

XUXA-16.6

XUXA Manual

Isode

Table of Contents

Chapter 1	Overview.....	1
	XUXA (X.400 User (X) Agent) is a demonstration cross-platform X.400 User Agent, provided by Isode to help Isode customers demonstrate and test Isode products and APIs. XUXA is a Java GUI, with look and feel modelled on the open source Thunderbird client.	
Chapter 2	Installing and running XUXA.....	3
Chapter 3	Configuration.....	4
Chapter 4	Preferences.....	5
Chapter 5	The Address Book.....	12
Chapter 6	Basic Operation.....	14
Chapter 7	Bodyparts (attachments).....	16
	This version of Xuxa provides better handling of bodyparts (attachments), both while composing a message and displaying bodyparts of received messages.	
Chapter 8	MTS Parameters.....	18
Chapter 9	The Sent folder, Outbox folder and Testing.....	20
Chapter 10	Log View.....	21
Chapter 11	Troubleshooting and FAQ.....	22
	Here are the typical problems you can encounter when running Xuxa.	
Chapter 12	Advanced Topics.....	24
Chapter 13	AMHS Features.....	27
Chapter 14	Running Multiples Copies of Xuxa in One Machine.....	29
Appendix A	Notes and Change Log.....	30

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 2002-2017, 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 2017.

1 Software version

This guide is published in support of Isode Xuxa R16.6. 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 Xuxa, a user agent provided to enable testing of the M-Switch products.

3 Related publications

Related topics are discussed in the volumes of the Isode documentation set listed below.

Volume	Title
SWADM-16.6	<i>M-Switch Administration Guide</i>
VAUADM-16.6	<i>M-Vault Administration Guide</i>

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

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/news-signup.php> [<http://www.isode.com/company/contact.php>]

7

Export controls

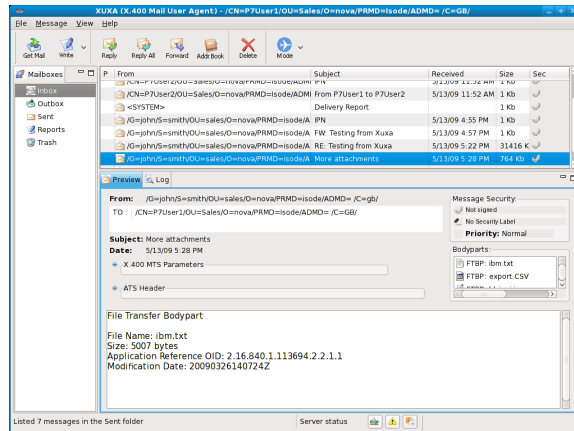
Isode products use the TLS protocol to encrypt data on connections. By default these operate with 56 or less bits of encryption. If you license the higher grade encryption (HGE) option for Isode products they will operate with higher levels of encryption which makes them subject to UK Export controls pursuant to Isode's open general license *EU GEA 001 (EU001)*.

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

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

Chapter 1 Overview

XUXA (X.400 User (X) Agent) is a demonstration cross-platform X.400 User Agent, provided by Isode to help Isode customers demonstrate and test Isode products and APIs. XUXA is a Java GUI, with look and feel modelled on the open source Thunderbird client.



1.1 Goals

Primary goals:

- To demonstrate features of the Isode API and Server products that cannot otherwise be easily shown.
- To help customers make use of these features in their own applications by providing a good sample application.
- To help customers set up, evaluate and demonstrate Isode products.

Secondary goals:

- To help Isode and Isode customers test Isode products.
- To provide a source code base which Isode customers can purchase from Isode to build specialized applications using some or all of the XUXA code.
- To showcase a pure “client only” application that holds all data on the server, which it accesses with standard protocols.

XUXA enables:

- Sending X.400 messages and probes using P7 to an X.400 Message Store
- Sending X.400 messages and probes using P3 to an X.400 MTA.
- Listing, fetching and deleting messages using X.400 P7.
- Retrieving messages using X.400 P3. (Note that these are held in memory, and not retained on exit). This is useful for testing M-Switch X.400 without a Message Store.
- Setting and displaying message priority.
- Control of delivery reports and IPNs (Inter-personal Notifications) on a per-recipient basis.
- Display of delivery reports and IPNs.
- Auto-generation of IPNs on message reception.

- Generate, Display and Save text encoded body parts (IA5 and General Text (with choice of character set)).
- Generate, Display and Save FTBP encoded body parts.
- Generate, Display and Save binary body parts.
- Recognize G3Fax and forwarded messages.
- Control of most X.400 Message Transport Service parameters.
- Generate and display X.411 Security Labels, using the DSA's Catalogue.
- Display the Trace Information and Internal Trace Information of received messages.
- Forwarding X.400 messages.

1.2 Non-goals

Things we are not trying to do or going to do:

- Build a User Agent product that we will sell.
- Implement local storage of data in XUXA.
- Implement features that do not directly help the goals (e.g., printing)
- Implement body part viewing or editing capabilities (e.g., don't display or edit flight plans – just show as a big icon)

1.3 Capabilities

XUXA provides access to a range of X.400 capabilities. XUXA has “modes”, that provide capabilities for markets that make extensive use of X.400. This description groups features by mode. Some features are specific to the markets (and mode). Others are general purpose X.400 features, of particular interest to the market.

Chapter 2 Installing and running XUXA

2.1 Installing Xuxa

2.1.1 Windows

Xuxa is shipped as part of the M-Switch package. Even if you want to run Xuxa on a stand alone machine without the servers, you will have to install M-Switch on the machine and select the **Client Only** option.

