

Caplin Trader Client 1.3

Composite Component Configuration XML Reference

December 2008

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..... | 3 |
| 1.5 | Feedback..... | 3 |
| 2 | Introduction to the composite component..... | 4 |
| 3 | Getting started..... | 5 |
| 3.1 | Technical assumptions and restrictions..... | 5 |
| 3.2 | Using the XML configuration markup..... | 5 |
| | An example configuration file | 5 |
| | Ordering and nesting of tags | 9 |
| 4 | XML Reference information..... | 10 |
| 4.1 | <component>..... | 10 |
| 4.2 | <compositeComponent>..... | 11 |
| 4.3 | <splitter>..... | 11 |
| 4.4 | <terrace>..... | 12 |
| 4.5 | <tower>..... | 13 |

1 Preface

1.1 What this document contains

This reference document describes the XML-based configuration that defines the layout and functionality of the composite display component in Caplin Trader Client.

The information in this document applies to Caplin Trader version 1.3.

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 Administrators and Software Developers who need to compose display components in Caplin Trader Client.

1.3 Related documents

- ◆ **Caplin Trader Client: Tree View Configuration XML Reference**
Describes the XML-based configuration that defines the layout and functionality of the tree views displayed in Caplin Trader Client.
- ◆ **Caplin Trader Client: Grid Configuration XML Reference**
Describes the XML-based configuration that defines the layout and functionality of grids displayed in Caplin Trader Client.
- ◆ **Caplin Trader Client: Simple Form Component Configuration Reference**
Describes the XML-based configuration that defines the layout and functionality of the simple form display component in Caplin Trader Client.

◆ **Caplin Trader XML Configuration Reference**

Describes the XML-based configuration that defines the layout and appearance of Caplin Trader Client through webcentric.

◆ **Caplin Trader Client: Customizing the Appearance**

Describes how to configure the on-screen layout and 'look and feel' of Caplin Trader Client.

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 |
| /AFolder/Afile.txt | 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 |
| 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.

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.

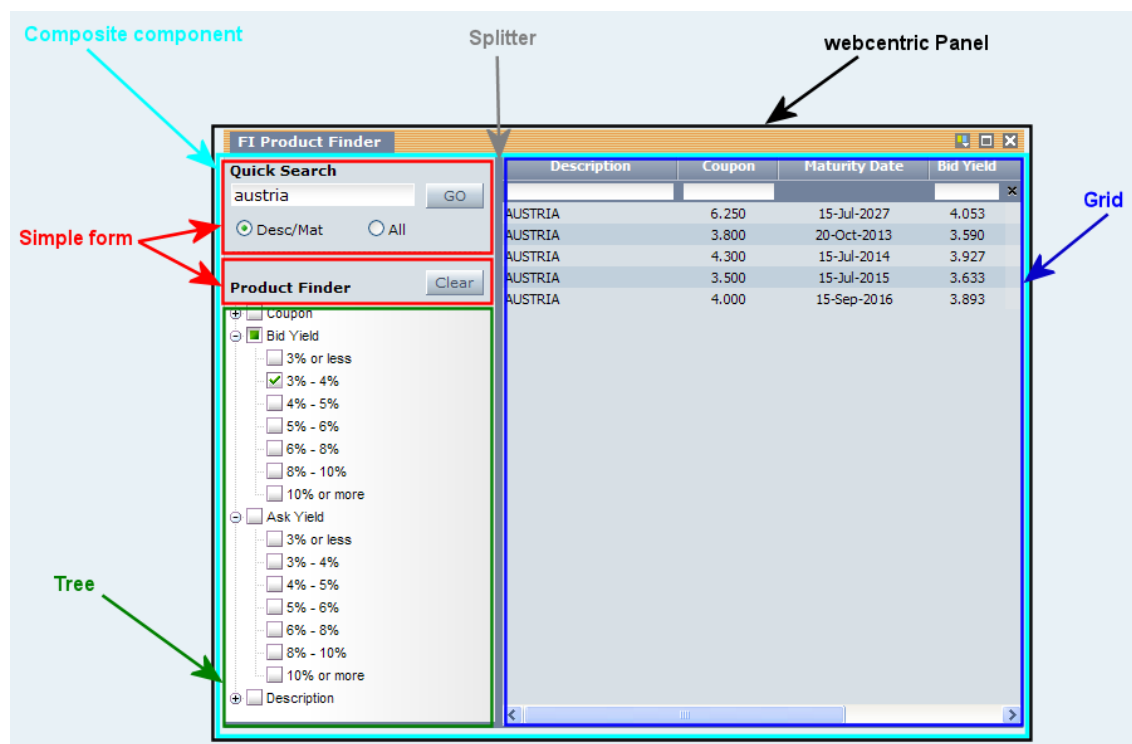
Please email your feedback to documentation@caplin.com.

