/CRL check

Description: Perform S/MIME OCSP/CRL checks when determining message signature status.

Default value: true

XML element name: `smime_crl_check`

Parent XML element: top level `<harrier>`

Example: `<smime_crl_check>true</smime_crl_check>`

2.5 Domains

This section describes options that can be configured under the `domains` element

The domains contains one or more `<domain>` elements, each of which describes distinct set of configuration options (like IMAP, SMTP and LDAP) for a specific user group (usually associated with internet mail domain).

Here is an example `<domain>` entry. The individual elements are described below:

```
<domain name="example.com">
  <imap_url>imaps://imap.example.com:993</imap_url>
  <imap_server_trustanchors>
    $(harrier.dir.etc)trusted-ca-cert.pem
  </imap_server_trustanchors>

  <smtp_url>smtp://smtp.example.com:587</smtp_url>
  <smtp_server_trustanchors>
    $(harrier.dir.etc)trusted-ca-cert.pem
  </smtp_server_trustanchors>

  <ldap_url>ldap://ldap.example.com:19389</ldap_url>
  <ldap_server_trustanchors>
    $(harrier.dir.etc)trusted-ca-cert.pem
  </ldap_server_trustanchors>

  <ldap_sasl_mechs>SCRAM-SHA-1</ldap_sasl_mechs>

  <ldap_user_mail_ats>mail,mailLocalAddress</ldap_user_mail_ats>
  <ldap_user_name_ats>displayName,CN</ldap_user_name_ats>
  <ldap_user_prefs_oc>isodeHarrierRole</ldap_user_prefs_oc>
  <ldap_user_prefs_at>harrierUserPreferences</ldap_user_prefs_at>

  <ldap_addressbook_base_dn>
    ou=users,c=us
  </ldap_addressbook_base_dn>
```

```
</domain>
```

2.5.1 Domain Name Attribute

Description: Specifies a domain name used to match the domain part of the username supplied by someone logging in to Harrier Web. For example:

- `<domain name="example.com">` matches any username that ends with `@example.com`.

Default value: -None-

XML attribute name: `name`

Parent XML element: `<domain>`

Example: `<domain name="example.com">`

2.5.2 Domain Pattern Attribute

Description: Replacement for `name` attribute allowing to define more advanced matching rules. Specifies an ECMAScript Regular Expression which is used to match the domain part of the username supplied by someone logging in to Harrier Web. For example:

- `<domain pattern="example\.(net|com)">` matches any username that ends in either `@example.com` or `@example.net`

See ECMAScript syntax documentation [http://cplusplus.com/reference/regex/ECMAScript/] for more detailed information about regular expressions.

Default value: -None-

XML attribute name: `pattern`

Parent XML element: `<domain>`

Example: `<domain pattern="example\.(net|com)">`

2.5.3 Limit Charset

Description: Specifies if composition of messages should restrict the user to a limited set of characters. This option has no effect in `internet` mode. Valid options are:

- `IA5` - 7-bit character encoding corresponding to International Reference Alphabet (IRA). See http://www.itu.int/rec/T-REC-T.50-199209-I/en/
- `ITA2` - 5-bit character encoding. See https://en.wikipedia.org/wiki/Bauddot_code#ITA2.

Default value: empty (no restriction)

XML element name: `limit_charset`

Parent XML element: `<domain>`

Example: `<limit_charset>IA5</limit_charset>`

2.5.4 Military Sort Order

Description: Specifies if the Inbox should be sorted using the military sort order algorithm. This option has no effect in `internet` mode.

Default value: false

XML element name: `military_sort_order`

Parent XML element: `<domain>`

Example: `<military_sort_order>true</military_sort_order>`

2.5.5 Mode

Description: Specifies the mode Harrier will operate in. Can be either `internet`, `military` or `acp127`. See Section 2.6, “Server Modes” for more info.

Default value: military

XML element name: `mode`

Parent XML element: `<domain>`

Example: `<mode>internet</mode>`

2.5.6 Available Precedence value set

Description: Specifies which precedence values can be selected in Compose window for recipients. `stanag4406` means that all 6 precedence values (DEFERRED, ROUTINE, PRIORITY, IMMEDIATE, FLASH, and OVERRIDE) are available. `acp127` means that the following 4 precedence values are available: ROUTINE, PRIORITY, IMMEDIATE and FLASH. `acp127+emergency` means that the 4 ACP 127 values are allowed, plus an extra value: EMERGENCY.

Default value: `stanag4406` (in "military" and "internet" modes) or `acp127` (in the "acp127" mode)

XML element name: `precedence_set`

Parent XML element: `<domain>`

Example: `<precedence_set>stanag4406</precedence_set>`

2.5.7 Session Timeout

Description: Specifies the period of inactivity before web client sessions are terminated by the Harrier Web Server. A value of `0` can be used to indicate that sessions should never be timed out.

The value may be defined as number of seconds or number with unit: s/sec/seconds, m/min/minutes, h/hours, d/days, w/weeks.

Default value: 30min

XML element name: `session_timeout`

Parent XML element: `<domain>`

Example: `<session_timeout>30min</session_timeout>`

2.5.8 User Preferences

Description: Specifies the path for default user preferences. This setting is reserved for use by Isode.

Default value: -None-

XML element name: `user_preferences`

Parent XML element: `<domain>`

Example: `<user_preferences/>`

2.5.9 Automatic Saving

Description: Option allows to enable periodical saving of edited messages (0 means disabled).

It must be shorter than `session_timeout` to help ensure that if a user session is timed out while a composing a message, any changes to the message would already have been saved.

The value may be defined as number of seconds or number with unit: s/sec/seconds, m/min/minutes, h/hours, d/days, w/weeks.

Default value: 5m

XML element name: `auto_save`

Parent XML element: `<domain>`

Example: `<auto_save>5m</auto_save>`

2.5.10 Action Thresholds

Description: Defines action thresholds (act by) for messages with given precedence. Each instance of this option defines multiple pairs of precedence and period specified as attributes. Action thresholds are used in message sorting in absence of Reply-By header fields. They are also used to display remaining time for taking action on a particular message.

The precedence is specified as a number or corresponding name: DEFERRED=0, ROUTINE=1, PRIORITY=2, IMMEDIATE=3, FLASH=4, OVERRIDE=5. (Use "5" for EMERGENCY.)

The period may be defined as number of seconds or number with unit: s/sec/seconds, m/min/minutes, h/hours, d/days, w/weeks.

Default value: -Empty-

XML element name: `act_by`

Parent XML element: `<domain>`

Example:

```
<act_by precedence="5" period="5m"/>
<act_by precedence="4" period="10m"/>
<act_by precedence="3" period="15m"/>
<act_by precedence="2" period="1h"/>
<act_by precedence="1" period="3h"/>
<act_by precedence="0" period="7d"/>
```

2.5.11 Controlling Message Disposition Notification (MDN) handling

Description: Option allows generation of MDNs for messages requesting MDN (e.g. Read Receipt) when a logged in user deletes (or moves to Trash) messages without reading them first.

This option should only be used in environments where senders are trusted.

Default value: false

XML element name: `notify_delete_before_read`

Parent XML element: `<domain>`

Example: `<notify_delete_before_read>true</notify_delete_before_read>`

2.5.12 IMAP

This section describes options related to IMAP service access and usage.

2.5.12.1 Host and Port

Description: Specifies the host and port of the IMAP server for this domain.

Default value: -None-

XML element name: `imap_url`

Parent XML element: `<domain>`

Example: `<imap_url>imap.example.com:143</imap_url>`

2.5.12.2 Keep Alive Interval

Description:IMAP servers may be configured with an autologout timer, which will drop connections to clients (including Harrier Web Server) if there has been no IMAP activity for a certain time. This option can be used to make Harrier Web Server keep IMAP sessions alive. IMAP servers which are RFC3501 conformant will not use timeout values of less than thirty minutes, in which case a setting of imap_keepalive_interval of 1740 (i.e. 29 minutes) will prevent the IMAP session from being disconnected.

Note that the <session_timeout> option may be used to have HWS disconnect idle web client sessions, and may be a more appropriate way to control idle users.

The value may be defined as number of seconds or number with unit: s/sec/seconds, m/min/minutes, h/hours, d/days, w/weeks.

Default value: 29min

XML element name: `imap_keepalive_interval`

Parent XML element: `<domain>`

Example: `<imap_keepalive_interval>5m</imap_keepalive_interval>`

2.5.12.3 Re-connect Interval

Description: Specifies how frequently Harrier Web Server should re-attempt to connect to an IMAP server after failed/lost connection to it.

The value may be defined as number of seconds or number with unit: s/sec/seconds, m/min/minutes, h/hours, d/days, w/weeks.

Value 0 disables automatic re-bind.

Default value: 5s

XML element name: `imap_reconnect_interval`

Parent XML element: `<domain>`

Example: `<imap_reconnect_interval>5s</imap_reconnect_interval>`

2.5.12.4 Server Pinned Certificates

Description: If IMAP server certificate is not issued by any trusted CA (see: <imap_server_trustanchors>) pinned certificates allows to specify trustworthy certificates directly. Any number of pinned certificates (or none) may be specified. Each <imap_server_pinned> element should contain the path to a file containing one or more X.509 certificates in PEM format.

Default value: -None-

XML element name: `imap_server_pinned`

Parent XML element: `<domain>`

Example: `<imap_server_pinned>$(harrier.dir.etc)remote.imap.pem</imap_server_pinned>`

2.5.12.5 Server Trust Anchors

Description: Specifies the path to a file that contains one or more CA certificates in PEM format. There can be multiple files specified in separate <imap_server_trustanchors> elements. Specifying this option forces Harrier Web to use `StartTLS` for all IMAP connections, and to require that the IMAP server provides a certificate which can be verified using this set of CA certificates.

Default value: -None-

XML element name: `imap_server_trustanchors`

Parent XML element: `<domain>`

Example: `<imap_server_trustanchors>$(harrier.dir.etc)imap.pem</imap_server_trustanchors>`

2.5.12.6 SASL PLAIN/LOGIN Usage

Description: Controls whether authentication using the PLAIN or the LOGIN SASL mechanisms can be used without TLS in IMAP.

Default value: off

XML element name: `imap_plain_over_cleartext`

Parent XML element: `<domain>`

Example: `<imap_plain_over_cleartext>on</imap_plain_over_cleartext>`

2.5.12.7 STARTTLS Usage Policy

Description: Controls use or non use of STARTTLS. It can have one of the following 3 values: "mandatory" (always use STARTTLS, fail the connection if not advertised), "opportunistic" (try to use STARTTLS if advertised, but carry on regardless of STARTTLS success) and "suppress" (never use STARTTLS, even if advertised).

Default value: "opportunistic" when activation includes "TLS" subfeature and "suppress" otherwise

XML element name: `imap_starttls_policy`

Parent XML element: `<domain>`

Example: `<imap_starttls_policy>opportunistic</imap_starttls_policy>`

2.5.13 LDAP

This section describes options related to all LDAP services.

2.5.13.1 Attributes Validation

Description: By default, all LDAP attribute names used in this file will be validated using the local directory schema, and if any unrecognised names are present then Harrier Web Server will not start. If the LDAP server is using a different schema then this option may be used to prevent this validation.

Default value: true

XML element name: `ldap_validate_attributes`

Parent XML element: `<domain>`

Example: `<ldap_validate_attributes>false</ldap_validate_attributes>`

2.5.14 LDAP Server

This section describes options specifying access to LDAP server.

2.5.14.1 Host and Port

Description: Specifies the host and port of the LDAP server for this domain.

Default value: -None-

XML element name: `ldap_url`

Parent XML element: `<domain>`

Example: `<ldap_url>ldap.example.com:19389</ldap_url>`

2.5.14.2 Re-bind Interval

Description: Specifies how frequently Harrier Web Server should re-attempt to connect to an LDAP server after losing connection to it.

The value may be defined as number of seconds or number with unit: s/sec/seconds, m/min/minutes, h/hours, d/days, w/weeks.