2.1.2 Linux

Xuxa is shipped as an RPM, and should be installed in exactly the same way as the other Isode RPMs. If you want to run it as a client, the following RPMs are the minimum you should install:

- *ISDxuxa-16.3v0-0.i386.rpm*
- *ISDx400-16.3v0-0.i386.rpm*
- *ISDbase-16.3v0 -0.i386.rpm*
- *ISDpp-16.3v0 -0.i386.rpm*
- *ISDtps-16.3v0 -0.i386.rpm*

2.2 Running Xuxa

2.2.1 Windows

You can running by choosing: **Start** → **Program Files** → **Isode** → **XUXA**

Depending on your settings, you may have to run Xuxa as an Administrator.

2.2.2 Linux

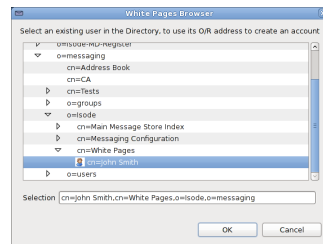
Run `/opt/isode/bin/xuxa`

Chapter 3 Configuration

Xuxa is configured using Java Preferences and not through configuration files. These preferences should be edited with the build-in editor, as their location depends on the operating systems. For example, on Windows they are saved in the Windows Registry.

The easiest way to configure Xuxa is by using the setup wizard that is run when no configuration is found. To do that, before you run Xuxa, configure a P3 or P7 account first using Mconsole.

Then run Xuxa, and when prompted, enter the host name where the DSA (M-Vault) is running. Xuxa will create an Isode Bind Profile to connect to the DSA on the host given and then it will then try connect to the DSA. If the DSA is running in the host name provided and there are no connection problems, the connection will succeed, and a directory browser dialogue like the one shown below will appear.



Select the user you want to create an account for, in the example above is **John Smith**. After you click on **OK**, you will be prompted for the user's password. Enter the password and click on **OK**. Xuxa will then do the following:

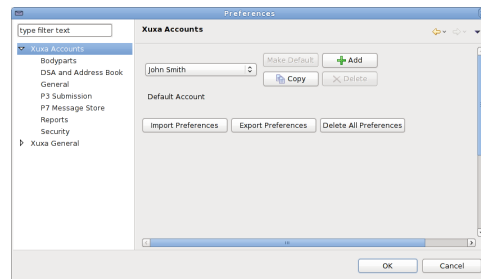
- Use the Directory Bind Profile for this user
- Use the O/R address of the selected entry, for both P3 and P7
- Use the password provided for both P3 and P7
- Attempt to bind to the X.400 P7 Message Store with the password provided

In case the bind doesn't work, you can use Xuxa's built-in **Preference editor** to set the values and try again. Many preference values are only read at start-up, so you will need to restart Xuxa for the new values to take effect.

Chapter 4 Preferences

4.1 Accounts

To set your preferences, select **File** → **Preferences** and you will see a window similar to the one below.



Xuxa can only be connected to one account at a time, but can have several accounts configured, and one of them is always a default account. The default account is the one that is used at startup. You can make Xuxa prompt for the account to use by selecting the option **Prompt to select account at startup** in the Xuxa **General** tab.

You can also **Add**, **Delete** or **Copy** a new account by clicking on the appropriate button. In case you add a new account, you will be prompted for the account name. After that, you should expand the other tabs under **Xuxa Accounts** and set the values for this new account.

In case that the new account has a lot in common with one of the accounts you have configured, you can click on **Copy**, enter the name and then modify the relevant data.

It is also possible to export the preferences to a file, which can then be imported later on, on the same machine or in another machine.

If you want to completely wipe out all Xuxa preferences, you can click on the **Delete All Preferences** button. The next time you run Xuxa, the setup wizard will be shown. This is also useful if you want to upgrade an old account from an previous (incompatible) version.

4.2 Bodyparts

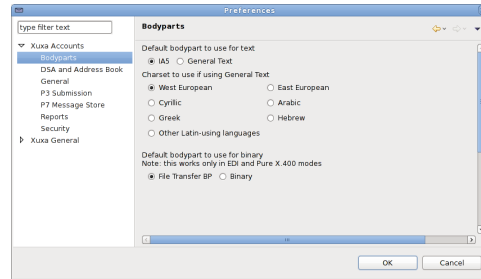
When you compose a message, it is possible to attach files to it, and so create a message with several bodyparts.

By default, text messages will be sent using IA5 bodyparts. All X.400 User Agents can deal with IA5 text, and so it's a sensible default, but bear in mind that some characters are not included in IA5, so the content of your message can be converted.

If you want to change the default to use **General Text**, you can change it in the **Bodyparts** preference page. You can also choose the encoding of the charset to use for General Text.

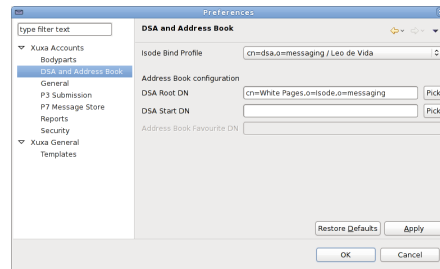
Finally, the default bodypart to use for binary attachments is the **File Transfer BP** (bodypart). Only for EDI and Pure X.400 modes, it is possible to change the default, and to use the Bilaterally Defined (**Binary**) bodypart. All files attached this way will be added

as a Binary bodypart, and so they won't have any extra information, like the file name or size.



4.3 DSA and Address Book

The other fundamental set of parameters that need to be configured is the X.500 Directory and Address Book information. This is needed so that Xuxa can find the addresses (O/R Names) of users to which it can send messages.



Here, the only mandatory attribute is the **Isode Bind Profile**, which could be the one generated by Xuxa, or it could have been generated by another application, like MConsole or Sodium. Sodium is the best application to use to manage the Isode Bind Profiles.

