

CAPLIN

Caplin Xaquu 1.0

Installing Permissioning Components

March 2012

CONFIDENTIAL

Contents

1	Preface.....	1
1.1	What this document contains.....	1
	About Caplin document formats	1
1.2	Who should read this document.....	1
1.3	Related documents.....	2
1.4	Typographical conventions.....	3
1.5	Feedback.....	3
1.6	Acknowledgments.....	4
2	Installation and Configuration.....	5
2.1	Checking the Liberator License.....	5
2.2	Installing the Permissioning Auth Module.....	6
2.3	Configuring Liberator to use the Permissioning Auth Module.....	6
	Enabling the Java Auth Module	6
	Setting the path to JVM	7
	Loading the Java Classes	7
	Identifying the Permissioning Auth Module	7
	Adding Field Mappings	8
	Configuring Liberator to work with KeyMaster	8
	Configuring Liberator to connect to the Permissioning DataSource	10
	Configuring Liberator to connect to multiple Permissioning DataSources	10
	Configuring Liberator for the TOBO switch message	12
2.4	Configuring the Permissioning Auth Module.....	13
	Permissioning Auth Module configuration for TOBO	14
2.5	Installing the Permissioning DataSource.....	15
	Creating a new directory for the Permissioning DataSource	16
2.6	Configuring the Permissioning DataSource.....	16
	Modifying the application name and id to match the Liberator configuration	16
	Setting the Liberator connection address and port number	17
	Modifying the scripts that start and stop Caplin Xaqua components	17
2.7	Verifying the Installation.....	18
3	Further Reading.....	20
4	Glossary of terms and acronyms.....	21

1 Preface

1.1 What this document contains

This document describes how to install the Permissioning Auth Module and Permissioning DataSource adapter in an existing Caplin Xaqua installation.

Note: You do not need to install or configure these permissioning components if your client application is based on Caplin Trader, because it includes them. However, if your Caplin Trader application uses the Permissioning System's support for TOBO, you may find it helpful to read the sections [Configuring Liberator for the TOBO switch message](#)^[12], [Configuring the Permissioning Auth Module](#)^[13], and [Permissioning Auth Module configuration for TOBO](#)^[14].

If your client application is *not* based on Caplin Trader, then you must install and configure these components to build permissioning capability into your Caplin Xaqua installation.

About Caplin document formats

This document is supplied in three formats:

- ◆ Portable document format (.PDF file), which you can read on-line using a suitable PDF reader such as Adobe Reader®. This version of the document is formatted as a printable manual; you can print it from the PDF reader.
- ◆ Web pages (.HTML files), which you can read on-line using a web browser. To read the web version of the document, navigate to the *HTMLDoc* folder and open the file *index.html*.
- ◆ Microsoft HTML Help (.CHM file), which is an HTML format contained in a single file. To read a .CHM file just open it – no web browser is needed.

For the best reading experience

On the machine where your browser or PDF reader runs, install the following Microsoft Windows® fonts: Arial, Courier New, Times New Roman, Tahoma. You must have a suitable Microsoft license to use these fonts.

Restrictions on viewing .CHM files

You can only read .CHM files from Microsoft Windows.

Microsoft Windows security restrictions may prevent you from viewing the content of .CHM files that are located on network drives. To fix this either copy the file to a local hard drive on your PC (for example the Desktop), or ask your System Administrator to grant access to the file across the network. For more information see the Microsoft knowledge base article at <http://support.microsoft.com/kb/896054/>.

1.2 Who should read this document

This document is intended for System Administrators and Software Developers who want to integrate Caplin Xaqua with a Permissioning System.

1.3 Related documents

- ◆ **Caplin Xaqua Overview**

A business and technical overview of Caplin Xaqua, including an explanation of its architecture.

- ◆ **Caplin Liberator: Administration Guide**

Describes how to install and configure Caplin Liberator and discusses the authentication modules that are provided with the server.

- ◆ **Caplin Xaqua: Permissioning Overview And Concepts**

Introduces permissioning concepts and terms, and shows the permissioning components of the Caplin Xaqua architecture.

- ◆ **Caplin Xaqua: How To Create A Permissioning DataSource**

Describes how to create a Permissioning DataSource adapter using the Permissioning DataSource API. A Permissioning DataSource adapter is required to integrate Caplin Xaqua with a Permissioning System. The document also discusses the Demo Permissioning DataSource provided with the reference implementation of Caplin Trader.

- ◆ **Caplin Trader: How To Add Permissioning At The Client**

Describes how to add permissioning to Caplin Trader.

