

# Caplin Trader 1.3

---

## How To Add Permissioning At The Client

January 2009

# Contents

<b>1</b>	<b>Preface.....</b>	<b>1</b>
1.1	What this document contains.....	1
	About Caplin document formats .....	1
1.2	Who should read this document.....	1
1.3	Related documents.....	1
1.4	Typographical conventions.....	3
1.5	Feedback.....	3
1.6	Acknowledgments.....	3
<b>2</b>	<b>Permissioning and Caplin Trader Client.....</b>	<b>4</b>
2.1	About Product Permissions .....	4
<b>3</b>	<b>Using the Permissioning API for Caplin Trader Client.....</b>	<b>6</b>
3.1	Methods that return permissioning information when called.....	7
3.2	Methods that register listeners.....	9
3.3	User attributes.....	10
<b>4</b>	<b>Further Reading.....</b>	<b>11</b>
<b>5</b>	<b>Glossary of terms and acronyms.....</b>	<b>12</b>
	<b>Index.....</b>	<b>13</b>

# 1 Preface

## 1.1 What this document contains

This document describes how to add permissioning to Caplin Trader Client so that display components behave in accordance with the permissions of the logged in user.

Before reading this document, make sure you are familiar with the document **Caplin Trader: Permissioning Overview and Concepts**.

### 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\_m\_n* 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.

#### 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 Architects and Software Developers who want to add or modify permissioning in Caplin Trader Client.

## 1.3 Related documents

- ◆ **Caplin Trader: Architecture**  
Describes the architecture of Caplin Trader. It focuses on the use of the Caplin Platform in trading applications. It also identifies the areas in which the Platform can be integrated with your company's own and third-party systems.
- ◆ **Caplin Liberator: Administration Guide**  
Describes how to install and configure Caplin Liberator and discusses the authentication modules that are provided with the server.

◆ **Caplin Trader: Permissioning Overview And Concepts**

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

◆ **Caplin Trader: Installing Permissioning Components**

Describes how to install the Permissioning Auth Module and Permissioning DataSource in an existing Caplin Trader installation. You only need to install these components if your installation of Caplin Trader is earlier than release 1.2.8, as later releases include these permissioning components.

◆ **Caplin Trader: 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 Trader with a Permissioning System. The document also discusses the Demo Permissioning DataSource provided with the reference implementation of Caplin Trader from release 1.2.8.

◆ **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 Trader.

◆ **Caplin Trader Client: API Reference**

The API reference documentation provided with Caplin Trader Client. 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 Client.

## 1.4 Typographical conventions

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

Type	Uses
<b>aMethod</b>	Function or method name
<i>aParameter</i>	Parameter or variable name
<i>/AFolder/Afile.txt</i>	File names, folders and directories
<div>Some code;</div>	Program output and code examples
The value=10 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
<b>XYZ Product Overview</b>	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.

## 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.

Please email your feedback to [documentation@caplin.com](mailto:documentation@caplin.com).

## 1.6 Acknowledgments

*Firefox* is a registered trademark of the Mozilla Foundation.

*Java*, *JavaScript*, and *JVM* are trademarks of Sun Microsystems, Inc. in the U.S. or other countries.

*Windows* and *Internet Explorer* are registered trademarks of Microsoft Corporation in the United States and other countries.

## 2 Permissioning and Caplin Trader Client

The display components of Caplin Trader Client can be tailored to match the permissions of the currently logged in user. An example would be to display information that the user is allowed to view, and to hide information that the user is not allowed to view. This information could be anything from pricing data for currency pairs, to menu items, data grids, trade tiles, and tenors. Another example would be to enable or disable the one-click trading button to reflect the permissions of the logged in user.

To add permissioning at the client, you must modify the client application by adding JavaScript code that uses the [Permissioning API](#) for Caplin Trader Client. The Permissioning API is part of the Caplin Trader Client JavaScript API, and is supplied with Caplin Trader Client from release 1.2.8.

For a complete description of the Permissioning API, please refer to the **caplin.security.permissioning** package of the **Caplin Trader Client: API Reference** documentation.

### Security

User authentication and permissioning is enforced by the Permissioning Auth Module at the Liberator server. It is not possible to modify these product permissions from Caplin Trader Client. The classes of the Permissioning API only allow you to query the permissions that have been assigned to the currently logged in user; they do not allow you to change these permissions.

### Latency of permissioning queries

When a user logs in to Caplin Trader Client, the application automatically subscribes to the permissioning data that Liberator holds for that user. At the start of the subscription, Liberator sends Caplin Trader Client all the permissioning data for that user, followed by subscription updates whenever a product permission changes.