Xuxa doesn't prompt for a password to connect to the DSA, so it connects to the DSA using an anonymous bind. So you will be able to connect to the DSA and browse your address book. However, if you want to edit the information in the address book, for example to create a new entry or modify an existing one, you will need to use another application, like Sodium.

You can use the **Pick** button to bind anonymously to the DSA and navigate to the user you want to use, and automatically set the **DSA Root DN** and **DSA Start DN** parameters.

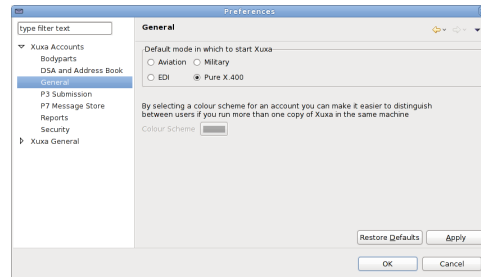
If the **DSA Root DN** is not set, when opening the Address Book, you will connect to the DSA and start browsing from the top of the DIT (The World). However, if you set the **DSA Root DN** the Address Book will start showing the DIT from the chosen Root DN.

Similarly, if you set the **DSA Start DN** attribute, your selected entry will be shown in the browser by default. These DN's can be entered manually, pasted or selected via the **Pick** button if a DSA connection is available.

4.4 General

If you want to change the default mode in which Xuxa starts, change it in the **General** page.

The colour scheme settings is disabled at present.



4.5 P3 Submission

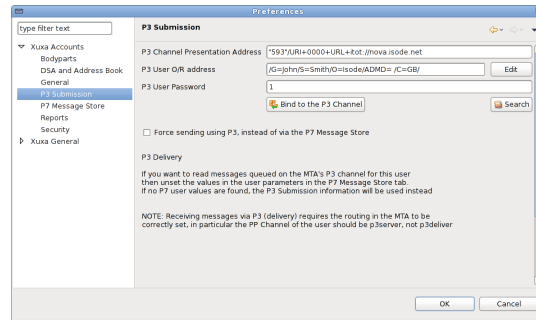
If you don't have access to a P7 Message Store, or if you want to test P3 Submission and Delivery, you can still use Xuxa, although P7 is preferred to P3.

You have to configure the connection information in the **P3 Submission** preference page. When you enter the correct information and check the **Force sending using P3** checkbox, all new messages will be sent using the P3 protocol directly to the MTA, instead of using P7 and sending it to the P7 Message Store.

Since version 4.1, it is possible to receive (accept the delivery) of the messages via P3. This allows Xuxa to work without a P7 Message Store. But as Xuxa doesn't store messages on disk, messages received via P3 will only be available as long as Xuxa is running. Once you restart the program, the messages are gone, as the MTA no longer has them, and Xuxa didn't save them.

To configure P3 delivery in the preferences, you will have to enter the correct **P3 Channel Presentation Address** (something like '593"/URI+0000+URL+itot://nova.isode.net), the **P3 User O/R address** (the same as the P7 Message Store user O/R address) and the **P3 User Password**.

Note: Bear in mind that once a user is created in MConsole as a P7 Message Store user, all messages will be delivered to the P7 Message Store, and therefore they won't be available for P3 Delivery. You should use users that have been created in MConsole as P3 users.



The **P3 Channel Presentation Address** is the full address of the P3 Server channel that the program will connect to in order to send and receive messages. If it is an Isode P3 Server channel, it will typically be listening on port 102, so it can be omitted, and will have a transport selector of “593”. Use MConsole’s **Switch Configuration View**, to look at your MTA’s the p3server channel’s **Presentation Address** in the **Inbound** tab.

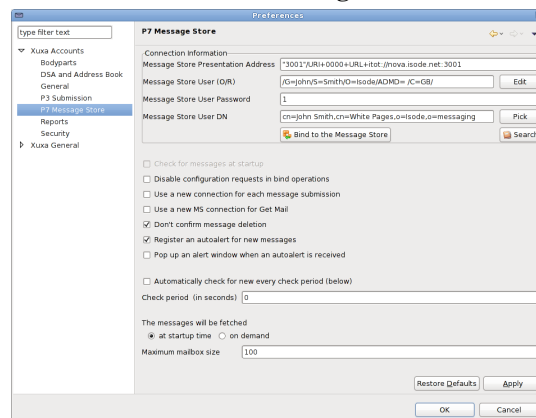
The **P3 User (O/R)** is the O/R address of the P3 user that Xuxa is going to use to send and receive messages. If you want to know what users are available and know their passwords, configure the DSA and access the Address Book by clicking on the **Search** button, or use EMMA to connect to the DSA and search for the available users.

You can also paste the value in the text editor, or use the **Edit** button to invoke the O/R address editor.

The **P3 User Password** is the password used for the **P3 User (O/R)** mentioned above. You can find this if you edit the user in EMMA, and check the MTS Password (P3).

4.6 P7 Message Store

To configure the P7 Message Store, select **File** → **Preferences** expand the **Xuxa Accounts** node, and select the **P7 Message Store** tab.



The **Message Store Presentation Address** is the full address of the P7 Message Store that the program will connect to in order to send and receive messages. If it is an Isode Message Store, it will typically be listening on port 3001, and will have a transport selector of “3001”.

The **Message Store User (O/R)** is the O/R address of the P7 Message Store user that Xuxa is going to use to send and receives messages. If you want to know what users are available and know their passwords, configure the DSA and access the Address Book by clicking on the **Search** button, or use EMMA to connect to the DSA and search for the available users.

You can also paste the value in the text editor, or use the **Edit** button to invoke the O/R address editor.

The **Message Store User Password** is the password used for the **Message Store User (O/R)** mentioned above.

