

CAPLIN

# FX Motif 2.0

---

## Upgrade Guide

December 2014



CONFIDENTIAL

## Contents

<b>1</b>	<b>Preface .....</b>	<b>1</b>
1.1	What this document contains .....	1
1.2	About Caplin document formats .....	1
1.3	Who should read this document .....	1
1.4	Related documents .....	1
1.5	Typographical conventions.....	3
1.6	Feedback.....	3
1.7	Acknowledgments.....	4
<b>2</b>	<b>Introduction.....</b>	<b>5</b>
2.1	Workspace Management .....	5
2.2	Caplin Trader 4 and BladeRunnerJS .....	8
<b>3</b>	<b>Information for Developers.....</b>	<b>10</b>
3.1	Caplin Trader 4 and BladeRunnerJS .....	10
3.2	FX Motif library changes .....	10
3.3	Browser-specific styles.....	12
3.4	Browser compatibility .....	13
	3.4.1 Pop-outs.....	13
	3.4.2 Multiple workspaces .....	13
<b>4</b>	<b>Information for Operations Personnel .....</b>	<b>15</b>
<b>5</b>	<b>Webcentric Database Migration .....</b>	<b>16</b>
5.1	Option 1: Reset the Webcentric Database.....	16
	5.1.1 Find the JDBC information for your Webcentric database.....	16
	5.1.2 Ensure that you have a local version of your application.....	17
	5.1.3 Deploy the JDBC driver for your database .....	17
	5.1.4 (Optional) Configure the SQL dialect.....	17
	5.1.5 Reset the Webcentric database .....	18
5.2	Option 2: Upgrade the Webcentric database .....	18
	5.2.1 Update the application XML.....	21
	5.2.2 Update the user theme XML .....	21
	5.2.3 Update the static theme XML.....	22
	5.2.4 Update the default template layout.....	22
	5.2.5 Update the blank layout .....	23
	5.2.6 Open the Layout Updater tool .....	23
	5.2.7 Configure the database connection.....	24

---

5.2.8 Deploy the JDBC driver for your database .....	24
5.2.9 (Optional) Customise the tool.....	25
5.2.10 Run the tool .....	25
<b>6 Retaining the Fixed Workspace.....</b>	<b>26</b>
6.1 Upgrade your application .....	26
6.2 Switch the application theme.....	26
6.3 Set the layout.....	27
6.4 Disable the toolbars .....	27
<b>7 Further Information.....</b>	<b>28</b>

# 1 Preface

## 1.1 What this document contains

This document describes the high level differences between version 2.0 of the FX Motif and the proceeding versions, and provides advice on how to upgrade applications built on older versions of the FX Motif.

## 1.2 About Caplin document formats

This document is supplied in Portable document format (*.PDF* file), which you can read on-line using a suitable PDF reader such as Adobe Reader®. The document is formatted as a printable manual; you can print it from the PDF reader.

## 1.3 Who should read this document

This document is intended for technical personnel who want to upgrade an existing application to the new version of the FX Motif. It is also of use to operations staff who manage the deployment of the application on an application server, due to some configuration changes that are required to files like *web.xml*.

## 1.4 Related documents

### [1] Caplin Trader V4.0 Release Highlights

A one page summary of the new functionality and benefits in Caplin Trader 4.0. Available on the Caplin website: [https://www.caplin.com/downloads/developer-docs/Caplin\\_Trader\\_V4.pdf](https://www.caplin.com/downloads/developer-docs/Caplin_Trader_V4.pdf)

### [2] CT/BRJS relationship

A brief description of the relationship between Caplin Trader and BladeRunnerJS. Available on the Caplin website: <https://www.caplin.com/developer/product/ct4-brjs-caplin-trader-4/ct4-brjs-ct-brjs-relationship2>

### [3] Changes from Caplin Trader 3.11 to 4.0

A high level overview of the changes in Caplin Trader 4.0. Available on the Caplin website: <https://www.caplin.com/developer/product/ct4-brjs-caplin-trader-4/ct4-brjs-differences-in-ct-4.0>

### [4] CT 4.0 Upgrade Guide – Configuration changes

Some detailed technical differences in how applications must be configured when they are migrated

from Caplin Trader 3.x to Caplin Trader 4.0. Available on the Caplin website:  
<https://www.caplin.com/developer/product/ct4-brjs-caplin-trader-4/ct4-brjs-ct-4.0-upgrade-guide-configuration-changes>

[5] **CT 4.0 Upgrade Guide – Code changes**

Some detailed technical differences in how code must now be written when applications are migrated from Caplin Trader 3.x to Caplin Trader 4.0. Available on the Caplin website:  
<https://www.caplin.com/developer/product/ct4-brjs-caplin-trader-4/ct4-brjs-ct4.0-upgrade-guide>

[6] **BRJS Documentation site**

The BladeRunnerJS project website contains a lot of information on how to use the framework and develop applications with it: <http://bladerunnerjs.org/docs>

## 1.5 Typographical conventions

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

Type	Uses
<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 value=10 attribute is...	Code fragment in line with normal text
Something typed in	User input – things you type at the computer keyboard
<b>XYZ Product Overview</b>	Document name

**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.

## 1.6 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).

## 1.7 Acknowledgments

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

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

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

*Linux*® is the registered trademark of Linus Torvalds in the U.S. and other countries.

*Hibernate* is a registered trademark of Red Hat, Inc. in the United States and other countries.

*Eclipse* is a trademark of Eclipse Foundation, Inc.

## 2 Introduction

This document is intended as a guide to the updates and changes in version 2.0 of the Caplin FX Motif. There are two main differences between this version and previous versions, one of which relates to the functionality available to the end user and the other relates to the underlying foundations of the application.

The document is divided into five main sections.

