CAPLIN

Caplin Trader Client 1.3

Customizing The Appearance

December 2008

Contents

1 Preface

1.1 What this document contains

This document describes how to configure the on-screen layout and 'look and feel' of the Caplin Trader Client (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 Developers who need to configure the appearance of the Caplin Trader Client. It assumes a basic working knowledge of CSS (to customize the look and feel) and XML (to modify the layout).

1.3 Related documents

- **Caplin Trader Overview**

  Describes how the Caplin Trader product provides real-time market data and trading capabilities.
- **Caplin Trader Client: Customizing the Content**

  Describes how to customize the content that Caplin Trader Client displays, and how to display new content.

1.4 Typographical conventions

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

Type	***Uses***
aMethod	Function or method name
aParameter	Parameter or variable name
/AFolder/Afile.txt	File names, folders and directories
`Some code;`	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.

1.6 Acknowledgments

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

Firefox is a registered trademark of the Mozilla Foundation.

1.7 Technical assumptions and restrictions

The tutorials in this document generate some example Caplin Trader Client applications. To run these applications you need a suitable Web browser. The supported Web browsers are:

- Mozilla Firefox® version 1.5
- Microsoft Internet Explorer® version 6
- Microsoft Internet Explorer® version 7

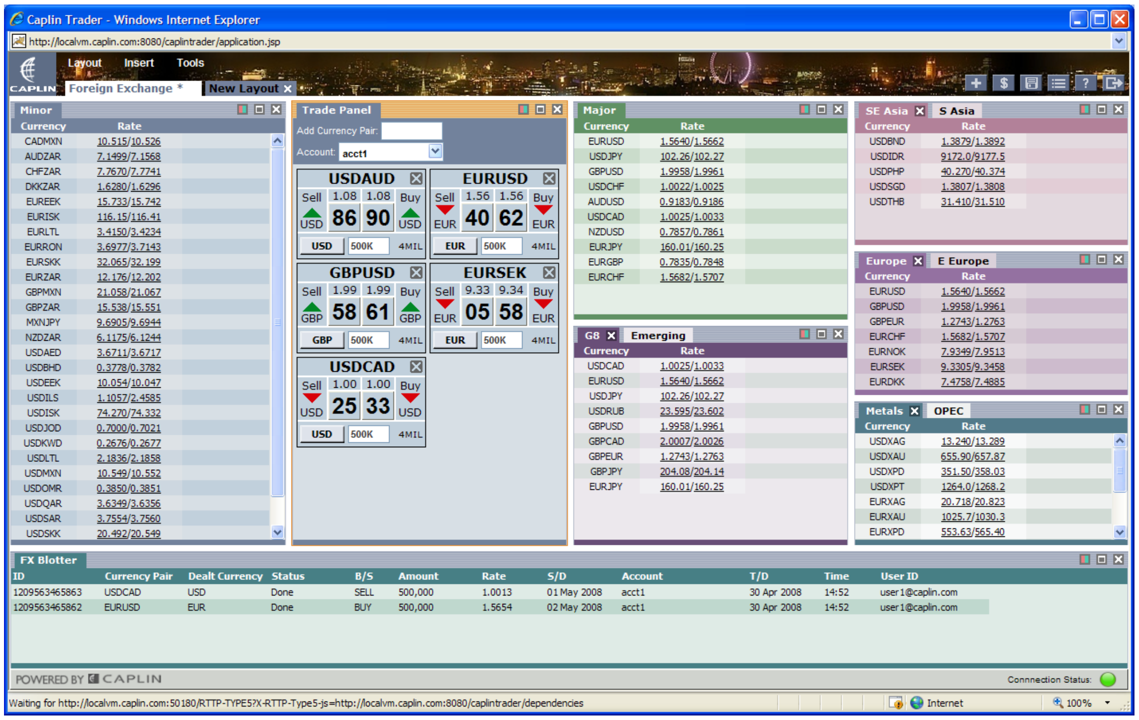
Note: When testing changes made in the tutorials, remember to *clear the browser cache* before refreshing the browser window. This ensures that code changes are reflected in the browser output.

2 What is the Caplin Trader Client?

The Caplin Trader Client is a Web browser based application for displaying real-time market data and placing trades on various financial instruments. Although the Caplin Trader Client is deployed as a web page in a browser it uses an Ajax programming framework that allows it to emulate many features of a desktop application including:

- Many windows on one screen, like a Multiple Document Interface (MDI).
- You can "drag and drop" windows to other parts of the application.
- You can resize, minimize and maximize windows.
- You can switch between layered windows via "tabs".

See also the **Caplin Trader Overview**.



Minor

Currency	Rate
CADMXN	10.515/10.526
AUDZAR	7.1499/7.1568
CHFZAR	7.7670/7.7741
DKKZAR	1.6280/1.6296
EUREEK	15.733/15.742
EURISK	116.15/116.41
EURLTL	3.4150/3.4234
EURRON	3.6977/3.7143
EURSKK	32.065/32.199
EURZAR	12.176/12.202
GBPMXN	21.058/21.067
GBPZAR	15.538/15.551
MXNJPY	9.6905/9.6944
NZDZAR	6.1175/6.1244
USDAED	3.6711/3.6717
USDBHD	0.3778/0.3782
USDEEK	10.054/10.047
USDILS	1.1057/2.4585
USDISK	74.270/74.332
USDJOD	0.7000/0.7021
USDKWD	0.2676/0.2677
USDLTL	2.1836/2.1858
USDMXN	10.549/10.552
USDOMR	0.3850/0.3851
USDQAR	3.6349/3.6356
USDSAR	3.7554/3.7560
USDSKK	20.492/20.549

Major

Currency	Rate
EURUSD	1.5640/1.5662
USDJPY	102.26/102.27
GBPUSD	1.9958/1.9961
USDCHF	1.0022/1.0025
AUDUSD	0.9183/0.9186
USDCAD	1.0025/1.0033
NZDUSD	0.7857/0.7861
EURJPY	160.01/160.25
EURGBP	0.7835/0.7848
EURCHF	1.5682/1.5707

G8 | Emerging

Currency	Rate
USDCAD	1.0025/1.0033
EURUSD	1.5640/1.5662
USDJPY	102.26/102.27
USDRUB	23.595/23.602
GBPUSD	1.9958/1.9961
GBPCAD	2.0007/2.0026
GBPEUR	1.2743/1.2763
GBPJPY	204.08/204.14
EURJPY	160.01/160.25

SE Asia | S Asia

Currency	Rate
USDBND	1.3879/1.3892
USDIDR	9172.0/9177.5
USDPHP	40.270/40.374
USDSGD	1.3807/1.3808
USDTHB	31.410/31.510

Europe | E Europe

Currency	Rate
EURUSD	1.5640/1.5662
GBPUSD	1.9958/1.9961
GBPEUR	1.2743/1.2763
EURCHF	1.5682/1.5707
EURNOK	7.9349/7.9513
EURSEK	9.3305/9.3458
EURDKK	7.4758/7.4885

Metals | OPEC

Currency	Rate
USDXAG	13.240/13.289
USDXAU	655.90/657.87
USDXPD	351.50/358.03
USDXPT	1264.0/1268.2
EURXAG	20.718/20.823
EURXAU	1025.7/1030.3
EURXPD	553.63/565.40

FX Blotter

ID	Currency Pair	Dealt Currency	Status	B/S	Amount	Rate	S/D	Account	T/D	Time	User ID
1209563465863	USDCAD	USD	Done	SELL	500,000	1.0013	01 May 2008	acct1	30 Apr 2008	14:52	user1@caplin.com
1209563465862	EURUSD	EUR	Done	BUY	500,000	1.5654	02 May 2008	acct1	30 Apr 2008	14:52	user1@caplin.com

Waiting for http://localvm.caplin.com:50180/RTTP-TYPE5?X-RTTP-Type5-js=http://localvm.caplin.com:8080/caplintrader/dependencies

The Caplin Trader Client Reference Implementation layout and appearance

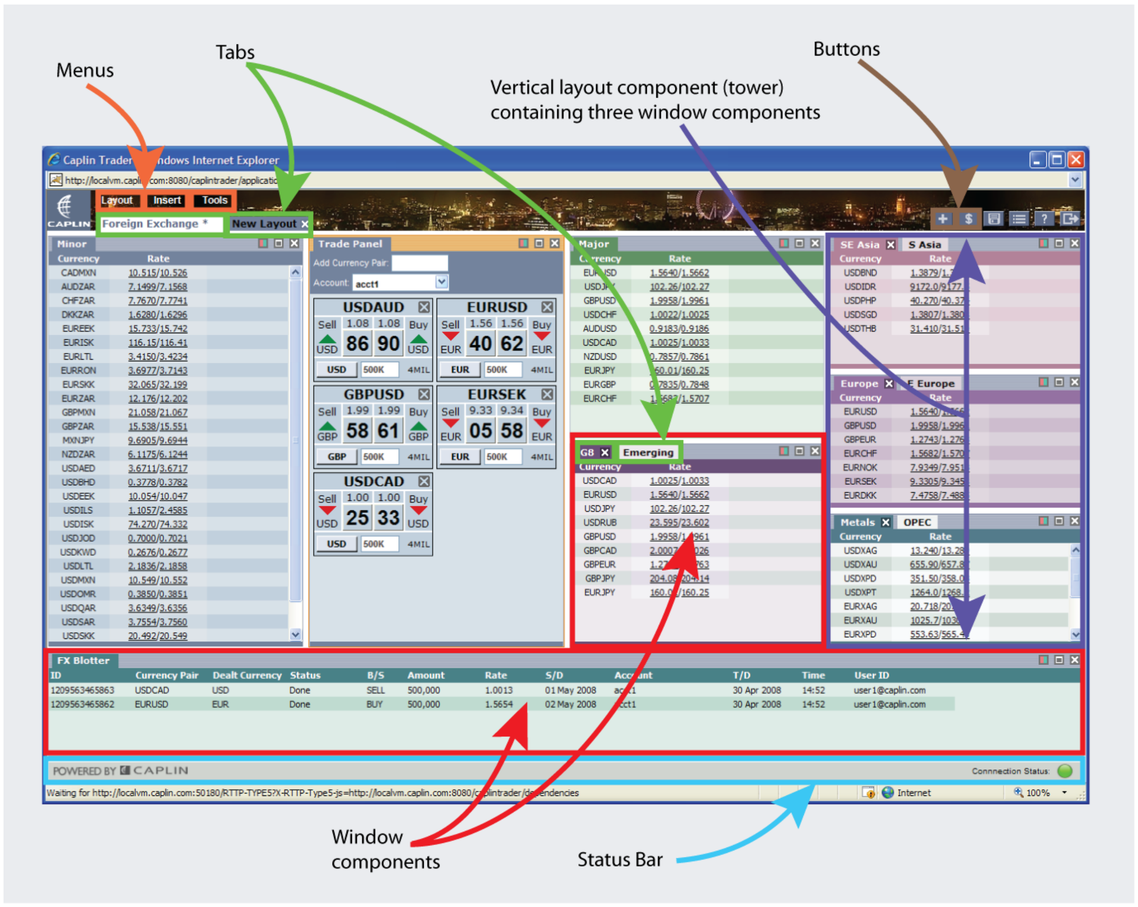
This document focuses on how to customize the *layout* and the *look and feel* of the Caplin Trader Client. You do this by editing a set of configuration files that determine the visual appearance of the interface. The Overview 5 introduces to you the main concepts behind the Caplin Trader Client interface, and the Look and Feel Tutorial 7 starts by showing you the files you will need to modify.

2.1 Overview of the Caplin Trader Client user interface

The Caplin Trader Client user interface is made from a number of tiled and nested components:

- Window components are rectangular areas that show various kinds of data in tabular and other formats.
- Layout components are used to arrange the layout of groups of windows.
- Tab, Menu, and Button components help the user interact with the windows.

These interface components are shown in the diagram below.