The classes of the Permissioning API provide an interface to this permissioning data, allowing you to tailor your copy of Caplin Trader Client to match the permissions of the currently logged in user.

Since Caplin Trader Client automatically subscribes to permissioning data, the latency of each subsequent permissioning query is reduced. This is because Caplin Trader Client holds a local copy of the latest product permissions, and does not need to contact Liberator each time your code initiates a permissioning query.

### 2.1 About Product Permissions

Product permissions are sent to Liberator by the Permissioning DataSource (see **Caplin Trader: Permissioning Overview And Concepts** for further information). A user can have any number of product permissions assigned to them, where each permission determines whether an action on a product will be allowed or denied.

Permissions for actions can reside in either the global (default) namespace or a specific namespace. Permissions are typically defined in a specific namespace to group related actions. An example would be to group tenor permissions by defining these permissions in a "tenor" namespace. Your client application could then query the tenor namespace to determine the tenors that are allowed and those that are denied.

The table below shows four such permissions, where each row of the table defines a single product permission. Note that the final column of the table (Significance) is not part of the permission, and is only included here to provide a description of the permission.

Namespace	Product	Action	Authorization	Significance
null	.*	VIEW	ALLOW	<i>Allow the user to view all products.</i>
null	/FX/.*	TRADE	ALLOW	<i>Allow the user to trade all FX products.</i>
tenor	/FX/.*	1_WEEK	ALLOW	<i>Allow the user 1-week tenor for FX products.</i>
tenor	/FX/.*	2_WEEK	DENY	<i>Deny the user 2-week tenor for FX products.</i>

You will see from the table above that JavaScript regular expressions have been used to define the product. In this way the same permission can be applied to a range of products. When a user attempts to view or trade a product, Liberator consults pre-defined permissioning rules that indicate the permissions that the user must have in order to complete the task.

The permissions in the first two rows of the table above allow the user to view all products, and to trade any FX product. In this case the namespace is null, which indicates that the permission resides in the global (default) namespace. This type of permission would typically be used at the client to enable or disable the display of products or product grids.

The permissions in the last two rows of the table are defined in the "tenor" namespace. These permissions allow the user to trade any FX product with a 1-week tenor, but do not allow the user to trade any FX product with a 2-week tenor. These permissions would typically be used at the client to:

- \* Enable the 1-week tenor option.
- \* Disable (or hide) the 2-week tenor option.

### 3 Using the Permissioning API for Caplin Trader Client

The Permissioning API for Caplin Trader Client provides two classes.

**caplin.security.permissioning.PermissionServiceListener:** An interface. The classes of your GUI application must implement this interface if they want to be notified when user permissions change.

**caplin.security.permissioning.PermissionService:** A singleton class. The methods in this class return permissioning information about the currently logged in user. The methods fall into three categories.

- ◆ The first category (see [Methods that return permissioning information when called](#)<sup>[7]</sup>) returns permissioning information when the method is called. This category can be used by the classes of your GUI application that respond to user interactions.
- ◆ The second category (see [Methods that register listeners](#)<sup>[9]</sup>) allows you to register listeners that subscribe to current and future product permissions. This category can be used to register the classes of your GUI application that need to be notified when user permissions change. Classes that you register as listeners must implement the `PermissionServiceListener` interface.
- ◆ The third category (see [User attributes](#)<sup>[10]</sup>) allows you to retrieve user attributes.



### 3.1 Methods that return permissioning information when called

Methods of the `PermissionService` class that return permissioning information when called are shown in the table below.

Query Type	Methods that can be used for this type of query	Return Value
Query an action on a single product in the given namespace.	<code>canUserPerformAction(product, namespace, action)</code>	<i>true</i> if action is allowed. <i>false</i> if action is denied.
Query an action on a single product in the <i>default</i> namespace.	<code>canUserPerformGlobalAction(product, action)</code>	<i>true</i> if action is allowed. <i>false</i> if action is denied.
Query an action on an array of products in the given namespace.	<code>getPermissionedProducts(products, namespace, action)</code>	A subset of products on which the action is <i>allowed</i> .
	<code>getUnpermissionedProducts(products, namespace, action)</code>	A subset of products on which the action is <i>denied</i> .
Query the permitted actions on a single product in the given namespace.	<code>getAllowPermissions(product, namespace)</code>	An array of <i>allowed</i> actions for this product.
	<code>getDenyPermissions(product, namespace)</code>	An array of <i>denied</i> actions for this product.
	<code>getAllPermissions(product, namespace)</code>	An array of <i>all</i> actions for this product ( <i>allowed</i> and <i>denied</i> ).