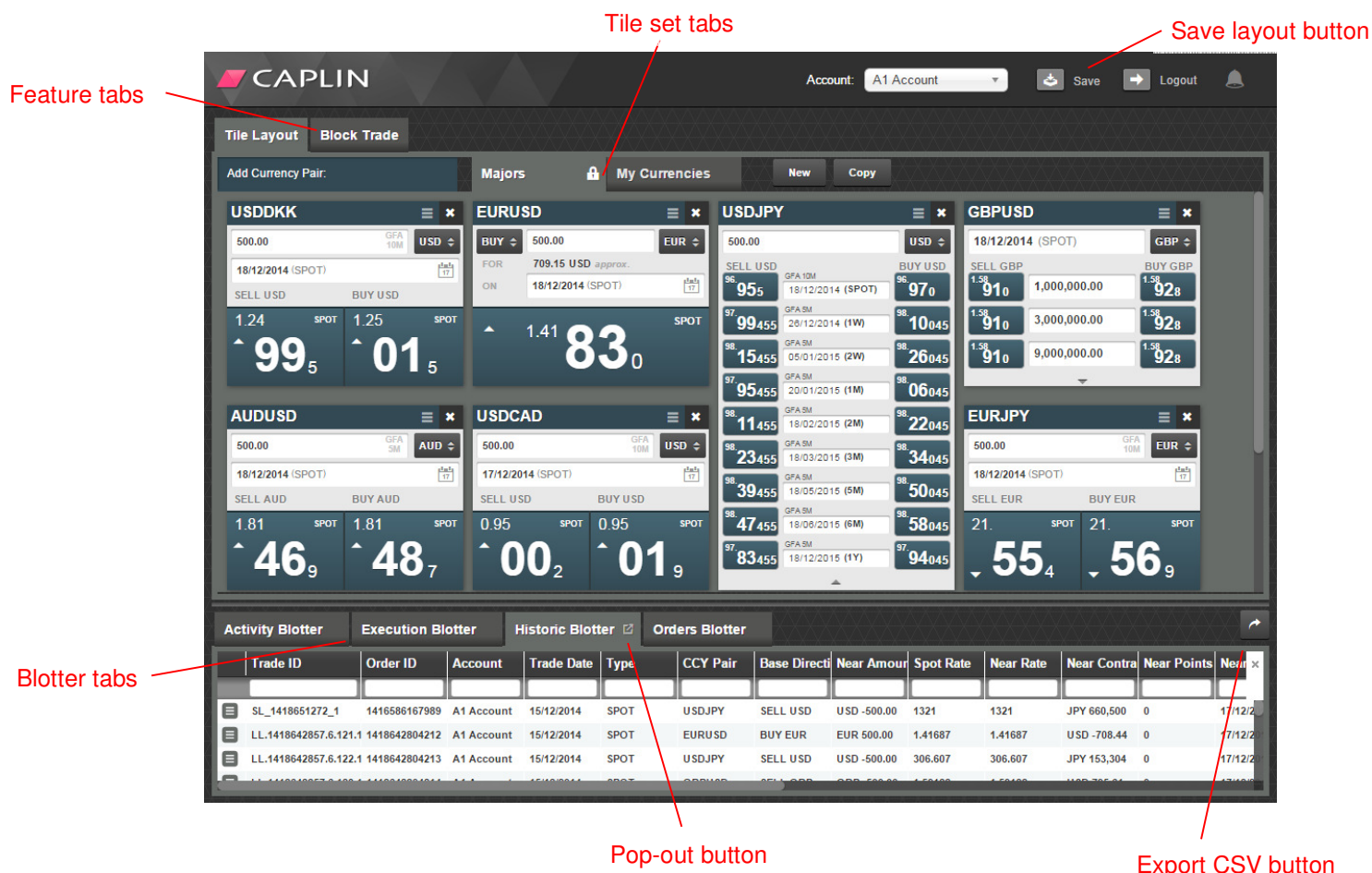
- This section describes the changes introduced in FX Motif 2.0 at a high level.
- Section 3 is intended for developers who are responsible for upgrading an existing application built on a previous version of the FX Motif to the new version. It covers topics like backwards compatibility issues in the code base.
- Section 4 is a short section describing the minor changes required to deploy the built application into a J2EE application server.
- Section 5 describes how to upgrade your existing Webcentric database to make it compatible with FX Motif 2.0. This is an important section. Your existing Webcentric database will not work with version 2.0 of the FX Motif, and you will need to carefully follow the steps described in this section if you want to upgrade and retain the saved layouts that your users have created so far.
- Section 6 explains how to upgrade your application to FX Motif 2.0 but retain the old fixed workspace application style rather than upgrade to the new workspace management style.

The remainder of this section will describe the two main changes in FX Motif 2.0.

### 2.1 Workspace Management

Below is an annotated screenshot of a previous version of the FX Motif.





There are a number of things to note about the application:

- The top section of the application is used primarily for the *Tile Layout*. If your application also includes the optional *Block Trading* blade, that will be a second tab in the top section.
- The Tile Layout contains one or more *Tile Sets*. The application starts with the *Majors* tile set and users can create their own tile sets. These are arranged in a second row of tabs, separate to the tabstrip containing the Tile Layout and Block Trading tabs. After creating or modifying a tile set the user must click the *Save Layout* button in the header bar to save the workspace.
- The lower section of the screen contains four blotters, arranged in a tabbed panel. The *Orders Blotter* will only be present if your application includes the optional Orders blade.
- You can pop the blotters out into a separate window or export the contents to a CSV file using the buttons provided. Individual trade tiles from the Tile Layout can also be popped out.

This version of the application has a number of restrictions:

- Components cannot be moved. This means that, for example, the blotters are always arranged in a tabbed panel. It is not possible to have two or more blotters in view at the same time.
- Components cannot be closed. If a user never places orders and is not interested in the order blotter, it is still not possible to close it.
- It is not possible to modify the default *Majors* tile set.

In FX Motif 2.0 we want to provide the user with far more flexibility and freedom to arrange their workspace however they want. We call this feature “workspace management”. Below is a screenshot of FX Motif 2.0.

The screenshot shows the FX Motif 2.0 interface with several components labeled:

- Feature tabs:** Located at the top left, showing 'Majors' and 'Block Trade' tabs.
- Panel toolbar:** Located at the top center, showing 'FX' and 'My Second Layout' tabs.
- Workspace tabs:** Located at the top right, showing 'A1 Account' and a dropdown menu.
- Application menu:** Located at the top right, showing a dropdown menu with options like 'New', 'Open', 'Save', 'Save All', and 'Duplicate'.
- Blotter tabs:** Located at the bottom, showing 'Activity Blotter', 'Execution Blotter', 'Historic Blotter', and 'Orders Blotter'.
- Pop-out button:** Located at the bottom right, near the 'Historic Blotter' tab.
- Export CSV button:** Located at the bottom right, near the 'Historic Blotter' tab.
- Maximize button:** Located at the bottom right, near the 'Historic Blotter' tab.