Menus
Tabs
Buttons
Vertical layout component (tower) containing three window components
Window components
Status Bar
Caplin Trader - Windows Internet Explorer
http://localvm.capli[illegible]com:8080/caplintrader/applicatio
CAPLIN
Layout
Insert
Tools
Foreign Exchange *
New Layout
Trade Panel
Add Currency Pair:
Account: acct1
USDAUD Sell 1.08 86 1.08 90 Buy USD USD USD 500K 4MIL
EURUSD Sell 1.56 40 1.56 62 Buy EUR EUR EUR 500K 4MIL
GBPUSD Sell 1.99 58 1.99 61 Buy GBP GBP GBP 500K 4MIL
EURSEK Sell 9.33 05 9.34 58 Buy EUR EUR EUR 500K 4MIL
USDCAD Sell 1.00 25 1.00 33 Buy USD USD USD 500K 4MIL
G8
Emerging
SE Asia
S Asia
Europe
E Europe
Metals
OPEC
POWERED BY CAPLIN
Connection Status:
Internet
100%

Minor

Currency	Rate
CADMXN	10.515/10.526
AUDZAR	7.1499/7.1568
CHFZAR	7.7670/7.7741
DKKZAR	1.6280/1.6296
EUREEK	15.733/15.742
EURISK	116.15/116.41
EURLTL	3.4150/3.4234
EURRON	3.6977/3.7143
EURSKK	32.065/32.199
EURZAR	12.176/12.202
GBPMXN	21.058/21.067
GBPZAR	15.538/15.551
MXNJPY	9.6905/9.6944
NZDZAR	6.1175/6.1244
USDAED	3.6711/3.6717
USDBHD	0.3778/0.3782
USDEEK	10.054/10.047
USDILS	1.1057/2.4585
USDISK	74.270/74.332
USDJOD	0.7000/0.7021
USDKWD	0.2676/0.2677
USDLTL	2.1836/2.1858
USDMXN	10.549/10.552
USDOMR	0.3850/0.3851
USDQAR	3.6349/3.6356
USDSAR	3.7554/3.7560
USDSKK	20.492/20.549

Major

Currency	Rate
EUR[illegible]SD	1.5640/1.5662
USDJ[illegible]	102.26/102.27
GBPUSD	1.9958/1.9961
USDCHF	1.0022/1.0025
AUDUSD	0.9183/0.9186
USDCAD	1.0025/1.0033
NZDUSD	0.7857/0.7861
EURJPY	[illegible]60.01/160.25
EURGBP	0[illegible]7835/0.7848
EURCHF	1[illegible]/1.5707

G8

Currency	Rate
USDCAD	1.0025/1.0033
EURUSD	1.5640/1.5662
USDJPY	102.26/102.27
USDRUB	23.595/23.602
GBPUSD	1.9958/1[illegible]961
GBPCAD	2.0007[illegible]026
GBPEUR	1.27[illegible]763
GBPJPY	204.08[illegible]204.14
EURJPY	160.0[illegible]/160.25

SE Asia

Currency	Rate
USDBND	1.3879/1.[illegible]
USDIDR	9172.0/9177.[illegible]
USDPHP	40.270/40.37[illegible]
USDSGD	1.3807/1.380[illegible]
USDTHB	31.410/31.51[illegible]

Europe

Currency	Rate
EURUSD	1.5640/[illegible]
GBPUSD	1.9958/1.996[illegible]
GBPEUR	1.2743/1.276[illegible]
EURCHF	1.5682/1.570[illegible]
EURNOK	7.9349/7.951[illegible]
EURSEK	9.3305/9.345[illegible]
EURDKK	7.4758/7.488[illegible]

Metals

Currency	Rate
USDXAG	13.240/13.28[illegible]
USDXAU	655.90/657.8[illegible]
USDXPD	351.50/358.0[illegible]
USDXPT	1264.0/1268.[illegible]
EURXAG	20.718/20[illegible]
EURXAU	1025.7/103[illegible]
EURXPD	553.63/565.4[illegible]

FX Blotter

ID	Currency Pair	Dealt Currency	Status	B/S	Amount	Rate	S/D	Account	T/D	Time	User ID
1209563465863	USDCAD	USD	Done	SELL	500,000	1.0013	01 May 2008	ac[illegible]t1	30 Apr 2008	14:52	user1@caplin.com
1209563465862	EURUSD	EUR	Done	BUY	500,000	1.5654	02 May 2008	[illegible]cct1	30 Apr 2008	14:52	user1@caplin.com

Waiting for http://localvm.caplin.com:50180/RTTP-TYPE5?X-RTTP-Type5-js=http://localvm.caplin.com:8080/c[illegible]plintrader/d[illegible]endencies

Different types of interface component

To change the layout or "look and feel" of Caplin Trader Client, you need to be familiar with **XML** files and **CSS** style sheets respectively. However, in our first tutorial Changing the look and feel of the Caplin Trader Client 7 you will get straight in and modify only the look and feel of your default Caplin Trader installation, as pictured above. You won't be able to change the layout at this stage, you will simply be altering the cosmetic and branding aspects of the interface. For this you will need mostly CSS skills but some very simple XML editing will also be needed.

Note: If however you wish to experiment with changing the layout of the interface, please go to the Layout Concepts 51 section where we will introduce you to the fundamentals and guide you in building a Caplin Trader GUI application from the ground up.

3 Tutorial: Changing the look and feel of the Caplin Trader Client

In this section we're going to dive straight in and explain how to change the color scheme and branding of your Caplin Trader Client. We're not changing the layout. Since the directory where Caplin Trader is installed may not be the same in every case, we suggest you follow the procedure below to record this in an environment variable.

- Find the directory where the installed Caplin Trader software is located (which contains the directory *apps*).

  For example */home/CaplinTrader*

- Define an environment variable **`CT_INSTALL_DIR`** that resolves to your chosen installation directory.

  For example:

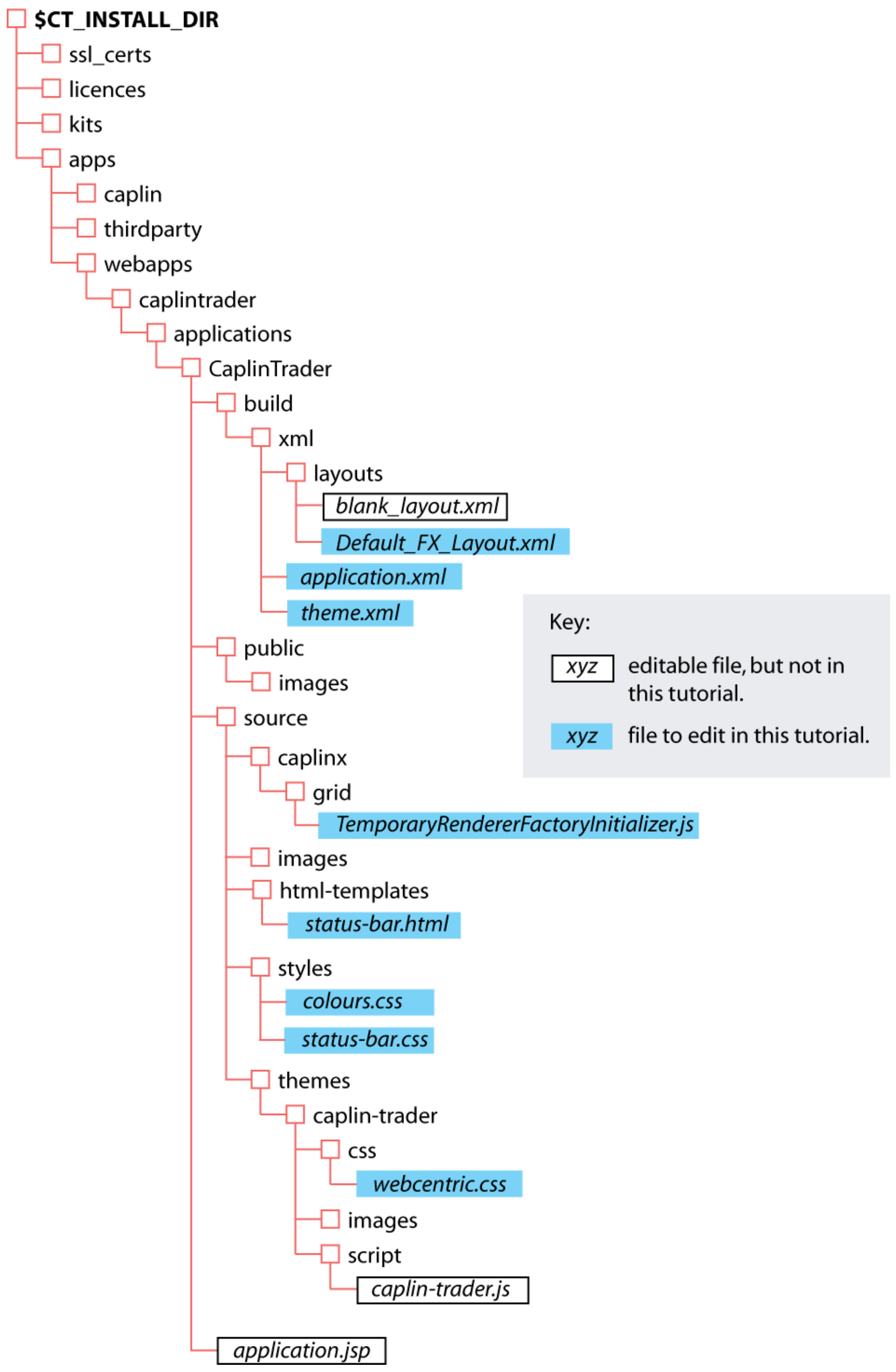
  ```
  export CT_INSTALL_DIR=/home/CaplinTrader
  ```

Note: In the rest of this guide the installation directory where the Caplin Trader evaluation software is located is referred to using the environment variable name ***$CT_INSTALL_DIR***

Under the folder ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader* you will find the folders *build* and *source*. Under the *build* folder hierarchy you will find a number of XML files that can be modified: *application.xml*, *blank_layout.xml*, *Default_FX_Layout.xml*, and *theme.xml*.

Under the *source* folder hierarchy you will find a number of CSS, HTML and JavaScript files that you can modify:

- *webcentric.css*
- *status-bar.css*
- *colors.css*
- *status-bar.html*
- *TemporaryRendererFactoryInitializer.js.*


$CT_INSTALL_DIR
ssl_certs
licences
kits
apps
caplin
thirdparty
webapps
caplintrader
applications
CaplinTrader
build
xml
layouts
blank_layout.xml
Default_FX_Layout.xml
application.xml
theme.xml
public
images
source
caplinx
grid
TemporaryRendererFactoryInitializer.js
images
html-templates
status-bar.html
styles
colours.css
status-bar.css
themes
caplin-trader
css
webcentric.css
images
script
caplin-trader.js
application.jsp
Key:
xyz editable file, but not in this tutorial.
xyz file to edit in this tutorial.

You will be modifying the following files:

- *application.xml*
- *Default_FX_Layout.xml*
- *theme.xml*
- *colours.css*
- *status-bar.css*
- *webcentric.css*
- *status-bar.html*
- *TemporaryRendererFactoryInitializer.js*

Before we begin the tutorial make sure you can run the Caplin Trader Client in your browser by going to the URL:
http://<your.domain.name>:8080/caplintrader/
(where <your.domain.name> is replaced with the appropriate domain for your installation).

We're going to show you five ways to choose your own branding:

- How to change the logo on the interface 10.
- How to change the default background images in the top bar 12.
- How to change the image and text in the status bar 13.
- How to fix colors of your choosing over the whole of the interface 15.
- How to add new background graphics for the title bars of the windows 28.

3.1 Changing the logo

The logo in the top left hand corner of the Caplin Trader Client can be changed to one of your choosing. To do this, open up the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/build/xml/theme.xml* in your favorite text editor.