Value 0 disables automatic re-bind.

Default value: 5s

XML element name: `ldap_rebind_interval`

Parent XML element: `<domain>`

Example: `<ldap_rebind_interval>5s</ldap_rebind_interval>`

2.5.14.3 SASL Mechanisms

Description: a space separated list of SASL mechanisms (in order of preference) to be used when authenticating to the LDAP server. Mechanisms not supported by the server are ignored from the list. The remaining mechanisms are tried in order. The ANONYMOUS and EXTERNAL mechanisms are not supported by Harrier (LDAP client) so not allowed in this option. If this option is not specified (or specified with an empty value), then Harrier Web will try to use all supported mechanisms one by one (it may be expensive so it is a good idea to specify a good mechanism explicitly).

Default value: -None-

XML element name: `ldap_sasl_mechs`

Parent XML element: `<domain>`

Example: `<ldap_sasl_mechs>SCRAM-SHA-1</ldap_sasl_mechs>`

2.5.14.4 Server Pinned Certificates

Description: If LDAP server certificate is not issued by any trusted CA (see: <ldap_server_trustanchors>) pinned certificates allows to specify trustworthy certificates directly. Any number of pinned certificates (or none) may be specified. Each <ldap_server_pinned> element should contain the path to a file containing one or more X.509 certificates in PEM format.

Default value: -None-

XML element name: `ldap_server_pinned`

Parent XML element: `<domain>`

Example: `<ldap_server_pinned>$(harrier.dir.etc)remote.ldap.pem</ldap_server_pinned>`

2.5.14.5 Server Trust Anchor

Description: Specifies the path to a file that contains one or more CA certificates in PEM format. There can be multiple files specified in separate <ldap_server_trustanchors> elements. Specifying this option forces Harrier Web to use `StartTLS` for all LDAP connections, and to require that the LDAP server provides a certificate which can be verified using this set of CA certificates.

Default value: -None-

XML element name: `ldap_server_trustanchors`

Parent XML element: `<domain>`

Example: `<ldap_server_trustanchors>$(harrier.dir.etc)ldap.pem</ldap_server_trustanchors>`

2.5.15 LDAP Authentication

This section describes options specifying directory bind and authentication details.

2.5.15.1 Simple Bind DN

Description: This option allows to enable simple bind to access local directory and organisational address book.

This option requires setting Section 2.5.15.3, “Bind Password” while Section 2.5.15.2, “SASL ID Bind” cannot be set at the same time.

XML element name: `ldap_auth_dn`

Parent XML element: `<domain>` for local directory (and address book), `<ldap_orgs_addressbook>` for organisational address book directory

Example: `<ldap_auth_dn>cn=user1,cn=users,o=example</ldap_auth_dn>`

2.5.15.2 SASL ID Bind

Description: This option allows to enable SASL ID based bind to access local directory and organisational address book.

This option requires setting Section 2.5.15.3, “Bind Password” while Section 2.5.15.1, “Simple Bind DN” cannot be set at the same time.

XML element name: `ldap_auth_sasl_id`

Parent XML element: `<domain>` for local directory (and address book), `<ldap_orgs_addressbook>` for organisational address book directory

Example: `<ldap_auth_sasl_id>user1@example.com</ldap_auth_sasl_id>`

2.5.15.3 Bind Password

Description: This option allows to specify password used by simple bind or SASL ID based bind.

This option is required when Section 2.5.15.1, “Simple Bind DN” or Section 2.5.15.1, “Simple Bind DN” is sepecified.

XML element name: `ldap_auth_sasl_pwd`

Parent XML element: `<domain>` for local directory (and address book), `<ldap_orgs_addressbook>` for organisational address book directory

Example: `<ldap_auth_sasl_pwd servpass:encrypt="true">secret</ldap_auth_sasl_pwd>`

2.5.16 LDAP User

This section describes options related to (logged in) user information stored in local LDAP directory.

2.5.16.1 User Mail Attributes

Description: Comma-separated list of attributes which can be used to find users' email addresses. The first attribute in the list will be used as the main user email address. Any other attributes are used to find alternate addresses for the user. These values are used by Harrier when showing a user's messages, to indicate whether a message was addressed "to" (action) or "cc" (info) that user.

Note that the first named attribute must be single valued.

Default value: mail,mailLocalAddress

XML element name: `ldap_user_mail_ats`

Parent XML element: `<domain>`

Example: `<ldap_user_mail_ats>mail,mailLocalAddress</ldap_user_mail_ats>`

2.5.16.2 User Name Attributes

Description: This option provides a way to specify attributes which may contain alternate user friendly user name (used both by logged in user and address book) Value of this attribute also appears in the From header field of sent messages.

Default value: in ACP 127 mode: "plaNameACP127,displayName,CN", in military mode: "displayName,plaNameACP127,CN", in internet mode: "displayName,CN".

XML element name: `ldap_user_name_ats`

Parent XML element: `<domain>`

Example: `<ldap_user_name_ats>displayName,CN</ldap_user_name_ats>`

2.5.16.3 User Preferences Object Class

Description: this is the objectclass value that must be present in a role's (or user's, if you don't use roles) own directory entry in order for Harrier Web to be able to save user preferences. See Section 2.5.16.4, “User Preferences Attribute”.

Default value: isodeHarrierRole

XML element name: `ldap_user_prefs_oc`

Parent XML element: `<domain>`

Example: `<ldap_user_prefs_oc>isodeHarrierRole</ldap_user_prefs_oc>`

2.5.16.4 User Preferences Attribute

Description: this option specifies the directory attribute in user entries that is used to store Harrier-specific user preferences for a Harrier user (see Section 2.5.16.3, “User Preferences Object Class”.

Default value: harrierUserPreferences

XML element name: `ldap_user_prefs_at`

Parent XML element: `<domain>`

Example: `<ldap_user_prefs_at>harrierUserPreferences</ldap_user_prefs_at>`

2.5.17 LDAP Address Book

Address-book lookups are performed when a user enters part of an address and the application provides matching names/addresses - for example when entering recipient names in the compose window, or when searching the address book for contacts.

The options determine how Harrier Web should search the LDAP directory when performing address-book lookups.

2.5.17.1 Base DN

Description: this option specifies base DN in the directory for searches. Only entries below this DN will be considered when searching the addressbook.

Default value: empty (i.e. the root DN)

XML element name: `ldap_addressbook_base_dn`

Parent XML element: `<domain>` for local address book, `<ldap_orgs_addressbook>` for organisational address book

Example: `<ldap_addressbook_base_dn>ou=staff,c=us</ldap_addressbook_base_dn>`

2.5.17.2 Search Filter Attributes

Description: This option contains a list of attributes which will be searched. This option is used if `<ldap_addressbook_filter>` is not specified. See Section 2.5.17.3, “Custom Search Filter”.

Default value: in ACP127 mode `displayName,sn,givenName,plaNameACP127,CN`; in other modes `displayName,sn,givenName,CN,mail,mailLocalAddress,mailRoutingAddress`.

XML element name: `ldap_addressbook_filter_ats`

Parent XML element: `<domain>` for local address book, `<ldap_orgs_addressbook>` for organisational address book

Example: `<ldap_addressbook_filter_ats>cn,givenname,sn,mail</ldap_addressbook_filter_ats>`

2.5.17.3 Custom Search Filter

Description: This option may be used to specify an LDAP search filter (see https://tools.ietf.org/search/rfc4515) that should be used when performing address book

searches. The special string `$1` may be used in the filter string to indicate the user's search string.

If this option is supplied, then the `<ldap_addressbook_filter_ats>` option is ignored (see Section 2.5.17.2, “Search Filter Attributes”).

Default value: empty, which means that the search simply looks for matches in attributes as specified by the `<ldap_addressbook_filter_ats>` option.

XML element name: `ldap_addressbook_filter`

Parent XML element: `<domain>` for local address book, `<ldap_orgs_addressbook>` for organisational address book

Example: `<ldap_addressbook_filter>(&(mail=*)(cn=*$1))</ldap_addressbook_filter>`

2.5.17.4 Mail Attributes

Description: Comma-separated list of attributes which can be used to locate user friendly names, photos or public certificates associated with email addresses present in messages.

Note: that the first named attribute must be single valued.

Default value: `mail, mailLocalAddress`.

XML element name: `ldap_addressbook_mail_ats`

Parent XML element: `<domain>` for local address book, `<ldap_orgs_addressbook>` for organisational address book

Example: `<ldap_addressbook_mail_ats>mail,mailLocalAddress</ldap_addressbook_mail_ats>`

2.5.17.5 User-friendly Name Attributes

Description: This option specifies User friendly name attributes. (First value found is being used where attributes order search is preserved).

Default value: in ACP127 mode `plaNameACP127,displayName,CN`; in military mode `displayName,plaNameACP127,CN`; in internet mode `displayName,CN`.

XML element name: `ldap_addressbook_name_ats`

Parent XML element: `<domain>` for local address book, `<ldap_orgs_addressbook>` for organisational address book

Example: `<ldap_addressbook_name_ats>displayName,CN</ldap_addressbook_name_ats>`

2.5.17.6 Routing Indicator Attribute

Description: Attribute used as an addresse's hint in ACP-127 mode.

Default value: rI

XML element name: `ldap_addressbook_routing_indicator_at`

Parent XML elements: `<domain>` for local address book, `<ldap_orgs_addressbook>` for organisational address book

Example: `<ldap_addressbook_routing_indicator_at>rI</ldap_addressbook_routing_indicator_at>`

2.5.18 LDAP Organisational Address Book

The domain may contain one or more `<ldap_orgs_addressbook>` elements, each of which describes distinct configuration options for a specific organisational address book.

Local address book (non-organisational) is specified directly under `domain` with options from Section 2.5.14, “LDAP Server”, Section 2.5.15, “LDAP Authentication”, Section 2.5.17, “LDAP Address Book” sections.

Each `<ldap_orgs_addressbook>` inherits by default all local address book settings and allows to redefine / overwrite differences. (In simple case organisational address book may be in the same directory but under distinct base DN or using different search filter for example). Yet it is flexible enough to use distinct directory, schema, and authentication (bind) details.

Here is an example `<ldap_orgs_addressbook>`:

```
<domain name="example.com">
...
  <ldap_url>ldap://ldap.example.com:19389</ldap_url>
  <ldap_server_trustanchors>
    $(harrier.dir.etc)trusted-ca-cert.pem
  </ldap_server_trustanchors>
  <ldap_sasl_mechs>SCRAM-SHA-1</ldap_sasl_mechs>

  <ldap_user_mail_ats>mail,mailLocalAddress</ldap_user_mail_ats>
  <ldap_user_name_ats>displayName,CN</ldap_user_name_ats>

  <!-- Non-Org AB (local directory only) --->
  <ldap_addressbook_base_dn>ou=users,c=us</ldap_addressbook_base_dn>
...
  <!-- Org AB in local directory --->
  <ldap_orgs_addressbook>
    <ldap_addressbook_base_dn>ou=orgs,c=us</ldap_addressbook_base_dn>
  </ldap_orgs_addressbook>
...
  <!-- Org AB in remote directory --->
  <ldap_orgs_addressbook>
    <ldap_url>ldap://ldap.nato.org:389</ldap_url>
    <ldap_auth_dn>cn=example,cn=users,o=nato</ldap_auth_dn>
    <ldap_auth_pwd servpass:encrypt="true">secret</ldap_auth_pwd>
    <ldap_addressbook_base_dn>ou=hqs,o=nato</ldap_addressbook_base_dn>
  </ldap_orgs_addressbook>
...
</domain>
```

2.5.19 Security Labels

This section describes configuration options used to control how security labels are presented to the user in the web browser. Note that these options are ignored for *internet* mode.

2.5.19.1 Catalog

Description: Specifies the full file path of a security label catalog. When a user composes a message, this catalog is used to populate the list of available security labels presented to the user.

This option is ignored unless a policy has been configured (see Section 2.5.19.2, “Policy”).