The main content area displays various currency pairs and their rates, including USDDKK, EURUSD, USDJPY, GBPUSD, AUDUSD, USDCHF, and EURJPY. Each pair has a 'BUY' and 'SELL' section with a price and a 'SPOT' button. The bottom section shows a table of trade data with columns: Trade ID, Order ID, Account, Trade Date, Type, CCY Pair, Base Dir., Near Amt., Spot Rate, Near Rate, and Near Cont.

Trade ID	Order ID	Account	Trade Date	Type	CCY Pair	Base Dir.	Near Amt.	Spot Rate	Near Rate	Near Cont.
SL_1418651272_1	1416586167989	A1 Account	15/12/2014	SPOT	USDJPY	SELL USD	USD -500.00	1321	1321	JPY 660,50
LL_1418642857.6.121.1	1418642804212	A1 Account	15/12/2014	SPOT	EURUSD	BUY EUR	EUR 500.00	1.41867	1.41867	USD -708.4
LL_1418642857.6.122.1	1418642804213	A1 Account	15/12/2014	SPOT	USDJPY	SELL USD	USD -500.00	306.607	306.607	JPY 153,30
LL_1418642857.6.123.1	1418642804214	A1 Account	15/12/2014	SPOT	GBPUSD	SELL GBP	GBP -500.00	1.59122	1.59122	USD 795.6
SL_1418461011.6_1	1416231632159	A1 Account	13/12/2014	SPOT	USDJPY	SELL USD	USD -500.00	999	999	JPY 499,50
SL_1418461011.6_2	1416233401275	A1 Account	13/12/2014	SPOT	USDJPY	SELL USD	USD -500.00	999	999	JPY 499,50

Some of the enhancements in the workspace management version of the FX Motif include:

- Components can now be moved around the workspace by drag and drop, and resized. For example, you can now drag a blotter tab out of the tabbed panel and place it somewhere else in the workspace.
- Each component can now be closed using the button provided on the tab.
- There is a new *Application Menu* that allows the user to insert components into the workspace. Initially the only new components available on the menu are pricing grids. However as the FX Motif grows to include new components for features like news and charting, they will become available from the menu.
- The *Tile Layout* no longer contains its own tile set sub-components arranged in its own tabstrip. Instead, the user can simply insert a new *Tiles Frame* from the menu and drop it into place.
- Each panel now has a toolbar. The functions available on the toolbar vary by component, but some functions will be common. For example, any component can be maximized but only the blotters provide the *Export CSV* button. The toolbar on the *Tiles Frame* component provides a text input box that is used for adding new tiles.
- The user can create their own workspaces rather than be restricted to one workspace. The open workspaces are arranged in tabs at the top of the screen. The *Workspace* section of the Application Menu is used to create new workspaces and manage the existing workspaces. Workspaces can be renamed by double-clicking the tab and typing a new name.

Workspace management provides a significantly more flexible user experience than previous versions of the FX Motif. We recommend that you migrate your application to the workspace management style if possible, however we also support the ability to upgrade your application to FX Motif 2.0 and retain the old fixed workspace. The section “Retaining the Fixed Workspace” goes into technical detail about how to achieve this.

## 2.2 Caplin Trader 4 and BladeRunnerJS

The FX Motif can be thought of as an application with three distinct layers.

FX Motif 1.x
Caplin Trader 3
BladeRunner

- The *FX Motif* is an FX trading application intended as a starting point for you to extend and customize in order to create your own unique offering. It is comprised of a number of *aspects*, *blades*, *bladesets* and *libraries* specific to FX trading, such as the trade tile, RFS ticket and Order Ticket.
- *Caplin Trader* is a development suite for building trading applications. It includes libraries such as *StreamLink* for receiving real-time data from the Caplin Platform and various core libraries such as *Webcentric* for layout management.
- *BladeRunner* is a toolkit for building large, modular web applications. It is not specific to trading or FX, but can be used for building an application in any domain.

BladeRunner, which was previously a proprietary Caplin toolkit, has been open-sourced and is now known as BladeRunnerJS (<http://bladerunnerjs.org>). The latest major version of Caplin Trader, Caplin Trader 4, is built on the new and open source BladeRunnerJS rather than the old BladeRunner toolkit. Therefore the FX Motif structure has now changed.

FX Motif 2.0
Caplin Trader 4
BladeRunnerJS

Moving to this structure should provide many benefits. By releasing BladeRunnerJS as an open source project we hope to increase community involvement and build a stronger foundation for Caplin Trader and the FX Motif. BladeRunnerJS has a plugin architecture that will allow developers in the community to provide plugins that support things like CSS pre-processors or NodeJS integration.

There are various other developer benefits and new features provided in BladeRunnerJS and Caplin Trader 4, such as the ability to write JavaScript in a modern “CommonJS” style rather than our old namespaced style. These changes are described in more detail in the “Information for Developers” section.

Caplin Trader, which contains all of the financial and trading libraries, will remain closed source because this is where the advanced domain-specific functionality resides. These are the libraries that provide real value when building trading applications.

Although it is a major version increase, the upgrade from Caplin Trader 3 to Caplin Trader 4 should be significantly easier than the upgrade from Caplin Trader 2 to Caplin Trader 3. We have endeavoured to retain backwards compatibility wherever possible in terms of the libraries and APIs provided by Caplin Trader.

A one-page release highlights summary of Caplin Trader 4 is available on the Caplin website at [https://www.caplin.com/downloads/developer-docs/Caplin\\_Trader\\_V4.pdf](https://www.caplin.com/downloads/developer-docs/Caplin_Trader_V4.pdf).

## 3 Information for Developers

For the most part, you should approach the upgrade from your current version of the FX Motif to FX Motif 2.0 according to your standard procedure. This usually involves a three-way merge between your current version of the FX Motif (the baseline), the changes you have made in your own application, and the new version of the FX Motif.

However, there are some extra considerations to be aware of.

### 3.1 Caplin Trader 4 and BladeRunnerJS

Your current application is built on Caplin Trader 3 and the proprietary BladeRunner toolkit. The new FX Motif is built on Caplin Trader 4 and the open source BladeRunnerJS toolkit.

This page provides a brief summary of the relationship between Caplin Trader and BladeRunnerJS and the features that each provide: <https://www.caplin.com/developer/product/ct4-brjs-caplin-trader-4/ct4-brjs-ct-brjs-relationship2>.