Search for:

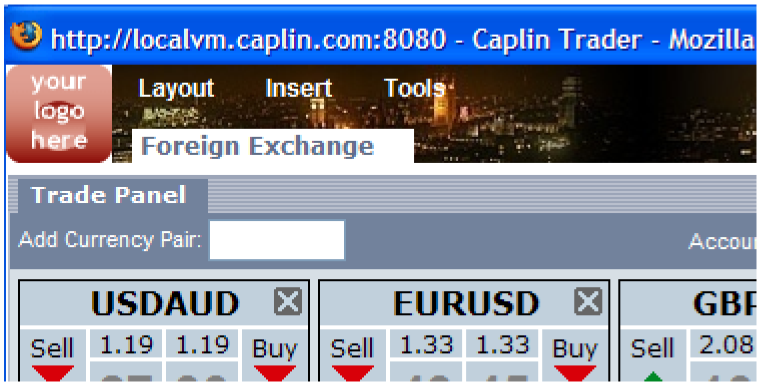
```
src="images/logo-tiny.gif"
```

You will find this path is given in an `<ImagePanel/>` tag. This path can be replaced to point to any image of your own. It is best to make your logo a maximum of 49 pixels high. We have one in this installation at ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/images/genericLogo.gif*, so change the `src` reference to be:

```
<ImagePanel src="images/genericLogo.gif" width="49" ...
```

Note: To execute the above XML change, you need to do the following:
Open a terminal shell (if you do not have one open already).
Type `cd $CT_INSTALL_DIR` followed by return.
Type `java -jar kits/CaplinTrader/webcentric_database_populator.jar apps/webapps/caplintrader/applications/CaplinTrader/build/xml`, followed by return.
This process will rebuild the user preferences database with the changes that you have just made.

Clear the cache in your browser and refresh the Caplin Trader Client and you should see the following **"your logo here"** branding in the top left of the web page:



Changes for wide Logos

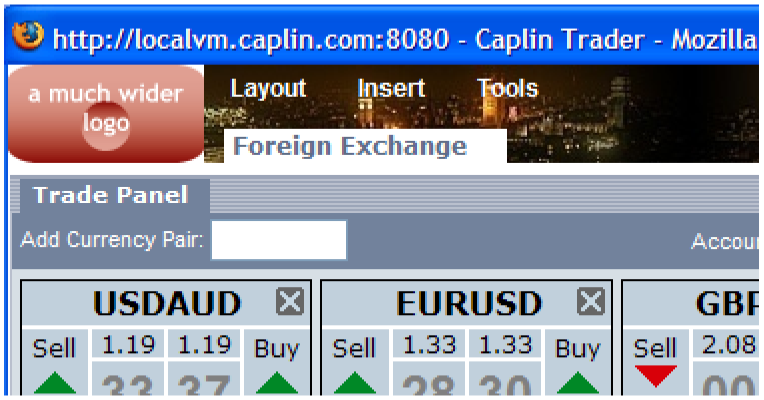
If you wish to use a very wide logo, then you will also need to change the value of the width attribute too. In the example below we will use a wide logo with a width of 98 pixels.

Change the same `<ImagePanel/>` tag to read as follows:

```
<ImagePanel src="images/genericLogoWide.gif" width="98" ...
```

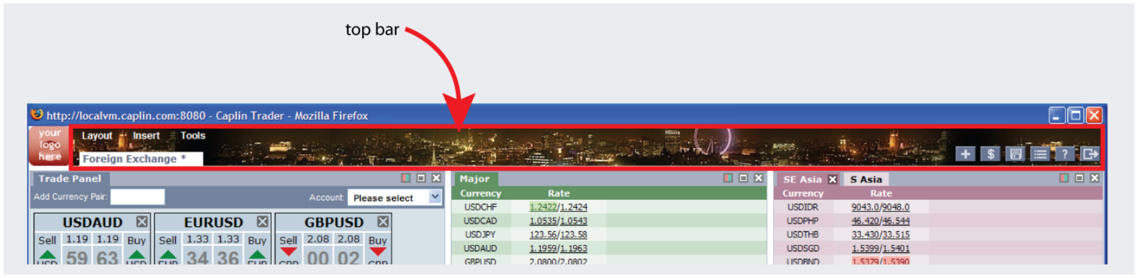
Note: To execute the above XML change, you need to do the following:
Open a terminal shell (if you do not have one open already).
Type `cd $CT_INSTALL_DIR` followed by return.
Type `java -jar kits/CaplinTrader/webcentric_database_populator.jar apps/webapps/caplintrader/applications/CaplinTrader/build/xml`, followed by return.
This process will rebuild the user preferences database with the changes that you have just made.

If you now clear your cache and refresh your browser, you will see that the menu bar and Foreign Exchange tab have been moved to the right to accommodate the wider logo.



3.2 Changing the background image in the top bar

Now we will look at changing the background image for the top bar. The 'top bar' area is marked in red in the figure below.



Top bar of Caplin Trader

This image is easy to change. Open up the file *$CT_INSTALL_DIR/apps/ webapps/caplintrader/applications/CaplinTrader/source/themes/caplin-trader/css/webcentric.css*, and search for: `Panel_header`

You should find a declaration like the following:

```
.Panel_header {
      background : url(../);
}
```

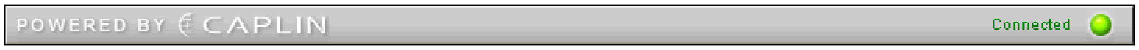
This file is located at *$CT_INSTALL_DIR/apps/ webapps/caplintrader/applications/CaplinTrader/source/themes/caplin-trader/images/newTopbars.jpg*

Replace `topbars.jpg` with the new alternative, `newTopbars.jpg`:

```
.Panel_header {
      background : url(../images/newTopbars.jpg);
}
```

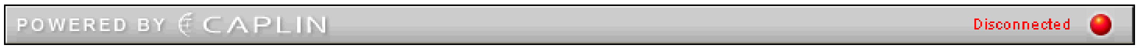
3.3 Changing the image and text in the status bar

The status bar is located at the bottom of Caplin Trader Client. On the right hand side of this bar you will find some text and a green, red, or amber image.



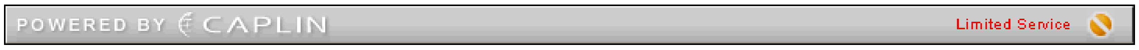
The Status bar is located at the bottom of Caplin Trader Client

A green image indicates that Caplin Trader Client is connected to a Liberator server, and a (flashing) red image indicates that Caplin Trader Client is not connected to a Liberator server.



A (flashing) red image is displayed when there is no Liberator connection

A flashing amber image is displayed when Liberator cannot connect to a DataSource provider.



A (flashing) amber image indicates Liberator cannot connect to a DataSource provider

You can change both the text that is displayed in the status bar and the image that indicates the connection status. First we will look at how to change the text.

Open the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/caplinx/grid/TemporaryRendererFactoryInitializer.js* in a text editor. Search for:

```
oRewiteMap[caplin.framework.connection.ConnectionStatusFieldModel.
                                        CONNECTION_STATUS_OK] = "Connected";
oRewiteMap[caplin.framework.connection.ConnectionStatusFieldModel.
                                  CONNECTION_STATUS_ERROR] = "Disconnected";
oRewiteMap[caplin.framework.connection.ConnectionStatusFieldModel.
                             CONNECTION_STATUS_LIMITED] = "Limited Service";
```

Modify the text as shown below and then save the file:

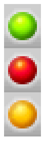
```
oRewiteMap[caplin.framework.connection.ConnectionStatusFieldModel.
                                    CONNECTION_STATUS_OK] = "Connection Up";
oRewiteMap[caplin.framework.connection.ConnectionStatusFieldModel.
                               CONNECTION_STATUS_ERROR] = "Connection Down";
oRewiteMap[caplin.framework.connection.ConnectionStatusFieldModel.
                          CONNECTION_STATUS_LIMITED] = "Connection Limited";
```

Now that we have changed the status bar text, we will look at how to change the image that indicates the connection status.

Search for:

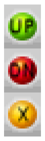
```
oConfig[caplin.renderer.control.StatefulImageControl.CONFIG_IMAGE] =
 "source/themes/caplin-trader/images/connection_status/three_part_connection.gif";
```

The text above defines the path to the connection status image. The upper part of the image is displayed when Caplin Trader Client is connected to the Liberator server, the middle part when Caplin Trader Client *is not* connected to the Liberator server, and the lower part when a connection to a DataSource provider cannot be made.



This multiple role of the image is necessary in case there is a network connection failure, when it would not be possible to contact the application server to get the image that would indicate the failure. A single connection status image overcomes this problem since it is sent to the browser when the application loads.

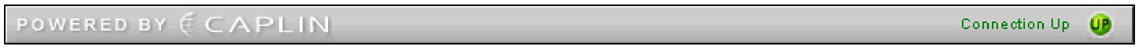
You can change the path to point to another image that you have created, or you can use the example alternative image provided with Caplin Trader Client.



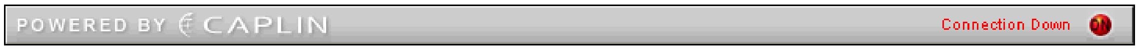
To use this alternative image, change the text as shown below and then save the file:

```
oConfig[caplin.renderer.control.StatefulImageControl.CONFIG_IMAGE] =
 "source/themes/caplin-trader/images/connection_status/alt_three_part_connection.gif";
```

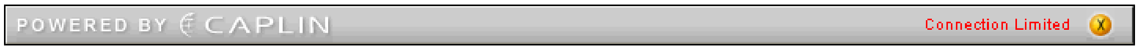
To see the effect of the changes that you have just made, clear your browser cache and refresh your browser.



Alternative text and image when connected to Liberator



Alternative text and image when not connected to Liberator

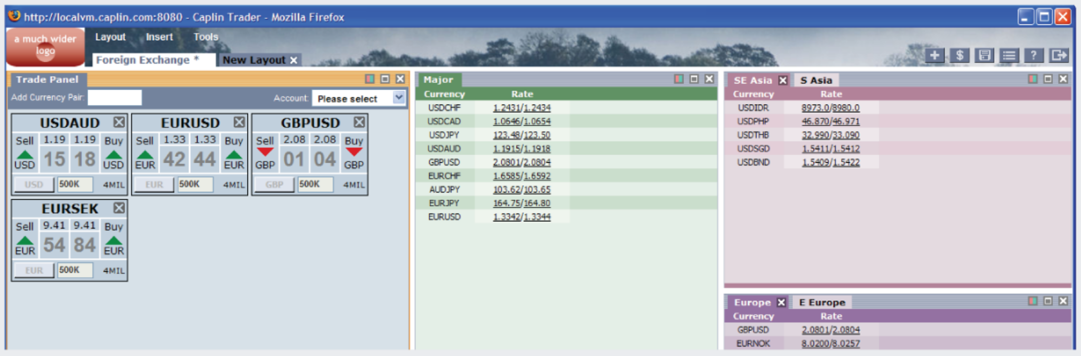


Alternative text and image when not connected to a DataSource provider

3.4 Changing the color scheme

Here we are going to show you how to change the color scheme of the Caplin Trader Client, as shown in the top half of the picture below, to the orange scheme shown underneath it.