2 Introduction to the composite component

In Caplin Trader Client, a composite display component is used to group together other display components, and to control and coordinate the behavior of these components in response to actions by the end user.

A typical example is the product finder, which allows an end user to filter the instruments that a grid displays by entering information into a 'quick search' text box and selecting options in a product finder tree.

The following picture shows the product finder composite component, identifying the display components and other elements that are used to construct it.



Composite component configured as a product finder

In the example above, if the end user types "Austria" (upper or lower case) in the 'Quick Search' box, then when GO is selected the grid only displays instruments with "AUSTRIA" in the 'Description' field. If the end user then selects a 'Bid Yield' of '3% - 4%' in the 'Product Finder' tree, the grid only displays instruments with "AUSTRIA" in the 'Description' field and a 'Bid Yield' of 3% to 4%.

A custom "Product Search Controller" JavaScript class filters the instruments in the grid according to the content of the 'Quick Search' box and the selected 'Product Finder' nodes.

3 Getting started

The following sections explain how the various XML tags may be combined to define composite display components for Caplin Trader Client.

3.1 Technical assumptions and restrictions

XML

The XML markup defined in this document conforms to XML version 1.0 and the XML schema version defined at <http://www.w3.org/2001/XMLSchema>.

3.2 Using the XML configuration markup

An example configuration file

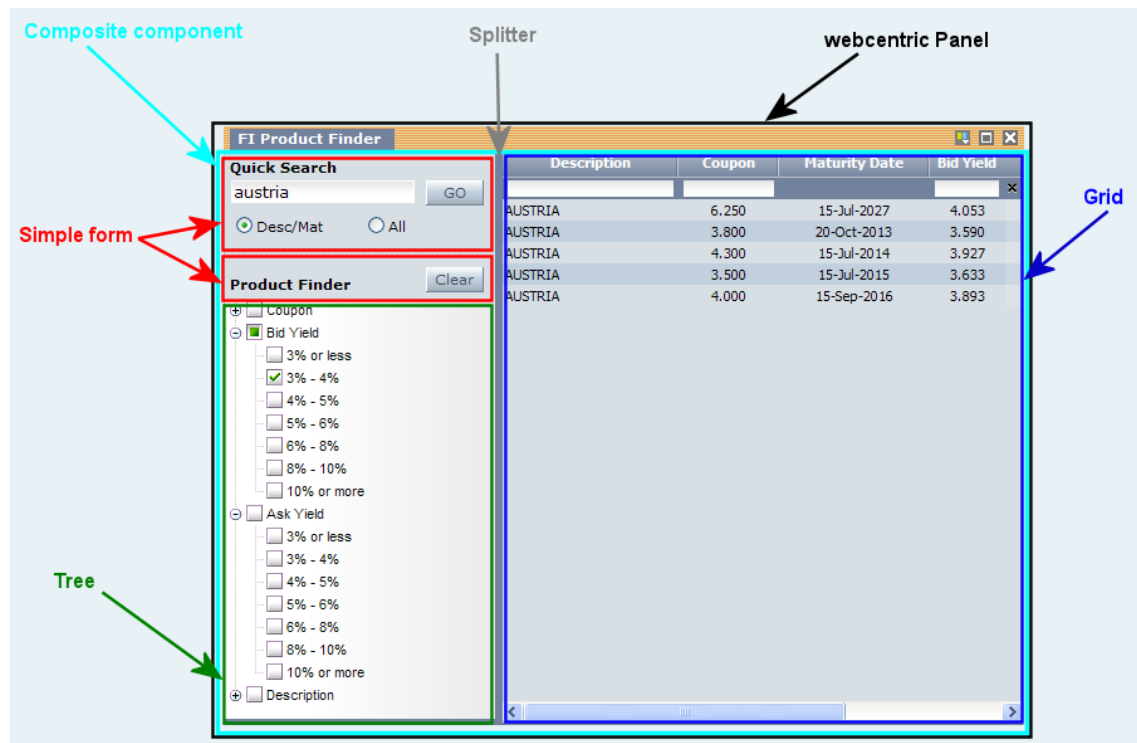
An example XML file describing a composite display component is shown below.