Where possible, backwards compatibility between Caplin Trader 3 and Caplin Trader 4 has been retained. However, some code changes are necessary. You should read this page for a high-level summary of the changes between Caplin Trader 3.11 (the final version of Caplin Trader 3) and Caplin Trader 4: <https://www.caplin.com/developer/product/ct4-brjs-caplin-trader-4/ct4-brjs-differences-in-ct-4.0>. Further differences are detailed in the FX Motif release note.

Be aware that, as described on the page linked above, some of the BladeRunner commands that you run while developing your application will now be different. For example, in previous versions you would run the command `bladerunner start` to launch the development server. The equivalent command in BladeRunnerJS is `brjs serve`. All BladeRunnerJS commands begin with `brjs` rather than `bladerunner`. You can execute the command `brjs help` for a full list of available commands.

This page contains some useful pointers about backwards compatibility issues you may run into while upgrading the code of your application: <https://www.caplin.com/developer/product/ct4-brjs-caplin-trader-4/ct4-brjs-ct4.0-upgrade-guide>. You should also take note of some of the changes described on this page, because they relate to files used at development time such as `jetty-env.xml`: <https://www.caplin.com/developer/product/ct4-brjs-caplin-trader-4/ct4-brjs-ct4.0-upgrade-guide-configuration-changes>.

In all cases the FX Motif has been upgraded as appropriate to run on Caplin Trader 4. You only need to be aware of these changes in case you run into problems trying to run your own custom code on Caplin Trader 4.

Finally, the BladeRunnerJS website contains a wealth of documentation on how to use the toolkit and develop applications with it: <http://bladerunnerjs.org/docs>.

### 3.2 FX Motif library changes

In addition to the *blades* and *bladesets* that provide business functionality such as RFS trading, the FX Motif also provides a number of *libraries*. The advantage of library code is that it be referenced from any blade, bladeset or *aspect*. It is therefore a good place to put code that is used by multiple

blades, such as formatters that are used for correctly formatting a price in both a ticket and a blotter.

You will notice a number of new libraries in the `lib` directory of the FX Motif. From this release onwards we plan to use more libraries that are smaller in size rather than just a few large libraries. This will prove beneficial when Caplin Trader fully supports a module system allowing your application to pull in just the libraries you need rather than a large library such as *FX*, which might contain a few classes you need plus a lot of classes that you don't need.

Please note that although the libraries have changed name and follow a new naming convention, in most cases the classes within the libraries have the same namespace and name as before. This means that your existing application code that references classes in the libraries should largely continue to work.

These libraries from previous versions of the FX Motif have been renamed:

- *caplinps* is now *cps*. This library contains generic application code written by the Caplin Professional Services team which is not specific to the FX asset class.
- *executebutton* is now *cps-fx-executebutton*. The execute button is used in the trade tiles and RFS ticket.
- *FX* is now *cps-fx*. This contains asset class specific code that relates to FX trading. However we have moved some classes out of this library and into other, more specific libraries.
- *fxhandlers* is now *cps-fxhandlers*. These classes are used by the trading library.
- *metal* is now *cps-metal*. This library contains classes that relate to precious metals trading.
- *model* is now *cps-model*. This library is also used for trading.
- *TOBO* is now *cps-tobo*. These classes relate to Trading On Behalf Of another user.
- *Trading* is now *cps-trading*.
- *UI* is now *cps-ui*. This is where you will find classes that relate to the user interface.
- *WorkbenchTools* is now *cps-workbench-tools*. These are common tools that are used by the workbenches of more than one blade.

As always, please bear in mind the usual restrictions that libraries have. A library cannot have a workbench (although this is under consideration for a future release) and you should also never modify library code directly. If you want to modify a class in one of the FX Motif libraries, you should extend it in your own child class instead. This is because any changes you make to the libraries will be lost when you next upgrade to a new version of the FX Motif.

From this release forward we intend to provide most functionality in libraries and we will try to reduce the amount of codes in blades, bladesets and aspects. Blades should contain only the minimum



amount of glue code, HTML templating, CSS and configuration required to compose a business feature out of functionality that the libraries provide.

The reason is that code in blades, bladesets and aspects must be upgraded via a three-way merge when you take a new version of the FX Motif, whereas the libraries are simply replaced in their entirety with no merge required. Therefore by reducing the amount of code that must be manually merged it should provide you with a simpler and better upgrade experience.

### 3.3 Browser-specific styles

In previous versions of the FX Motif it was possible to create CSS files in your application with a browser-specific suffix. For example, this file existed:

```
<root>/fxexecution-bladeset/blades/fxtile/themes/cotton/tile_ie8.css
```

A server-side filter in BladeRunner would detect the suffix and only serve this file to the browser if the browser was Internet Explorer 8. This mechanism allowed you to define additional styling rules that only affect certain browsers, to work around problems where some browsers render elements differently to others. In this case, the file contained only one rule:

```
.tilecontainer .buy_sell_row_headers span {  
    padding: 0 2px;  
}
```

BladeRunnerJS allows you to export your application as a flat file structure rather than a WAR file for J2EE application servers. Unfortunately, as a side effect of this new feature, the old mechanism for browser-specific styling is no longer supported. This is because if the application is served as a directory structure rather than a WAR file there are not necessarily any server-side filters that can detect the browser and decide which CSS files to serve back.

There is a new mechanism for browser-specific styling. All browser-specific styles in the FX Motif have been migrated to the new mechanism. However you will need to perform the same steps if you have any browser-specific styling of your own.

The new mechanism works like this:

- There is some code in the main entry class *FXTraderApp.js* which detects the browser the user is using and, if it is Internet Explorer, adds the version as a CSS class to the `body` element. For example, if the browser is Internet Explorer 8, the class *ie8* will be added to the `body` element.
- All styling rules that previously resided in browser-specific stylesheets must be moved into the main stylesheet. The relevant browser class must be pre-pended to the rule(s). Due to CSS specificity, this rule will override the other because it has one additional level of specificity.
- The browser-specific stylesheets must be deleted, otherwise they will be served to all browsers rather than just the intended browser and may cause regressions in the styling of your application.

As a worked example, in the case above this rule has been added to the main stylesheet *tile.css*:

```
.ie8 .tilecontainer .buy_sell_row_headers span {  
    padding: 0 2px;  
}
```

The file *tile\_ie8.css* has been deleted and is no longer present in FX Motif 2.0.