Color scheme before:

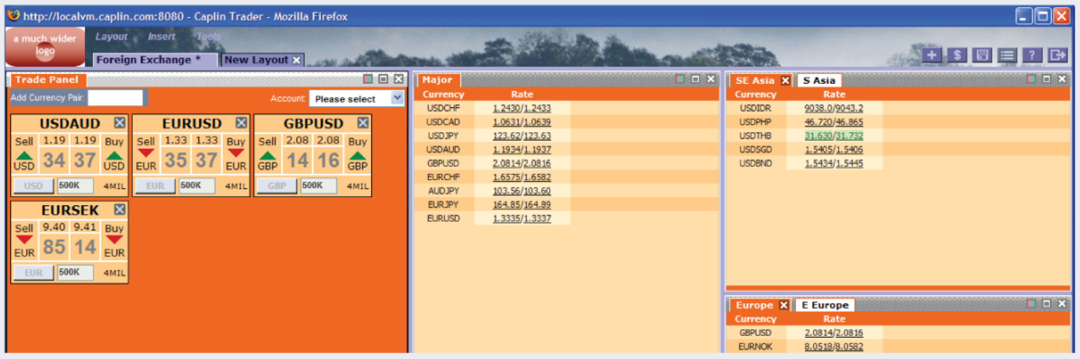

http://localvm.caplin.com:8080 - Caplin Trader - Mozilla Firefox
a much wider logo
Layout
Insert
Tools
Foreign Exchange *
New Layout
Trade Panel
Add Currency Pair:
Account:
Please select
USDAUD Sell 1.19 15 1.19 18 Buy USD USD USD 500K 4MIL
EURUSD Sell 1.33 42 1.33 44 Buy EUR EUR EUR 500K 4MIL
GBPUSD Sell 2.08 01 2.08 04 Buy GBP GBP GBP 500K 4MIL
EURSEK Sell 9.41 54 9.41 84 Buy EUR EUR EUR 500K 4MIL
Major
SE Asia
S Asia
Europe
E Europe

Currency	Rate
USDCHF	1.2431/1.2434
USDCAD	1.0646/1.0654
USDJPY	123.48/123.50
USDAUD	1.1915/1.1918
GBPUSD	2.0801/2.0804
EURCHF	1.6585/1.6592
AUDJPY	103.62/103.65
EURJPY	164.75/164.80
EURUSD	1.3342/1.3344

Currency	Rate
USDIDR	8973.0/8980.0
USDPHP	46.870/46.971
USDTHB	32.990/33.090
USDSGD	1.5411/1.5412
USDBND	1.5409/1.5422

Currency	Rate
GBPUSD	2.0801/2.0804
EURNOK	8.0200/8.0257

Color scheme after:


http://localvm.caplin.com:8080 - Caplin Trader - Mozilla Firefox
a much wider logo
Layout
Insert
Tools
Foreign Exchange *
New Layout
Trade Panel
Add Currency Pair:
Account:
Please select
USDAUD Sell 1.19 34 1.19 37 Buy USD USD USD 500K 4MIL
EURUSD Sell 1.33 35 1.33 37 Buy EUR EUR EUR 500K 4MIL
GBPUSD Sell 2.08 14 2.08 16 Buy GBP GBP GBP 500K 4MIL
EURSEK Sell 9.40 85 9.41 14 Buy EUR EUR EUR 500K 4MIL
Major
SE Asia
S Asia
Europe
E Europe

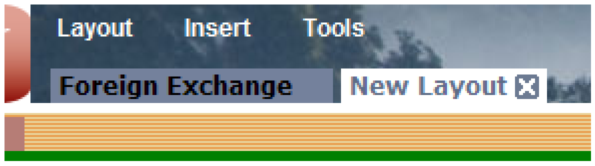
Currency	Rate
USDCHF	1.2430/1.2433
USDCAD	1.0631/1.0639
USDJPY	123.62/123.63
USDAUD	1.1934/1.1937
GBPUSD	2.0814/2.0816
EURCHF	1.6575/1.6582
AUDJPY	103.56/103.60
EURJPY	164.85/164.89
EURUSD	1.3335/1.3337

Currency	Rate
USDIDR	9038.0/9043.2
USDPHP	46.720/46.865
USDTHB	31.630/31.732
USDSGD	1.5405/1.5406
USDBND	1.5434/1.5445

Currency	Rate
GBPUSD	2.0814/2.0816
EURNOK	8.0518/8.0582

Changing the layout tab colors

In the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/themes/caplin-trader/css/webcentric.css* you will find some commented out code which refers to the tabs at the top of the layout. Each tab allows the user to switch between 'layouts'. There is always at least one tab, in this case "Foreign Exchange". But if you click on the "Layout" menu in the application and then select "New...", a new layout will be created with its own tab.



New layout tab after "New" is selected from the "Layout" menu

You can see the difference in colors between the selected tab (that represents the current layout) and the unselected tabs for those layouts behind. To change the look and feel of the selected tab, search for the following code:

```
/* Selected layout tab: background color */
.Tabstrip_layout .tab_body_selected,
.Tabstrip_layout .tab_close_selected,
.Tabstrip_layout .tab_tail_selected,
.Tabstrip_layout .tabs_tail_selected {
      background-color    : #fff;
      color               : #5E6D86;
}
/* Selected layout tabs: borders */
.Tabstrip_layout .tab_body_selected {
      border-left: 1px solid #fff;
      border-top: 1px solid #fff;
}
.Tabstrip_layout .tab_tail_selected,
.Tabstrip_layout .tabs_tail_selected,
.Tabstrip_layout .tab_close_selected {
      border-right: 1px solid #fff;
      border-top: 1px solid #fff;
}
```

These CSS declarations set the background color, the font color and the border style and colors of the selected tab. Change the colors used in this code as follows:

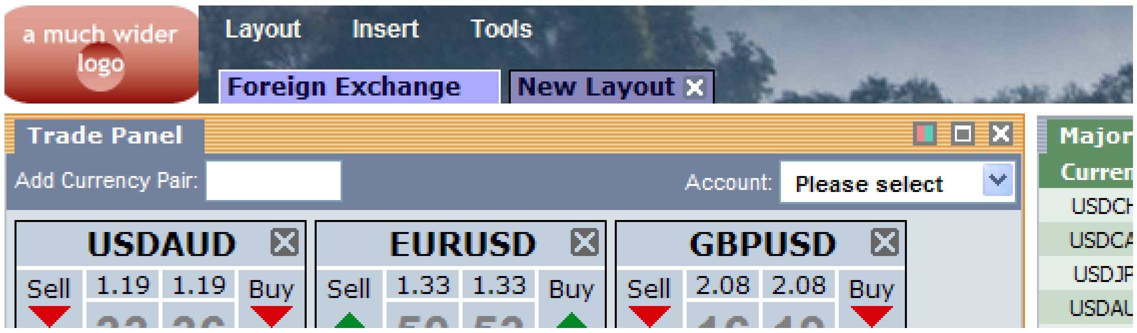
```
/* Selected layout tab: background color */
.Tabstrip_layout .tab_body_selected,
.Tabstrip_layout .tab_close_selected,
.Tabstrip_layout .tab_tail_selected,
.Tabstrip_layout .tabs_tail_selected {
      background-color    : #aaf;
      color               : #003;
}
/* Selected layout tabs: borders */
.Tabstrip_layout .tab_body_selected {
      border-left: 1px solid #fff;
      border-top: 1px solid #fff;
}
.Tabstrip_layout .tab_tail_selected,
.Tabstrip_layout .tabs_tail_selected,
.Tabstrip_layout .tab_close_selected {
      border-right: 1px solid #fff;
      border-top: 1px solid #fff;
```

In the above code, we have left the colors of the tab borders the same as they were (white, using the hex code `#fff`), but we have set the background color to a light purple color (`#aaf`), and the text color to dark blue (`#003`).

Now we also need to choose a color scheme for the unselected tabs; so find the next set declarations and change the colors as follows:

```
/* Unselected layout tabs: background color */
.Tabstrip_layout .tab_body,
.Tabstrip_layout .tab_close,
.Tabstrip_layout .tab_tail,
.Tabstrip_layout .tabs_tail {
      background-color    : #88b;
      color               : #003;
      cursor              : pointer;
}
/* Unselected layout tabs: borders */
.Tabstrip_layout .tab_body {
      border-left: 1px solid #000;
      border-top: 1px solid #000;
}
.Tabstrip_layout .tab_tail,
.Tabstrip_layout .tabs_tail,
.Tabstrip_layout .tab_close {
      border-right: 1px solid #000;
      border-top: 1px solid #000;
}
```

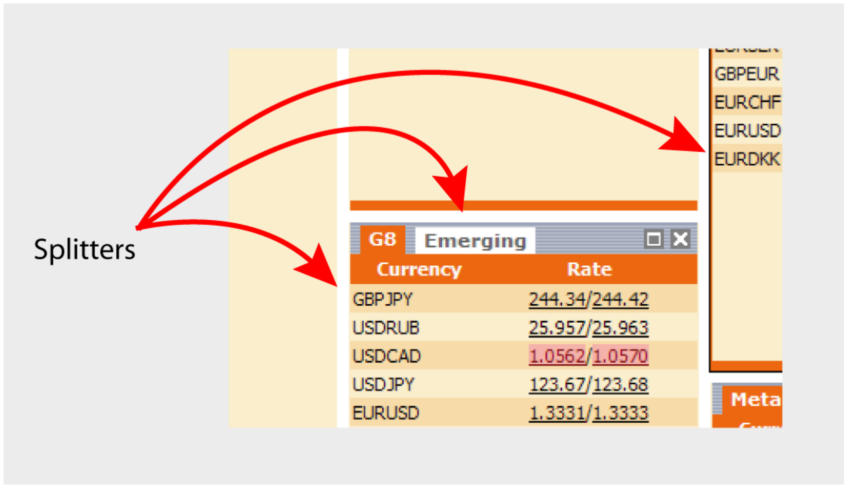
This code will color unselected layout tabs with a darker shade of light purple (`#88b`), and use black for the top, left and right borders (`#000`). As before, we use very dark blue (`#003`) for the text color. Remember you need to have at least two layouts (and hence two tabs). The following screenshot shows the effect of these changes.



This screenshot shows the new light blue selected layout tab on the left and the new dark blue unselected tabs

Splitter colors

Splitters are the bars or gaps between window components. They come in horizontal and vertical varieties and when you click and drag them you can resize the components.



Currency	Rate
GBPJPY	244.34/244.42
USDRUB	25.957/25.963
USDCAD	1.0562/1.0570
USDJPY	123.67/123.68
EURUSD	1.3331/1.3333

GBPEUR
EURCHF
EURUSD
EURDKK

The splitters are the horizontal and vertical bars that separate other components

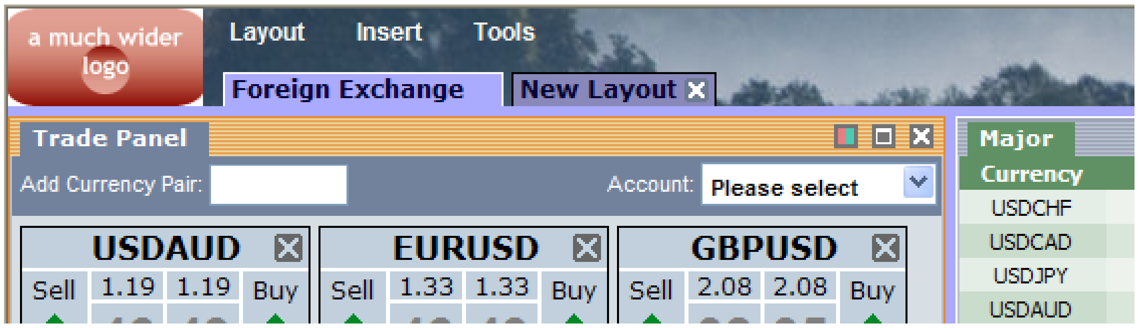
We're going to change these to a light purple. In ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/themes/caplin-trader/css/webcentric.css*, search for the text `/* splitter color */` and change `background-color` as follows:

```
/* splitter color */
.Splitter_horizontal,
.Splitter_vertical,
.stretcher_splitterDrag {
      background-color: #aaf;
}
```

Note that we also change the class `.stretcher_splitterDrag` here which sets the color of the splitters while they are being dragged.

Now, if you clear your cache and refresh your browser, you may notice that whilst the splitter colors have been changed, the very top border which separates the layout tabs from the window components is still white. We want this border to be the same color as the splitters and the selected layout tab, so search for the text `/* top layout  border color */` and change `background-color` as follows:

```
/* top layout border color */
.border_layout_top,
.border_layout_top .border_left,
.border_layout_top .border_right {
      background-color: #aaf !important;
}
```



Changing the splitters and top border to light purple

Changing the layout menu colors

You can change the color of the menus at the top of the application ("Layout", "Insert" and "Tools"), and the color of the menu background, as you move the mouse over them.