The **Message Store User (DN)** is the DN address of the P7 Message Store user that Xuxa is going to use to send and receives messages.

You can also paste the value in the text editor, or use the **Edit** button to invoke the **DN Picker**, which is a DSA Browser.

If you click on the **Search** button, and you have the correct DSA connection details configured, you will have able to search on your address book, and select a user from the DSA. In that case, both the users's O/R address and DN will be copied to the relevant fields, making switching accounts easier. For example, you can copy an existing account, and then use the **Search** button to quickly change the information.

If you want to connect to the Deutsche Telekom P7 Message Store, select the option **Disable configuration requests in bind operations** as the values returned by the bind operation are invalid and make the bind operation fail. Bear in mind that this means that you won't have access to your sent messages (in the **Sent** folder).

Some P7 Message Stores don't work well when using the same connection to both send and receive message. In that case, select the option **Use a new connection for each message submission** and in that way a new bind to the Message Store will be performed when needed.

Similarly, the option **Use a new MS connection for Get Mail** opens a new connection to the Message Store every time a user clicks on the Get Mail icon. This can be useful if, for example, the connection to the Message Store is lost because it was restarted or the connection was dropped.

The option **Don't confirm message deletion** does just that.

When the option **Register an autoalert for new messages** is set, Xuxa will detect that a new message has been delivered, and will fetch it automatically in the background.

The options **Pop up an alert window when an autoalert is received** and **Don't confirm message deletion** will do just that.

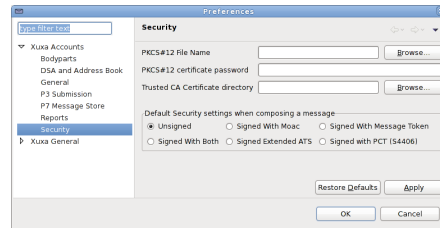
The other options are there to set up timers to make the program check every *X* number of seconds.

If you are planning to test the X.400 Message Store with very large mailboxes, the option **The messages will be fetched: at startup time or on demand** allow you to optimize the MS operations, as if configured to be fetched on demand, messages will only be fetched when the user wants to see them. The default behaviour is to read all the messages at startup time.

A relatively new feature in XUXA is the ability to configure the maximum number of messages that the program will be able to handle. This can be configured with **Maximum mailbox size** the default value of 0 means that the internal limit will be used instead. Bear in mind that when the maximum is reached, all new messages are ignored.

4.7 Security

The following settings are optional, but must be set if digitally sign your messages, or to check the signatures of received X.400 messages.



The value **PKCS#12 File Name** is the fully qualified file name containing the PKCS#12 certificates.

The value **PKCS#12 certificate password** is the passphrase protecting the private key in the PKCS#12 file. The need for this attribute can be removed by creating a file in the same directory as your PKCS#12 file with the same name and the *pphr* suffix. This file must contain the passphrase with no CR or LF.

The **Trusted CA Certificate directory** is a directory name which contains a set of CA certificates, whose Certificate are to be trusted when verifying signatures. If you want to make sure all your messages are signed (or signed to the Extended ATS), you can select the default security settings to be used when composing a new message.

There are several default security settings available:

Unsigned

No digital signatures are added to the message

Signed with MOAC

This provides a Message Originator Authentication Check.

Signed with Message Token

This provides content integrity, and message sequence integrity.

Signed With Both

This is a combination of **Signed with MOAC** and **Signed with Message Token**, and provides all the features of both.

Signed Extended ATS

This is the recommended way of digitally signing Aviation messages according to the ICAO SARPs.

Signed with PCT (S4406)

PCT is Protected Content Type. This is the recommended way of digitally signing STANAG 4406 messages.

See more detailed information in the **Security** section.

4.8 Xuxa General

This page allows you to set some general preferences. The checkbox **Prompt to select account at startup** can be useful in a demo situation, when you have several accounts configured and want to choose with one to use before connecting to an account.

4.8.1 Default way to presenting O/R Names

Most X.400 User Agents display users as O/R addresses, but it's possible to use Directory Names (DNs) which are more friendly and provide a good abstraction.

The preference control in this page allows you to choose which one is the default way of presenting O/R Names: as O/R addresses only (the default), as a DN if present, or as both. In this last case, what is shown is the DN, followed by a \$ and then the O/R address.

Chapter 5 The Address Book

5.1 General Info

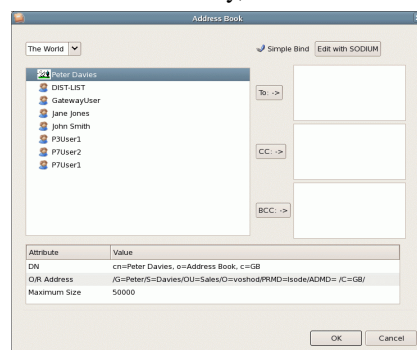
Xuxa doesn't allow you to enter the O/R address manually, this is a design feature to prevent errors due to mistyping and to enforce the concept of Directory based user agents. This version connects to the Directory (DSA) to allow Xuxa users to browse and search for O/R addresses.

The information in the Address Book has to be created and maintained by an external program (like MConsole or Sodium).

When you create a user in Mconsole, you should also set up a White Pages entry for the user. By default, the White Pages entry are created at the same level of the **cn=Messaging Configuration** entry, but the value can be changed with MConsole.

For example, a typical White Pages DN looks like this: **cn=White Pages,o=Isode,o=messaging**.

When you click on the Address Book icon in Xuxa, if you have configured the DSA information correctly, a window similar to the one shown below will pop up.



This window shows that Xuxa is acting as a Directory User Agent, because the information show is retrieved from the DSA. Once you select an entry, some of the entry's attributes will be shown in the display at the bottom of the **Address Book** window.

