

SWIFT-USER-6.2

Swift User Guide

Isode

Table of Contents

Chapter 1	Usage.....	1
------------------	-------------------	----------

This chapter describes how to use the Swift chat interface.

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.

Chapter 1 Usage

This chapter describes how to use the Swift chat interface.

1.1 Signing in

As a user of Swift, the first step is to sign in. This requires supplying an XMPP user address and password on the Sign In screen and clicking "Sign In".

1.2 Changing and viewing status

Swift offers the ability to set and view status and optional status messages. This is sometimes known as "presence" and can be a useful way to gauge the availability of other users of the XMPP service. Statuses are presented in Swift in the chat list, alongside the avatar of each entry, as well as in the header of the chat. Status messages are presented in the header of the chat. To set your status in Swift you can click on your avatar in the top-left corner. Here you will be able to choose between three statuses; "Available", "Busy" and "Away". To set a status message in Swift you can click on the placeholder text below your username and type your message.

1.3 Running ad-hoc commands

Swift has the ability to browse and run ad-hoc commands on an XMPP server. To access this feature, click on your avatar in the top-left corner and choose "Run commands". By default, Swift will present a list of commands available on your local XMPP server. You can also choose to browse commands on remote XMPP servers by supplying a server address in the input at the top of the dialog. Depending on the command that you choose, Swift will either prompt for additional input, or present the output of the command.

1.4 Managing notifications

When you first sign in to Swift, you will be presented with an option to enable system notifications at the top of the chat list. You will receive notifications when mentioned in rooms, or when receiving direct messages from other users of the XMPP service. If you choose to decline notifications, you can enable them later by clicking on your avatar in the top-left corner, and selecting "Preferences". Here you will find a toggle to enable notifications, and optionally notification sound. If these options are blocked by system permissions, a warning will be displayed under the option in question.

1.5

Managing chats

In order to add a contact, click the "Add" button in the chat list (denoted by the plus icon) and select "Add contact". This will open a dialog where you can enter the XMPP address of the user that you wish to add. Contacts that you add will appear under "People" in the chat list.

In order to join a room, click the "Add" button and choose "Join room". This will open a dialog where you can either enter the XMPP address of the room you wish to join, or browse for available rooms on the selected server (by default, the selected server is the local XMPP server). Rooms that you join will appear under "Rooms" in the chat list.

By default, Swift will use your global nickname when joining a room, but you can choose to override this by clicking the down arrow next to 'Join room', selecting 'Join with options...' and supplying a per-room nickname in the 'Join with options' dialog. If no override is specified, the nickname used when joining a room is the one specified in the user's profile at the time the user logs in to Swift, and this can be updated by logging out and back in again. To modify or set a per-room nickname after joining the room, open the context menu behind the vertical ellipsis for the room's entry in the chat list, and select 'Edit join options'. Changes to the user's per-room nicknames will take effect immediately.

Some rooms may require a password to join. To supply a password when joining a room, use the 'Join with options' dialog and enter it in the 'Room password' field. To later change the password used for a room, open the context menu behind the vertical ellipsis for the room's entry in the chat list, and select 'Edit join options' (note that if the 'Room password' is left blank the existing password will not be changed - to remove an existing password select 'Clear Password').

Some chats that you add may exist in a "pending" state until the other party chooses to either accept or decline your request to add them to your chat list. Similarly, you may receive requests from other users to add you to their chat list too. This is presented as a "Pending request" in the chat list, where you will have the option to either accept or decline their request. If you decline the request, the entry is removed from the chat list (the requestee will simply see the request as pending in this instance). If you accept the request, the "Pending" label is removed from the entry and the chat will persist in your chat list.

To search through your available chats, Swift provides a search palette in the top bar. Here you can type to filter through the chats in your chat list, and hit "Enter" to quickly switch to the highlighted result. There are also groupings available in the chat list that you can use to filter chats by type ("All", "People" and "Rooms").

To remove a chat from your chat list in Swift you can either select the chat, and then click on the menu icon in the top-right of the chat, or you can hover over the entry in your chat list and click on the menu icon there. Here you will find an option to exit the chat and remove it from your list of chats.