- ◆ **Permissioning DataSource: API Reference**

The API reference documentation provided with the Permissioning DataSource SDK (Software Development Kit). The classes and interfaces presented by this API allow you to write a Java application that will integrate a Permissioning System with Caplin Xaqua.

- ◆ **Caplin Trader: API Reference**

The API reference documentation provided with Caplin Trader. The classes and interfaces of the `caplin.security.permissioning` package allow you to write JavaScript classes that extend the live permissioning capabilities of Caplin Trader.

- ◆ **KeyMaster: Administration Guide**

Describes how to configure and operate Caplin KeyMaster to provide a secure and reliable user authentication service.

1.4 Typographical conventions

The following typographical conventions are used to identify particular elements within the text.

Type	Uses
aMethod	Function or method name
<i>aParameter</i>	Parameter or variable name
<i>/AFolder/Afile.txt</i>	File names, folders and directories
<code>Some code;</code>	Program output and code examples
The <code>value=10</code> attribute is...	Code fragment in line with normal text
Some text in a dialog box	Dialog box output
Something typed in	User input – things you type at the computer keyboard
Glossary term	Items that appear in the “Glossary of terms and acronyms”
XYZ Product Overview	Document name
◆	Information bullet point
■	Action bullet point – an action you should perform

Note: Important Notes are enclosed within a box like this.
Please pay particular attention to these points to ensure proper configuration and operation of the solution.

Tip: Useful information is enclosed within a box like this.
Use these points to find out where to get more help on a topic.

Information about the applicability of a section is enclosed in a box like this.
For example: “This section only applies to version 1.3 of the product.”

1.5 Feedback

Customer feedback can only improve the quality of our product documentation, and we would welcome any comments, criticisms or suggestions you may have regarding this document.

Visit our feedback web page at <https://support.caplin.com/documentfeedback/>.

1.6 Acknowledgments

Adobe® Reader is a registered trademark of Adobe Systems Incorporated in the United States and/or other countries.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Java, *JavaScript*, and *JVM* are trademarks or registered trademarks of Oracle® Corporation in the U.S. and other countries.

Adobe Flex is a trademark of Adobe Systems Incorporated in the United States and/or other countries.

Silverlight is a trademark of Microsoft Corporation in the United States and other countries.

2 Installation and Configuration

When you install the reference implementation of **Caplin Trader**, you also install **Caplin Xaqua** and some third party software components.

In order to integrate Caplin Xaqua with a **permissioning system**, the **Liberator** server must be configured to use the services of a **Permissioning Auth Module**. You must also create and install a **Permissioning DataSource** adapter, and then configure the adapter to send permissioning data to the correct Liberator address and port number.

Note: The reference implementation of Caplin Trader is installed with a Permissioning Auth Module and **Demo Permissioning DataSource** adapter. This software is configured during the Caplin Trader installation process.
If your client application is *not* based on Caplin Trader, then you must manually install and configure these components as described in the following sections.

In the instructions that follow, file paths are relative to the directory in which you installed Caplin Xaqua. This installation directory contains the following sub-directories.

- ◆ *apps*
- ◆ *doc*
- ◆ *kits*
- ◆ *licenses*
- ◆ *ssl_certs*

2.1 Checking the Liberator License

- Before you install the permissioning software, make sure that you have the correct license for Liberator. Open the file *licenses/license-rtttd.conf* in a text editor and search for the following line of text.

```
module          javaauth auth
```

If this line is missing then you do not have the correct license. Please contact Caplin Support before proceeding with the installation.

2.2 Installing the Permissioning Auth Module

- Locate the zipped Permissioning Auth Module and unzip it to the *apps/caplin/kits* directory. You received this zip file with the permissioning software.

```
cd apps/caplin/kits
unzip permissioning-auth-module.zip
```

When you unzip this file, the directory *apps/caplin/kits/permissioning-auth-module-<version>* will be created (where *<version>* = version number).

- Now add two symbolic links that link this version of the Permissioning Auth Module to a "latest" version. This will make it easier to apply future upgrades.

First link the installed directory:

```
ln -s permissioning-auth-module-<version> permissioning-auth-module-latest
```

Now link the installed JAR file:

```
ln -s permissioning-auth-module-<version>/
permissioning-auth-module-<version>-jar-with-dependencies.jar
permissioning-auth-module-latest-jar-with-dependencies.jar
```

When you apply an upgrade, simply modify the symbolic links.

2.3 Configuring Liberator to use the Permissioning Auth Module

To configure Liberator to use the Permissioning Auth Module you must edit a number of files.