In the following example, we add code that responds to a user interacting with a GUI component. In particular, we add code to the part of the client application that responds to attempts by the user to open a trade ticket.

```
/* check that user is permitted to open this trade ticket */
if(caplin.security.permissioning.PermissionService.
    canUserPerformGlobalAction("/FX/EURUSD", "TRADE"))
{
    // open the trade ticket
}
```

In above example, the trade ticket will only open if the user is permitted "TRADE" action for the product "/FX/EURUSD" in the "default" namespace.

The screenshot shows the 'FX Trade Ticket' window. At the top, there is a currency pair 'EURUSD' with a search icon, a 'SPOT/FWD' dropdown menu, and an 'Account' field with 'acct 1'. Below this, there are two main sections: 'Amount' and 'Settlement'. The 'Amount' section has a currency dropdown set to 'EUR' and a text input field containing '500,000'. The 'Settlement' section has a 'Spot' dropdown menu and a date field set to '25 Jun 2008' with a calendar icon. At the bottom center of the window is a large blue button labeled 'GET QUOTE'.

Typical Opened Trade Ticket

## 3.2 Methods that register listeners

Methods of the `PermissionService` class that are used to register listeners are shown in the table below. The listeners that you register will typically be classes in your GUI application that need to receive notification of user permissions for a particular type of subscription. Listeners get notified of user permissions when the classes are first registered, and also when permissions change for the subscribed elements.

To receive permissioning notifications, the registered listener must implement one or more of the callback methods of the `PermissionServiceListener` interface, as indicated in the table.

Subscription Type	PermissionService method to register the subscription listener	Callback method the listener must implement	Notification
Subscribe to permissioning changes for an action on a single product in the given namespace.	<code>addPermissionListener(     product,     namespace,     action,     listener)</code>	<code>onSinglePermissionChanged(     bIsAuthorized)</code>	<code>bIsAuthorized:</code>  <i>true</i> if action is allowed.  <i>false</i> if action is denied.
Subscribe to permissioning changes for an action on a single product in the <i>default</i> namespace.	<code>addGlobalPermissionListener(     product,     action,     listener)</code>	<code>onSinglePermissionChanged(     bIsAuthorized)</code>	<code>bIsAuthorized:</code>  <i>true</i> if action is allowed.  <i>false</i> if action is denied.
Subscribe to permissioning changes for an action on an array of products in the given namespace.	<code>addProductPermissionsListener(     products,     namespace,     action,     isAuthorized     listener)</code>	<code>onPermissionsChanged(     permissions)</code>	<code>permissions:</code>  An array (subset) of products for which the action is <i>allowed</i> , <i>denied</i> , or <i>allowed and denied</i> , depending on the value of <code>isAuthorized</code> (ALLOW, DENY, or ALL).
Subscribe to permissioning changes for permitted actions on a single product in the given namespace.	<code>addPermissionSetListener(     product,     namespace,     authtype,     listener)</code>	<code>onPermissionsChanged(     permissions)</code>	<code>permissions:</code>  An array of <i>allowed</i> , <i>denied</i> , or <i>all</i> actions on the product, depending on the value of <code>authtype</code> (ALLOW, DENY, or ALL).

In the following example, we add code that responds to permissioning changes for an action on a single product. In particular, we add code to the part of the client application that is responsible for displaying a trade tile.

Unlike the previous example (see [Methods that return permissioning information when called](#)), where the trade ticket does not open until the user selects the currency pair that they want to trade, a trade tile is typically on display whether or not the end user uses it to trade. The tile must therefore always reflect the latest permissions that apply to it. It is for this reason that a listener must be registered to enable or disable the trade tile, rather than calling a method in response to a user action.

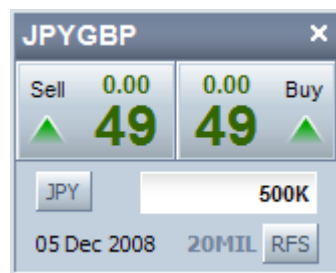
Register the listener,

```
/* register a listener for the "TRADE" action */
tile()
{
    addPermissionListener(product, namespace, "TRADE", this)
}
```

and implement the callback method.

```
/* implement the callback method for this listener */
onSinglePermissionChanged(bIsAuthorized)
{
    if(bIsAuthorized)
    {
        /* enable the trade tile */
    }
    else
    {
        /* disable the trade tile */
    }
}
```

In above example, the trade tile is either enabled or disabled, depending on the permission that the user has for "TRADE" action on the product. If the action is allowed, then the trade tile is enabled. If the action is denied, then the trade tile is disabled.