In the case shown above, you can see that the user Peter Davies is different from the other users in the window, because it has the **Maximum Size** attribute set, and that means that a different icon is used to show this difference.

If you want to compose a message to a user, simply select it and then click on the **To**, **CC** or **BCC** buttons, and then click on **OK**.

5.1.1 Adding users manually to the Address Book

There are cases when you may want to add a user to the Address Book manually. This is the case, for example, of an external user (that wasn't created with MConsole), or one that is local but was created by a user template that didn't have the White Pages set.

In that case, you have to use Sodium to create the entry in the DSA for that user. In previous versions of Xuxa, you could click on the **Edit with SODIUM** button to launch Sodium. This is no longer available, as Xuxa doesn't store the Isode Bind Profile password, so the connection to the DSA will be unauthenticated and you won't be able to create or modify

entries. Therefore, Sodium has to be launched separately, and an authenticated session needs to be established to create or modify White Pages entries.

Once you have Sodium running, you can easily create a new user by cloning an existing one. Select an existing user, click on the **Clone** button at the bottom of the screen. Then enter the new user as it will appear in the Address Book (a short, easy to identify name), and click on **OK**. A new editor will appear, and you will have to complete all mandatory fields before you can click on **Add**. In case you have used cloning, remember to change the existing O/R address in the **Messaging** tab to the new one. Use copy-and-paste to avoid typing errors.

Chapter 6 Basic Operation

6.1 Listing the content of a mailbox

If you have configured Xuxa correctly, when you run it, the first thing it will do is to automatically connect to the Message Store and list the contents of the mailbox. If there are no messages in the user mailbox, the message table will appear empty.

6.2 Sending a message

You can start by testing sending a message to yourself. Provided that you have your own O/R address entered in the Address Book, you can do this by clicking on the Write icon. A new window appears, click on the small orange book icon and this will make the **Address Book** window pop up.

Select the address of the user you want to compose a message to (yourself ?) and then click on the **To:**, **Cc:** or **Bcc:** buttons. You can also double click on the address to have it added automatically to the **To:** field. After you have finished, you have to click on **OK**.

Enter the **Subject** of the message, and type something on the content editor, and finally click on the Send icon. A confirmation window with **Message submitted** will appear. You will also see the Message Submission Identifier returned by the MTA, and also the submission time. The Message Submission Identifier allows you to correlate sent messages with delivery reports.

6.3 Receive your message

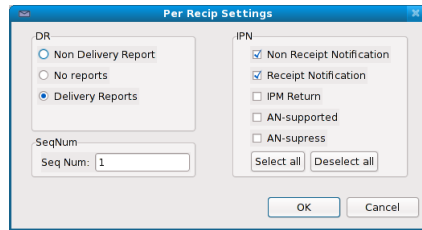
To receive the message you have sent to yourself, click on the Get Mail icon. If you have configured the Register an AutoAlert option in the Preferences, messages will be automatically fetched in the background.

6.4 Send a message requesting Delivery Reports or IPNs

By default, you will only get a Delivery Report (DR) if the message fails to be delivered. To send a message requesting a positive Delivery Report, compose a message like before, and click on the **Per Recip** button that appears on the same line as the destination address.

In X.400 Delivery Reports and IPNs are requested on a per recipient basis, so you can have a combination of addresses, for some you request a positive DR, for some just the negative DR and for some you can request receiving no reports at all.

If you want, you can change the default value for both the Delivery Reports or IPN requests, by setting the value you want in the **Preferences**.



Notifications, also known as read receipts are formally called Inter Personal Notifications (IPNs). The current version of Xuxa allows you to request the IPNs, displays the values of the IPNs received, and since v2.1 it also generates IPNs when receiving a message.

The generation of Delivery Reports (positive and negative) is handled by the MTA, and Xuxa displays the received DRs and IPNs as English text messages.

If you are going to sign the message using a Message Token, you can also set the message sequence number. Start with 1, and increase it for every message you send to the user.

Chapter 7 Bodyparts (attachments)

This version of Xuxa provides better handling of bodyparts (attachments), both while composing a message and displaying bodyparts of received messages.

7.1 Generation

By default, all messages are sent using an IA5 bodypart (which is basically plain text without special characters) with the content of what the user types in the editor. This can be changed in the preferences to make it send **General Text BP** (bodypart) by default. Another preferences option allows you to change the encoding of attachments in Pure X.400 and EDI modes. By default attachments are sent as File Transfer Bodyparts but it can be changed to use Binary.

Attachments can be added to a message, the type of bodypart used depends on the mode Xuxa is in, an also on the file extension of the boypart.

In the case of Aviation mode messages using the Flight Plan, NOTAM and MET templates they are presumed to be *replacing* the content typed. Flight plans are expected to end in *.fpl*, NOTAM messages in *.notam*, BUFR messages in *.bufr* and OPMET messages in *.opmet*.

All other attachments (including BUFR messages) are sent using the File Transfer Bodypart (FTBP).

For demo purposes, if the attachment file name ends with *.text* IA5 will be used, and if it ends in *.gentext* General Text will be used instead (this last encoding is not always working correctly at the moment).

7.2 Reception

If you compose a message with Xuxa or another X.400 user agent, this version of Xuxa will be able to recognize and handle the following bodyparts:

- IA5
- General Text
- Binary
- FTBP

Selecting the bodypart from the list will display the content of the bodypart for IA5 and General Text. For FTBP a summary of the information will be displayed, but not the content itself.

If you want to save the bodypart, you can right click on the bodypart in the Bodyparts list, or select the menu option **Messages** → **Save Bodypart**.