Enabling the Java Auth Module

- Open the file *apps/caplin/Liberator/etc/rtpd.conf* in a text editor.
Enable Java and instruct Liberator to use the Java Auth Module by adding the following lines.

```
java-file      java.conf
auth-module    javaauth
```

After you have added these lines, save and close the file.

Setting the path to JVM

- Open the file *apps/caplin/Liberator/etc/java.conf* in a text editor.

Add a line of text that sets `jvm-location` to the path of the Java Virtual Machine (JVM™). In the example below, the path to the JVM is

```
/usr/local/java/jre/lib/i386/server/libjvm.so.
```

```
jvm-location    /usr/local/java/jre/lib/i386/server/libjvm.so
```

Change this text to identify the path of your JVM, and then save and close the file.

Loading the Java Classes

- Open the file *apps/caplin/Liberator/etc/java.conf* in a text editor.

You must add some lines of text that will instruct Liberato to load the Java classes for the Permissioning Auth Module. In the example shown below, the path to the JAR file that contains the Permissioning Auth Module uses the symbolic link that we created in [Installing the Permissioning Auth Module](#) ⁶.

```
add-javaclass
  class-name    com.caplin.permissioning.auth.PermissioningAuthModule
  class-id      authenticator
  classpath     %r/../../kits/permissioning-auth-module-latest-jar-with
                                     -dependencies.jar
end-javaclass
```

After you have added these lines, save and close the file .

Identifying the Permissioning Auth Module

- Open the file *apps/caplin/Liberator/etc/javaauth.conf* in a text editor and add the following line of text (and optional comment line).

```
# Identifier for the class to load
javaauth-classid    authenticator
```

This will instruct the Java Auth Module to use the classes of the Permissioning Auth Module to authenticate **users** and user transactions (see [Loading the Java Classes](#) ⁷). After you have added this line, save and close the file.

Adding Field Mappings

- Create a symbolic link to the field mappings that are used by the Permissioning Auth Module.

```
ln -s apps/caplin/kits/permissioning-auth-module-latest/  
fields-permissioning.conf  
apps/caplin/CommonConfig/fields-permissioning-latest.conf
```

- Now open the file *apps/caplin/CommonConfig/fields-caplintrader.conf* in a text editor and add the following line of text. This will configure Liberator with the same field mappings.

```
include-file    fields-permissioning-latest.conf
```

After you have added this line, save and close the file. The Permissioning DataSource that is included with the permissioning software is already configured with the same field mappings.

Configuring Liberator to work with KeyMaster

If you are using **KeyMaster** with Liberator, then you must make sure that Liberator is correctly configured. To do this you must make sure that:

- the KeyMaster public key is available to Liberator
- the Liberator configuration file (*apps/caplin/Liberator/etc/rtpd.conf*) has been modified for KeyMaster

If Liberator has been correctly configured for KeyMaster, the configuration file *rtpd.conf* will have an entry that looks like the following:

```
add-sigkey  
    key-id          Caplin  
    timeout        600  
    keyfile         %r/etc/publickey.der  
end-sigkey
```

In this example, the identifier of the signature key is set to `Caplin`, which is the identifier the Permissioning Auth module uses by default. If you want to use another identifier, you must configure Liberator and the Permissioning Auth module to use the same identifier.

The following example configures Liberator to use `Novobank` as the identifier of the signature key `publickey2.der`:

```
add-sigkey  
    key-id          Novobank  
    timeout        600  
    keyfile         %r/etc/publickey2.der  
end-sigkey
```

To configure the Permissioning Auth module to use the same identifier, add the following line to the properties file *javaauth.properties*:

```
KeymasterKeyIdentifier=Novobank
```

The *javaauth.properties* file configures the Permissioning Auth Module, and may have been supplied with the Liberator kit.

If *javaauth.properties* exists, it will be located in a directory that Liberator can access. This could be:

1. In a directory defined by a `classpath` of the Permissioning Auth Module in the Liberator configuration file *java.conf*.
2. In a location referred to by a symbolic link. The symbolic link would also be located in a directory defined by a `classpath` of the Permissioning Auth Module.

If *javaauth.properties* does not exist, you will need to create it and deploy it to a classpath of the Permissioning Auth Module.

Note: Only one copy of *javaauth.properties* must be deployed, otherwise the configuration that is applied cannot be determined.

Deploying the *javaauth.properties* file

To deploy *javaauth.properties*:

- Copy the file to a directory that Liberator can access (or create the file in that directory).
- In the Liberator configuration file *java.conf*, add the directory as a classpath of the Permissioning Auth Module.