Default value: -None-

XML element name: `sio_catalog`

Parent XML element: `<domain>`

Example: `<sio_catalog>
$(harrier.dir.etc_or_share)label_catalog.xml</sio_catalog>`

2.5.19.2 Policy

Description: Specifies the full file path of the security policy file to be used when interpreting security labels that are presented to the user, either when composing a message or when viewing a message that has been received.

Default value: -None-

XML element name: `sio_policy`

Parent XML element: `<domain>`

Example: `<sio_policy>$(harrier.dir.etc_or_share)policy.xml</
sio_policy>`

2.5.19.3 Controlling Clearance Check

Description: Option controls whether the clearance check is performed on message send. Harrier will display a modal dialog if one or more recipients doesn't have clearance compatible with the currently selected security label.

Default value: false

XML element name: `sio_clearance_check`

Parent XML element: `<domain>`

Example: `<sio_clearance_check>on</sio_clearance_check>`

2.5.19.4 Default Clearance and Clearance Override

Description: Each option instance describes which Clearance applies to a specific pattern of recipients. This option takes the following parameters:

- `recipient_pattern` specifies an ECMAScript Regular Expression used to match a recipient email address (for example "alfa@example\.com") or domain (".*@example \.com"). If this parameter is empty, the whole option instance is ignored.
- `clearance_name` specified the clearance item name to use for the specified recpients. Values are as defined in the clearance catalog (for example: "DEMO-NATO SECRET").
- `override` is a boolean parameter. The value "false" means that the sio_clearance_map option specified the default clearance, which is going to be used if no clearance (for the same policy as label) is found in the corresponding recipient's directory entry. The value "true" means that the sio_clearance_map option specifies the override clearance, which is going to be used irrespective of any other clearances matching the corresponding recipient's (whether specified in recipient's directory entry or in another sio_clearance_map with override=false).

This option can appear multiple times.

Default value: N/A

XML element name: `sio_clearance_map`

Parent XML element: `<domain>`

Example: `<sio_clearance_map recipient_pattern="user34@example\.com"
clearance_name="Isode Confidential" override="false"/>`

2.5.20 Subject Indicator Codes

This section describes configuration options used to control subject indicator codes support.

2.5.20.1 Catalog

Description: Specifies the full file path of a subject indicator codes catalog.

Default value: $(harrier.dir.etc_or_share)/sic_catalog.xml

XML element name: `sics`

Parent XML element: `<domain>`

Example: `<sics>$$(harrier.dir.etc_or_share)/sic_catalog.xml</sics>`

2.5.20.2 Is at least one SIC required in every message sent?

Description: Specifies whether every message sent from Harrier requires at least one SIC code to be included. Note that this option should only be used in the military or ACP127 mode, as there is no way of entering SICs in the internet mode.

Default value: false

XML element name: `sic_required`

Parent XML element: `<domain>`

Example: `<sic_required>true</sic_required>`

2.5.20.3 Maximum number of SICs in a message

Description: Specifies the maximum number of allowed SICs in a message sent from Harrier.

Default value: 0 (i.e. no limit)

XML element name: `sic_max`

Parent XML element: `<domain>`

Example: `<sic_max>3</sic_max>`

2.5.21 S/MIME

This section describes configuration options that can be used to control S/MIME. Harrier supports S/MIME signing of outgoing email, signature verification on incoming email, email encryption/decryption, as well as automatic issuance of user certificates.

2.5.21.1 Encrypt outgoing email by default

Description: This option controls whether or not by default, all messages sent by users of the domain will be S/MIME encrypted. This option can be overridden by setting the harrierSmimeEncrypt LDAP attribute to TRUE in user's entry. In order to be able to S/MIME encrypt an email message, Harrier server needs to have LDAP access configured for the domain, it needs to have read access to userPKCS12 attribute in user's LDAP entry (which contain PKCS#12 object encrypted with user's login password) or to have read access to harrierSignIdentity attribute in role's LDAP entry (when HSM/PKCS#11 is used), and each recipient of the message must have S/MIME certificate published in userCertificate attribute of her respective LDAP entry. Note that expired certificates and certificates that don't correspond to the current From email address (in userPKCS12/userCertificate) are ignored when deciding whether a particular user can send encrypted or receive encrypted email.

Default value: false, which means that outgoing messages from the domain will not be S/MIME encrypted, unless explicitly enabled for a user account in LDAP.

XML element name: `smime_encrypt`

Parent XML element: `<domain>`

Example: `<smime_encrypt>true</smime_encrypt>`

2.5.21.2 Encryption algorithm

Description: Specifies symmetric cipher used for encrypting S/MIME messages.

Default value: aes-256-cbc

XML element name: `smime_encrypt_algorithm`

Parent XML element: `<domain>`

Example: `<smime_encrypt_algorithm>aes-128-cbc</smime_encrypt_algorithm>`

2.5.21.3 Sign outgoing email by default

Description: This option controls whether or not by default, all messages sent by users of the domain will be S/MIME signed. This option can be overridden by setting the harrierSmimeSign LDAP attribute to TRUE in user's entry. In order to be able to S/MIME sign email, Harrier server needs to have LDAP access configured for the domain, it needs to have read access to the userPKCS12 attribute in the user's LDAP entry (which contain PKCS#12 object encrypted with user's login password) and/or to have read access to harrierSignIdentity & userCertificate attributes in the role's LDAP entry (when HSM/PKCS#11 is used).

Default value: false, which means that outgoing messages from the domain will not be S/MIME signed, unless explicitly enabled for a user account in LDAP.

XML element name: `smime_sign`

Parent XML element: `<domain>`

Example: `<smime_sign>true</smime_sign>`

2.5.21.4 Triple Wrap outgoing email by default

Description: This option controls whether or not by default, all messages sent by users of the domain will be S/MIME triple-wrapped, which means signed, then encrypted, then signed again. You can find more information about triple-wrap in RFC 2634. In order to be able to S/MIME triple-wrap an email message, Harrier server needs to have LDAP access configured for the domain, it needs to have read access to userPKCS12 attribute in user's LDAP entry (which contain PKCS#12 object encrypted with user's login password) or to have read access to harrierSignIdentity attribute in role's LDAP entry (when HSM/PKCS#11 is used), and each recipient of the message must have S/MIME certificate published in userCertificate attribute of her respective LDAP entry. Note that expired certificates and certificates that don't correspond to the current From email address (in userPKCS12/userCertificate) are ignored when deciding whether a particular user can send triple-wrapped or receive triple-wrapped email. This option is automatically set to false if S/MIME signing and/or S/MIME encryption is explicitly disabled or not configured.

Default value: false, which means that outgoing messages from the domain will not be S/MIME triple-wrapped.

XML element name: `smime_triple_wrap`

Parent XML element: `<domain>`

Example: `<smime_triple_wrap>true</smime_triple_wrap>`

2.5.21.5 Protect message header when signing and/or encrypting outgoing email

Description: This option controls whether or not by default, headers of email messages are protected from changes in transit and disclosure to unintended readers. This is done by wrapping any to-be-signed/to-be-encrypted email message within message/rfc822 MIME body part in order to apply S/MIME security services to the message header fields. This procedure is described in RFC 5751, however it is not always implemented by common Email Clients. Administrator should set this option to false if compatibility with commone Email clients such as Thunderbird or Apple Mail is required at the expense of less secure handling of S/MIME email messages.

This option can be overridden by setting harrierHeaderProtect LDAP attribute in user's entry. This option is ignored, unless S/MIME signing and/or encryption is also enabled for the user.

Default value: true, which means that header of outgoing messages from the domain will be S/MIME protected, unless explicitly disabled for a user account in LDAP.

XML element name: `smime_protect_header`

Parent XML element: `<domain>`

Example: `<smime_protect_header>false</smime_protect_header>`

2.5.21.6 Automatically generate user's S/MIME certificate requests

Description: This option controls whether or not Certificate Signing Requests (CSR) are generated for users that don't have any valid S/MIME certificate in userPKCS12 attributes of their LDAP entries. If this option is enabled, S/MIME certificates corresponding to issued CSRs will also be uploaded to users' LDAP entries.

When this option is enabled for a domain, when a user logs in for the first time, CSR is generated in the `smime_csr_path` directory and the corresponding private key is saved in the `smime_pkcs12_path` directory. At suitable intervals, a suitably privileged administrator should review pending CSRs, and either issue certificates for them (using a tool such as Sodium CA), or submit them to an external CA. Once certificates are generated and saved in the `smime_csr_path` directory, the subsequent login attempt by the user will update the PKCS#12 file to include the user's certificate and upload both PKCS#12 file and certificate to the userPKCS12 and userCertificate attributes (respectively) in user's LDAP entry. After that, the user will be able to S/MIME sign and/or encrypt her messages.

This option is ignored, unless S/MIME signing and/or encryption is also enabled for the user. When this option is set to true, both `smime_csr_path` and `smime_pkcs12_path` must be set to non empty values.

Default value: false, which means that user S/MIME certificate requests will not automatically be generated.

XML element name: `smime_auto_generate_csrs`

Parent XML element: `<domain>`

Example: `<smime_auto_generate_csrs>true</smime_auto_generate_csrs>`

2.5.21.7 CSR and Certificate location path

Description: This option specifies the filesystem directory where S/MIME CSRs will be generated by Harrier Web Server and where the corresponding certificate (PEM) files should also be saved. This option is ignored, unless S/MIME signing and/or encryption is also enabled for the user and `smime_auto_generate_csrs` is enabled.

Default value: -None-

XML element name: `smime_csr_path`

Parent XML element: `<domain>`

Example: `<smime_csr_path>$(harrier.dir.var)csr</smime_csr_path>`

2.5.21.8 PKCS#12 location path

Description: This option specifies the filesystem directory where S/MIME private keys and final PKCS#12 files (which also include the corresponding certificates and certificate chains) will be generated by Harrier Web Server. This option is ignored, unless S/MIME signing and/or encryption is also enabled for the user and `smime_auto_generate_csrs` is enabled.

Default value: -None-

XML element name: `smime_pkcs12_path`

Parent XML element: `<domain>`

Example: `<smime_pkcs12_path>$(harrier.dir.var)pkcs12</smime_pkcs12_path>`

2.5.21.9 RSA Key Size

Description: S/MIME RSA key size (in bits) when S/MIME private keys are automatically generated. This option is ignored when the smime_ec_name is set to a non empty value.

Default value: 0 which means RSA default (2048 bits)

XML element name: `smime_key_size`

Parent XML element: `<domain>`

Example: `<smime_key_size>2048</smime_key_size>`

2.5.21.10 Elliptic Curve Name

Description: S/MIME Elliptic Curve name that will be used when S/MIME private keys are automatically generated. Use one of the curves listed in "openssl ecparam -list_curve".

Default value: "" which means that the RSA key will be generated instead

XML element name: `smime_ec_name`

Parent XML element: `<domain>`

Example: `<smime_ec_name>secp256k1</smime_ec_name>`

2.5.22 SMTP

This section describes options related to SMTP service connection and usage.

2.5.22.1 Host and Port

Description: Specifies the host and port of the SMTP server for this domain.

Default value: -None-

XML element name: `smtp_url`

Parent XML element: `<domain>`

Example: `<smtp_url>smtp.example.com:587</smtp_url>`

2.5.22.2 Server Pinned Certificates

Description: If SMTP server certificate is not issued by any trusted CA (see: <smtp_server_trustanchors>) pinned certificates allows to specify trustworthy certificates directly. Any number of pinned certificates (or none) may be specified. Each <smtp_server_pinned> element should contain the path to a file containing one or more X.509 certificates in PEM format.

Default value: -None-

XML element name: `smtp_server_pinned`

Parent XML element: `<domain>`

Example: `<smtp_server_pinned>$(harrier.dir.etc)remote.smtp.pem</smtp_server_pinned>`

2.5.22.3 Server Trust Anchor

Description: Specifies the path to a file that contains one or more CA certificates in PEM format. There can be multiple files specified in separate <smtp_server_trustanchors> elements. Specifying this option forces Harrier Web to use `StartTLS` for all SMTP connections, and to require that the SMTP server provides a certificate which can be verified using this set of CA certificates.

