

CAPLIN

Trading 6.0

Trade Model Configuration XML Reference

March 2013



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.....	1
1.4	Typographical conventions.....	2
1.5	Feedback.....	2
1.6	Acknowledgments.....	2
2	Getting started.....	3
2.1	Ordering and nesting of tags	3
3	XML tag reference.....	4
3.1	<field>.....	4
3.2	<state>.....	5
3.3	<tradeModel>.....	5
3.4	<tradeModels>.....	5
3.5	<transition>.....	6
4	Glossary of terms and acronyms.....	7

1 Preface

1.1 What this document contains

This reference document describes the XML-based configuration that defines the **Trade Models** used by **Trading Adapters**, where the Trading Adapter is built using Caplin's **Trading Integration API**. The Trade Model XML is also used by **client applications**, such as applications built using **Caplin Trader**.

About Caplin document formats

This document is supplied in two 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*.

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.

1.2 Who should read this document

This document is intended for System Administrators and Software Developers who need to configure **Trade Models** for use by **Trading Adapters** and **client applications** (such as applications built using **Caplin Trader**).

1.3 Related documents

- ◆ **Caplin Trading: Integrating The Caplin Platform With A Trading System.**
Describes how a Caplin Trading Adapter is used to integrate the Caplin Platform with an existing trading system. It includes information on how to use the XML-based configuration to define Trade Models.

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.

2 Getting started

A Trade Model is used to define and control the life-cycle of a trade. It specifies all the states that a trade can be in, and the transitions between those states. Typical Trade Models are Request for Quote (RFQ) or Executable Streaming Price (ESP). The **Caplin Platform** and **client applications** are not tied to any particular Trade Model, and it is the purpose of this document to allow you to define your own.

Specifically this section explains briefly how the various XML tags may be combined to define the Trade Models. Once your Trade Model has been defined, it can be used to ensure the **Trading Adapter** and the client application's Trading GUI maintain a consistent state with each other during the lifetime of the trade.

For further details please see the document
Caplin Trading: Integrating The Caplin Platform With A Trading System.

2.1 Ordering and nesting of tags

Each top level tag is shown below, together with the child tags that it can typically contain (the children are in no particular order).

Tip: Advanced users may wish to consult the Schema (*state-model-schema.xsd*) for definitive information on the ordering and nesting of tags.

For a description of each tag and its attributes, see the [XML Tag reference](#) ⁴ section.

<tradeModels>

This is the outermost tag.

```
<tradeModels>
  <tradeModel></tradeModel> (one or more)
</tradeModels>
```

<tradeModel>

```
<tradeModel>
  <state></state> (one or more)
</tradeModel>
```

<state>

```
<state>
  <transition /> (zero or more)
</state>
```

<transition>

```
<transition>
  <field /> (zero or more)
</transition>
```

<field>

```
<field /> (no children)
```

3 XML tag reference

This is the reference information for the trade model configuration XML.

Default attribute values

In the tables that follow, if an attribute is not required (Req? = 'N') and there is a default value specified, then not supplying the attribute is equivalent to setting the attribute to this default value. If an attribute is not required and the default is '(none)', then not supplying the attribute can result in one of two behaviors, depending on the particular attribute – either the behavior is as specified in the description column of the table, or there is no effect on the appearance or behavior of the component.

3.1 <field>

<field>

Defines a DataSource message field that is relevant to the state <transition> containing this <field> tag.

Attributes:

Name	Type	Default	Req?	Description
default	string	(none)	N	Defines a default value for the field. When sending a message to the client in response to the state transition, if the field is not present in the data, this default value is sent instead. The default value does not apply to messages received from a client. Note: Default field values are applied by generated code, not the core Trading Integration API. For more information, see the section about the Trading Code Generator in the document "Caplin Trading: Integrating The Caplin Platform With A Trading System".
description	string	(none)	N	A description of the field.
name	string	(none)	Y	The name of the field.
required	boolean	(none)	N	If true, for the transition to occur, the field must be present in the DataSource message, or (when sending a message to the client) a default value must be defined. If the field is not present and (when sending a message to the client) no default is defined, a TradeException is generated.