## 3.4 Browser compatibility

As with previous versions of the FX Motif, the minimum supported browser is Internet Explorer 8. In all cases we try to ensure that every feature is fully compatible with all of our supported browsers. Unfortunately, due to flaws in the way Internet Explorer 8 manages memory, it has not been possible to make two of the more advanced features of FX Motif 2.0 available in that browser.

Users will need a more modern browser to fully access these features:

### 3.4.1 Pop-outs

When a pop-out is closed, we ensure that all JavaScript references are cleaned up and nulled down as appropriate. Most browsers perform garbage collection before closing a browser window, which means that as long as the appropriate clean-up has happened in the code, no memory will be leaked when you close the window.

However, closing an Internet Explorer 8 browser window does not trigger garbage collection. Although we still clean up our references correctly, garbage collection will not take place until some unknown time in the future. This means that it is possible for the window to close with JavaScript objects still in memory, which is then leaked.

Since we cannot prevent the browser window containing the pop-out from closing until garbage collection has taken place, and in any case it isn't possible to even know when garbage collection has taken place, we cannot solve this problem.

Rather than disable pop-outs in Internet Explorer 8 completely, we display a warning telling the user that opening and closing pop-outs repeatedly is likely to cause the application performance to gradually degrade.

### 3.4.2 Multiple workspaces

FX Motif 2.0 allows the user to create multiple workspaces. Some or all of these workspaces can be opened at the same time and the user is able to switch between them using the tabs at the top of the application.

Unfortunately, performing functions such as creating a new workspace and duplicating an existing workspace cause large memory spikes in Internet Explorer 8. Duplicating workspaces is particularly problematic, because we have observed some cases in which performing a small number of



duplicate operations (two to five) can render the application almost completely unusable.

Due to the severity of this problem, creating new workspaces and duplicating workspaces are disabled completely in Internet Explorer 8. Users who hover the mouse pointer over these menu options will see a tool-tip informing them that they need to upgrade to a modern browser to use these features.

## 4 Information for Operations Personnel

In most respects the deployment procedure for applications built on FX Motif 2.0 is unchanged. The application is still packaged into a WAR file for deployment into a J2EE application server.

Some changes are required to the *web.xml* file. These changes have already been made in the FX Motif, so you can use the *web.xml* file provided as a reference. All of the required changes are described on this page of the Caplin website: <https://www.caplin.com/developer/product/ct4-brjs-caplin-trader-4/ct4-brjs-ct-4.0-upgrade-guide-configuration-changes>.

## 5 Webcentric Database Migration

The FX Motif uses a layout manager called Webcentric, which allows you to define the positions and sizes of components in an XML format. The FX Motif includes several Webcentric XML files, such as a template file that defines the default layout for new users to be presented with when they first login to the application.

As part of the FX Motif deployment process, the contents of the Webcentric XML files are stored in a database. At runtime, the FX Motif communicates with this database via a servlet whenever the user wants to perform an action such as creating a new layout or saving a modified layout. For example, when the user first logs in to the FX Motif a request is made to the Webcentric servlet for the user's current layout. The servlet will retrieve the relevant XML from the database and serve it back to the browser.

If you have an existing version of the FX Motif, you will already have one or more Webcentric database instances. These databases will contain the saved layouts for your existing users.

Unfortunately, implementing the workspace management feature in FX Motif 2.0 required quite significant changes to the Webcentric XML. FX Motif 2.0 **will not work with your existing Webcentric database**. In order to use the new FX Motif, you need to migrate your existing database to the new format. This guide will take you through the options.

**Note:** The steps in this section assume that you want to **upgrade to the workspace management version of the application**. If you want to keep the fixed layout, skip ahead to section 6.

**Tip:** "Layout" is the term used by Webcentric to describe what we have previously been calling a "workspace". The terms "layout" and "workspace" can be used interchangeably.

### 5.1 Option 1: Reset the Webcentric Database

The simplest option is to run a BRJS command that will wipe your existing Webcentric database and re-initialise it based on the new XML that ships with FX Motif 2.0. This is easy to achieve and guarantees compatibility between your Webcentric database and the FX Motif, but has the obvious downside that any saved layouts your users have created will be lost.

If you want to preserve your existing saved layouts, skip the rest of this section and go to section 5.2.

#### 5.1.1 Find the JDBC information for your Webcentric database

The command that you are going to run requires the JDBC URL for your database, and valid credentials. You will need to locate this information first.

### 5.1.2 Ensure that you have a local version of your application

You are not allowed to reset the database for an application that you don't have a local version of. For example, if the database you want to reset is for an application called `novotrader`, you will not be able to reset that database unless you have the `novotrader` application locally.

The FX Motif application is called `fxtrader`. You can import it with your own application name, to allow you to reset your database.

You will need an extracted Caplin Trader 4.0 kit and the `fxtrader` zip file.

#### Example: importing the FX Motif application as “novotrader” with the “novo” namespace

```
cd <Caplin Trader root>/sdk  
brjs import-app fxtrader-2.0.0-6369-3f4a61c.zip novotrader novo
```

### 5.1.3 Deploy the JDBC driver for your database

To make a JDBC connection to your Webcentric database you will need to use a library that is specific to your database implementation. You must make this library available to Caplin Trader by placing it in the `conf/java` directory.

For example, if you are using Microsoft SQL Server to host your Webcentric database, you will need a library with a name like `sqljdbc.jar`, `sqljdbc4.jar` or `sqljdbc41.jar` (see <http://msdn.microsoft.com/en-us/library/ms378526%28v=sql.110%29.aspx> for details).

Place a copy of this jar file in the `conf/java` directory of your Caplin Trader installation.

You will also need to make a note of the main driver class within this library. For example, if you are using the Microsoft SQL Server driver the main class is `com.microsoft.sqlserver.jdbc.SQLServerDriver`.

### 5.1.4 (Optional) Configure the SQL dialect

Open this file:

```
<Caplin Trader root>/apps/novotrader/WEB-INF/classes/hibernate.cfg.xml
```

This file contains the Hibernate configuration for your Webcentric database. The file does not specify which SQL dialect should be used. Testing at Caplin has shown this to work correctly for both H2 and Microsoft SQL Server, because Hibernate can infer the correct dialect to use based on the JDBC URL. However if you have any problems with your database connection you might need to add a line to this file that explicitly specifies the correct SQL dialect. For example, if your Webcentric database is hosted on Microsoft SQL Server you could add this line:

```
<property  
name="hibernate.dialect">org.hibernate.dialect.SQLServerDialect</property>
```

### 5.1.5 Reset the Webcentric database

You are now ready to reset your Webcentric database. Make sure you take a back-up first!