The XML Reference information section defines the XML tags and attributes you can use to define composite display components in Caplin Trader Client pages. Also see the section [Ordering and nesting of tags](#).

XML that configures a composite display component

```
<Panel>
  <compositeComponent controller="caplinx.composite.TreeGridComboController">
    <terrace>
      <tower width="300px">
        <component id="fiQuickSearch" height="88px">
          <simpleForm src="source/html-templates/quick-search.html" />
        </component>
        <component id="productFinderButtons" height="30px">
          <simpleForm src="source/html-templates/product-finder-buttons.html" />
        </component>
        <component id="fiTree">
          <tree baseTemplate="FiProductSearchTree" />
        </component>
      </tower>
      <splitter />
      <component id="fiGrid">
        <grid baseGrid="FiProductSearchGrid" />
      </component>
    </terrace>
  </compositeComponent>
</Panel>
```

This configuration defines a composite component that consists of two simple forms (a 'Quick Search' box and a 'Product Finder' clear button), a tree and a product grid.



Composite component configured as a product finder

An explanation of the example XML configuration

Here is an explanation of what the example XML configuration contains and how this relates to what the end-user sees on the screen:

- ◆ **<Panel>** The webcentric **<Panel>** tag is the container for the **<compositeComponent>**.

```
<Panel>
  <compositeComponent>
    ...
  </compositeComponent>
</Panel>
```

There can only be one **<compositeComponent>** tag inside a **<Panel>** tag.

For a definition of the **<Panel>** tag, see **Caplin Trader XML Configuration Reference**.

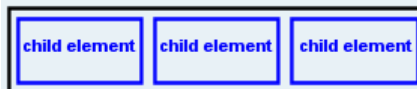
For an explanation of the webcentric Panel, see **Caplin Trader Client: Customizing the Appearance**.

- ◆ **<compositeComponent>** Contains a single **<templates>** tag and starts the composite component definition.

```
<compositeComponent controller="caplinx.composite.TreeGridComboController">
  <terrace>
    ...
  </terrace>
</compositeComponent>
```

`controller="caplinx.composite.TreeGridComboController"`: Defines a custom JavaScript class that controls the composite component. In this case the controller filters the instruments displayed in the grid according to the nodes that are selected in the 'Product Finder' tree, and the text that is typed in the 'Quick Search' box.

- ◆ **<terrace>** A layout element that arranges its children horizontally from left to right.



```
<terrace>
  <tower width="300px">
    ...
  </tower>
  <splitter />
  <component>
    ...
  </component>
</terrace>
```

This **<terrace>** is 300 pixels wide (`width="300px"`) and contains a **<tower>**, a **<splitter>**, and a **<component>**.

- ◆ **<tower>** A layout element that arranges its children vertically from top to bottom.



```
<tower>
  <component id="fiQuickSearch" height="88px">
    ...
  </component>
  <component id="productFinderButtons" height="30px">
    <simpleForm src="source/html-templates/product-finder-buttons.html" />
  </component>
  <component id="fiTree">
    ...
  </component>
</tower>
```

This tower contains three components (`id="fiQuickSearch"`, `id="productFinderButtons"`, and `id="fiTree"`).