In the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/themes/caplin-trader/css/webcentric.css*, find the following declarations and modify the `font-style` as suggested:

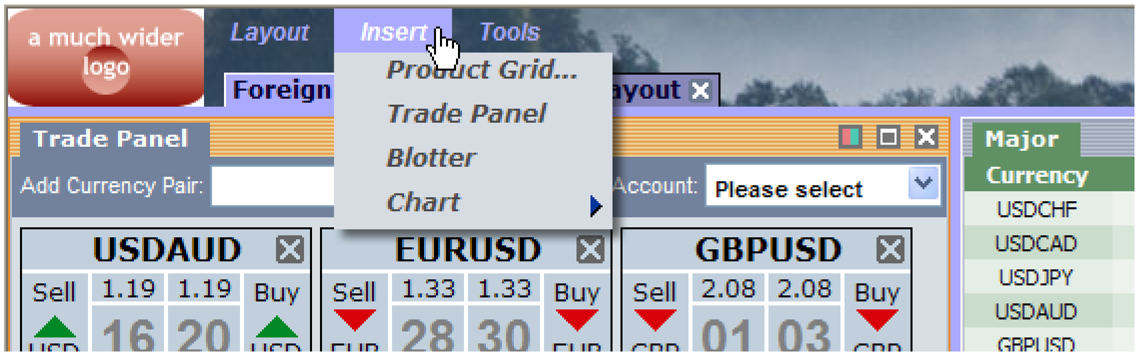
```
/* menu buttons */
.menuButton,
.menuButton_mouseout,
.menuButton_mouseover,
.menuButton_mousedown {
      font-family     : arial;
      font-size       : 12px;
      font-weight     : bold;
      font-style      : italic;
      cursor          : pointer;
      text-align      : center;
      white-space     : nowrap;
      line-height     : 23px;
      padding         : 4px 13px 4px 13px;
}
```

The first set of declarations in this code sets the font styles and sizes for menus. By grouping the three menu states `.menuButton`, `.menuButton_mouseover`, and `.menuButton_mousedown`, we can give them all the same typographic style. In the case below we use the same settings that the menus had already, with the exception of `font-style`, which we now set to `italic`.

However, for the colors and backgrounds, we are going to set the three states separately. We use the light purple font for the menu buttons (`#aaf`). When the user places the mouse over the menu button the button background changes to a darker shade of purple (#88b) and the font color is set to white. Lastly, when the user clicks on the menu, the menu button font stays white but we lighten the background color to `#aaf`.

To complete these changes, find the declarations below and modify as follows:

```
.menuButton,
.menuButton_mouseout
{
      background-color    : transparent;
      color               : #aaf;
}
.menuButton_mouseover
{
      background-color    : #88b;
      color               : #fff;
}
.menuButton_mousedown
{
      background-color    : #aaf;
      color               : #fff;
}
```



The purple and italic scheme for the menus

If you want to change the colors of the menu items themselves (which are currently charcoal grey on light grey), then you will need to change a few more lines of CSS. Scroll down in the CSS code for the following declarations:

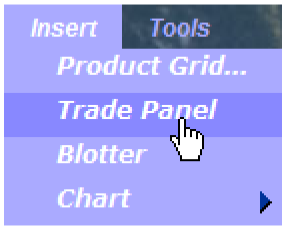
```
.menuitem_default,
.menuitem_selected {
      overflow            : hidden;
      font-family         : verdana;
      font-size           : 12px;
      font-weight         : bolder;
      CURSOR              : default;
      background-color    : transparent;
}

.menuitem_default .menuitem_left,
.menuitem_default .menuitem_main,
.menuitem_default .menuitem_right {
      background-color    : #D7DBE2;
      color               : #494848;
}

.menuitem_selected .menuitem_left,
.menuitem_selected .menuitem_main,
.menuitem_selected .menuitem_right {
      background-color    : #000;
      color               : #fff;
}
```

The first block defines the default settings for the menu items; to make this italic like the menu buttons, add the following line to this block:

```
.menuitem_default,
.menuitem_selected {
      overflow          : hidden;
      font-family       : verdana;
      font-size         : 12px;
      font-style        : italic;
      font-weight       : bolder;
      CURSOR            : default;
      background-color  : transparent;
}
```

Now change the other two blocks to set the menu items to have a light purple background color with white text, but darken the tint of purple when you roll over them:

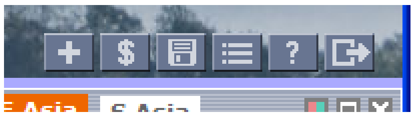
```
.menuitem_default .menuitem_left,
.menuitem_default .menuitem_main,
.menuitem_default .menuitem_right {
      background-color    : #aaf;
      color               : #fff;
}

.menuitem_selected .menuitem_left,
.menuitem_selected .menuitem_main,
.menuitem_selected .menuitem_right {
      background-color    : #88f;
      color               : #fff;
}
```



Changing the color of the menu items

Changing the layout button colors

The layout buttons on the far right-hand side of the application can also be changed.



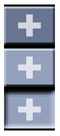
Since these are graphical images you will need to edit them in an image editor. The buttons can be found in the following directory:

$CT_INSTALL_DIR*/apps/*
webapps/caplintrader/applications/CaplinTrader/source/themes/caplin-trader/images/tabstrip

The files to be edited are:

ButtonPlus.png
ButtonDollar.png
ButtonSave.png
ButtonPreferences.png
ButtonHelp.png
ButtonExit.png

For example, *ButtonPlus.png*, when viewed in an image editor looks like this:



It is composed of three images, one for each of the three interaction (mouseover) states that can occur. The top image is how the button looks if the mouse pointer is not over it, the second is for when the mouse pointer is over the button and the third is for when the mouse button is pressed while the mouse is over the button.

To change the color of these buttons, you can either edit the above files directly, or create a new set of images files and refer to them in the XML file:
$CT_INSTALL_DIR*/apps/webapps/caplintrader/applications/CaplinTrader/build/xml/theme.xml*

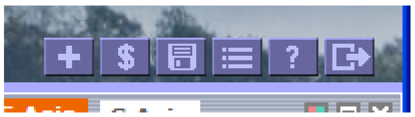
For example, we have created a second set of images with "*_blue*" appended to the files names. So, *ButtonPlus.png* becomes *ButtonPlus_blue.png*. Edit the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/build/xml/theme.xml*, searching for the reference to `ButtonPlus.png`. Replace `ButtonPlus.png` with `ButtonPlus_blue.png`. Do the same for the other buttons too, so `ButtonDollar.png` becomes `ButtonDollar_blue.png`, etc.

The markup should look like this:

```
<Tabstrip  style="layout" handle_height="25" align="top">
   <Button ... img="%theme%/images/tabstrip/ButtonPlus_blue.png"
               tooltip="Add Product Grid">
      <showDialog ... />
   </Button>
   <Button ... img="%theme%/images/tabstrip/ButtonDollar_blue.png"
               tooltip="Product Search">
      <showDialog ... />
   </Button>
   <Button ... img="%theme%/images/tabstrip/ButtonSave_blue.png"
               tooltip="Save">
      <saveLayout event="DOMActivate"/>
   </Button>
   <Button ... img="%theme%/images/tabstrip/ButtonPreferences_blue.png"
               tooltip="Preferences">
      <message ... />
   </Button>
   <Button ... img="%theme%/images/tabstrip/ButtonHelp_blue.png"
               tooltip="Help">
      <action xref="//*[@id='open-help-dialog']" defer="true" />
   </Button>
   <Button ... img="%theme%/images/tabstrip/ButtonExit_blue.png"
               tooltip="Log Out">
      <eval event="DOMActivate" xref="Declarations/eval[@id='logout']" defer="true"/>
   </Button>
</Tabstrip>
```

Note: To execute the above XML change, you need to do the following:
Open a terminal shell (if you do not have one open already).
Type `cd $CT_INSTALL_DIR` followed by return.
Type `java -jar kits/CaplinTrader/webcentric_database_populator.jar apps/webapps/caplintrader/applications/CaplinTrader/build/xml`, followed by return.
This process will rebuild the user preferences database with the changes that you have just made.

Now clear the cache and refresh the browser and the layout buttons will now appear in a shade of blue:



Additional border color changes

There are some other elements whose colors we can change. If you select (by clicking on) a window component such as a Grid or Trade Panel, you will notice a thin light orange border around it. We're going to make this black so it stands out from the rest of the scheme.

In the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/themes/caplin-trader/css/webcentric.css*, find the following code:

```
/* component outer border */
.border_outer_active {background-color:#e68b2c;}
.border_outer_inactive {background-color:white;}
```

Change this to be as follows:

```
/* component outer border */
.border_outer_active {background-color:black;}
.border_outer_inactive {background-color:white;}
```

Now we make one last change on the borders. If you look closely you'll notice that there are two thin vertical gray lines in between the outer component border and the content of the component. These *internal* border lines can also be colored. Here we leave them set to light gray when the component is not selected and also set them to black when it is, for added emphasis:

```
/* component inner borders */
.border_inner_active {background-color:black;}
.border_inner_inactive {background-color:#B9BFCD;}
```

Changing the border color of dialog boxes

In the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/themes/caplin-trader/css/webcentric.css* you can define the colors you wish to see.

There are two borders on the dialogs, a thin black outer border and a thicker white inner border. These colors are not currently affected by selecting or unselecting the dialog.

Find the following code fragment.

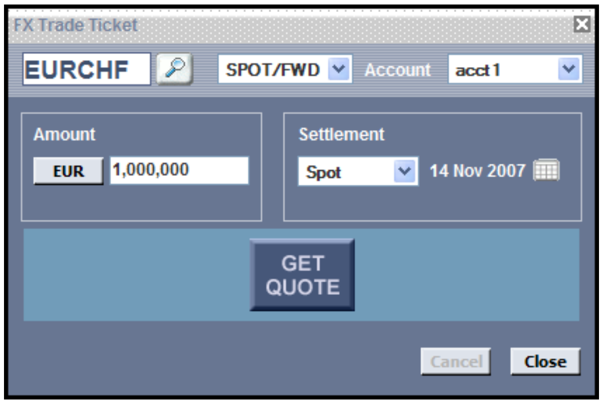
```
/* dialog outer border */
.border_dialogouter_active {background-color:#000;}
.border_dialogouter_inactive {background-color:#000;}

/* dialog inner border */
.border_dialoginner_active {background-color:#fff;}
.border_dialoginner_inactive {background-color:#fff;}
```

We are going to change this to mimic the behaviour of the borders on the window components in the previous section Additional border changes 26

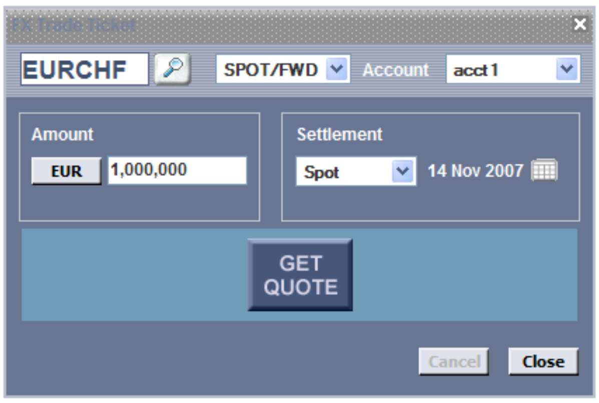
When the dialog is selected, we will set the outer *and* inner border to black. When the dialog is not selected, we will set the outer border to white and the inner border to light gray. Change the code as follows:

```
/* dialog outer border */
.border_dialogouter_active {background-color:#000;}
.border_dialogouter_inactive {background-color:#fff;}

/* dialog inner border */
.border_dialoginner_active {background-color:#000;}
.border_dialoginner_inactive {background-color:#B9BFCD;}
```

The dialog boxes will now look like this:



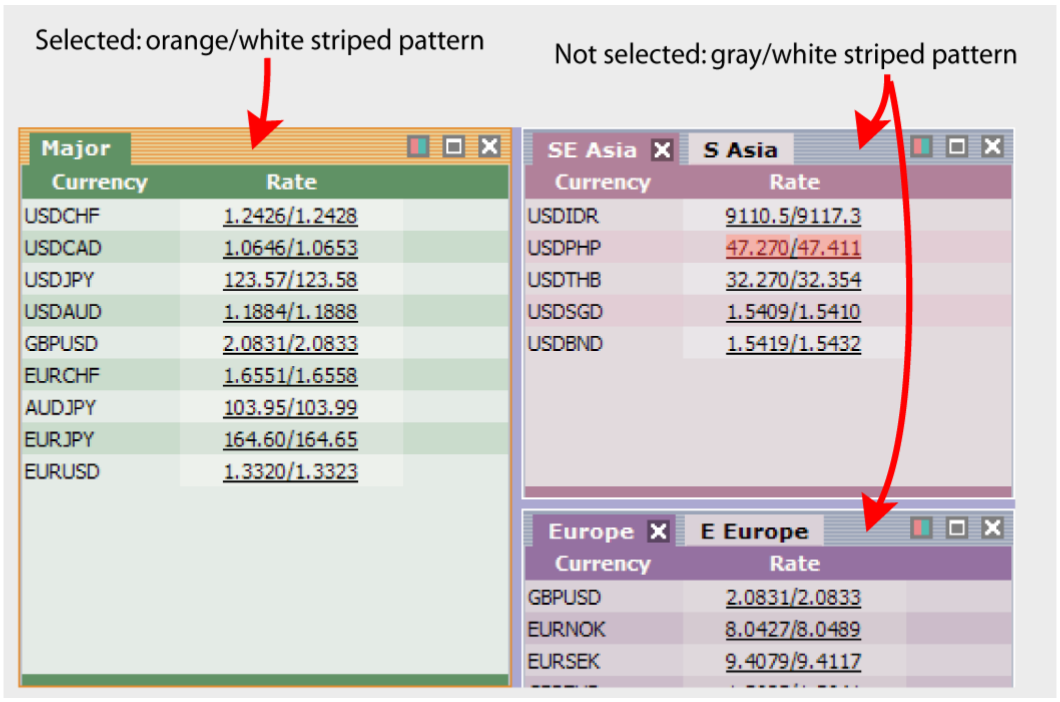
An active (selected) dialog box



An inactive (unselected) dialog box

Changing the background image in the window and dialog title bars

We shall now change the background of the title bars in the window components. In the image below the title bar there is an orange/white striped pattern when the componenrt is selected, and a grey/white stripe where the component is not selected. See the image below:


Selected: orange/white striped pattern
Not selected: gray/white striped pattern
Major
SE Asia
S Asia
Europe
E Europe

Currency	Rate
USDCHF	1.2426/1.2428
USDCAD	1.0646/1.0653
USDJPY	123.57/123.58
USDAUD	1.1884/1.1888
GBPUSD	2.0831/2.0833
EURCHF	1.6551/1.6558
AUDJPY	103.95/103.99
EURJPY	164.60/164.65
EURUSD	1.3320/1.3323

Currency	Rate
USDIDR	9110.5/9117.3
USDPHP	47.270/47.411
USDTHB	32.270/32.354
USDSGD	1.5409/1.5410
USDBND	1.5419/1.5432

Currency	Rate
GBPUSD	2.0831/2.0833
EURNOK	8.0427/8.0489
EURSEK	9.4079/9.4117

When the title bar in a Panel has only one tab it is called a **Handle**, and when it has more than one tab it is called a **Tabstrip**. With the CSS code below (in the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/themes/caplin-trader/css/webcentric.css)*, we're going to replace the two background image patterns with a light blue pattern for when the Panel is selected and dark blue pattern when it isn't. This code will also use this pattern to change the background used in the dialog title bars (note the .Handle_dialog and .Handle_dialog_active class names). So, first of all find the following declarations for the Handle:

```
.Handle {
      background : url(../images/header_gray_stripe.gif) repeat-x;
      cursor : move;
}
.Handle_active {
      background : url(../images/header_gray_stripe8.gif) repeat-x;
      cursor : move;
}
```

Change these to:

```
.Handle {
      background : url(../images/handleInactive.png) repeat-x;
      cursor : move;
}
.Handle_active {
      background : url(../images/handleActive.png) repeat-x;
      cursor : move;
}
```

Now find the following declarations for the Tabstrip:

```
.Tabstrip {
      background : url(../images/header_gray_stripe.gif) repeat-x;
      cursor : move;
}
.Tabstrip_active {
      background : url(../images/header_gray_stripe8.gif) repeat-x;
      cursor : move;
}
```

Change these to:

```
.Tabstrip {
      background : url(../images/handleInactive.png) repeat-x;
      cursor : move;
}
.Tabstrip_active {
      background : url(../images/handleActive.png) repeat-x;
      cursor : move;
}
```

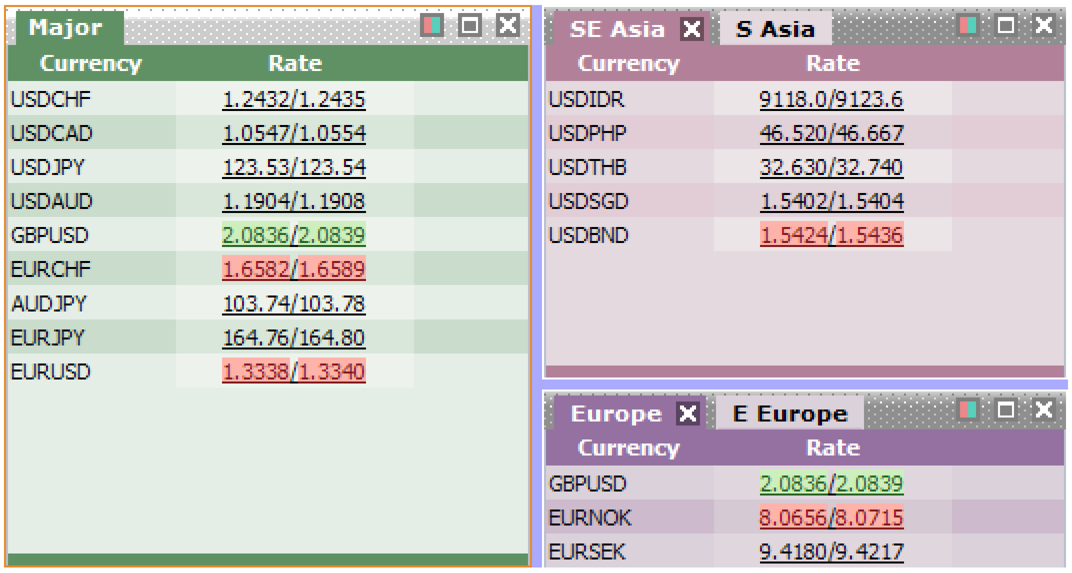
Finally, find the following declarations for the dialogs:

```
.Handle_dialog {
      background          : none;
      background-color    : #D7DAE1;
      cursor : move;
}
.Handle_dialog_active {
      background          : none;
      background-color    : #D7DAE1;
      cursor : move;
}
```

Change these to:

```
.Handle_dialog {
      background            : url(../images/handleInactive.png) repeat-x;
      background-color      : #D7DAE1;
      cursor : move;
}
.Handle_dialog_active {
      background            : url(../images/handleActive.png) repeat-x;
      background-color      : #D7DAE1;
      cursor : move;
}
```

The title bars should now look like this:



Major

Currency	Rate
USDCHF	1.2432/1.2435
USDCAD	1.0547/1.0554
USDJPY	123.53/123.54
USDAUD	1.1904/1.1908
GBPUSD	2.0836/2.0839
EURCHF	1.6582/1.6589
AUDJPY	103.74/103.78
EURJPY	164.76/164.80
EURUSD	1.3338/1.3340

SE Asia | S Asia

Currency	Rate
USDIDR	9118.0/9123.6
USDPHP	46.520/46.667
USDTHB	32.630/32.740
USDSGD	1.5402/1.5404
USDBND	1.5424/1.5436

Europe | E Europe

Currency	Rate
GBPUSD	2.0836/2.0839
EURNOK	8.0656/8.0715
EURSEK	9.4180/9.4217

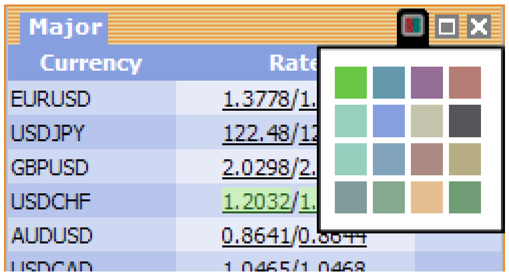
You are welcome to change any of the colors or images in the code discussed so far in this section to get a color scheme of your choice.

3.5 Changing the colors used in the components

The components, such as grids and trade panels, have their own set of colors too. In the sections Additional border color changes 26 and Changing the background image in the window and dialog title bars 28 we discussed changing the colors or graphics associated with a small part of the component windows. However, in this section we look at all the other colors in a component and how they can be changed independently of other components in the application.

Changing the colors interactively

The first, and easiest way to change the color of a component is to click on the color chooser gadget on the top left-most button on the top right-hand side of any component. The color chooser gadget presents you with number of colors, each of which is representative of a color scheme.



Currency	Rate
EURUSD	1.3778/1.
USDJPY	122.48/12
GBPUSD	2.0298/2.
USDCHF	1.2032/1.
AUDUSD	0.8641/0.8644
USDCAD	1.0465/1.0468

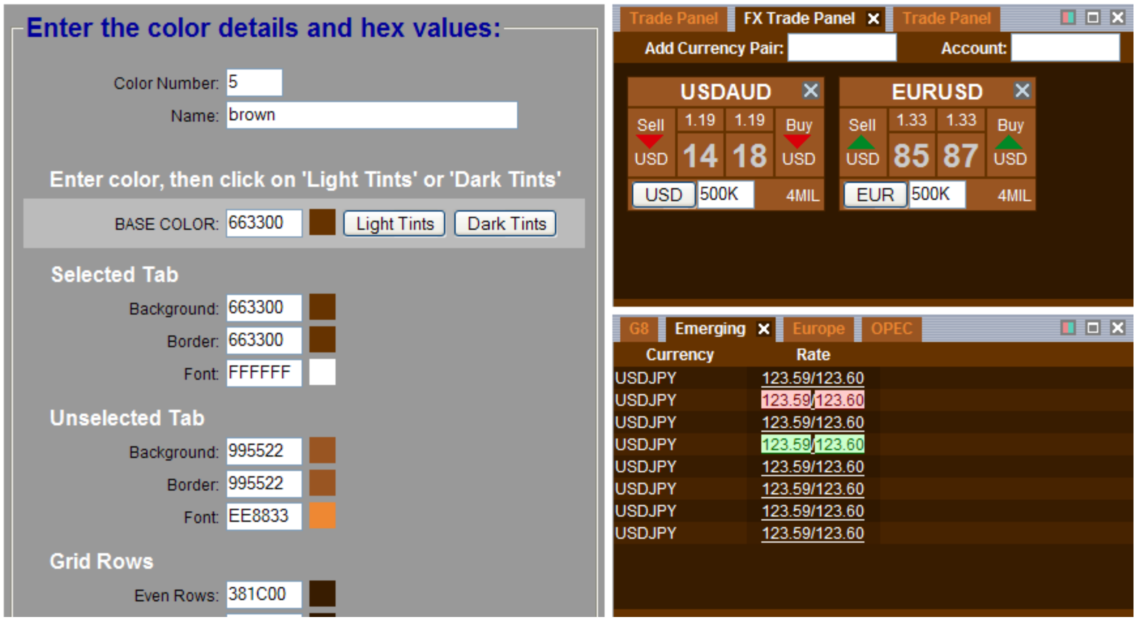
Color chooser

Moving the mouse over any of these colors will dynamically update the color scheme used in the component. Simply click once you're happy with particular scheme and the color chooser will go away, leaving you with your new scheme for that component.

Changing the colors used in a scheme

There are two ways of changing the 16 defined color schemes in the Caplin Trader Client:

1. Manually editing the colors for one or more schemes in the CSS file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/styles/colors.css*. If you wish to do this, see the following section Changing the colors manually 37.
2. Using our Color Swatch Generator, which allows you to define and preview color schemes of your own in an interactive fashion. To do this, please see the section Changing the colors using the Color Scheme Generator tool 33.