The following example shows what the entry in *java.conf* would look like if the *javaauth.properties* file is copied to the directory *%r/etc*.

```
add-javaclass
  class-name    com.caplin.permissioning.PermissioningAuthModule
  class-id      authenticator
  classpath     %r/../kits/permissioning-auth-module-latest-jar-
                with-dependencies.jar
  classpath     %r/etc/
end-javaclass
```

In the example configuration above, *%r* is a symbolic reference to the Liberator installation directory.

For further information about configuring Liberator to work with KeyMaster, please refer to the **KeyMaster: Administration Guide**.

Configuring Liberator to connect to the Permissioning DataSource

The following example shows you how to configure Liberator to connect to one Permissioning DataSource.

- Open the file `apps/caplin/Liberator/etc/rtpd.conf` and add the following lines to add and configure the **DataSource peer**.

```
# permissioningsrc
add-peer
    remote-id          16
    remote-name        permissioningsrc
    label              permissioningsrc
end-peer
```

The `remote-id` and `remote-name` of the peer must match the `appName` and `id` attributes of the `<dataSource>` tags in the Permissioning DataSource configuration file (see [Modifying the application name and id to match the Liberator configuration](#) ^[16]).

The `label` of the peer must match the `label` in the data service configuration (see below).

- Now add the following lines to add and configure the data service.

```
add-data-service
    service-name        permissioning-data
    include-pattern     ^/PERMISSIONS

    add-source-group
        required
        add-priority
            label        permissioningsrc
        end-priority
    end-source-group
end-data-service
```

The `label` of the data service must match the `label` in the DataSource peer configuration (see above).

See the **Caplin Liberator: Administration Guide** for further information about configuring peers and data services.

Configuring Liberator to connect to multiple Permissioning DataSources

When Permissioning data is sent to Liberator from more than one Permissioning DataSource, one of the Permissioning DataSources is designated the **master** and each of the remaining Permissioning DataSources are designated as **slaves**.

The following example shows you how to configure Liberator to connect to one master and one slave Permissioning DataSource.

- Open the file `apps/caplin/Liberator/etc/rttpd.conf` and add the following lines to add and configure the DataSource peers.

```
# master permissioning src peer
add-peer
    remote-id                16
    remote-name              master-src
    label                    master-permissioningsrc
end-peer

# FX slave permissioning src peer
add-peer
    remote-id                17
    remote-name              fx-src
    label                    fx-permissioningsrc
end-peer
```

The `remote-id` and `remote-name` of the master and slave peers must match the `appName` and `id` attributes of the `<dataSource>` tags in the master and slave Permissioning DataSource configuration files

(see [Modifying the application name and id to match the Liberator configuration](#) ^[16]).

The `label` of the master and slave peers must match the `label` in the master and slave data service configurations (see below).

- Now add the following lines to add and configure the master and slave data services.

```
# master permissioning src data service
add-data-service
    service-name              master-permsrc
    include-pattern           ^/PERMISSIONS/MASTER

    add-source-group
        required              true
        add-priority
            label              master-permissioningsrc
        end-priority
    end-source-group
end-data-service

# FX slave permissioning src service
add-data-service
    service-name              fx-permsrc
    include-pattern           ^/PERMISSIONS/FX

    add-source-group
        required              false
        add-priority
            label              fx-permissioningsrc
        end-priority
    end-source-group
end-data-service
```

The `label` of the master and slave data services must match the `label` in the master and slave `DataSource` peer configurations (see above). In addition, the "FX" at the end of the `include-pattern` of the slave (`^/PERMISSIONS/FX`) must match the name of the slave Permissioning `DataSource` that provides the service (see "Setting the Slave Role" in **Caplin Xaqua: How To Create A Permissioning DataSource Adapter**).

The master data service is configured as a `required` data service because end-users will be unable to log in to Liberator unless the master has provided permissioning data to Liberator. Slaves are typically not configured as `required` services to avoid a failed slave from preventing end-users with permissions from another slave from trading.

See the **Caplin Liberator: Administration Guide** for further information about configuring peers and data services.

Configuring Liberator for the TOBO switch message

This section only applies if the release of the Permissioning Auth Module you are installing is 4.5.9 or later.

Client is not based on Caplin Trader

If you are not using a Caplin Xaqua client that is based on Caplin Trader, and your client application is to make use of the Permissioning System's support for **TOBO** ("trading on behalf of"), you must configure a subject mapping for the **TOBO switch message**. This is done using the Liberator configuration item called **object-map**.