It will recognize but not handle these bodyparts:

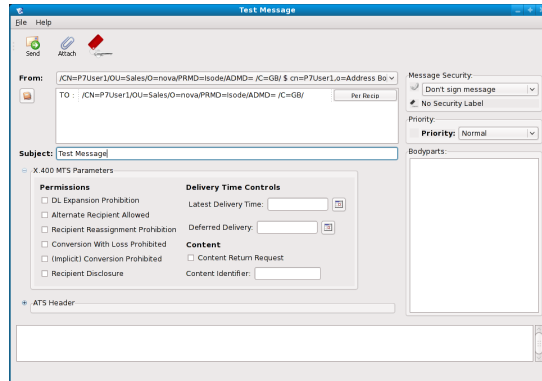
- G3Fax

- Message (as in Forwarded message)

Future versions will allow you to handle other bodyparts better.

Chapter 8 MTS Parameters

XUXA can be used for capability testing and demonstration of M-Switch X.400, using MTS and other features shown below.



The following M-Switch capabilities can be tested using the addressing settings:

- Generation of delivery and non-delivery reports in normal operation to good and bad addresses.
- Generation of delivery reports on operator actions, such as message timeout and deletion.
- Alternate recipient. This gives an alternate address to use, if the primary one fails (either by being invalid, or if delivery is not possible in an appropriate time frame). It can be tested by using an invalid primary address and setting an alternate recipient.

The following M-Switch capabilities can be tested using the **X.400 MTA Parameter** settings:

- **DL Expansion Prohibition.** Will give DR if message is sent to a distribution list.
- **Alternate Recipient Allowed.** If set, the MTS may send the message to an alternate recipient (specified on the receiving system), in the event that the originator specified address is invalid. This can be demonstrated by setting "Admin Alt. Recipient" in M-Switch.
- **Recipient Re-assignment Prohibition.** This prevents the message from being sent to another recipient by use of a redirect. This can be demonstrated by use of a redirected address.
- **Conversion with loss prohibited.** This prohibits message conversion that will lead to information loss. None of the standard M-Switch channels performs a "lossy" conversion. However, any M-Switch conversion channel can be configured to say that it loses information. This can be used to demonstrate this flag.
- **(Implicit) Conversion Prohibited.** This prohibits all conversion. It can be tested by sending a message to a recipient behind a gateway, such as a MIXER gateway, which will be rejected with this parameter set.
- **Recipient Disclosure.** This allows the message recipient to see all the MTS recipients. This cannot be demonstrated with the current XUXA version, as other MTS recipients are not shown.
- **Latest Delivery Time.** This can be used to set the latest time at which a message may be delivered. It can be tested by stopping M-Store X.400 to prevent message delivery.
- **Deferred Delivery.** The capability to defer delivery of a message can be demonstrated easily.
- **Content Return Request.** This requests that the original message is returned with negative delivery reports. Although the current version of XUXA does not display returned content, the capability can be demonstrated by requesting content return on a large message to a bad address. Content return can be inferred from the substantial increase in size of the delivery report.

- **Content Identifier.** This can be set to arbitrary values in a message, and viewed in associated DRs.
- **Original EITs (Encoded Information Types).** XUXA sets this when sending a message. On reception, this can be viewed, to see both body part types, and character set types used in general text. FTBP content types are not shown in the current version of XUXA.

Chapter 9 The Sent folder, Outbox folder and Testing

Since R14.0 the P7 Message Store supports keeping copies of sent messages in the user's mailbox. This feature is now used by Xuxa to show the **Sent** messages, which are the ones that are provided by the Message Store.

This feature allows a new method of testing the messaging infrastructure with Xuxa: one or more **Sent** messages can be selected, and then using the menu option **Messages** → **Copy to Outbox**, it will place a copy of the sent message in the **Outbox**, ready to be sent to the MTA.

In this way, test batches of messages can be prepared and then run at the same time.

Chapter 10 Log View

Right next to the **Preview** tab in the middle of the screen, there's a new tab called **Log**.

Up until R14.4, Xuxa used to log the main operational events to `stdout`. This was available for Unix users, but not for Windows users.

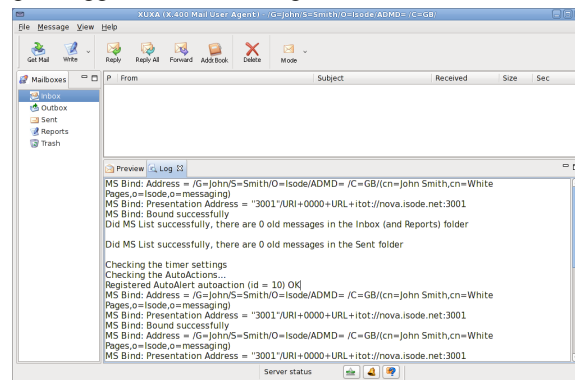
The new **Log** tab captures all the information that used to go to the `stdout`, and displays it in a standard text editor.

What gets logged varies from feature to feature, but it basically shows more low level information that is not normally available in the graphical interface.

Here's a typical logging:

```
MS Bind:Address = /G=John/S=Smith/O=Isode/ADMD= /C=GB/(cn=John Smith,cn=White
MS Bind: Presentation Address = "3001"/URI+0000+URL+itot://nova.isode.net:3001
MS Bind: Bound successfully
Did MS List successfully, there are 0 old messages in the Inbox (and Reports)
Did MS List successfully, there are 0 old messages in the Sent folder
Checking the timer settings
Checking the AutoActions...
Registered AutoAlert autoaction (id = 10) OK
MS Bind: Address = /G=John/S=Smith/O=Isode/ADMD= /C=GB/(cn=John Smith,cn=White
MS Bind: Presentation Address = "3001"/URI+0000+URL+itot://nova.isode.net:3001
MS Bind: Bound successfully
MS Bind: Address = /G=John/S=Smith/O=Isode/ADMD= /C=GB/(cn=John Smith,cn=White
MS Bind: Presentation Address = "3001"/URI+0000+URL+itot://nova.isode.net:3001
MS Bind: Bound successfully
Did MS List successfully, there are 0 old messages in the Inbox (and Reports)
Did MS List successfully, there are 0 old messages in the Sent folder
```

One interesting feature is that you can still use the other aspects of Xuxa as you see the activity being logged. For example, you can send a message, delete a message, or see what gets logged when a message is received.