Default value: -None-

XML element name: `smtp_server_trustanchors`

Parent XML element: `<domain>`

Example: `<smtp_server_trustanchors>$(harrier.dir.etc)smtp.pem</smtp_server_trustanchors>`

2.5.22.4 SASL PLAIN/LOGIN Usage

Description: Controls whether SASL PLAIN and SASL LOGIN can be used without TLS in SMTP.

Default value: off

XML element name: `smtp_plain_over_cleartext`

Parent XML element: `<domain>`

Example: `<smtp_plain_over_cleartext>on</smtp_plain_over_cleartext>`

2.5.22.5 STARTTLS Usage Policy

Description: Controls use or non use of STARTTLS. It can have one of the following 3 values: "mandatory" (always use STARTTLS, fail the connection if not advertised), "opportunistic" (try to use STARTTLS if advertised, but carry on regardless of STARTTLS success) and "suppress" (never use STARTTLS, even if advertised).

Default value: "opportunistic" when activation includes "TLS" subfeature and "suppress" otherwise

XML element name: `smtp_starttls_policy`

Parent XML element: `<domain>`

Example: `<smtp_starttls_policy>opportunistic</smtp_starttls_policy>`

2.5.23 Draft & Release and Draft & Review

This section describes configuration options related to Draft & Release workflow support.

2.5.23.1 Releaser Address

Description: This option specifies the email address of a Releaser that is used for releasing Draft & Release messages. When this option is set all messages sent by users in the domain will be subject to Draft & Release procedure, except for messages created by the Releaser or senders specified in `exempted_address`. When Draft & Release procedure is in effect, two extra fields appear in the Compose window: one for selected Reviewers (can be empty) and another one for selected Releasers. The latter field allows one or more Releaser to be selected from the list of Releaser email addresses configured for the domain. Reviewer(s) will receive such messages in the order specified in the message. Each Reviewer can reject (return to Drafter with some comments), pass to the next Reviewer or Releaser, change Reviewer(s), or take ownership of the message and become the Drafter (to be reviewed by remaining Reviewer(s) and Releaser(s)). Releaser(s) will receive such messages once they are approved by Reviewer(s) (if any), in the order specified in the message. Each Releaser can reject (return to Drafter), pass to the next Releaser (if not the last Releaser), release (send to the originally intended recipients. Only if the last Releaser), change Drafter (reassign authorship of the message to another Drafter), change Reviewer(s), change Releaser(s), or edit and send each individual message.

This option can appear multiple times. If this option appears multiple times it specifies alternative Releasers that can be selected in the Compose window. Any one of them will be able to release messages.

Default value: empty, which means that the Draft & Release procedure is not used for the domain, unless `releaser_required` is also set to "on" or "true".

XML element name: `releaser_address`

Parent XML element: `<domain>`

Example: `<releaser_address>releaser@example.org</releaser_address>`

2.5.23.2 Enable Draft & Release procedure

Description: This option controls whether or not Draft & Release procedure is required for the domain. When it is required, this means that each message sent by any domain user needs to be approved for release by an authorized Releaser. When Draft & Release procedure is in effect, two extra fields appear in the Compose window: one for selected Reviewers (can be empty) and another one for selected Releasers. See Section 2.5.23.1, “Releaser Address” for more details on the meaning of these fields and operations allowed.

If `releaser_required` is set to true and `releaser_address` option is also set, the email address(es) specified in that option will be used as Releaser(s) for all messages: the Releaser field in the Compose window will display the specified email address(es), and the user will be able to pick one of the Releasers, but will not be able to leave the Releaser field empty. If `releaser_required` is set to true and no `releaser_address` option is set, the Releaser field remains empty and it needs to be entered manually. If `releaser_required` is set to false and no `releaser_address` option is set, then Draft & Release procedure is disabled for the domain. If `releaser_required` is set to false and `releaser_address` option is also set, the email address(es) specified in that option will be used as Releaser(s) for all messages (as above), plus an extra choice "No Releaser / Default Releaser" is given to the user.

Note, irrespective of the value of this option, the Draft & Release procedure is enabled if the `releaser_address` option is set.

Default value: off, which means that the Draft & Release procedure is not used for the domain, unless the `releaser_address` option is also set.

XML element name: `releaser_required`

Parent XML element: `<domain>`

Example: `<releaser_required>false</releaser_required>`

2.5.23.3 Conditional Draft & Release

It is possible to specify Draft & Release configuration that will only apply to certain messages (e.g. messages with certain SICs, messages from certain senders or messages above certain priority). Releasers specified using such conditional rules can be selected in the Compose window, or the user can select "No releaser" if the releaser is optional or an exception rule applies.

Each conditional rule is defined using `release_policy_rule` XML option with some attributes (see below). The order of conditional rules is important: the first matching rule applies to any sent message. (Or if none of them apply, the message is sent without Draft & Release.)

Each XML attribute specifies condition that must be true for the submitted message in order for the corresponding rule to be triggered. If multiple such conditions are specified, they all must be true for the rule to apply (i.e. they are ANDed), with the exception of "default", which is treated specially.

If no condition is specified, the `release_policy_rule` XML option behaves the same way as the corresponding `releaser_address` XML option.

Supported conditions are defined below:

a) sics=<optional-negate-char><list of ;-separated SIC codes that trigger this rule>. Where <optional-negate-char> is the character "!". If optional-negate-char is absent, any of the SICs has to match for the rule to be triggered. If optional-negate-char is present, none of the SICs have to match for the rule to be triggered.

Examples:

```
<release_policy_rule sics="AAA;BBB;CCC">steve@example.com</
release_policy_rule>

<release_policy_rule sics="!ZZZ;YYY">steve@example.com</
release_policy_rule>
```

b) precedence=<optional-negate-char><number or a precedence keyword, such as 'flash' or 'override'>. Where <optional-negate-char> is the character "!". If optional-negate-char is absent, all messages with the action precedence which is the same or higher than this value trigger this rule. If optional-negate-char is present, all messages with the action precedence which is the same or lower than this value trigger this rule.

Examples:

```
<release_policy_rule precedence="flash">nick@example.com</
release_policy_rule>

<release_policy_rule precedence="!3">nick@example.com</
release_policy_rule>
```

c) domain=<optional-negate-char><destination domain>. Where <optional-negate-char> is the character "!". If optional-negate-char is absent and any of the message recipients is

in the specified domain, this rule is triggered. If optional-negate-char is present and any of the message recipients is not in the specified domain, this rule is triggered.

Example: `<release_policy_rule domain="isode.net">alex@example.com</release_policy_rule>`

d) sender=<optional-negate-char><email address>. Where <optional-negate-char> is the character "!". If optional-negate-char is absent and the message sent by the specified (role) email address, this rule will be triggered. If optional-negate-char is present and the message is not sent by the specified (role) email address, this rule will be triggered.

Example: `<release_policy_rule sender="alex@example.com">damy@example.com</release_policy_rule>`

All messages that were sent by alex@example.com will use damy@example.com as the Releaser.

e) from=<optional-negate-char><email address>. Where <optional-negate-char> is the character "!". If optional-negate-char is absent and the message contains the specified email address in the From header field, this rule will be triggered. If optional-negate-char is present and the message doesn't contain the specified email address in the From header field, this rule will be triggered. (Note that in the case of an Organizational email message, this will be the Organization email address, while "sender" would be the role that generated the email message.)

Example: `<release_policy_rule from="!alex@example.com">damy@example.com</release_policy_rule>`

All messages that don't contain From alex@example.com will use damy@example.com as the Releaser.

e) label=<optional-negate-char><selector>. Where <optional-negate-char> is the character "!". If optional-negate-char is absent and the message contains the specified Security Label, this rule will be triggered. If optional-negate-char is present and the message doesn't contain the specified Security Label, this rule will be triggered. It is also possible to specially handle messages with unrecognized/invalid or missing SIO-Label header field values using special ":missing" and ":unknown" selectors.

Examples:

```
<release_policy_rule label="NATO|Secret">jasper@example.com</release_policy_rule>
```

```
<release_policy_rule label=":missing">collin@example.com</release_policy_rule>
```

There is also another attribute, which doesn't define a condition:

g) default="true". Used to reference the default set of releasers (all releasers specified with no conditions). The releaser value is omitted in this case, i.e. the `release_policy_rule` element content is empty.

Example: `<release_policy_rule sics="AAA" default="true"/>`

All messages that contain SIC "AAA" will be releasable by the default set of Releasers. (If this set is empty, then the rule is effectively removed.)

More complicated example 1: `<release_policy_rule precedence="flash" sics="ZZZ;WWW">mir@example.com</release_policy_rule>`

This rule will use mir@example.com as the Releaser if the message has action precedence "flash" (or higher) AND contains one or more of SICs {"ZZZ", "WWW"}.

More complicated example 2: `<release_policy_rule domain="example.net" sender="jengo@example.com">mir@example.com</release_policy_rule>`

This rule will use mir@example.com as the Releaser if the message is sent to any recipients in domain "example.net" and has From/Sender "jengo@example.com".

Draft & Release rules specified in the Harrier configuration file can be extended by setting the harrierDraftReleaseRules LDAP attribute in domain's configuration entry. This LDAP attribute contains XML fragment with `release_policy_rule` and/or `exempted_address` XML elements.

Default value: empty, which means that the Draft & Release procedure is not used for the domain, unless the `releaser_address` option is also set.

XML element name: `release_policy_rule`

Parent XML element: `<domain>`

2.5.23.4 Addresses Exempt from Draft & Release procedure

Description: This option includes an email address exempted from Draft & Release. This option can appear multiple times.

Draft & Release rules specified in the Harrier's configuration file can be extended by setting the harrierDraftReleaseRules LDAP attribute in the domain's and/or organization's configuration entry. This LDAP attribute contains XML fragment with `release_policy_rule`, `exempted_address` and/or `optional_releaser` XML elements.

Default value: empty, which means that no user (other than Releasers specified in `releaser_address` or `release_policy_rule`) is exempt from the Draft & Release procedure.

XML element name: `exempted_address`

Parent XML element: `<domain>`

Example: `<exempted_address>co@example.org</exempted_address>`

2.5.23.5 Addresses for which Draft & Release procedure is optional

Description: This option includes an email address for which Draft & Release procedure is optional, i.e. the email address can send direct email messages, as well as draft messages that are subject to Draft & Release procedure. The email address appears in the "sender" attribute. This option can appear multiple times.

Draft & Release rules specified in the Harrier Server's configuration file can be extended by setting the harrierDraftReleaseRules LDAP attribute in the domain's and/or organization's configuration entry. This LDAP attribute contains XML fragment with `release_policy_rule`, `exempted_address` and/or `optional_releaser` XML elements.

Default value: empty, which means that no user (other than Releasers specified in `releaser_address` or `release_policy_rule`) is exempt from the Draft & Release procedure.

XML element name: `optional_releaser`

Parent XML element: `<domain>`

Example: `<optional_releaser sender="xo@example.org"/>`

2.5.24 Organizational Messaging

This section describes configuration options related to Organizational Messaging. Organizational Messaging allows logged in users to send email messages on behalf of organizations, for example as specific roles within organizations.

2.5.24.1 Organizational From Addresses

Description: This option specifies an email address and associated display name that will be used as a From header field value in all emails sent by users of the domain, with the exception of outer From in Draft & Release messages. If this option is not set or set to empty value and `enable_org_messaging` is set to false, Organizational Messaging is not enabled for the domain.

Multiple instances of this option may be used to specify a number of alternative Organizational addresses.

There can be explicitly specified optional `name` attribute to define associated organisation name - when it is not specified or empty Harrier will try to get the name from the corresponding directory entry and the first attribute specified in `ldap_user_name_ats`.

Administrators should use Cobalt to configure Organizations in LDAP, as that provides more flexibility.

Default value: empty, which means that no domain-wide Organization is defined for the domain.