Enter the color details and hex values:

Color Number:	5
Name:	brown

Enter color, then click on 'Light Tints' or 'Dark Tints'

BASE COLOR:	663300	🟫

Selected Tab

Background:	663300	🟫
Border:	663300	🟫
Font:	FFFFFF	⬜

Unselected Tab

Background:	995522	🟫
Border:	995522	🟫
Font:	EE8833	🟧

Grid Rows

Even Rows:	381C00	🟫

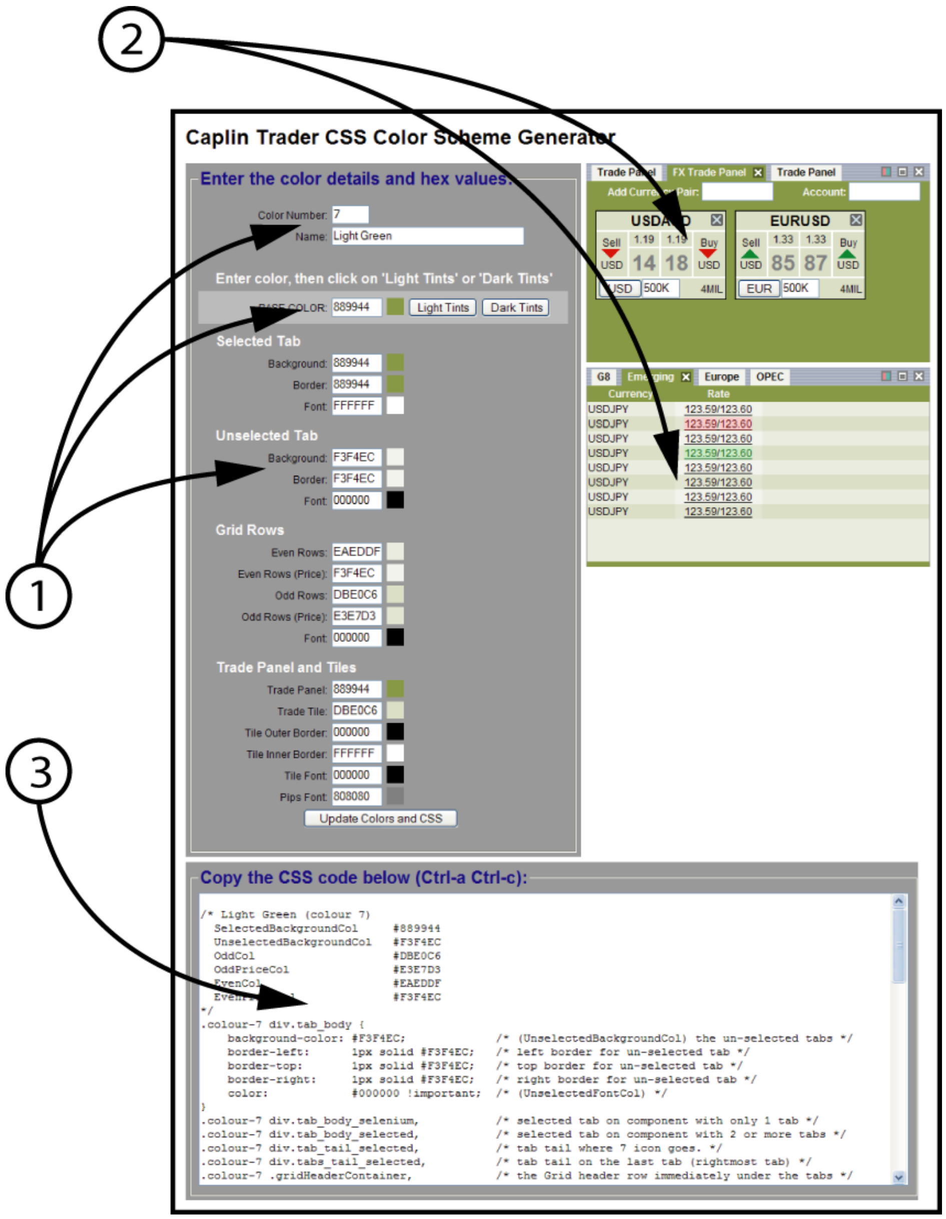
Currency	Rate
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60

The Caplin Trader CSS Color Swatch Generator allows the user to interactively design and preview color schemes for use in the Caplin Trader Client

Changing the colors using the Color Scheme Generator tool

The Caplin Trader Color Scheme Generator is a simple stand-alone browser application that helps you to design a color scheme for Caplin Trader GUI components.It then generates the CSS code for the chosen color scheme and you can apply this code to your Caplin Trader Client, thus implementing the scheme. To use it, navigate to the following file in your file browser and double click to open the tool in the web browser:
$CT_INSTALL_DIR*/apps/webapps/caplintrader/applications/CaplinTrader/source/styles/colorSchemeGenerator/colorSchemeGenerator.html*

The Color Scheme Generator looks like this:



Caplin Trader CSS Color Scheme Generator

Enter the color details and hex values:

Color Number:	7
Name:	Light Green

Enter color, then click on 'Light Tints' or 'Dark Tints'

BASE COLOR: 889944 [Light Tints] [Dark Tints]

Selected Tab		
Background:	889944	🟩
Border:	889944	🟩
Font:	FFFFFF	⬜
Unselected Tab		
Background:	F3F4EC	⬜
Border:	F3F4EC	⬜
Font:	000000	⬛
Grid Rows		
Even Rows:	EAEDDF	⬜
Even Rows (Price):	F3F4EC	⬜
Odd Rows:	DBE0C6	⬜
Odd Rows (Price):	E3E7D3	⬜
Font:	000000	⬛
Trade Panel and Tiles		
Trade Panel:	889944	🟩
Trade Tile:	DBE0C6	⬜
Tile Outer Border:	000000	⬛
Tile Inner Border:	FFFFFF	⬜
Tile Font:	000000	⬛
Pips Font:	808080	⬛

[Update Colors and CSS]

Copy the CSS code below (Ctrl-a Ctrl-c):

```
/* Light Green (colour 7)
  SelectedBackgroundCol      #889944
  UnselectedBackgroundCol    #F3F4EC
  OddCol                     #DBE0C6
  OddPriceCol                #E3E7D3
  EvenCol                    #EAEDDF
  Even[illegible]            #F3F4EC
*/
.colour-7 div.tab_body {
    background-color: #F3F4EC;           /* (UnselectedBackgroundCol) the un-selected tabs */
    border-left:      1px solid #F3F4EC;  /* left border for un-selected tab */
    border-top:       1px solid #F3F4EC;  /* top border for un-selected tab */
    border-right:     1px solid #F3F4EC;  /* right border for un-selected tab */
    color:            #000000 !important; /* (UnselectedFontCol) */
}
.colour-7 div.tab_body_selenium,          /* selected tab on component with only 1 tab */
.colour-7 div.tab_body_selected,          /* selected tab on component with 2 or more tabs */
.colour-7 div.tab_tail_selected,          /* tab tail where 7 icon goes. */
.colour-7 div.tabs_tail_selected,         /* tab tail on the last tab (rightmost tab) */
.colour-7 .gridHeaderContainer,           /* the Grid header row immediately under the tabs */
```

The Color Scheme Generator tool

In the figure above, the three main areas are:

1. The edit panel, containing color number and name, base color and individual color fields. This is where you enter a base color from which a preliminary light or dark tinted scheme is generated.
2. The preview Trade Panel and Grid components. This is where you see how the scheme will look in the Caplin Trader Client.
3. The generated CSS code. You will copy the code in this text area and paste it into the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/styles/colours.css*, taking care to overwrite the previous CSS code for the particular color scheme number in question.

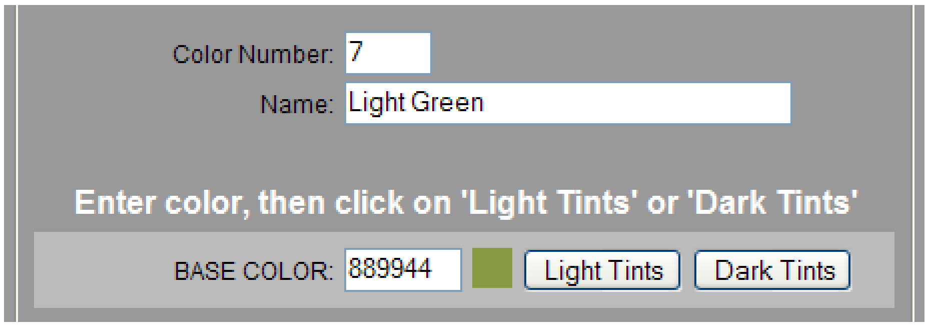
Using the Color Scheme Generator

First of all you need to enter a color number. Normally this will be somewhere in the range 1 to 16, since the Caplin Trader Client comes pre-configured with 16 color schemes. However, there is no reason why you cannot define more than 16 color schemes.

You should then associate a name with this color scheme; this is not mandatory, but the name you choose will be added to a comment in the generated CSS code, making it easier to refer to the scheme later on.

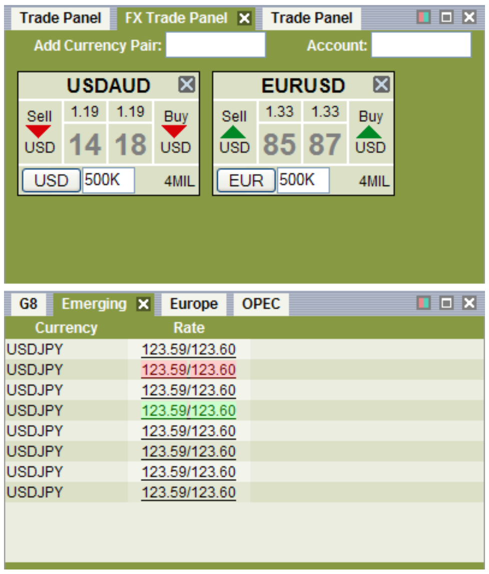
The next stage is to choose the base color for your new scheme. Type in a hexadecimal color either in 6 digit form or in 3 digit *shorthand* form. The 6 digit form comprises three pairs of digits, each pair comprising a value between 0 and 255 in hexadecimal. The first pair represents the amount of red in the color, the second pair the amount of green and the last pair represents the amount of blue.

With a 3 digit form, the amounts of red, green and blue are each defined by one hexadecimal digit only. The Color Scheme Generator automatically converts a three digit value to a 6 digit one by doubling up the digits (so 3A9, for example, is an equivalent color to 33AA99).


Color Number: 7
Name: Light Green
Enter color, then click on 'Light Tints' or 'Dark Tints'
BASE COLOR: 889944

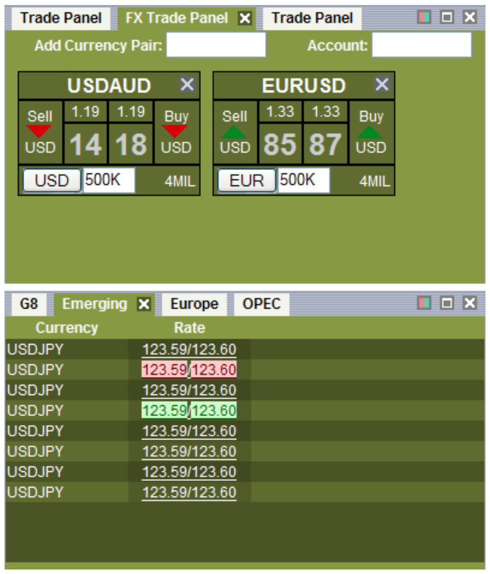
The primary details that must be entered for any color scheme

Once you have entered these details, click on either the "Light Tints" button or the "Dark Tints" button. The Color Scheme Generator will then create a scheme based on your chosen base color and tint preference. See the previews in the figure below.