- ◆ **<component>** A container for display components.

```
<tower>
  <component id="fiQuickSearch" height="88px">
    <simpleForm src="source/html-templates/quick-search.html" />
  </component>
  <component id="productFinderButtons" height="30px">
    <simpleForm src="source/html-templates/product-finder-buttons.html" />
  </component>
  <component id="fiTree">
    <tree baseTemplate="FiProductSearchTree" />
  </component>
</tower>

<splitter/>

<component id="fiGrid">
  <grid baseGrid="FiProductSearchGrid" />
</component>
</tower>
```

The top component of the terrace (`id="fiQuickSearch"`) is 88 pixels high (`height="88px"`) and contains a simple HTML form (`<simpleForm>`) that provides the 'Quick Search' box and Go button. The attribute `src="source/html-templates/quick-search.html"` identifies the file that contains the XHTML for the form. See **Caplin Trader Client: Simple Form Component Configuration Reference** for reference information on simple form configuration.

The middle component of the terrace (`id="productFinderButtons"`) is 30 pixels high (`height="30px"`) and contains another simple HTML form (`<simpleForm>`) that provides a Clear button. The attribute `src="source/html-templates/product-finder-buttons.html"` identifies the file that contains the XHTML for the form.

The bottom component of the terrace (`id="fiTree"`) contains the 'Product Finder' tree and takes up the remaining height of the parent tower. The tree is constructed from a base template (`baseTemplate="FiProductSearchTree"`).

See **Caplin Trader Client: Tree View Configuration XML Reference** for reference information on tree configuration.

The component at the right hand side of the terrace (`id="fiGrid"`) contains the product grid. The grid is constructed from a base grid (`baseGrid="FiProductSearchGrid"`).

See **Caplin Trader Client: Grid Configuration XML Reference** for reference information on grid configuration.

- ◆ **<splitter>** Defines a bar that can be used to space display components apart. In this case it provides a gray vertical bar between the grid and the elements in the tower (two simple forms and a tree).

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 Relax NG Schema (*compositeComponent.rnc*) for definitive information on the ordering and nesting of tags.

For a description of each tag and its attributes, see the XML Reference information section.

<compositeComponent>

This is the outermost tag

```
<compositeComponent> (one only of the following)
  <tower></tower>
  <terrace></terrace>
  <component></component>
</compositeComponent>
```

<tower>

```
<tower> (at least one of the following)
  <terrace></terrace> (zero or more)
  <tower></tower> (zero or more)
  <component></component> (zero or more)
  <splitter /> (zero or more)
</tower>
```

<terrace>

```
<terrace> (at least one of the following)
  <terrace></terrace> (zero or more)
  <tower></tower> (zero or more)
  <component></component> (zero or more)
  <splitter /> (zero or more)
</terrace>
```

<component>

```
<component> (one only of the following)
  <grid></grid> (see Caplin Trader Client: Grid Configuration XML Reference)
  <tree></tree> (see Caplin Trader Client: Tree View Configuration XML Reference)
  <simpleForm /> (see Caplin Trader Client: Simple Form Component Configuration Reference)
</component>
```

4 XML Reference information

This is the reference information for the configuration XML that describes the composite display component.

Note: The composite component configuration has XML tags called `<terrace>` and `<tower>`. The webcentric configuration defined in **Caplin Trader XML Configuration Reference** has similar tags called `<Terrace>` and `<Tower>`. Although these two sets of tags have similar effects on the appearance of Caplin Trader Client, they are distinct and operate at different levels. Always use the `<terrace>` and `<tower>` tags to configure a composite component; *never* use the `<Terrace>` or `<Tower>` tags.

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.

4.1 `<component>`

`<component>`

A component is a container for a registered display component, such as a `<grid>`, a `<tree>`, or a `<simpleForm>`, and must contain only one of these tags.