XML element name: `org_address`

Parent XML element: `<domain>`

Example: `<org_address name="Customer Support">support@example.org</org_address>`

2.5.24.2 Allow sending as self when Organizational Messaging is enabled

Description: When Organisation Messaging is enabled by `org_address`, this option is used to control whether a logged in user can send email messages with selected role address or just on behalf of the Organization.

Default value: false.

XML element name: `org_allow_send_as_self`

Parent XML element: `<domain>`

Example: `<org_allow_send_as_self>true</org_allow_send_as_self>`

2.5.24.3 Controlling loading of Organizational entries from LDAP

Description: This option is used to control whether or not Harrier attempts loading of Organizations the currently logged in role belongs to.

Default value: true.

XML element name: `enable_org_messaging`

Parent XML element: `<domain>`

Example: `<enable_org_messaging>false</enable_org_messaging>`

2.5.25 Message Printing and Message Display

This section describes options related to message view and message printing.

2.5.25.1 Controlling Message Printing

Description: Option controls presence of the print button in message view. When enabled and the user presses the print button, Harrier would open a new message print tab in browser and would show message print dialog.

Default value: true

XML element name: `print_allowed`

Parent XML element: `<domain>`

Example: `<print_allowed>true</print_allowed>`

2.5.25.2 Message Print Template

Description: Option controls what appears on the message print tab and how message information is formatted. It contains path to the message print Twig template, which is basically HTML file with special control operators to control insertion of different fields from message, such as message subject, message recipients, S/MIME verification status or message content.

Default value: $(harrier.dir.etc_or_share)/webapps/harrier/mail-print.html

XML element name: `message_print_template`

Parent XML element: `<domain>`

Example: `<message_print_template>$(harrier.dir.etc)/webapps/harrier/mail-print.html</message_print_template>`

2.6 Server Modes

Harrier Web Server can be configured to run in one of three separate server modes.

2.6.1 General Purpose ("Internet") Mode

Harrier can operate as a general purpose SMTP/IMAP client, providing a high performance easy to use Web interface to an IMAP/SMTP service. Harrier provides a useful set of general purpose email capabilities, but in addition will display any MMHS military headers (including security labels, message types, expires, reply-by, deliver-by etc.) that are present on received messages.

Internet mode is configured using the *mode* option in the configuration file. See Section 2.5.5, “Mode”.

2.6.2 Military Mode

In Military mode, *To* and *CC* recipients are always labelled as *Action* and *Info* respectively.

Users are able to specify values for the following fields when composing a message:

- Priorities can be specified for Action and Information recipients
- Exempted recipients can be specified
- Zen Action and Zen Info recipients can be specified
- Security Label selection and display
- SIC (Subject Indicator Code) support with server side configuration of SIC catalogue. See Section 2.5.20, “Subject Indicator Codes”.
- Message Type support with server side configuration of MMHS types catalogue. See mmhs-types at Section 2.4.4.3, “URL mapping”.
- Time controls (Filing Time; DTG; Expires; Reply-By; Deliver-By)
- Handling and Message Instructions can be specified

- Line length and charset can be restricted (e.g. to interoperate with ACP 127 recipients). See Section 2.6.3, “ACP127”. The Server returns limits per recipient in response to ldap-address-book command. Client restricts charset/line length/ attachments in the compose window.

Other changes from internet mode:

- The "Junk" folder is not visible
- Priorities are displayed in the list of messages in a folder.
- All dates are displayed in DTG format
- Some IMAP keywords can't be set, such as TODO, Later, Work.
- By default, military sort order is used, unless disabled in configuration. Military sort order sorts first by the precedence, then by delivery date, then by message UID. See Section 2.5.4, “Military Sort Order”

Military mode is configured using the *mode* option in the configuration file. See Section 2.5.5, “Mode”.

2.6.3 ACP127

ACP127 mode is largely the same as *Military* mode, with some additional changes/ restrictions:

- Both in scan listing and Compose window addresses are presented as ACP 127 PLA (Plain Language Address) and RI (Routing Indicator) as "hints". SMTP addresses are hidden from the user.
- When composing a message lines are limited to 69 characters.
- When composing a message character set is restricted to ITA2 or IA5 (ASCII). The default is IA5.
- When composing a message attachments are disabled.
- Display names for senders and recipients are searched for in the LDAP directory.
- Forwarding is disallowed (as attachments are disallowed).
- DEFERRED and OVERRIDE priorities are not supported.

ACP127 mode is configured using the *mode* option in the configuration file. See Section 2.5.5, “Mode”.

2.7 Logging

Harrier Web Server generates event and audit logs. When it starts, the server looks for the file *harrierlogging.xml* (first in *(ETCDIR)* and then *(SHAREDIR)*). If this file is found, it is used to determine where and what types of log records should be preserved. The version of *(SHAREDIR)/harrierlogging.xml* provided by Isode will cause Harrier to generate event and audit log files in *(LOGDIR)*.

To modify the default logging settings for the Harrier Web Server application, you should do the following:

Copy the file *(SHAREDIR)/harrierlogging.xml* into *(ETCDIR)harrierlogging.xml*, and then use the Log Configuration tool to make changes to the file in *(ETCDIR)*.

On Windows, a shortcut to the Log Configuration Tool will have been set up in the Isode folder on your Start menu.

On Unix, run */opt/isode/sbin/logconfig*.

Once the GUI is running, open *(ETCDIR)harrierlogging.xml*. You will see a display of a number of predefined logging streams used by the Harrier Web Server, which can be modified as required.

2.8 Storing passwords

The *(ETCDIR)/harrier_web_conf.xml* file which configures how Harrier Web Server connects to other servers can contain passwords. These can be obfuscated using the Service Key facility. To do this:

- The `servpass` option must be specified in the configuration file. See Section 2.4.3.1, "servpass:info". For example:

  ```
  <servpass:info service="isode.harrier"/>
  ```

- Any configuration option element containing sensitive data like password to be obfuscated should have a `servpass:encrypt` attribute:

  ```
  <password servpass:encrypt="true">secret</password>
  ```

- Create a service password using the command line option *(SBINDIR)/isode.harrierwebserver --passphrase <secret>*

  ```
  % isode.harrierwebserver ---passphrase -"myse£cret_84@76_"
  ```

  The passphrase must contain at least three out of uppercase, lowercase, numeric digits and punctuation).

  On Unix systems, you need to run this command as whatever userid the Harrier Web Server process is using (see Section 2.1.3, "Harrier Web Server runtime user").

- Harrier server ensures that all sensitive data is encrypted automatically. Every time Harrier server reads configuration file (on start or modification detection) checks also if all required data is encrypted. If needed encrypts unencrypted data and rewrites config file.

  Notice: Harrier server must have write access to the configuration file!

Once the Harrier Server performs encryption, the passwords and other sensitive data in the configuration file will no longer be stored in plain text, and will appear something like this:

```
<password servpass:encrypt="true">{spcrypt3}rnak0U/xxDw6W/P8lF+MN
+f/lUf0AK0C7Ln1vjDs8U7ZB2CyBQ==</password>
```

Chapter 3 Features

This section talks about certain features and how they are configured.

3.1 User and Role configuration/preferences in LDAP

Harrier Web Server relies on user specific information stored in LDAP in order to implement some features, such as Section 6.1.2.1, “S/MIME signing/encryption/triple-wrap” or Section 3.6, “Recipients Capability”.

In order to be able to use full functionality provided by Harrier, you must ensure all user entries have the **isodeHarrierUser** auxiliary object class. This is typically combined with the **person** or the **inetOrgPerson** structural object class.

If you use Roles, you must ensure all role entries have the **isodeHarrierRole** auxiliary object class. This is typically combined with the **inetOrgRole** structural object class.

3.2 Security Labels

Harrier Web Server supports security labels for messages as per RFC7444 [https://tools.ieft.org/html/rfc7444]. When viewing a message that contains a security label, that label will be always shown to the user. When Harrier is suitably configured, then users will be able to select a security label to be applied to any message that they compose (*military* and *ACP127* modes only).

If a security policy and label catalog are configured (see the Section 2.5.19, “Security Labels”), then users will be able to select a label when composing a new message. The catalog contains the labels which will be presented to the user; the policy contains information about how each label should be displayed (name and colors). A sample policy and label catalog are provided in

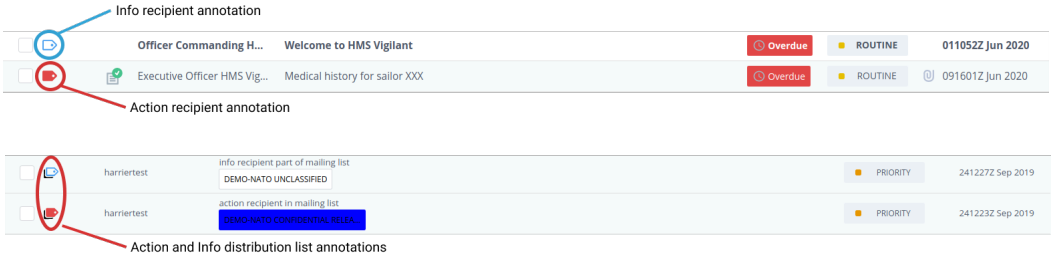
```
$(SHAREDIR)/policy.xml
$(SHAREDIR)/label_catalog.xml
```

3.3 Recipient Type annotation

Each message displayed by Harrier Web is annotated to show whether you are an *action* recipient (you need to do something about this message) or *info* recipient (this message is for your information). No icon will appear if the recipient is BCCed or in some uncommon cases. This feature enables the user to get a better understanding of which message needs action and which is just informative. The annotation can also help the user get a better understanding if the message was sent directly to them or they received it as part of a distribution list.

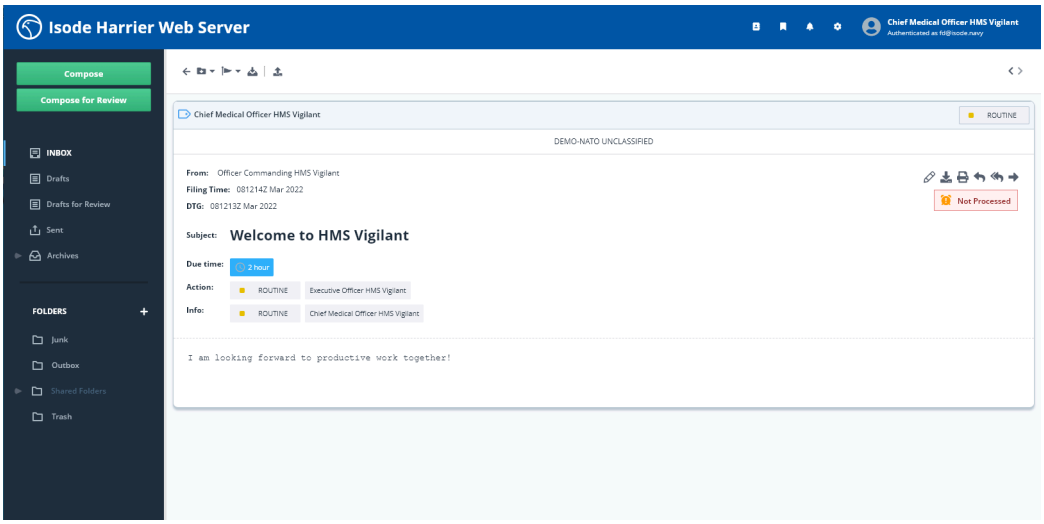
On the scan-list, the user can look at all of the messages and see the recipient type as an icon.

Figure 3.1. Scan-list view of recipient type bar



For example, if the user sees the action recipient annotation in the scan-list they would have to take action on the message. Whereas, if they see the info recipient annotation, they know that this is not as urgent on their side since the message serves more of an informative purpose. And if its one of the action/info distribution annotations, that would mean that they have received the message as part of a mailing list. On the scan-list, the user can hover over the icon to get more information and on an opened message this is presented with more detail. On the message pane, this information is displayed in a bar at the top of the message.