3.2 <state>

<state>

Defines a particular state; its children define the possible transitions from this state to another.

Attributes:

Name	Type	Default	Req?	Description
name	string	(none)	Y	The name of the state, must be unique.
timeout	positive Integer	(none)	N	The time in seconds before triggering a transition to the state defined in the timeoutState attribute. This attribute only effects client applications, not the Trading Adapter.
timeoutState	string	(none)	N	The state to transition to if this state lasts longer than the time defined in timeout. This attribute only effects client applications, not the Trading Adapter.

3.3 <tradeModel>

<tradeModel>

This element, together with its children, fully defines all the states that are available for a particular trading model, and the events that can occur to move from one state to another.

Attributes:

Name	Type	Default	Req?	Description
initialState	string	(none)	Y	The initial state of the state machine generated from this model. This must be the name of one of the states defined for this model.
name	string	(none)	Y	The name of the state model. This is usually the trading protocol, for example RFQ.

3.4 <tradeModels>

<tradeModels>

Defines the state models for one or more trading protocols. This element has no attributes.

Attributes: This tag has no attributes.

3.5 <transition>

<transition>

This element defines a transition to another state; specifically it defines the event that causes it to happen and the state that it will move to. You can optionally place one or more <field> tags inside a <transition> to define the message fields that are relevant to that state transition.

Attributes:

Name	Type	Default	Req?	Description
source	string	(none)	Y	Defines the source of the transition. This can either be "client" or "server". A client transition is one that is initiated by an end user, such as the opening or execution of a trade. A server transition is one that is initiated by a response from the trading system.
target	string	(none)	Y	Defines the state that will be moved into when this transition occurs. This must be a valid state defined within this model, and may be this state.
trigger	string	(none)	Y	Defines the event that is raised to cause the transition to occur.

4 Glossary of terms and acronyms

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

Term	Definition
App	An application that runs in a web browser or on a mobile device.
Caplin Platform	An integrated suite of software that supports the services and distribution capabilities needed for web trading. It consists of Caplin Liberator , Caplin Transformer, Caplin KeyMaster, Caplin Director, and Caplin Management Console.
Caplin Trader	A complete development suite for creating HTML5 trading apps .
Caplin Integration Suite (CIS)	A set of APIs and tools for creating adapters that integrate the Caplin Platform with external systems. It includes: <ul style="list-style-type: none">◆ a Pricing Integration API◆ a Trading Integration API◆ a Permissioning Integration API◆ an Alerts Integration API◆ the Blade toolkit
Caplin Liberator	A financial internet hub that delivers data and messages in real time to and from subscribers over any network.
Client application	A browser-based or desktop application that communicates with the Caplin Platform (Caplin Liberator) through the StreamLink API .
DataSource	DataSource is the messaging infrastructure used by the Caplin Platform and Integration Adapters .
DataSource API	An API that allows server applications (including Integration Adapters) to communicate with the Caplin Platform .
DataSource application	An application that uses the DataSource API . Caplin Liberator , Caplin Transformer, and Integration Adapters are all DataSource applications.
ESP	<u>Executable Streaming Price Trade Model</u> .
Integration Adapter	A server application that allows an external system to communicate with the Caplin Platform . An Integration Adapter is a DataSource application and is created using the Caplin Integration Suite .
RFQ	<u>Request for Quote Trade Model</u> .
StreamLink API	An API that allows a client application to communicate with a Caplin Liberator . There are StreamLink APIs for various technologies; for example, Java, JavaScript, .NET and Silverlight applications, and Objective-C running on iOS.
Trading Adapter	An Integration Adapter that uses the Trading Integration API to integrate the Caplin Platform with a trading system .
Trading Integration API	An API for creating an Integration Adapter for trade messages that intelligently manages the Trade model for each trade. The Trading Integration API is part of the Caplin Integration Suite .

Term	Definition
Trade Model	The definition of a particular trading workflow. It specifies all the states that a trade can be in, and the transitions between those states. Examples of trade models are Request for Quote (RFQ) and Executable Streaming Price (ESP).
Trading system	A system, external to the Caplin Platform , that supports trading of financial instruments.

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.