From the `sdk` directory of your Caplin Trader installation, run this command:

```
brjs reset-db <app-name> --jdbc <jdbc-url> <driver-class> <username>  
<password>
```

For example, if you are using Microsoft SQL Server your command might be something like this:

```
brjs reset-db novotrader --jdbc  
jdbc:sqlserver://mydatabasehost:55591;databaseName=novotrader  
com.microsoft.sqlserver.jdbc.SQLServerDriver dbuser dbpass
```

If it worked correctly, you should see some output like this:

```
Using D:\CaplinTrader\apps\novotrader\default-  
aspect\webcentric\layout\layoutConfiguration.xml XML specification file.
```

You should now find that your database has been reset to an empty state that is compatible with version 2.0 of the FX Motif.

## 5.2 Option 2: Upgrade the Webcentric database

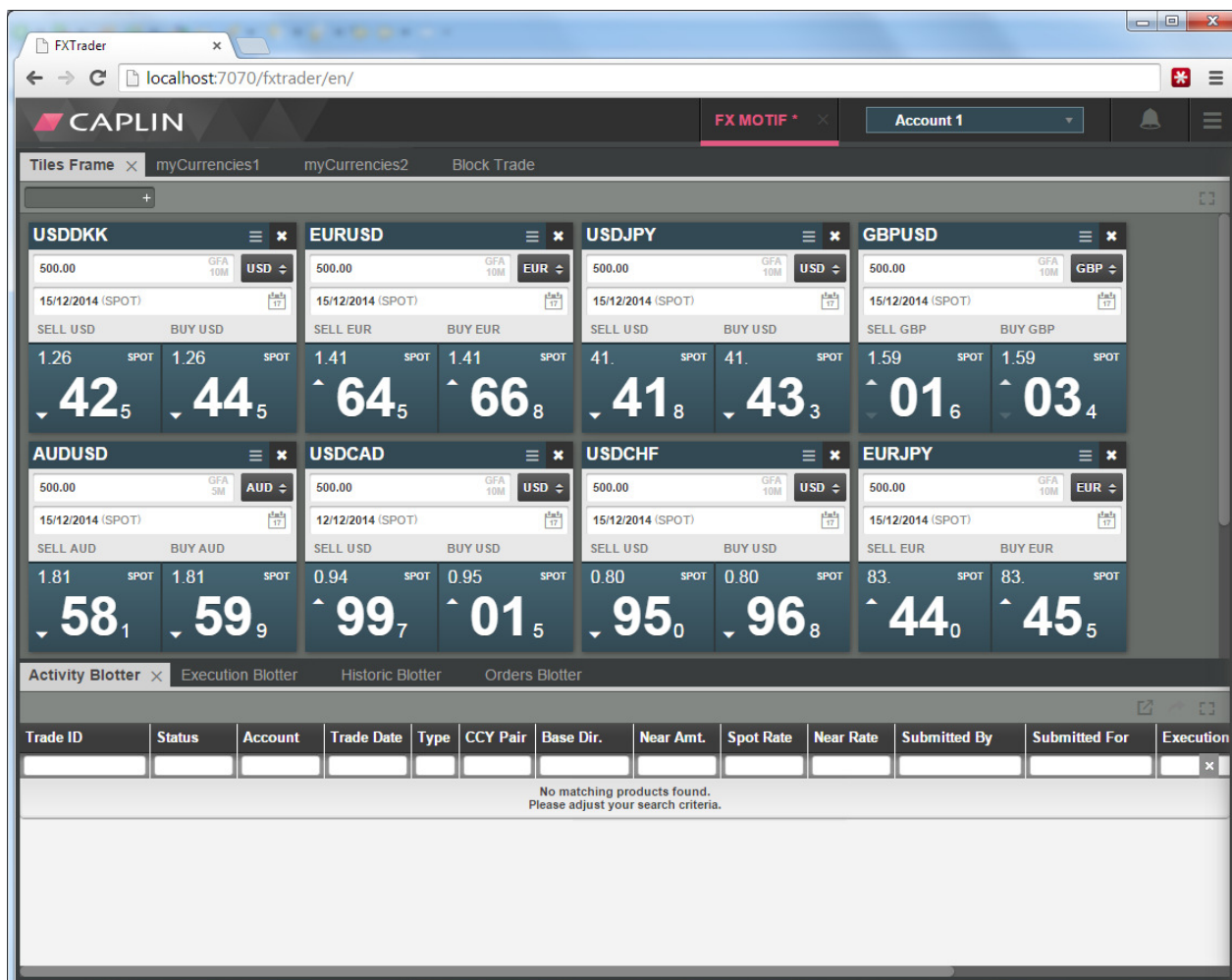
We provide XML files and a tool that should allow you to update the layouts in your database to the new schema expected by FX Motif 2.0, and retain your users existing saved layouts.

In the old fixed application style the only customization possible was for users to create new tile sets and modify the amounts, tenors etc. within the tiles. If you follow the process described in this section, each user layout should be converted to the new format but the saved tile sets will be retained.

For example, here is an FX Motif 1.8.2 layout that has been customized with two tile sets called "myCurrencies1" and "myCurrencies2". There are two top level tabs, one for the Tile Layout and one for Block Trading. The Tile Layout component contains its own tab strip with the three tile sets – the default system tile (*Majors*) set and the two custom tile sets.



If you follow the steps in this section, the upgraded FX Motif 2.0 workspace will look like this:



You can see that the Tile Layout component no longer has its own tabstrip, but there are now four top level tabs instead of two – the system tile set, the two custom tile sets and the block trading tab. The tiles that were saved in the two custom tile sets will also have been preserved.

It is important to note that some of the data in the Webcentric database exists in a single place and can easily be upgraded by simply overwriting the content with new XML. This is the case for the default layout template that every new user gets when they first login, for example.

The modified layouts created by your users cannot simply be overwritten with new XML, otherwise the modifications would be lost. To upgrade these layouts we provide a Java tool that will parse each saved layout from the database in turn, extract the modified tilesets, then build a new XML document retaining the modifications and replace the layout in the database.

The steps in this section will cover the simple replacements first, then how to use the tool to convert each user layout.