Figure 3.2. Message pane view of recipient type bar



3.4 Profiled Messages

Harrier Web Server supports Profiled Messaging in which the messages are redistributed by M-Switch to a set of users depending on the content, and the rules of the Profiler configured in M-Switch.

3.5 Organizational Messaging

Harrier Web Server supports Organizational Messaging. Organizational Messaging allows users to send email messages on behalf of one or more organizations they are associated with, for example an Engineering Officer on a ship HMS Kent sending a message to HQ on behalf of HMS Kent.

Information about Organizations associated with a particular email domain can be provided in either Harrier configuration file and/or retrieved from the configured LDAP server. When storing Organizations in LDAP the mmhsOrganization object class is used.

3.6 Recipients Capability

When operating in *military* or *acp127* mode Harrier Web Server will check any limitations that apply to message recipients. When a user is composing a message they will see the enforced limits at the top of the message content. Invalid content will be highlighted to the user and they will be unable to send the message until this has been corrected.

Limits may be specified on any user entry in the directory which has the objectClass `isodeHarrierUser`, using the attributes described below.

3.6.1 Charset

Description: The charset option can be be set to either `ITA2` or `IA5`

Default value: -None-

Example: ITA2

LDAP attribute name: `harrierCharsetRestrictions`

3.6.2 Max Line Length

Description: The Max Line Length option can be set to any positive integer. Once set user will not be able to send messages with lines that exceed this length. Note that in *ACP127* mode, a default maximum line length of 69 is assumed.

Default value: -None-

Example: 80

LDAP attribute name: `harrierMaxTextPlainLineLength`

3.6.3 Attachments

Description: The attachment option is a boolean value to indicate if users can receive attachments. If it is set to `false` then users will not be able to send them attachments.

Default value: true

Example: false

LDAP attribute name: `harrierAllowAttachments`

3.7 Allowing remote content in HTML messages

External content, such as images, videos and CSS, is frequently used by spammers and attackers for tracking when an email message was read. By default, Harrier blocks external content in HTML messages. This doesn't apply to content generated by Harrier Web Server itself or to content embedded into HTML messages. When such external content is found, a button appears on the message, allowing the user to show remote content.

Once the user allows remote content for a specific message, Harrier Web Server remembers this decision by storing it on the IMAP server using $ShowRemoteContent IMAP keyword. Next time the user views the same message (using the same or different instance of Harrier Web Server), remote content will be shown automatically.

3.8 Message Search

Harrier Web Server supports search for messages that match a specific search criteria. For example, a user can search for messages sent or received by specific senders/ recipients, messages containing certain text in subject or body of the message, messages with a particular SIC, security label and/or of specific precedence.

3.9 Military Sort Order

Military sort order can be enabled in `military` and `acp127` modes. This is controlled by the `military_sort_order` option.

By default INBOX and other folders are sorted by the message arrival time, which is the time they were delivered (using SMTP) or uploaded to the corresponding IMAP folder. When military sort order is enabled, messages in INBOX are sorted so that they appear in the order that they should be actioned. This is done by sorting first on the message Action or Info precedence (see below), with highest precedence messages appearing first. When messages have the same precedence, they are sorted in order of "due time" (see below) so that messages with the soonest "due time" appear first. Messages with the same precedence and "due time" are sorted by arrival time.

The "Due time" for a message is calculated as the shortest time period of the following: a) time left till the Reply-By time (if any); b) time left till the Expires time (if any); c) default processing time as prescribed by the message precedence that applies to the logged in role. See the `act_by` option for more details.

Message precedence is calculated as follows:

1. If the logged in role is an Action recipient, then the Action precedence is used;
2. Otherwise, if the logged in role is an Info recipient, then the Info precedence is used;
3. Otherwise Harrier assumes that the message was received through a distribution list, so the Action precedence is used.

3.10 Integration with IRIS WebForms

Harrier Web Server supports close integration with Systematic IRIS WebForms for display and creation of MTF (Message Text Format) attachments.

In order to integrate IRIS WebForms into Harrier, 2 steps need to be completed:

1. The following top level option needs to be added to the Harrier configuration file: `<proxy_map pattern="^\/iris\/(.*)" uri="http://localhost/webforms/${1}"/>` where http://localhost/webforms is the location of the IIS web server running in front of the IRIS WebForms, and "/iris/" is the URL prefix reserved on Harrier Web Server for IRIS specific forms and related documents.

2. To enable IRIS WebForms integration for a particular domain use the following option under the corresponding <domain> XML element:
   `<adatp3_webforms_uri>/iris/webforms.html</adatp3_webforms_uri>`

Chapter 4 Drafter-centric Review

This section introduces Drafter-centric Review and explains how this feature of Harrier is presented to Harrier Users.

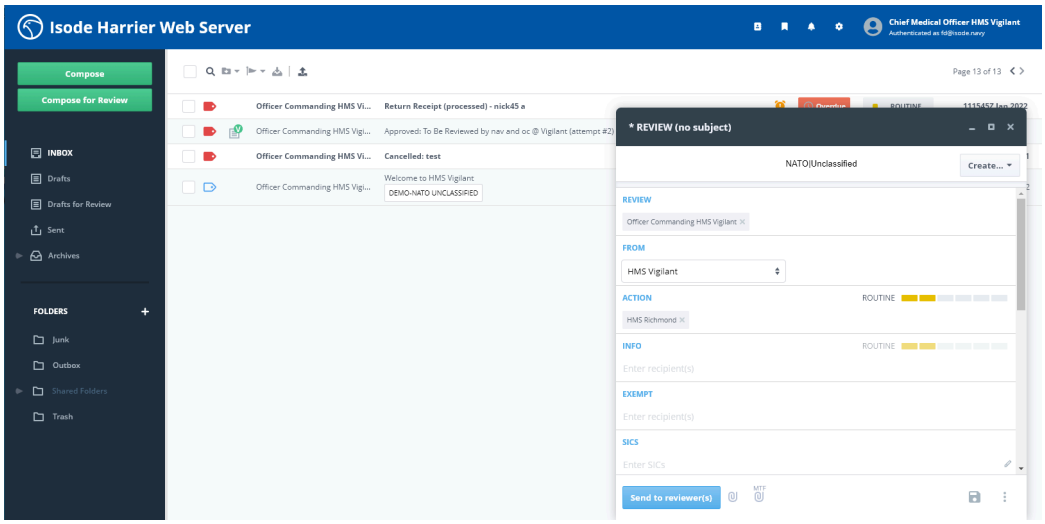
4.1 Drafter-centric Review

Drafter-centric Review is an optional process that allows outgoing messages to be reviewed by one or more reviewer before they are sent out to intended recipients or being subjected to Draft & Release process. Drafter-centric Review requires no Harrier server configuration. Any recipient can be specified as a reviewer.

Reviewers can approve a draft message, or make comments on it. The drafter does not need to wait for any reviewers to respond before sending the message, and does not need to make changes as a result of reviewer comments. If the drafter decides to change the message in response to reviewer comments, then the re-drafted message may be submitted without further review, or sent to reviewers again. The drafter can also cancel a message before sending it, which will notify reviewers that the message has been abandoned.

In Drafter-centric Review mode, Drafters compose messages in the usual manner, but will press "Compose for Review" instead of "Compose" button. The "Compose for Review" window is going to be almost identical to the regular Compose window, except that it will contain an extra "Reviewer" field.

Figure 4.1. Compose for Review window



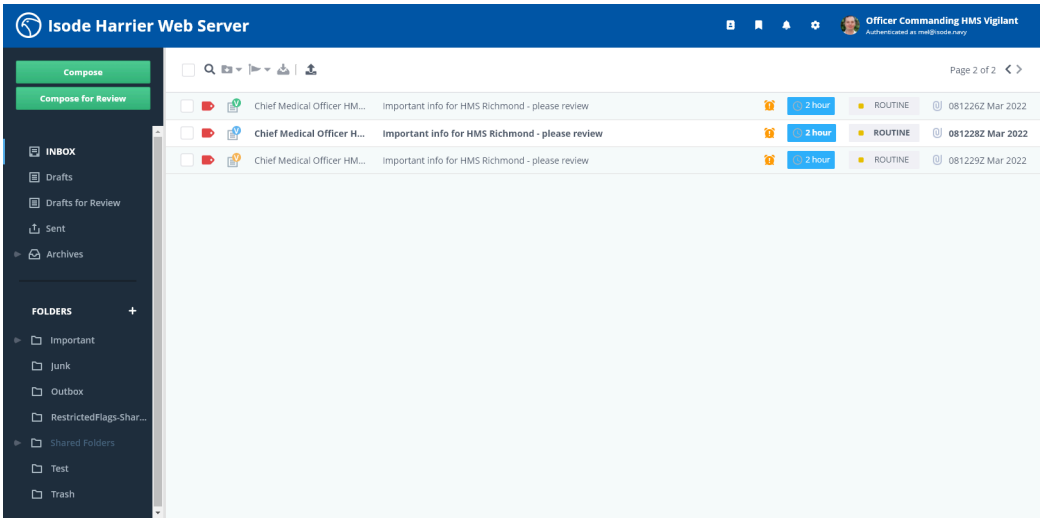
A Reviewer's message list contains icons to indicate which messages are "draft-for-review" messages, and whether they need to be reviewed, or have been reviewed already. The screenshot below has examples of 3 icons you may see on a Drafter-centric Review message.

The green icon with the letter "V" indicates that the message has been reviewed.

The blue "info" icon indicates messages that have yet to be processed.

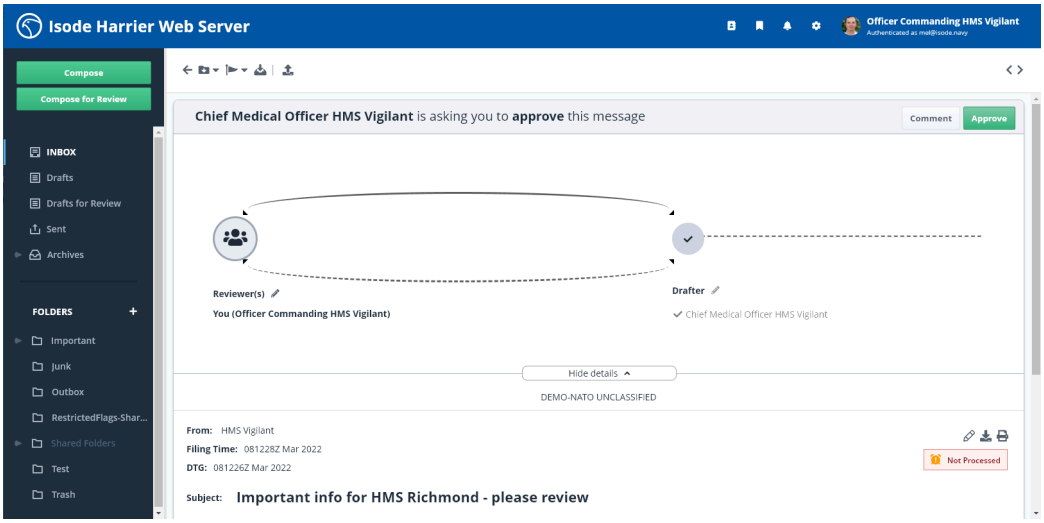
The yellow icon indicates a message that has been commented on.

Figure 4.2. Message list with examples of Drafter-centric Review icons



Extra buttons are displayed in the message view to allow a Reviewer to comment on or approve the message.