Chapter 11 Troubleshooting and FAQ

Here are the typical problems you can encounter when running Xuxa.

11.1 The version of Java installed is wrong

A Java interpreter is required, both versions 7 and 8 are supported. If you are running Java 1.4, 1.5 or 1.6 you will get an error like this:

```
java.lang.UnsupportedClassVersionError: com/isode/simplex400api/MStore (Unsup
```

11.2 The Message Store is not running

You will get an error box with this message:

```
It is not possible to connect to the Message Store.  
Reason: x400_ms_open failed: Internal config error
```

Action:

Check that the `presentation_address` key of the configuration file is correctly specified.

If it is, check that Message Store is running on the target system (look for the `pumice` process).

If it is running, check that you can connect from the machine where you are running Xuxa to the port where the Message Store is running.

Check if you can ping the machine, and also if you can telnet to the Message Store port (`telnet myserver.mycompany.com 3001`)

11.3 The Message Store user + password is incorrect

You will get an error box with this message:

```
It is not possible to connect to the Message Store.  
Reason: invalid credentials for connection
```

Action: Check the user and password combination is correct by double checking your configuration file and the configuration of the Message Store using EMMA (if it is an Isode server).

11.4 Cannot open the Address Book

If you get an error box with this message:

The DSA access details have to be configured before trying to access the Address Book.

Action: Follow the instructions, and set the DSA and Address Book information to connect correctly. You can copy the information from another DUA like SODIUM or DDM.

You need to restart the program for the new DSA parameters to take effect.

11.5 The DSA connection parameters are incorrect

This can happen if, for example, you have mis-configured the DSA User name and password.

11.6 Do I need a valid Isode license file to run Xuxa?

No, you don't need an Isode *license.dat* file in the machine for the present version. The copyrights of the program still apply nevertheless.

Chapter 12 Advanced Topics

12.1 Xuxa modes

In order to make the program adapt better to the needs of the different markets that use X.400, Xuxa provides three specialized modes of operation. This can be changed by clicking on the small black triangle that is part of the Mode icon.

Once the mode is selected, two things will happen: one is that the new message templates will change to suit the mode the program is in. The other is that any special behaviour for the mode will be activated.

12.1.1 Aviation mode

This mode allows you to compose AMHS messages by attaching pre-formatted content from a file. There are templates for Flight Plans (*.fpl*), NOTAMs (*.notam*), MET (*.met*) and BUFR (*.buf*). See [Section 12.2, “Use of templates for composing messages”](#) for an explanation.

Once the attachment file is selected, its contents will be used to compose an AMHS message. The standard X.400 attributes can be set as normal, but in Aviation mode also the ATS Headers can be set. The current version (v2.1) supports both Extended and Basic Encoding.

12.1.2 Military Mode

The following Military features are supported:

- In military mode XUXA encodes Inter-Personal Messages as P772 according to STANAG 4406. Arbitrary P772 messages may be imported from files. This enables testing that an X.400 messaging infrastructure correctly supports P772 message transfer.
- Six level military message priority (deferred; routine; immediate; priority; flash; override) can be used.
- One P772 X.400 Heading Extension (Message Type) can be set and displayed in the Military Header section. More heading extensions will be added in the future.

Note: A new Military Header has been added in version 3.0, for now it only has the parameter `Message Type`, which has four values: `Exercise`, `Operation`, `Project` and `Drill`.

12.1.3 EDI mode

XUXA supports two standard X.400 features of specific interest to EDI customers:

- Display of general text body parts, as well as IA5.
- Setting general text to: "West European", "East European", "Cyrillic", "Arabic", "Greek", "Hebrew", "Other Latin-using languages".
- Support of FTBP (file transfer body part), and the ability to send arbitrary named files, such as Word documents.

12.1.4 Pure X.400 mode

This mode should be used when no market-specific behaviour is required, just the pure X.400 standard behaviour.

12.2 Use of templates for composing messages

Once you have set the mode in which you want to work, the templates available will reflect the mode of operation. For example, when operating in Aviation mode, the templates to create a new message will be: Flight Plan, NOTAM, MET and BUFR.

12.3 Digital Signatures

If Xuxa is correctly configured with security information (see [Section 4.7, "Security"](#)), it is possible to request digitally signing a message when in aviation mode.

On reception of a message, if the message is digitally signed, the information about the digital signature appear on the message **Preview** window. You will see the Distinguished Name (DN) of the user who signed the message, and the DN of the Certification Authority (CA).

You can create the PKCS#12 certificates using the Isode Sodium CA tool or any other third party tool. If you are using Sodium CA, you need to create a CA first. After that, generating X.509 Certificates is very easy, as you can use the build-in Directory browser to navigate to the user's White Pages entry, and then use the **Generate X.509 Identity** wizard.

12.4 Security Labels

XUXA can generate and display X.411 Security Labels.

In order to add a security label to a message, you have to click on the Security Label icon on the compose window and select a file that contains the security label that you want to use.