**Note:** Please be aware that our scripts and upgrade tool have been tested with the standard FX Motif Webcentric XML only. If your application extends the standard saved layout XML with your own serialized components and attributes, the scripts and tool that we provide will not retain that information. You may need to customize the scripts and/or the tool in order to perform an upgrade that works for your application.

### 5.2.1 Update the application XML

In your Webcentric database you will have a table called `apps` with a single row. The value of the `name` field should be “caplintrader” and the `xmlcontent` field should contain a blob of XML starting with the `<Application>` tag.

In the FX Motif kit you will find this file:

```
<root>/fxtrader/default-aspect/webcentric/layout/caplintrader.xml
```

You need to completely replace the existing `xmlcontent` in your database with the contents of this file, by running a query like this:

```
UPDATE [novotrader].[cap].[apps] SET xmlcontent='<Application...>' WHERE  
name='caplintrader'
```

In this and each subsequent step that involves updating the database, be sure to escape any single quotes in the XML that you are trying to insert. In Transact-SQL this can be done by changing each single quote to two consecutive single quotes.

### 5.2.2 Update the user theme XML

FX Motif 2.0 supports two themes:

- “user” – the new workspace management layout.
- “static” - the old fixed layout.

The XML for both themes can be found in the `themes` table of your Webcentric database. You should have two rows, one for the static theme and one for the user theme (even though the user theme was never supported until now).

In the FX Motif kit you will find this file:

```
<root>/fxtrader/default-aspect/webcentric/themes/user/layout-  
themes/user.xml
```

You need to completely replace the existing `xmlcontent` in the row relating to the user theme with the contents of this file, but there is a very important step you must do first.

The `user.xml` file contains references to classes with a *namespace*, for example:



```
<presenter templateId="caplinx.scaffold.header.main"
presentationModel="caplinx.scaffold.header.HeaderPM"/>
```

When you import the FX Motif, you choose your own application name, such as `novotrader`, and application namespace such as `novo`. The references in `user.xml` will therefore not be correct for your application.

Before updating the database, do a find and replace on the contents of `user.xml` and replace all instances of `caplinx` with **the correct namespace for your application**. So you should end up with lines like this, if your application used the `novo` namespace:

```
<presenter templateId="novo.scaffold.header.main"
presentationModel="novo.scaffold.header.HeaderPM"/>
```

When the text has been updated, you can update your database by running a query like this:

```
UPDATE [novotrader].[cap].[themes] SET xmlcontent='<Application...>' WHERE
name='user'
```

### 5.2.3 Update the static theme XML

Follow the same process as in section 5.2.2, except use the contents of this file:

```
<root>/fxtrader/default-aspect/webcentric/themes/static/layout-
themes/static.xml
```

Note that once again, this file contains references to classes in the `caplinx` namespace. Before updating your database you must update all of these references to use the correct namespace for your application, just as you did in step 5.2.2.

Once that has been done, in your update query replace the `xmlcontent` in the row where the value of `name` is "static":

```
UPDATE [novotrader].[cap].[themes] SET xmlcontent='<Application...>' WHERE
name='static'
```

### 5.2.4 Update the default template layout

In your Webcentric database you will have a `layouts` table containing the template layout that every new user gets when they first login to the application, plus all of the saved layouts created by your users.

We now need to update the default template. It will be one of the first few rows in the table and the value of the `name` field will be "caplintrader\_fxtrader\_layout". The first thing you need to do is change this name to be "caplintrader\_user\_layout":

```
UPDATE [novotrader].[cap].[layouts] SET name='caplintrader_user_layout'  
WHERE name='caplintrader_fxtrader_layout'
```

Now find this file in the FX Motif kit:

```
<root>/fxtrader/default-aspect/webcentric/layout/layouts/user_layout.xml
```

This is another file that contains references to the `caplinx` namespace. Replace all of these with the correct namespace for your application, as you did in the two previous steps.

When done, you need to replace the existing `xmlcontent` of the row we just modified with the modified contents of this file, by running a query like this:

```
UPDATE [novotrader].[cap].[layouts] SET xmlcontent=<Tower... ' WHERE  
name='caplintrader_user_layout'
```

### 5.2.5 Update the blank layout

There will be two blank layouts in your database that have a reference to something called the “blankLayoutDecorator”. This decorator no longer exists, so we need to update the blank layouts to remove the reference. Open this file:

```
<root>/fxtrader/default-aspect/webcentric/layout/layouts/blank_layout.xml
```

You need to replace two layouts in your database with the contents of this file. There are no references to namespaced code in this file, so you don't need to find and replace anything before updating the database.

The database query should use this `WHERE` clause, and it should affect two rows:

```
UPDATE [novotrader].[cap].[layouts] SET xmlcontent=<Panel... ' WHERE  
(name='caplintrader_blank' OR name='caplintrader_blank_layout')
```

At this point we have finished updating the static XML in the database to the new format, which means that any new users who login to the application should be given a workspace management workspace that works correctly with version 2.0 of the FX Motif.

### 5.2.6 Open the Layout Updater tool

To help upgrade the layouts that have been modified and saved by your users, we provide a Java tool that will perform the following steps:

- Connect to your Webcentric database

- For each saved layout in your database, identify the saved tile sets
- Apply an XML transformation that upgrades the layout to the new format, but retains the saved tile sets
- Write the layout back to the database

The tool is provided as a Java project rather than an executable, because it is quite important for you to understand how the tool works. You may need to modify the tool if your saved layouts differ significantly from the standard FX Motif layouts. For example, if your saved layouts contain serialized forms of your own components, you might want to modify the tool so that these saved components are translated.

The source code of the tool is heavily annotated. You might also find it useful to look at the test cases in the file *LayoutUpdaterTest.java*, which will show you how the old layouts will be transformed to the new format.

The tool is provided as a zip file containing an *Eclipse* project. You can download the Eclipse IDE from <https://eclipse.org>.

Extract *LayoutUpdater.zip* to a location on disk. Then launch Eclipse and follow these steps to import the project:

- From the menu, select *File > Import...*
- In the Import dialog, expand *General* and select *Existing Projects into Workspace*
- On the next screen, click *Browse...* and navigate to the *LayoutUpdateTool* directory on disk
- Click *Finish*

The *LayoutUpdateTool* project should now be visible in the *Package Explorer* pane.

**Note:** When you run the tool it will attempt to make a JDBC connection to your Webcentric database. This means you must either run the tool on the same machine that hosts your Webcentric database, or ensure that there are no network or firewall restrictions that will prevent the tool from connecting to the correct host and port for your database.