**Enabled Trade Tile**




**Disabled Trade Tile**

### 3.3 User attributes

User attributes are name/value pairs that are not processed by the Permissioning Auth Module and therefore do not affect permissioning directly. A typical use is to send information to Caplin Trader Client about the maximum tradable amount that a user is permitted to trade. Your application could then use this information to limit the tradeable amounts shown to the user.

You will find further information about user attributes in **Caplin Trader: Permissioning Overview and Concepts**, and in the **caplin.security.permissioning** package of the **Caplin Trader Client API** documentation.

## 4 Further Reading

If you would like an introduction to permissioning concepts and terms, or to consult reference documentation for the Permissioning API, then the following documents provide this information. You may also be interested in reading some of the other [Related documents](#) .

### An introduction to permissioning concepts and terms

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

### Reference documentation for the Permissioning API

Reference documentation for the Permissioning API can be found in the **caplin.security.permissioning** package of the **Caplin Trader Client: API Reference**.

## 5 Glossary of terms and acronyms

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

Term	Definition
<b>Action</b>	The interaction that a user can have with a <b>product</b> .
<b>API</b>	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
<b>Caplin Platform</b>	A suite of software products for on-line financial trading and Web delivery of real-time market data.
<b>Caplin Trader</b>	Caplin Trader is a complete platform and toolkit for building multi-product trading portals. It is built on the <b>Caplin Platform</b> .
<b>Caplin Trader Client</b>	Caplin Trader Client is a Web application written in Ajax that provides a rich trading workstation in a browser.
<b>DataSource</b>	DataSources are software adapters within the <b>Caplin Platform</b> that connect the Platform to external sources of real time data and external <b>Permissioning Systems</b> . In other Caplin documents DataSources are also called DataSource adapters.
<b>Demo Permissioning DataSource</b>	The Demo Permissioning DataSource is an example of a <b>Permissioning DataSource</b> application that gets its permissioning data from an XML file.
<b>Liberator</b>	Caplin Liberator is a bidirectional streaming push server designed to deliver market data and trade messages over any network that supports Web traffic.
<b>Permission</b>	Determines whether an <b>action</b> on a <b>product</b> will be allowed or denied.
<b>Permissioning Auth Module</b>	One of several authentication modules that are supplied with <b>Caplin Trader</b> .
<b>Permissioning DataSource</b>	A <b>DataSource</b> adapter that acts as the interface between <b>Caplin Trader</b> and your <b>Permissioning System</b> .
<b>Permissioning System</b>	The source of the permissioning data that you want to integrate with <b>Caplin Trader</b> .
<b>Product</b>	In permissioning documentation (including this document) a "product" is any entity on which a <b>User</b> may be assigned <b>permissions</b> (including financial instruments). In other <b>Caplin Trader</b> documentation a "product" is a term that refers only to a financial instrument.
<b>Rule</b>	Rules link <b>permissions</b> to user interactions, and are used by <b>Liberator</b> to decide which of the many permissions that have been defined will apply when a <b>user</b> attempts to interact with a <b>product</b> .
<b>SDK</b>	<u>S</u> oftware <u>D</u> evelopment <u>K</u> it
<b>User</b>	An end user of <b>Caplin Trader Client</b> .

# Index

## - A -

Abbreviations, definitions 12  
Acronyms, definitions 12  
actions 4  
API 4, 6

## - C -

Caplin Trader Client  
    permissioning 4  
class  
    PermissionService 7, 9  
classes and interfaces 6

## - E -

example code 7, 9

## - G -

Glossary 12  
GUI components 4

## - L -

latency 4  
listeners 9

## - M -

Matching GUI components to permissions 4

## - N -

name/value pairs 10  
namespaces 4  
notification 9

## - P -

Permissioning at the client 4  
Permissioning query 7  
PermissionService class 7, 9  
Product permissions 4

## - Q -

query 7

## - R -

Readership 1  
register listener 9  
regular expressions 4  
respond to user interaction 7  
rules 4

## - S -

security 4  
subscribe to user permissions 9

## - T -

Terms, glossary of 12

## - U -

user attributes 10  
user interaction 7

## Contact Us

Caplin Systems Ltd  
Triton Court  
14 Finsbury Square  
London EC2A 1BR  
Telephone: +44 20 7826 9600  
Fax: +44 20 7826 9610  
[www.caplin.com](http://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.