- Open the file `apps/caplin/Liberator/etc/rttpd.conf` and add the following line:

```
object-map <value-of-TOBOSubject> <value-of-TOBOSubject>/%u
```

where `<value-of-TOBOSubject>` is the value of the Permissioning Auth Module's property **TOBOSubject**, as defined in the `javaauth.properties` file.

For example, using the conventional value of **TOBOSubject**, add the line:

```
object-map /TOBOCHANGEUSER /TOBOCHANGEUSER/%u
```

Client is based on Caplin Trader

If you *are* using a Caplin Xaqua client that is based on Caplin Trader, and your client application is to make use of the Permissioning System's support for TOBO:

- Check that the subject of the TOBO switch message has the desired value.

By convention, the subject is `/TOBOCHANGEUSER`

This is the value set in the Caplin Xaqua installation supplied with Caplin Trader (in the Liberator **object-map** configuration, and in the Permissioning Auth Module's **TOBOSubject** property).

If you wish to change the subject of the TOBO switch message from its conventional value:

- Change the value of the **TOBOSubject** property in the `javaauth.properties` file of the Permissioning Auth Module (see [Configuring the Permissioning Auth Module](#)^[13] and [Permissioning Auth Module configuration for TOBO](#)^[14]).
- Open the file `apps/caplin/Liberator/etc/rttpd.conf` and change the following line:

```
object-map /TOBOCHANGEUSER /TOBOCHANGEUSER/%u  
to
```

```
object-map /TOBOCHANGEUSER <value-of-TOBOSubject>/%u
```

where `<value-of-TOBOSubject>` is the new value of the Permissioning Auth Module's **TOBOSubject** property.

For example:

```
object-map /TOBOCHANGEUSER /TOBOSWITCH/%u
```

More information

For more information about TOBO, see the document **Caplin Xaqua: Permissioning Overview And Concepts** (section "Permissioning for TOBO" in "Additional permissioning capabilities").

For a description of **TOBOSubject**, see [Permissioning Auth Module configuration for TOBO](#)^[14].

2.4 Configuring the Permissioning Auth Module

This section only applies if the release of the Permissioning Auth Module you are installing is 4.5.9 or later.

Client is not based on Caplin Trader

If you are not using a Caplin Xaqua client that is based on Caplin Trader, and you wish to use the Permissioning System's support for **TOBO** ("trading on behalf of"), you must configure the Permissioning Auth Module so that the TOBO capability operates according to your requirements.

The configuration items are defined as properties in the Permissioning Auth Module's *javaauth.properties* file. A *javaauth.properties* file may have been supplied with the Liberator kit.

If *javaauth.properties* exists, it will be located in a directory that Liberator can access. This could be:

1. In a directory defined by a `classpath` of the Permissioning Auth Module in the Liberator configuration file *java.conf*.
 2. In a location referred to by a symbolic link. The symbolic link would also be located in a directory defined by a `classpath` of the Permissioning Auth Module.
- If *javaauth.properties* does not exist, you will need to create it and deploy it to a classpath of the Permissioning Auth Module. See [Deploying the javaauth.properties file](#)^[9] in [Configuring Liberator to work with KeyMaster](#)^[8].

Note: Only one copy of *javaauth.properties* must be deployed, otherwise the configuration that is applied cannot be determined.

Client is based on Caplin Trader

If you are using a Caplin Xaqua client that is based on Caplin Trader, the Permissioning Auth Module in the Caplin Xaqua installation supplied with Caplin Trader has a suitably configured *javaauth.properties* file. You do not normally need to change this configuration unless your client application uses TOBO in a different way – see [Permissioning Auth Module configuration for TOBO](#)^[14].

For more information about TOBO, see the document **Caplin Xaqua: Permissioning Overview And Concepts** (section "Permissioning for TOBO" in "Additional permissioning capabilities").

Permissioning Auth Module configuration for TOBO

This section only applies if the release of the Permissioning Auth Module you are installing is 4.5.9 or later.

- Set the following configuration properties in *javaauth.properties* to define how the Permissioning Auth Module implements permissioning for TOBO.

If you are using a Caplin Xaqua client that is based on Caplin Trader, you do not normally need to change this configuration. Reasons why you might need to change the configuration include:

- You have changed the Caplin Trader application to use TOBO in `SalesUser` mode.
- You need to use different subject and/or field names in the TOBO switch message, because the existing names clash with other messages that your Caplin Trader application generates.
- You need to resolve a namespace clash in your permissioning set up.