Figure 4.3. Message view with Drafter-centric Review related action buttons



Chapter 5 Draft And Release

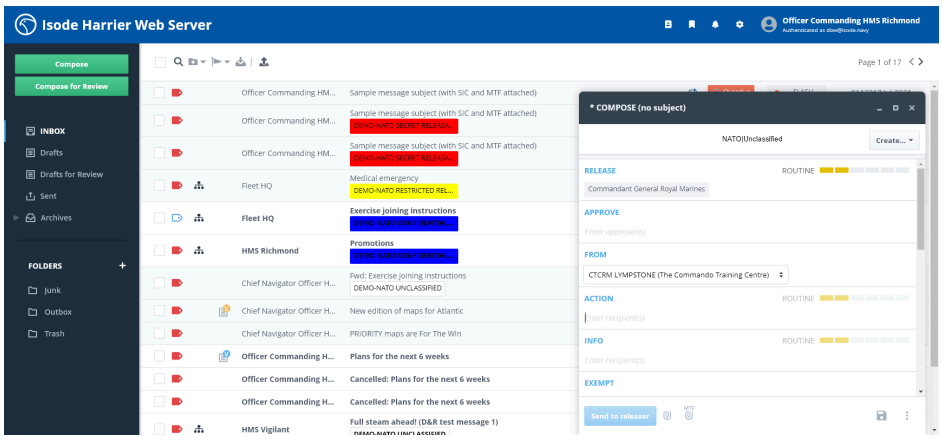
This section introduces Draft and Release and explains how this feature of Harrier is presented to Harrier Users.

5.1 Draft & Release

Draft & Release allows outgoing messages to be approved before they are sent out to intended recipients. When a Draft & Release policy has been configured, messages that are sent by any users who have not been designated as "exempted" will be sent to zero or more approvers, and then to one or more "releasers". Approvers and releasers are able to reject a message (in which case they can supply a reason for rejection), or to approve a message (in which case they can optionally edit the message before approving). A releaser can modify the list of approvers or releasers, or reassign the message to another drafter. The final releaser decides whether the message is released (sent) to the recipient(s) specified by the drafter. Messages which are rejected will be returned to the drafter, who can then choose to edit the message and re-submit, reassign editing to another drafter, or to abandon the attempt.

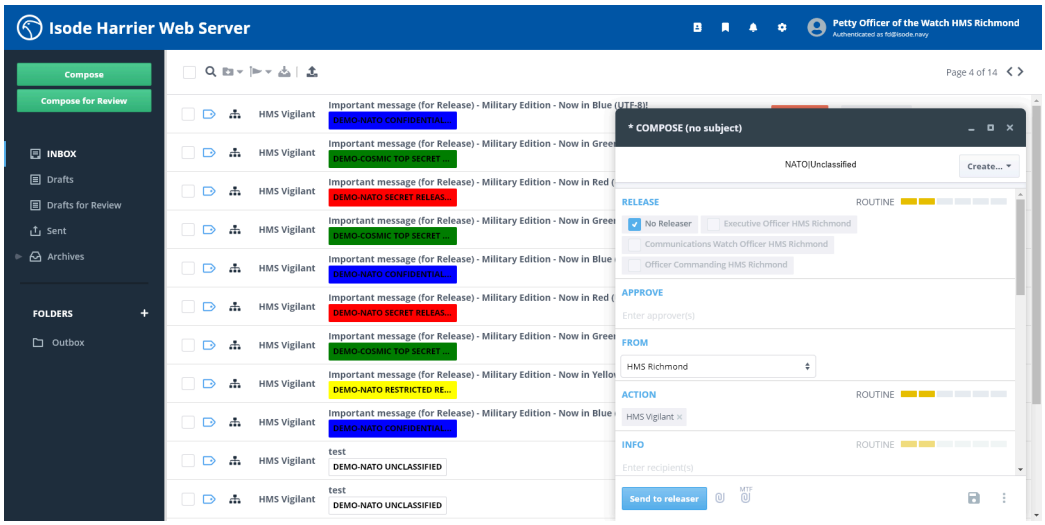
In Draft & Release mode, Drafters compose messages in the usual manner, but will see two extra fields in the Compose window: "Releaser" field and (optional) "Approver" field. Depending on policy, the Drafter may be forced to use a specific Releaser, or may be able to choose from a list of Releasers. The screenshot below shows an example of the "Compose" screen where the drafter is being required to use the releaser configured for this particular "From" organization

Figure 5.1. Compose Draft & Release message with required releaser



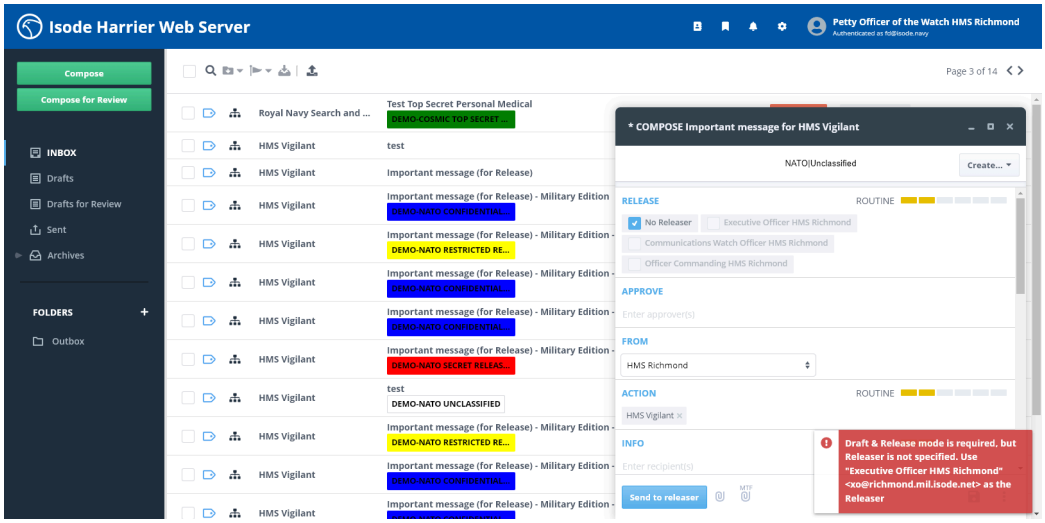
The following screenshot shows what the "Compose" screen may look like for a "From" organization which has multiple releasers configured, and a drafter who is configured to be able to draft and also to send direct in certain circumstances. The drafter may select from the releasers listed and can also choose "No Releaser".

Figure 5.2. Compose Draft & Release message with multiple releaser options



"No Releaser" is only a valid option if the rules for sending directly have been followed. If any of the rules are contravened, Harrier will provide a prompt when the drafter tries to send, and suggest the use of the default releaser, as seen in the screenshot below.

Figure 5.3. Draft & Release message with required releaser



A Releaser's/Approver's message list contains icons to indicate which messages are "draft" messages, and whether they need to be approved, or have been approved already. The screenshot below has examples of 4 icons you may see on a Draft & Release message.

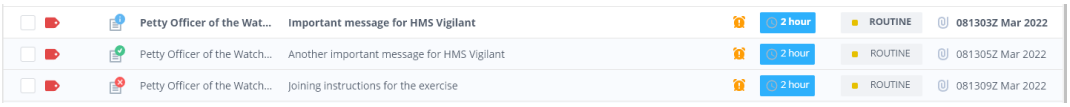
The green icon with a tick indicates that the message has completed it's journey through Draft & Release and has been released to the recipients.

The blue "info" icon indicates messages that have yet to be processed.

The green icon with an "R" indicates that the message has been processed and a receipt has been sent to the appropriate address but this user is not the final releaser and the message may still need actions performed by someone else.

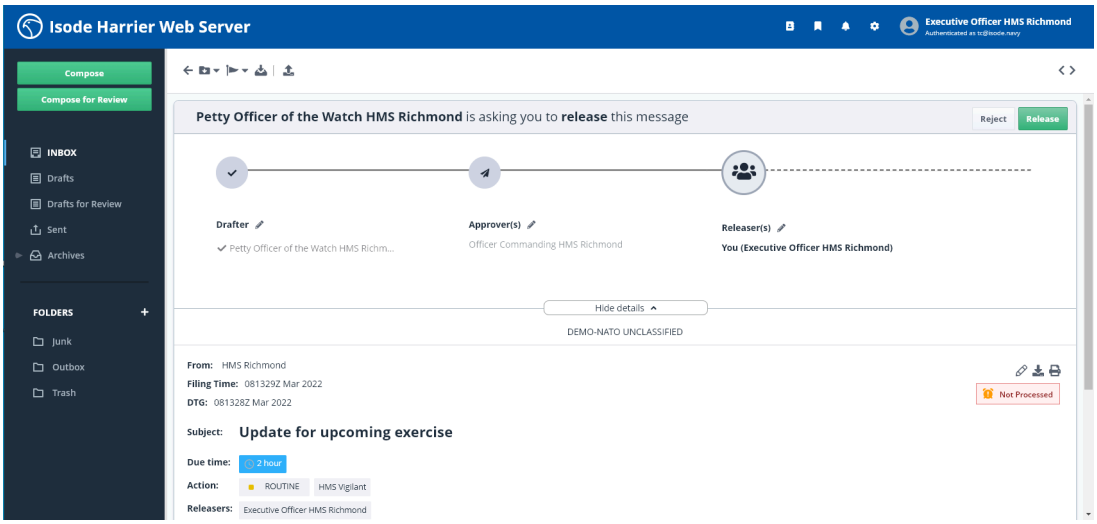
The red icon indicates a message that has been rejected.

Figure 5.4. Message list with examples of Draft & Release icons



Extra buttons are displayed in the message view to allow a Releaser / Approver to release/approve/reject the message. In order to change the drafter, approvers, or releasers the Releaser / Approver needs to expand the Draft & Release panel.

Figure 5.5. Message view with Draft & Release related action buttons



In order to enable Draft & Release, the Harrier administrator should use Isode's User Provisioning tool Cobalt. See *Cobalt Administration Guide* for information on how to do this.

In Cobalt, the administrator can enable Draft & Release for certain domains, create Profiled Addresses for organizations, and add members to those organizations who can be configured as Drafters and/or Releasers. Rules for sending directly can also be configured here.

Note that, in order to have an organization as an Action or Info recipient, the administrator must also provide the profiler.json file with a rule that allows this. Information on the Profiler channel can be found in the *M-Switch Administration Guide*.

For advanced configurations, the Harrier administrator could directly edit the harrierDraftReleaseRules LDAP attribute in the organization's configuration entry. This attribute contains an XML document with `exempted_address`, `release_policy_rule`, `optional_releaser` and/or `exempted_address` XML elements. For example, the `exempted_address` option can be used to exclude some senders from the Draft & Release procedure.

Releasers can be configured to be conditional or unconditional. Unconditional Releasers are authorised to approve all messages sent. Conditional Releasers are responsible for approving messages that satisfy a certain criteria. For example all messages with a particular SIC code, all messages with FLASH precedence or all messages from particular senders. `release_policy_rule` option can be used to define one or more conditional or unconditional Releaser, or an exception condition when no releaser is required. `releaser_address` option is a simplified version of a `release_policy_rule` option, which can only define an unconditional releaser.

5.1.1 Expected Behaviours

The following sub-sections describe the various behaviours that are expected when performing actions on a Draft & Release message.

5.1.1.1 Expected behaviour when composing a message for release

- The user is presented with a standard compose form with additional fields for releasers and optional approvers.
- The user can pick one or more releaser from the list. If multiple are selected, their order is important.
- In some configurations the user can also pick "No Releaser" choice. This is only available when a conditional Draft and Release rule is configured.
- When the user sends a new message the Harrier Web Server constructs a message with the drafted message as a nested message. Harrier Web Server includes a hardcoded explanatory note with instructions for approvers/releasers, in case they use a Mail User Agent different from Harrier. (The generated message is multipart/mixed with the first body part being text/plain (containing the explanatory note) and the second part message/rfc822 with the message to be released inside it).
- The release and reject controls will be hidden from the drafter when they view the message in their Drafts, Sent or Trash folders.
- If Draft and Release is enabled and required, the Harrier server will disallow sending a message which does not have a releaser.

5.1.1.2 Expected behaviour when receiving and Draft and Release message for approval

- The scan list will indicate the message is for Draft and Release, and whether it has already been released or rejected.
- When the message is opened the user is presented with options to Approve or Reject, as well as options to change Drafter or edit the list of Approvers.
- The release and reject controls will be hidden once one of these actions has been chosen. The server also rejects multiple actions on the same Draft & Release message if they are performed simultaneously by different users.
- Users are not allowed to release their own messages. Release/Reject controls are also not shown in "Drafts", "Sent" and "Trash" folders.

