SWIFT-ADMIN-6.2

Swift Administration Guide

Table of Contents

Chapter 1 Usage..... 1

This chapter describes how to use Swift Administration Interface.

Chapter 2 TLS Configuration..... 3

This chapter describes how Swift is configured to use TLS.

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

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

Isode software is © copyright Isode Limited 2002-2026, all rights reserved.

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

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

This manual is © copyright Isode Limited 2026, all rights reserved.

This guide is published in support of Swift 6.2. It may also be pertinent to later releases.

1 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	*/var/isode/log*
Program and macro names	isode.xmppd
Input to the system	`cd newdir`
Cross references	see Section 2, “Support queries and bug reporting”
Additional information to note, or a warning that the system could be damaged by certain actions.	Notes are additional information; cautions are warnings.

2 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: isode.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: https://www.isode.com/news/

3 Export Controls

Swift uses TLS (Transport Layer Security) to encrypt data in transit. This means that Swift is 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 from Isode.

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

XMPP servers are generally deployed with TLS for security reasons and Isode strongly recommends that all operational deployments use the export-controlled TLS feature.

You must ensure that you comply with these Export Controls where applicable, i.e. if you are licensing or re-selling Swift. Swift is subject to an Isode license agreement and your attention is also called to the export terms of the license agreement.

Chapter 1 Usage

This chapter describes how to use Swift Administration Interface.

1.1 Getting started

When first setting up Swift, you will need to complete a bootstrap process using Swift Administration Interface and then activate the instance of Swift with an Isode Product Activation Key.

In order to setup the administrator account, you will need to start the Swift service. This should happen automatically after installation, but failing that, you can start the service from the Windows Services app on Windows, or via systemctl on Linux (as swift). When running Swift Administration Interface for the first time it will be configured with a self-signed TLS certificate, and can be accessed on port 2080 (e.g. https://localhost:2080 from a browser on the local machine). Note that some web browsers may present a warning where self-signed certificates are used that you will need to acknowledge before the browser will load the page (if you intend to run Swift with TLS activated, you will likely want to replace the self-signed certificate when the product has been activated via Swift Administration Interface with a production-ready one).

After starting the service and visiting port 2080 for the first time, you will be presented with the bootstrap screen. This will ask for a username and password that you wish to use to manage the service. After supplying these details, you will be prompted to sign in.

After signing in, you will be prompted to activate Swift. To do this, click on the "Request activation key" option, and send the generated key to Isode Support (isode.support@isode.com). You will then be sent a Product Activation Key for your Swift installation that can be used to activate the product by clicking on the "Enter activation key" option. Paste in the key and click "Activate". Once complete, you will be presented with the administration interface.

1.2 Managing the service

Once signed in via Swift Administration Interface, options for Swift are presented in a form. Unless stated otherwise, changes to these options will apply immediately, without requiring a restart of the service.

Each option is presented alongside a description that explains what it does, and where applicable, options can be reverted to their default values by clicking on the "Use default" checkbox. Changes to the configuration must by applied by clicking the submit button, or cancelled (in which case, any changes to the configuration will be discarded).

1.3 Security Labelling (STANAG 4774/4778)

Swift supports security labelling of messages in accordance with STANAG 4774 and STANAG 4778. When security labelling is enabled, users can select a classification label when composing messages, and received messages display their classification marking with the associated colours. Labels are transported using the STANAG 4778

envelope format, which carries a STANAG 4774 Confidentiality Label validated against the configured security policy. When this is enabled, XEP-0258 labels are not used.

To configure security labelling:

1. Enable the feature by setting the *Enable S4774 Security Labelling* option in Swift Administration Interface. The *Server Security Policy* and *Server Security Label Catalog* fields will then appear.
2. Upload the STANAG 4774 Security Policy Information File (SPIF) XML file using the *Server Security Policy* field.
3. Optionally, upload a label catalog XML file using the *Server Security Label Catalog* field. The catalog describes the set of security labels that clients may select when composing messages, and must be compatible with the SPIF provided above. If omitted, Swift will automatically request a label catalog from the XMPP server using XEP-0258 Security Labels when each conversation is opened.

Configuration changes take effect for new sessions; existing sessions continue to use the policy and catalog that were active when they were established.

> **Note:** Disabling S4774 does not immediately affect sessions that are already connected. Users who are logged in at the time of the change will continue to use the security policy and label catalog that were active when their session was established, until they disconnect and reconnect.

1.4 Firewall Considerations

By default, Swift will listen on a select number of TCP ports. All ports are configurable through the Swift Administration Interface. Depending on the environment, firewall rules will need to be created to block or expose these ports. Isode recommends that ports are only exposed as-needed to provide the service, as in the following table.

Table 1.1. TCP ports used by Swift

Port	Recommended exposure	Description
2080	Administration only	Port on which the Swift Administration Interface is reachable. This port should not be exposed generally, but only to machines that will be used to administer the service.
1080	Users	Port on which the end-user web interface of Swift is accessible. By default this port serves the HTTPS protocol.

Chapter 2 TLS Configuration

This chapter describes how Swift is configured to use TLS.

Transport Layer Security (TLS) is used by Swift to provide privacy, data integrity and authentication for client-to-server connections. Use of TLS requires the configuration of three main classes of information:

Public Key Certificate Chain
: A Public Key Certificate Chain certifies the ownership of a public key (by the named subject of the certificate).

Private Key
: The Private Key corresponding to the Public Key Certificate Chain noted above.

A Public Key Certificate Chain and corresponding Private Key must be configured before Swift can act as a TLS server (that is to say, the responding side of a TLS handshake) and provide a TLS-encrypted connection.

Private Keys are normally stored in encrypted form. In this case, a passphrase which will be used when decrypting the Private Key must also be configured.

2.1 Client configuration

For the client to start up using TLS the option ‘Use HTTPS for web interfaces’ needs to be set to ‘true’ and a TLS key pair must be selected under ‘TLS Key Pair for HTTPS Interface’. If a key pair is not specified the client will run over insecure HTTP.

Refer to Adding TLS key pairs for instructions on adding TLS key pairs.

2.2 Swift Administration Interface configuration

For the Swift Administration Interface to start up using TLS the option ‘Use HTTPS for the Swift Administration Interface’ needs to be set to ‘true’ and a TLS key pair must be selected under ‘TLS Key Pair for Administrator Interface HTTPS’. If a key pair is not specified the Swift Administration Interface will run over insecure HTTP.

Refer to Adding TLS key pairs for instructions on adding TLS key pairs.

2.3 Adding TLS key pairs

Key pairs can be added either through the key pair selection dialog by clicking on the 'Add New' button or via the 'Key Pair Store' screen by clicking on the 'Add Item' button.

The following fields may be set for each key pair:

Identity or Certificate Chain

Private Key

Passphrase

Key Chain Name

2.4 Key pair store

Uploaded key pairs will be stored in the key pair store. To view this store navigate to 'Stores' > 'Key Pair Store' using the sidebar. A new key pair can be added by clicking on the ‘Add Item’ button. Details of a listed key pair can be viewed by clicking on the ‘View Details’ button next to it. If a key pair is no longer in use it can be deleted by clicking on the ‘Delete’ button.

2.5 XMPP Connection TLS Configuration

The *Minimum TLS Version for XMPP* option sets the minimum TLS protocol version accepted for connections from &swift; to the XMPP server; the default is TLS 1.2. This option has no effect on already active sessions, and changes will only apply to new sessions established after the change is made.