TOBO properties in *javaauth.properties*

Property	Conventional value or [Permitted values]	Description
TOBOField	UserName	The field in a TOBO switch message that contains the name of the customer-user on whose behalf the sales-user wishes to trade.
TOBOPermissionMode	[SalesIntersectCustomerUser or SalesUser]	The TOBO permission mode that the Permissioning Auth Module runs in. For a definition of these modes and the permissioning behavior they control, see the section “TOBO permission modes” in the document Caplin Xaqua: Permissioning Overview And Concepts . If this property is omitted, the TOBO permission mode is set to <code>SalesUser</code> .
TOBOSubject	/TOBOCHANGEUSER	The RTTP message subject that identifies a TOBO switch message .

Property	Conventional value or [Permitted values]	Description
TOBOSubjectPattern	/.*	When trading on behalf of a customer-user, all subjects within the scope of this regular expression are mapped by the customer-user's Subject Mapper, and any other subjects are mapped by the sales-user's own Subject Mapper. For more about subject mappings in permissioning, see the document Caplin Xaqua: Permissioning Overview And Concepts .
TOBOSwitchAction	ChangeTradeOnBehalfOfUser	This is the name of the action that, together with the namespace defined by TOBOSwitchNamespace , identifies the permission that allows or denies a "TOBO switch" (allows the sales-user to trade on behalf of a customer-user, or denies the sales-user from doing so).
TOBOSwitchNamespace	TradeOnBehalfOf	This is the namespace that, together with the action defined by TOBOSwitchAction , identifies the permission that allows or denies a "TOBO switch" (allows the sales-user to trade on behalf of a customer-user, or denies the sales-user from doing so).

2.5 Installing the Permissioning DataSource

- Locate the zipped Permissioning DataSource and unzip it to the *apps/caplin/kits* directory. You received this zip file with the permissioning software.

```
cd apps/caplin/kits
unzip permissioning-datasource.zip
```

When you unzip this file, the directory *apps/caplin/kits/permissioning-datasource-<version>* will be created (where *<version>* = version number).

- Add a symbolic link that links this version of the Permissioning DataSource to a "latest" version. This will make it easier to apply future upgrades.

```
ln -s permissioning-datasource-<version> permissioning-datasource-latest
```

When you apply an upgrade, simply modify this symbolic link.

Creating a new directory for the Permissioning DataSource

- Create the directory *apps/caplin/PermissioningDataSource*. Navigate to this directory and then copy and link the following files and directories.

```
cp -R ../kits/permissioning-datasource-latest/conf/ conf
cp ../kits/permissioning-datasource-latest/start.sh start.sh
cp ../kits/permissioning-datasource-latest/stop.sh stop.sh

ln -s ../kits/permissioning-datasource-latest/lib/ lib
ln -s ../kits/permissioning-datasource-latest/permissioning.sh
permissioning.sh
```

The directory *apps/caplin/PermissioningDataSource* will now contain the following sub-directories and files.

- ◆ *conf* (configuration files)
- ◆ *lib* (library files)
- ◆ *logs* (log files will be placed here)
- ◆ *start.sh* (script to start the Demo Permissioning DataSource adapter)
- ◆ *stop.sh* (script to stop the Demo Permissioning DataSource adapter)
- ◆ *permissioning.sh* (a non Caplin Xaqua specific script to start the Demo Permissioning DataSource)

2.6 Configuring the Permissioning DataSource

To configure the Permissioning DataSource you must edit a number of files.

Modifying the application name and id to match the Liberator configuration

The modifications below apply if you have a single Permissioning DataSource. If you have multiple (master/slave) Permissioning DataSources to configure, then you will need to modify the master and slave configuration files.

- Open the file *apps/caplin/kits/PermissioningDataSource/conf/DataSource.xml* in a text editor.
Find the first line of the XML file. The line contains the `appName` and `id` attributes and will look something like the following.

```
<dataSource xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              appName="permissionsrc" id="1">
```

- Modify `appName` and `id` to match the `remote-name` and `remote-id` that you set up in [Configuring Liberator to connect to the Permissioning DataSource](#) ^[10].

After making these changes the line will look something like the following.

```
<dataSource xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            appName="permissioningsrc" id="16">
```

If you are configuring a master or slave Permissioning DataSource, then `appName` and `id` must match the `remote-name` and `remote-id` that you set up in [Configuring Liberator to connect to multiple Permissioning DataSources](#).

After you have made these changes, save and close the file.

Setting the Liberator connection address and port number

The modifications below apply if you have a single Permissioning DataSource. If you have multiple (master/slave) Permissioning DataSources to configure, then you will need to modify the master and slave configuration files.