### 5.2.7 Configure the database connection

In the Package Explorer pane in Eclipse, open the file *LayoutUpdateTool/etc/hibernate.cfg.xml*. This is where you must configure the JDBC connection, JDBC driver class and credentials for your Webcentric database.

### 5.2.8 Deploy the JDBC driver for your database

In the *hibernate.cfg.xml* file you specified a driver class for your database. This class must be available to the tool when it runs.

You will notice that the *lib* directory of the project already contains the file *h2-1.3.167.jar*. This contains the database driver for H2, so if your Webcentric database runs on H2 you don't need to do

anything else in this step.

If your database does not run on H2, the first thing to do is find the JAR that contains the appropriate driver for your database. For example, if you are using Microsoft SQL Server to host your Webcentric database, you will need a library with a name like `sqljdbc.jar`, `sqljdbc4.jar` or `sqljdbc41.jar` (see <http://msdn.microsoft.com/en-us/library/ms378526%28v=sql.110%29.aspx> for details).

Place the JAR in the `lib` directory of the `LayoutUpdateTool` directory. Then follow these steps in Eclipse:

- If necessary, click on the project in the Package Explorer pane and press F5 to refresh it.
- Expand the `lib` directory.
- Right-click on the new JAR file and select *Build Path > Add to Build Path*

### 5.2.9 (Optional) Customise the tool

If you have configured the JDBC connection correctly and deployed the relevant JARs for your database, the tool is now ready to upgrade the layouts in your database. If you want to make any modifications to how the tool works first, this is a good time to do it. In most cases this won't be necessary.

The main file to look at is `LayoutUpdater.java`. You can also refer to the test cases in `LayoutUpdaterTest.java` to see what sort of transformations the tool will perform.

#### 5.2.10 Run the tool

Now you can run the tool and it will upgrade the saved layouts in your Webcentric database. It is easiest to run the tool from Eclipse. Before you do, it's useful to open the *Console* pane so that you can see any output or error messages generated by the tool:

- From the menu, click *Menu > Show View > Console*

The console will appear in the tabbed panel in the lower part of the screen.

Now you can run the tool:

- In the Package Explorer, expand `LayoutUpdateTool > src/main/java > com.caplin`
- Right-click on `LayoutUpdater.java` and select *Run As > Java Application*

When the tool finishes running, all of the saved layouts in your database should have been migrated to the new form.

## 6 Retaining the Fixed Workspace

Although we would encourage all clients to upgrade to the workspace management version of the FX Motif, we do support the ability to upgrade while retaining the old fixed layout. If you take this approach, your application will appear unchanged to the end user but you are still able to get the benefits of the Caplin Trader 4.0 upgrade and all of the non-workspace management related bug fixes and changes that have gone into this release.

This section will take you through the steps required to retain the fixed layout.

### 6.1 Upgrade your application

Follow your normal process to upgrade your application to FX Motif 2.0. This usually involves keeping a “vanilla” version of the FX Motif in a separate branch of your source control system, which you can merge into the branch containing your application.

### 6.2 Switch the application theme

In the file *AppConfig.js* you will notice a new line:

```
"CAPLIN.WEBCENTRIC.THEME" : "user",
```

You must override this configuration option to use the *static* theme instead. We recommend that you make this change in the file *ExtendedAppConfig.js* rather than edit *AppConfig.js* directly, so that you don't have three-way merge conflicts when you take new versions of the FX Motif containing modifications to *AppConfig.js*. Any configuration option that you add to *ExtendedAppConfig.js* will override the equivalent option in *AppConfig.js*.

In *ExtendedAppConfig.js*, add this line to your *extendedOptions* block:

```
"CAPLIN.WEBCENTRIC.THEME" : "static",
```

**Tip:** Setting this config option does three things:

Firstly, it affects which Webcentric theme the application loads. The application will use the theme from *static.xml* instead of *user.xml*.

Secondly, setting this option will make the application add the *static-layout* CSS class to the *body* element. This allows us to use CSS rules that specifically target elements that are present in both the workspace management and static versions of the application, but must be styled differently.

Lastly, certain components such as the tile layout and the block trading panel use the value of this configuration option to determine whether or not to display the grey toolbar that is only present in the workspace management version of the application.

## 6.3 Set the layout

Open this file:

```
<root>/fxtrader/default-aspect/webcentric/layout/layoutConfiguration.xml
```

This file determines which Webcentric layout to use. It currently has a line like this:

```
<layout name="@{layout.name.majors} TEMPLATE-PAGE" src="user_layout.xml"
defaultLayout="true"/>
```

Edit the line so that the default template page is *static\_layout.xml* instead of *user\_layout.xml*:

```
<layout name="@{layout.name.majors} TEMPLATE-PAGE" src="static_layout.xml"
defaultLayout="true"/>
```

## 6.4 Disable the toolbars

Open these files:

- <root>/fxblotters-bladeset/resources/xml/gridDefinitions.xml
- <root>/fxblotters-bladeset/blades/orders/resources/xml/gridDefinitions.xml

Note that the second file will only be present if your application includes the optional Orders blade.

In both of those files you will find this line:

```
<caplinps.grid.toolbar-decorator />
```

This line adds the grey toolbar containing the maximize, export CSV and pop-out buttons. This toolbar is only needed for the workspace management version of the application, the fixed layout provides other ways to access this functionality.

Comment out or remove the line that adds the decorator in both files.

Once these three changes have been made, your upgraded application should work correctly and still provide the same fixed layout as in previous versions of the FX Motif. You do not need to make any changes to your Webcentric database.

If you want to move to the workspace management version of the application later on, please remember that in addition to reverting the four changes described in this section you will also need to migrate your Webcentric database by following the instructions in the "Webcentric Database Migration" section.

## **7 Further Information**

All of the links throughout this document to pages on the Caplin website are collected in section 1.4, “Related Documents”.

As always, if you encounter problems or have questions about FX Motif 2.0 and the upgrade process, please contact Caplin Support by the usual channels.

## Contact Us

### Caplin Systems Ltd.

Cutlers Court  
115 Houndsditch  
London EC3A 7BR  
Telephone: +44 20 7826 9600

### Caplin Systems, Inc.

7 World Trade Center  
46th Floor  
New York, NY 10007  
Telephone: +1 (212) 266 0198

[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.