5.1.1.3 Expected behaviour when releasing a message

- The inner (to be released) message includes a header field indicating one or more releaser.
- The inner message is sent to the specified recipients.
- The inner message is copied into the releaser's "Sent" folder
- An IMAP flag is added to the outer Draft & Release message to indicate that it has been released.
- If drafter has requested confirmation of release, an MDN is sent to the drafter. This option is turned on by default.

5.1.1.4 Expected behaviour when rejecting a message

- A rejection message is sent to the drafter. A multipart/mixed message with the first part being a rejection note and the second part being message/rfc822 with the original message.
- The rejection message is copied into the releaser's "Sent" folder.
- An IMAP flag is added to the outer Draft & Release message to indicate that it has been rejected.

5.1.1.5 Expected behaviour when a releaser edits a Draft & Release message

- If the drafter has requested confirmation of release, an MDN is sent to the drafter and the releaser's copy is marked with $MDNSent IMAP keyword. This option is turned on by default.
- The releaser takes ownership of the message, so the From header field of the edited message will be the releaser's email address.
- The releaser is able to edit the earlier drafted message.
- If the releaser who edited the message is not exempted from Draft & Release procedure, the edited message will include all releasers who haven't yet approved the message.
- A releaser can edit a Draft & Release message any number of times. A releaser can also edit a message that was already released or rejected. This allows for flexible workflow, for example to allow releaser to split a single message from a drafter into multiple independent pieces.
- An edited Draft & Release message can be saved to "Drafts" as any other draft message. No special handling occurs.

Chapter 6 S/MIME

This section describes sending and receiving signed/encrypted messages using S/MIME.

6.1 S/MIME

Harrier Web Server supports S/MIME signing, verification of signed messages, encryption/decryption as specified in RFC 5750, RFC 5751 and RFC 1847.

When S/MIME has been configured, it provides the user with a method to send and receive secure MIME messages. Depending on the server configuration and the sender/recipient certificates, a user can choose to sign, encrypt, sign and encrypt, triple-wrap a message or not use S/MIME at all. If S/MIME is not configured or a user's certificate is not valid, the user won't be able to use partial or full functionality of the feature. Based on the setup and environment, some users may always need to send signed/encrypted/triple-wrapped messages or may never be able to send messages with a particular S/MIME option.

In an S/MIME configured environment, a sender would compose messages in the usual manner, but will have the option to enable S/MIME signing, encryption and triple-wrap by toggling the S/MIME checkboxes from the 'More options' menu in the Compose window. The following combinations of user options are available, if everything is enabled and the sender and recipients have valid certificates:

None	S/MIME Sign	S/MIME Encrypt	S/MIME Triple-wrap
x			
	x		
		x	
			x
	x	x	
	x		x
		x	x
	x	x	x

6.1.1 S/MIME Message Status

After the message has been delivered, the receiver can look at the message and see the S/MIME status. This shows irrespective of whether S/MIME signing and/or encryption is enabled for the logged in user. For example, if a signed and encrypted message was encrypted and signed and if the receiver was able to decrypt the message and verify the signature. The following options are most common:

<empty string>	Non S/MIME message
encrypted	S/MIME encrypted message. This doesn't convey whether or not the message was successfully decrypted.
encrypted+signed/failed	The message was both signed and encrypted. While decryption has succeeded, signature verification has failed. See **signed/failed**.
encrypted+signed/verified	The message was both signed and encrypted. The signature was verified (see **signed/verified**)

encrypted/failed	S/MIME encrypted message failed on decryption. The message might or might not contain other S/MIME messages inside.
signed	S/MIME signed message, but the signature was not yet verified
signed/failed	S/MIME signed message, but the signature failed to verify. This could be due to untrusted certificate authority(CA) or incomplete chain of certificates to a Trusted Anchor, expired or revoked signing certificate and broken signature due to message modification.
signed/verified	S/MIME signed message and the sender's signature was successfully verified, sender matches the From header field and the sender's certificate is trusted for signing.
encrypted+signed/verified/triple-wrapped	The message was signed, encrypted and then signed again. The two signatures were verified (see **signed/verified**)
unknown	S/MIME message, but it is neither signed, nor encrypted. Sometimes this is reported for corrupted S/MIME messages. This is also displayed for OpenPGP messages, which are not being handled for the time being.

If any of the failures above have occurred, the message will have an S/MIME Warning or Error displayed, explaining in a more comprehensive way what has gone wrong.

6.1.2 Configuration of S/MIME operations

6.1.2.1 S/MIME signing/encryption/triple-wrap

Signing and/or encryption can be enabled per user by setting harrierSmimeSign/ harrierSmimeEncrypt LDAP attributes. It is also possible to just enable signing/ encryption per domain, but setting harrierSmimeSign/harrierSmimeEncrypt will override it for a specific user. To enable signing for a domain use the following option under the corresponding <domain> XML element: `<smime_sign>true</smime_sign>` To enable encryption for a domain use the following option under the corresponding <domain> XML element: `<smime_encrypt>true</smime_encrypt>` To enable triple-wrap for a domain use the following option under the corresponding <domain> XML element: `<smime_triple_wrap>true</smime_triple_wrap>`. Note that setting harrierSmimeSign and/or harrierSmimeEncrypt LDAP attributes to FALSE for a user will also disable ability to send triple wrapped messages by that user.

When sending messages, it is possible to disable S/MIME signing/encryption/triple-wrap on per message basis. (It is not possible to enable signing/encryption/triple-wrap for a message, if it is disabled for the user.) A Harrier user can see whether S/MIME signing/ encryption/triple-wrap is enabled by viewing message options in the Compose window.

Note that in order for S/MIME signing to be enabled by default, all of the following conditions must be met: 1) signing must be enabled for the domain (in XML configuration) or for the specific user (in LDAP); 2) the domain has LDAP configuration and all users can bind to Directory using the same username/password as used for IMAP login; 3) the logged in user's LDAP entry must have a userPKCS12 attribute, containing the private key and corresponding certificate, encrypted with the user's login password. Alternatively PKCS#11 configuration is present in the Harrier configuration file, the user's (or role's) LDAP entry has both a userCertificate attribute and a

harrierSignIdentity attribute that describes how a particular user certificate related to a private key stored on an HSM. Because of the last point, you either need to upload an existing PKCS#12 object (encrypted with the user's login password) to the userPKCS12 attribute or you need to configure Harrier to automatically generate a CSR and key pair (see below), in order to be able to obtain a certificate for the user. (HSM/PKCS#11 private keys and certificates can be managed with Cobalt.)

Encryption for a message depends on sender's ability to encrypt, as well as the ability of each recipient to receive encrypted messages. In order to be able to encrypt a message, all of the following conditions must be met: 1) Harrier activation includes "encrypt" sub-feature; 2) encryption must be enabled for the domain (in XML configuration) or for the specific user (in LDAP); 3) the domain has LDAP configuration and all users can bind to Directory using the same username/password as used for IMAP login; 4) the logged in user's LDAP entry must have a userPKCS12 attribute, containing the private key and corresponding certificate, encrypted with the user's login password. Alternatively PKCS#11 configuration is present in the Harrier configuration file, the user's (or role's) LDAP entry has both a userCertificate attribute and a harrierSignIdentity attribute that describes how a particular user certificate related to a private key stored on an HSM. 5) each recipient's LDAP entry must contain userCertificate attribute (an unexpired certificate) that can be used to encrypt the message.

Triple-wrap is the process of sending a message that is signed then encrypted then wrapped in another signature. In order to do triple-wrap, the sender must have the necessary certificates (and corresponding private keys) for signing and encryption but it is not needed for the user to have signing and encryption enabled in the configuration. Triple-wrap for a message depends on sender's ability to triple-wrap (sign $\rightarrow$ encrypt $\rightarrow$ sign), as well as the ability of each recipient to receive encrypted messages. In order to be able to triple-wrap a message, all of the following conditions must be met: 1) Harrier activation includes "encrypt" sub-feature; 2) triple-wrap must be enabled for the domain (in XML configuration); 3) the domain has LDAP configuration and all users can bind to Directory using the same username/password as used for IMAP login; 4) the logged in user's LDAP entry must have a userPKCS12 attribute, containing the private key and corresponding certificate, encrypted with the user's login password. Alternatively PKCS#11 configuration is present in the Harrier configuration file, the user's (or role's) LDAP entry has both a userCertificate attribute and a harrierSignIdentity attribute that describes how a particular user certificate related to a private key stored on an HSM. 5) the user's LDAP entry must not have harrierSmimeSign and/or harrierSmimeEncrypt LDAP attributes set to FALSE; 6) each recipient's LDAP entry must contain userCertificate attribute (an unexpired certificate) that can be used to encrypt the message.

Note that when using S/MIME encryption together with Draft & Release procedure, Harrier needs to be able to encrypt the message to each recipient plus the selected Releaser(s) and Reviewer(s).

6.1.2.2 Automatic CSR generation/certificate import

In order to enable automatic private key and CSR generation with the subsequent installation of an issued certificate, you first need to enable S/MIME signing and/or encryption and second set the following 3 options:

```
<smime_auto_generate_csrs>on</smime_auto_generate_csrs>
<smime_csr_path>$(isode.dir.var)csr</smime_csr_path>
<smime_pkcs12_path>$(isode.dir.var)pkcs12</smime_pkcs12_path>
```

`smime_csr_path` specifies filesystem directory where automatically generated S/MIME CSR files (and corresponding PEM files) will be written. The directory must exist.

`smime_pkcs12_path` specifies filesystem directory where automatically generated S/MIME PKCS#12 files (initially containing private key) will be written. The directory must also exist.

Provided these three options are set, then whenever a user logs in, Harrier will check to see whether a certificate is configured for the user. If not, Harrier will generate a key pair and corresponding CSR, which will be written to <smime_csr_path> and <smime_pkcs12_path> respectively. (Note that the system administrator doesn't need access to the <smime_pkcs12_path> directory.) The system administrator should review the <smime_csr_path> directory for CSR files (which are named using the user's canonical email address). These CSRs should be submitted to an appropriate Certificate Authority or be processed by a tool like Sodium CA, and once corresponding certificates have been issued, they should be placed in the <smime_csr_path> directory as PEM files. (Note that original CSR files must not be deleted, as they are used by Harrier.) Subsequently, when the user logs in, Harrier will import the certificate and private key as a userPKCS12 attribute value, and also import the certificate as the userCertificate value. Following this, the user will be able to sign and encrypt messages, and to receive encrypted messages from other users.

Note that when using S/MIME encryption in a configuration where users are allowed to assume various roles, the userPKCS12 attribute is read from the logged in user entry, while userCertificate attribute is read from role (recipient) entries. For example, let's consider a configuration where user user1@example.com is able to select one of 2 roles upon login: postmaster@example.net and support@example.net. In order for other senders to send encrypted messages to postmaster@example.net the postmaster@example.net's LDAP entry must contain certificate with postmaster@example.net email in the userCertificate attribute. Similarly for support@example.net. When S/MIME encryption is used together with the automatic CSR generation/certificate import feature, this means that postmaster@example.net's certificate will be imported into user1@example.com's LDAP entry. It needs to be manually copied to the postmaster@example.net entry.

6.1.2.3 S/MIME verification/decryption

This happens automatically for all received messages.

In order for S/MIME signature verification to work Harrier needs to be configured with Trust Anchors (the certificates of Certificate Authorities (CAs) which are trusted) as well as any other CA certificates that may be needed when building a certificate chain from a Trust Anchor to sender's end-entity certificates. List all Trust Anchors in `smime_trusted_certificate` XML elements under the top level `harrier` element. List all intermediate Certification Authorities in `smime_intermediate_certificate`.

In order to be able to decrypt messages, Harrier needs to access to user's private key stored in LDAP Directory in the userPKCS12 attribute. This uses the same configuration that is needed for encryption. Note that decryption can be performed even if the certificate that corresponds to a private key is expired.