Attributes:

| Name | Type | Default | Req? | Description |
|--------|--------|---------|------|---|
| height | string | (none) | N | The height of the component as a positive integer, suffixed by 'px' (for pixels) or '%' (for percentage of the parent element's height). Omit this attribute if the parent is a <code><terrace></code> . If the parent is a <code><tower></code> and the height is omitted, then the height is determined by the height of the tower and the height of the other elements inside the tower. |
| id | string | (none) | Y | An identifier for this component. The identifier must be unique across all composite components (<code><compositeComponent></code>). |
| width | string | (none) | N | The width of the component as a positive integer, suffixed by 'px' (for pixels) or '%' (for percentage of the parent element's width). Omit this attribute if the parent is a <code><tower></code> . If the parent is a <code><terrace></code> and the width is omitted, then the width is determined by the width of the terrace and the width of the other elements inside the terrace. |

4.2 <compositeComponent>

<compositeComponent>

The outermost tag of the composite component definition XML. This tag must be placed inside a webcentric <Panel> tag, and must contain only one child tag. Valid children are the <tower>, <terrace>, and <component> tags.

Attributes:

| Name | Type | Default | Req? | Description |
|------------|--------|---------|------|--|
| controller | string | (none) | Y | The fully qualified name of the JavaScript class that controls the composite component. This class must implement the "caplin.component.composite.CompositeComponentController" interface of the Caplin Trader Client API. |

4.3 <splitter>

<splitter>

A splitter is a horizontal or vertical bar that can be inserted between components (<component>), terraces (<terrace>), and towers (<tower>) to space them apart. A splitter has no children.

Attributes:

| Name | Type | Default | Req? | Description |
|--------|--------|---------|------|--|
| height | string | '6px' | N | The height of the splitter as a positive integer, suffixed by 'px' (for pixels) or '%' (for percentage of the parent element's height). Omit this attribute if the splitter is inserted in a <terrace>. In a terrace, the height of the splitter is the height of the terrace. |
| id | string | (none) | N | An identifier for this splitter. The identifier must be unique within the scope of the composite component (<compositeComponent>). |
| width | string | '6px' | N | The width of the splitter as a positive integer, suffixed by 'px' (for pixels) or '%' (for percentage of the parent element's width). Omit this attribute if the splitter is inserted in a <tower>. In a tower, the width of the splitter is the width of the tower. |

4.4 <terrace>

<terrace>

A terrace is used to arrange child components horizontally from left to right, and must contain at least one child. Valid children are the <terrace>, <tower>, <component>, and <splitter>. At least one of the children inside a terrace must have its width attribute omitted. You can set the width or height of a terrace, but not both.

Attributes:

| Name | Type | Default | Req? | Description |
|--------|--------|---------|------|---|
| height | string | (none) | N | The height of the terrace as a positive integer, suffixed by 'px' (for pixels) or '%' (for percentage of the parent element's height). Omit this attribute if the parent of the terrace is a <compositeComponent> or <terrace>. If the parent is a <tower> and the height attribute is omitted, then the height is determined by the height of the tower and the height of the other elements inside the tower. |
| width | string | (none) | N | The width of the terrace as a positive integer, suffixed by 'px' (for pixels) or '%' (for percentage of the parent element's width). Omit this attribute if the parent of the terrace is a <compositeComponent> or <tower>. If the parent is a <terrace> and the width attribute is omitted, then the width is determined by the width of the terrace and the width of the other elements inside the terrace. |

4.5 <tower>

<tower>

A tower is used to arrange child components vertically on top of each other, and must contain at least one child. Valid children are the <tower>, <terrace>, <component>, and <splitter>. At least one of the children inside a tower must have its height attribute omitted. You can set the width or height of a tower, but not both.

Attributes:

| Name | Type | Default | Req? | Description |
|--------|--------|---------|------|---|
| height | string | (none) | N | The height of the tower as a positive integer, suffixed by 'px' (for pixels) or '%' (for percentage of the parent element's height). Omit this attribute if the parent of the tower is a <compositeComponent> or <terrace>. If the parent is a <tower> and the height attribute is omitted, then the height is determined by the height of the tower and the height of the other elements inside the tower. |
| width | string | (none) | N | The width of the tower as a positive integer, suffixed by 'px' (for pixels) or '%' (for percentage of the parent element's width). Omit this attribute if the parent of the tower is a <compositeComponent> or <tower>. If the parent is a <terrace> and the width attribute is omitted, then the width is determined by the width of the terrace and the width of the other elements inside the terrace. |

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

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.