- Open the file `apps/caplin/kits/PermissioningDataSource/conf/DataSource.xml` in a text editor.

Search for the `<peer>` tag. The text that you find will look something like the following.

```
<peer>
  <destination address="liberator.example.com" port="44310" />
</peer>
```

- Change `destination address` to the hostname (or IP address) of Liberator, and `port` to the port number that Liberator listens on for peer connections (see the **Caplin Liberator: Administration Guide** for further information about obtaining and setting these values).

A typical entry after making these changes will look something like the following.

```
<peer>
  <destination address="myliberator.mydomain.com" port="50100" />
</peer>
```

After you have made these changes, save and close the file.

Modifying the scripts that start and stop Caplin Xaqua components

You can add text to the scripts that start and stop the components of Caplin Xaqua, so that the Demo Permissioning DataSource starts or stops when the other components start or stop (see **Caplin Xaqua: How To Create A Permissioning DataSource** for further information about the Demo Permissioning DataSource).

- To modify the start script, open the file `apps/caplin/start-all-components.sh` in a text editor and add the following lines at the end of the file.

```
if [ -d PermissioningDatasource ]; then
    cd PermissioningDatasource
    echo "Starting PermissioningDatasource: "
    ./start.sh
    cd ..
fi
```

- To modify the stop script, open the file `apps/caplin/stop-all-components.sh` in a text editor and add the following lines at the end of the file.

```
if [ -d PermissioningDatasource
    -a -f PermissioningDatasource/permissioning-datasource.pid ]; then
    cd PermissioningDatasource
    echo "Shutting down PermissioningDatasource: "
    ./stop.sh
    cd ..
fi
```

After you have made these changes, save and close the files.

2.7 Verifying the Installation

- When you have installed and configured the permissioning software, start the Demo Permissioning DataSource (see **Caplin Xaqua: How To Create A Permissioning DataSource**) and Liberator server.
- Now open your browser and navigate to the Liberator status page at `http://<liberator.host>:<liberator.http.port>/status` where `<liberator.host>` is the hostname and `<liberator.http.port>` the Liberator port number that you specified when you installed Caplin Xaqua.
A typical address would be:
`http://linux1.domain1.com:50180/status`
- When the Liberator login page is displayed, log in to Liberator with the following credentials:
Username: admin
Password: admin
The Liberator status page will be displayed.
- In the `Data Services` part of this page, look for a service with the name `permissioning-data`. If the Status of this service is OK, then the software has been successfully installed and configured. If you have configured master and slave `Data Services`, then you should see a Status entry for the master and each slave.

```
Name: permissioning-data
Status: OK
Last Change: Jun 05 23:02:29
```

Extract from the Liberator status page

If you are unable to confirm that the software was successfully installed and configured, then the troubleshooting guide below will help you to resolve the problem.

Troubleshooting guide

Symptom	Issue	Action
The Liberator status page is unavailable.	Liberator has not started.	Check the Liberator console output for the reason.
<p>The Liberator status page loads, but the following message is displayed:</p> <pre>Sorry. You do not have the authorization to view live data from this source. Please contact the web site host if you have received this message in error.</pre>	Permissioning data did not reach the Liberator server.	<p>Check the process id to see that the Permissioning DataSource is still running.</p> <p>The process id for the Permissioning DataSource can be found in the file: <i>permissioning-datasource.pid</i> in the <i>apps/caplin/PermissioningDatasource</i> directory.</p> <p>Check <i>logs/stdout.log</i> and <i>logs/permissioningsrc.log</i>.</p> <p>If the logs are OK and the Permissioning DataSource is running, then there is a connection issue with Liberator. Check the packet logs on both sides of the connection. The section "Monitoring Performance" in the Caplin Liberator: Administration Guide describes how to check packet logs.</p>

3 Further Reading

If you would like an introduction to permissioning concepts and terms, or would like to know how to create a custom Permissioning DataSource or how to add permissioning to Caplin Trader, then the following documents provide this information. You may also be interested in reading some of the other [Related documents](#)².

An introduction to permissioning concepts and terms

The document **Caplin Xaqua: Permissioning Overview And Concepts** introduces permissioning concepts and terms, and shows the permissioning components of the Caplin Xaqua architecture.

How to create a Permissioning DataSource Adapter

A Permissioning DataSource adapter is required to integrate Caplin Xaqua with a Permissioning System. The document **Caplin Xaqua: How To Create A Permissioning DataSource** describes how to create a custom Permissioning DataSource adapter by writing an application that uses the Permissioning DataSource API. The document also discusses the Demo Permissioning DataSource that is provided with the reference implementation of Caplin Trader.