1.6

Participating in chat

In order to send or read messages in Swift, you first need to select the chat from the chat list. From here you will be able to read a log of recent messages in the chat, as well as send messages into the chat via the text input at the bottom. To send a message, simply type and hit "Enter" (or click on the send icon in the input). On chat services that support file upload, you can also attach files to your messages by clicking on the "Attach file"

option in the input, by dragging and dropping files over the chat, or by copying and pasting files or images into the input.

When a user is participating in a one-to-one chat with another user, their typing activity is indicated to the other party via a typing hint below the chat input.

Messages that you receive that have not yet been read will be indicated in the chat list as "Unreads", along with a count of the unread messages. If you have the chat open and visible, or enter into an unread chat, visible unread messages will be marked as read. If the number of unread messages exceeds the available screen space, the chat will automatically scroll to the first unread, at which point you can either scroll to the bottom, or click "Mark all read" to immediately mark them as read.

1.7

Viewing archived chat history

When you open a chat in Swift, you will be presented with a log of recent messages in the chat. Depending on the server configuration, you may be able to view older archived messages. If the chat supports message archiving, you will see a "Load more messages" button at the top of the chat allowing you to view earlier messages.

1.8

Submitting and viewing FDP forms in rooms

Swift also has the ability to submit and receive Form Discovery and Publishing (FDP) forms in rooms. Where supported, the chat input will include a "plus" icon that you can click to either attach files, or submit forms. When submitting forms, you will be presented with a list of available form templates, and you can choose to fill out the form and submit it to the chat. When a form is submitted to a chat, it will be presented in the chat log.

1.9

Color modes

Swift can display its User Interface (UI) using either a light or dark color mode. To select between modes, navigate to the preferences dialog by clicking the arrow next to your status icon and selecting 'Preferences' from the menu. On the preferences dialog, there will be a 'Color mode' label near the top with three option (radio) buttons underneath. Each option button represents light mode, dark mode and system default respectively. When set to system default, the color mode will be inferred using the current selection in the underlying OS or device settings.

1.10

User Preferences

Certain preferences in Swift can be stored for each user, allowing them to persist after signing out, and synchronise across multiple client instances. This is facilitated with the Personal Eventing Protocol (XEP-0163) extension, via a Publish-Subscribe node on the XMPP server. The following preferences are currently saved to the user's storage area on the server:

Table 1.1.

Preference	Description
Compact roster	When selected, this option hides avatars for contacts / rooms in the roster allowing it to fit into a smaller screen space.
Hide offline chats	When selected, this option hides offline contacts from the roster.

1.11 Trellis board mode

Swift includes an option to enable a trellis board layout. This option can be enabled in your user preferences, and changes the view from a single chat view, to a board that can contain multiple pinned chats. To pin a chat, click on the chat's entry in the chat list. Once pinned, chats can be interacted with much like in the single chat view. To unpin a chat, click on the close icon in the corner of the card.

To expand the board such that it fills the viewport, there is an option in the top-right corner. This will toggle everything except the board, allowing for a focused and spacious view of the chats that you have pinned.

Note that pinned chats can be reordered by dragging from the grab handle in the top-left corner of the card.

1.12 Keyboard shortcuts

Table 1.2. Shortcuts used by Swift

Shortcut	Action	Description
ctrl/cmd+k	Open search palette	Opens the search palette allowing you to search for users and rooms.
alt/option+shift+up	Open next unread chat	Opens the next chat containing unread messages.
alt/option+shift+down	Open previous unread chat	Opens the previous chat containing unread messages.

1.13 Muting chats for specified duration or indefinitely

In order to mute a chat in Swift, you can click on the 'Mute room/Mute conversation' option in the chat menu. The chat menu is accessed either by clicking on the menu icon in the top-right of the chat or by hovering over the chat in the sidebar and clicking on the menu icon there. After clicking on the 'Mute room/Mute conversation' option, you will be presented with a dialog where you can choose to mute the chat for a specific amount of time or indefinitely until you choose to unmute it. Muted chats are indicated by a muted icon and are faded out. The time until the chat is unmuted is displayed within a tooltip that's shown when you hover over the muted icon and within the chat menu. When a chat is muted, notifications will not be shown for the chat, and it will also be

prevented from being displayed as unread within the sidebar. To unmute a chat, simply click on the 'Unmute room/Unmute conversation' option in the chat menu.