At the moment, Xuxa can only access the Security Labels available in the Security Label Catalogue that the DSA uses. This has the advantage that you use Security Label catalogue browser, which is very user-friendly. However, it also means that the DSA has to be set up with the correct Access Controls so that anonymous binds can read the DSA's Security Policy. For security reasons, that is not the default way DSAs are created, so if you want to use this feature, you will need to first load a Security Label policy, catalogue and clearance in the **cn=config** entry first, and also set the Access Controls so that you can read it with an anonymous bind.

12.5 Message Priority

Xuxa allows you to set the priority of a message that you are composing. Just select the priority you want to give the message from the pull down list in the **New Message** window. There are three options: **Normal Priority**, **High Priority** and **Low Priority**. In Military mode there are more priority options.

When receiving messages, the priority will be displayed in the **Preview** window. Also an icon will be shown in the Inbox's message table (red exclamation mark for High Priority messages, and blue arrow pointing down for Low Priority)

Do not confuse the X.400 Priority value with the AMHS specific value Message Priority value that is part of the AMHS Headers. Also note that Delivery Reports have no priority.

When in Military mode, the set of priorities is different, and this is reflected in both their names and their icons.

12.6 Probes

When composing a message, you can select the menu option **File** → **Send as Probe**. The current message being composed will be sent as a probe instead of as a message. You will get a Delivery Report (if the MTA supports probes) indicating what would happen if you sent the message properly.

12.7 Optimizing the startup time for large mailboxes

By default, XUXA lists the contents of the mailbox and reads all the messages in memory. While this works well with small mailboxes, it can delay the startup if you have a large number of messages in your inbox.

In that case, you can set up a Preference option in the **P7 Message Store** page, indicating that you want the messages to be fetched on demand instead of at startup (the default).

Chapter 13 AMHS Features

13.1 Templates for attaching common AMHS messages

In case you already have the content of a common AMHS message (Flight Plans, NOTAM, MET and BUFR), you can easily attach the content to compose the message. You do this by clicking on the right hand side of the Write icon, and select the appropriate message type from the pull down list. Alternatively, you can choose **Message** → **New Message** → **NOTAM** for example.

Xuxa offers you to attach a file from a default directory, and this can be configured in the preferences (**Xuxa General** → **Templates**)

Some sample messages (flight plans, NOTAM, EDI, General Text and BUFR) are available by contacting Isode support.

One thing that Xuxa will do is to recognize that an incoming message is of a given type, and show the corresponding GIF file. If you click on the GIF, you will see the actual content. If you find problems with the recognition, please let us know by sending the message in question, as we need feedback.

13.2 Templates for composing common AMHS messages

In Aviation Mode, when sending and receiving Flight Plans, NOTAM, MET and BUFR messages, Xuxa will allow you to set AMHS specific values, like Filing Time, Message Priority and Originator's Reference.

The behaviour of the application will be different in this mode. For example, if you set the ATS Header Message Priority to be 107 [AFTN (SS) / CIDIN (2)] then the message that is sent will have automatically requested both Delivery Reports and IPNs, and also it will be set to have High Priority.

Similarly, if you choose ATS Header Message Priority to be 14 [AFTN (KK) / CIDIN (7)] the priority will be set to Low.

13.3 AMHS Encoding choice

Xuxa can send messages in AMHS Basic Encoding or in Extended Encoding. You can see the difference by looking, for example, by looking at the format of the ATS Header value Filing Time. If it is just 6 digits, it is basic encoding, but if it has a UTC date then it's extended encoding.

According to the ICAO SARPs, it is not possible to mix recipients that require Basic Encoding and Extended Encoding. Xuxa will check the capabilities of all users in the Address Book, if they are present, and warn the user in case there is an inconsistency.

If you have to send a message to a mix of Basic and Extended users, then you have to force the use of Basic Encoding, by selecting the option **Force Basic Encoding** in the **Encoding Choice** parameter under **ATS Header**.

Chapter 14 Running Multiples Copies of Xuxa in One Machine

It is now much easier to run multiples copies of Xuxa in one machine. If you are running on Unix, the simplest thing is to run them as different Unix users.

If you are running on Windows, or don't want to have two different Unix users, then you can run a copy of Xuxa, which for example is configured to connect to user A.

Then run a second copy of Xuxa, which will initially connect to the same user A ,and configure it to use user B. Quit the program and start again, and it will connect to user B, so you will have two programs running, one connected to user A and the other one to user B. The disadvantage of this method is that if, for whatever reason, you quit the user A program, you will have to repeat the configuration.

Appendix A Notes and Change Log

A.1 Notes

The following Preferences are not currently used:

- **Colour scheme**
 - **Default date format**
 - **DSA Start DN and DSA Favourite DN**
-

A.2 Change log

A.2.1 v15.3, March 2015 (R16.3)

New features:

- Implement the message "Forward" action

Bug fixes:

- Display X.411 Security Labels on messages received via P3
- Fix invalid warning about invalid security environment

A.2.2 v15.1, November 2011 (R15.1)

New features:

- Display the Content Type of messages#
- Display the Trace Information and Internal Trace Information
- Use the new Security Environment parameters

Bug fixes:

- Correct the management of FTBP that don't have file attributes

A.2.3 v15.0, June 2011 (R15.0)

New features:

- Use the common Isode Bind Profile to access the Directory
- New account setup wizard to make
- Implement STANAG 4406 PCT signatures
- ATN DS Directory schema upgrade

Bug fixes:

- Make XUXA more resilient to starting with no configuration

- Make XUXA use GeneralizedTime instead of UTCTime for Authorization Time
- Don't do the Sequence Integrity Check for IPNs and Drs
- Improve the display of security information of messages signed with PCT
- Use STANAG 4406 compliant message identifiers in XUXA
- Make Xuxa generate an IPM Identifier that includes the user's O/R address
- Set the originator when using Xuxa with P3 users