How to add Permissioning at the Client

The appearance and behavior of Caplin Trader can be tailored to match the permissions of the currently logged in user. You will find further information about how to do this in the document **Caplin Trader: How To Add Permissioning At The Client**.

4 Glossary of terms and acronyms

This section contains a glossary of terms, abbreviations, and acronyms used in this document.

Term	Definition
API	Application Programming Interface
Caplin Trader	A web application framework for constructing browser-based financial trading applications (Caplin Trader applications).
Caplin Trader application	A Caplin Xaqua client that has been built using Caplin Trader .
Caplin Xaqua	A framework for building single-dealer platforms that enables banks to deliver multi-product trading direct to client desktops. Caplin Xaqua can also be short for a Caplin Xaqua system .
Caplin Xaqua client	A client desktop or web application that interfaces with Caplin Xaqua to deliver multi-product trading to end-users.
Caplin Xaqua system	A single-dealer platform that is built using Caplin Xaqua .
Customer-user	An end-user of a Caplin Xaqua client who instructs a sales-user to trade on their behalf.
DataSource	An API and underlying code library that allows DataSource applications to communicate with each other.
DataSource adapter	A DataSource application that acts as the interface between Caplin Xaqua and an external (non-Caplin) system, exchanging data and/or messages with that system.
DataSource application	A Caplin Xaqua application that uses the DataSource API and code library to communicate with other Caplin Xaqua applications.
DataSource peer	A DataSource application that another DataSource application is configured to communicate with.
Demo Permissioning DataSource	The Demo Permissioning DataSource is an example of a Permissioning DataSource that gets its permissioning data from an XML file.
KeyMaster	Caplin KeyMaster integrates Caplin Xaqua with any existing authentication system, so that end-users or web applications do not need to explicitly log in to Liberator in addition to their normal login procedure.
Liberator	A real-time financial internet hub that delivers trade messages and market data to and from subscribers over any network that supports web traffic.
Master	When permissioning data is sent to Liberator from multiple Permissioning DataSource adapters, one of the Permissioning DataSource adapters is designated the master, and the others are designated as slaves .
Permissioning Auth Module	One of several authentication modules that are supplied with Caplin Xaqua .
Permissioning DataSource	A DataSource adapter that acts as the interface between Caplin Xaqua and your Permissioning System .
Permissioning System	The source of the permissioning data that you want to integrate with Caplin Xaqua .

<u>Term</u>	<u>Definition</u>
Sales-user	An end-user of a Caplin Xaqua client who takes instructions from a customer-user to trade on their behalf.
SDK	<u>Software Development Kit</u>
Slave	When permissioning data is sent to Liberator from multiple Permissioning DataSource adapters, one of the Permissioning DataSource adapters is designated the master , and the others are designated as slaves.
TOBO	<u>Trading On Behalf Of</u>
TOBO switch message	A message that is sent from a Caplin Xaqua client to a Liberator , requesting that the current user (a sales-user) be allowed to trade on behalf of a different user (a customer-user).
Trading On Behalf Of	This is a facility that allows a user who is logged in to a Caplin Xaqua client to execute trades on instruction from a customer (for example, the customer may give instructions by telephone). The logged-in user (sales-user) trades on behalf of the customer (customer-user). For more information, see the document Caplin Xaqua: Permissioning Overview And Concepts .
User	An end-user of a Caplin Xaqua client application such as Caplin Trader .

Contact Us

Caplin Systems Ltd
Cutlers Court
115 Houndsditch
London EC3A 7BR
Telephone: +44 20 7826 9600
www.caplin.com

The information contained in this publication is subject to UK, US and international copyright laws and treaties and all rights are reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the written authorization of an Officer of Caplin Systems Limited.

Various Caplin technologies described in this document are the subject of patent applications. All trademarks, company names, logos and service marks/names ("Marks") displayed in this publication are the property of Caplin or other third parties and may be registered trademarks. You are not permitted to use any Mark without the prior written consent of Caplin or the owner of that Mark.

This publication is provided "as is" without warranty of any kind, either express or implied, including, but not limited to, warranties of merchantability, fitness for a particular purpose, or non-infringement.

This publication could include technical inaccuracies or typographical errors and is subject to change without notice. Changes are periodically added to the information herein; these changes will be incorporated in new editions of this publication.

Caplin Systems Limited may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

This publication may contain links to third-party web sites; Caplin Systems Limited is not responsible for the content of such sites.