Currency	Rate
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60

A light tinted scheme from the base color 889944

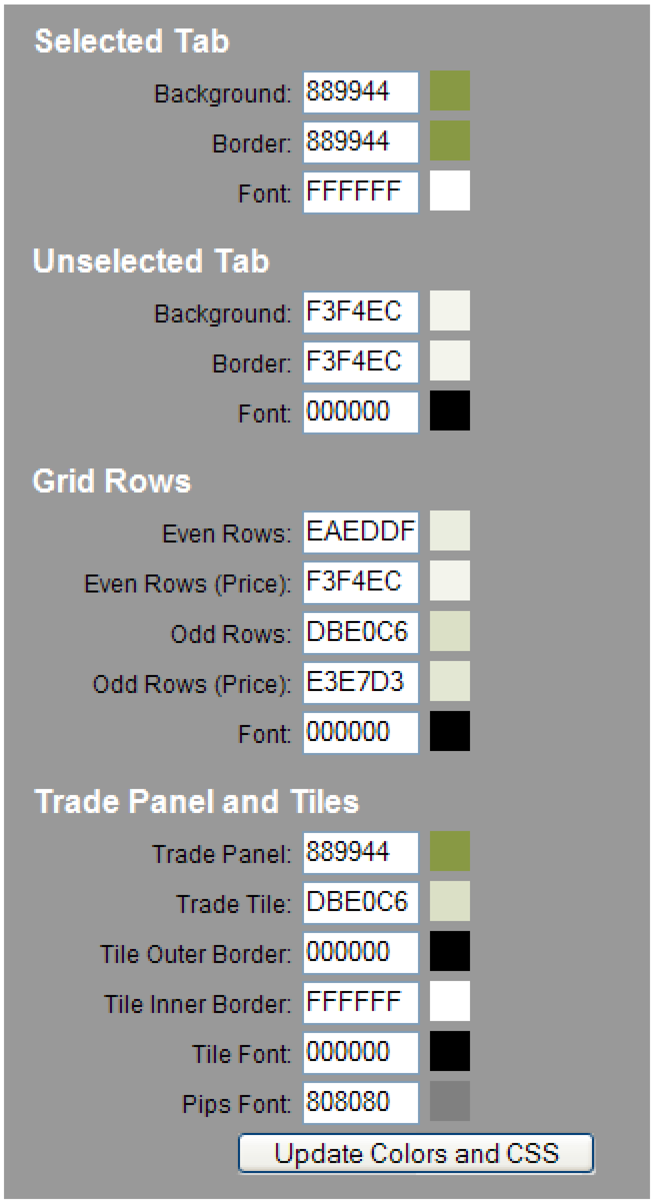


Currency	Rate
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60
USDJPY	123.59/123.60

A dark tinted scheme from the base color 889944

Once you are happy with the general look and feel of the scheme as shown by the preview components, you can then change specific details if you so wish.

Further down on the edit panel are the individual color settings for specific parts of the grid and trade panels.



Selected Tab

Background:	889944	🟩
Border:	889944	🟩
Font:	FFFFFF	⬜

Unselected Tab

Background:	F3F4EC	⬜
Border:	F3F4EC	⬜
Font:	000000	⬛

Grid Rows

Even Rows:	EAEDDF	⬜
Even Rows (Price):	F3F4EC	⬜
Odd Rows:	DBE0C6	⬜
Odd Rows (Price):	E3E7D3	⬜
Font:	000000	⬛

Trade Panel and Tiles

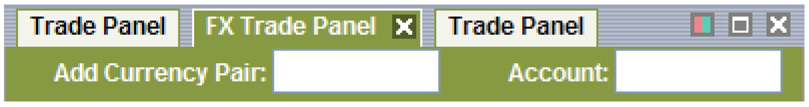
Trade Panel:	889944	🟩
Trade Tile:	DBE0C6	⬜
Tile Outer Border:	000000	⬛
Tile Inner Border:	FFFFFF	⬜
Tile Font:	000000	⬛
Pips Font:	808080	⬛

Color settings for specific parts of the grid and trade panels

To edit these values:

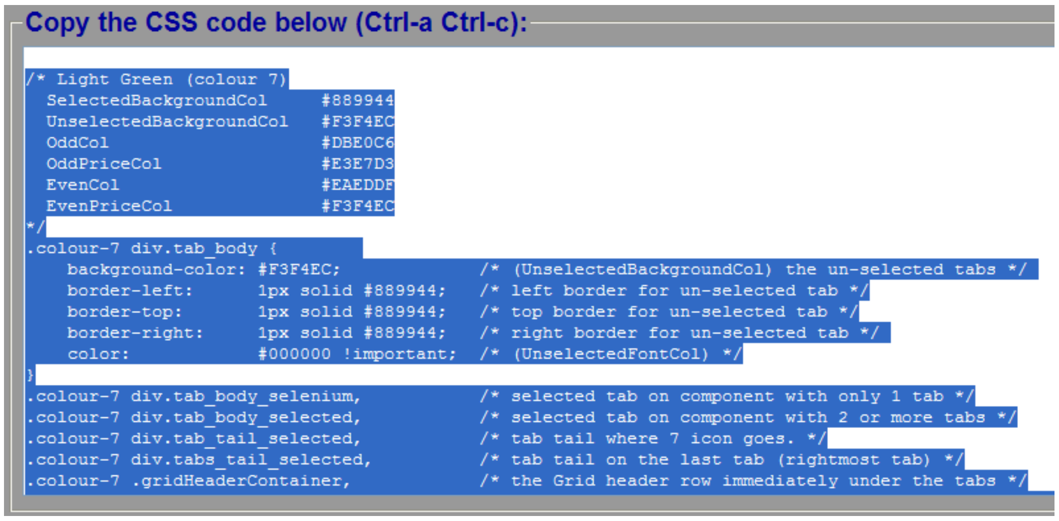
1. Enter the new color value in the field (you can also copy and paste a color value from another field, or even paste in a hex color value from different graphics application or picture editing program).
2. To set the effect of the new color, press the enter key, tab out of the field or click on the Update Colors and CSS button to activate your changes.

Using our example above, if you change the color value for the selected tab border to white (FFFFFF) then you will see a white border on the left, top and right hand sides of the selected tabs on the preview grid and trade panels. Now copy the color from the selected tab background value (889944) into the unselected tab border value, and you will see a green border appear around the unselected tabs. This is shown below:



Changing color values specific to the tab borders

Once you have achieved your desired final color scheme, you copy the generated CSS code and paste it into the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/styles/colors.css*, taking care to replace the color of the same number in the file.



Copy the CSS code below (Ctrl-a Ctrl-c):

```
/* Light Green (colour 7)
  SelectedBackgroundCol      #889944
  UnselectedBackgroundCol    #F3F4EC
  OddCol                     #DBE0C6
  OddPriceCol                #E3E7D3
  EvenCol                    #EAEDDF
  EvenPriceCol               #F3F4EC
*/
.colour-7 div.tab_body {
    background-color: #F3F4EC;             /* (UnselectedBackgroundCol) the un-selected tabs */
    border-left:      1px solid #889944;   /* left border for un-selected tab */
    border-top:       1px solid #889944;   /* top border for un-selected tab */
    border-right:     1px solid #889944;   /* right border for un-selected tab */
    color:            #000000 !important;  /* (UnselectedFontCol) */
}
.colour-7 div.tab_body_selenium,           /* selected tab on component with only 1 tab */
.colour-7 div.tab_body_selected,           /* selected tab on component with 2 or more tabs */
.colour-7 div.tab_tail_selected,           /* tab tail where 7 icon goes. */
.colour-7 div.tabs_tail_selected,          /* tab tail on the last tab (rightmost tab) */
.colour-7 .gridHeaderContainer,            /* the Grid header row immediately under the tabs */
```

Copy and paste the CSS code into the colours.css file

Changing the colors manually

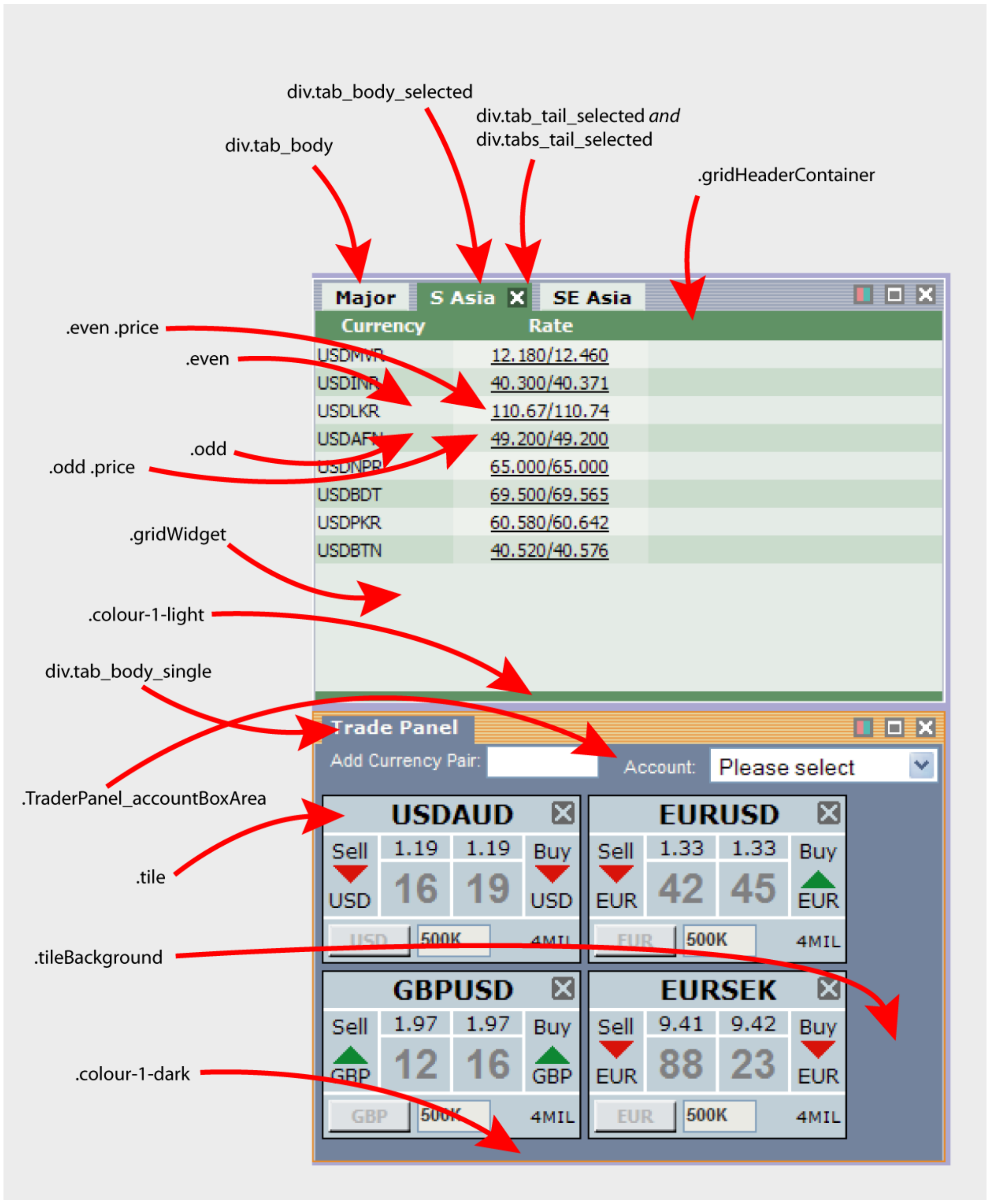
If you want to change the individual colors in a particular scheme manually, you need to edit ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/styles/colors.css*. In this file you will find a number of sets of CSS definitions, each of which defines a color scheme.

Let's look at color-1:

```
/* Grey Blue (colour 1)
      Base   #72829C
      Extra  #39414E
      Odd    #C1D0DC
      OddHi  #CDD7E0
      Even   #D7DFE5
      EvenHi #DDE2E6
*/

.colour-1 div.tab_body {
      background-color:#D7DFE5;
      border-left:       1px solid #D7DFE5;
      border-top:        1px solid #D7DFE5;
      border-right:      1px solid #D7DFE5;
      border-bottom:     1px solid #C1D0DC;
}
.colour-1 div.tab_body_single,
.colour-1 div.tab_body_selected,
.colour-1 div.tab_tail_selected,
.colour-1 div.tabs_tail_selected,
.colour-1 .gridHeaderContainer,
.colour-1 .TraderPanel_accountBoxArea {
      background-color:#72829C !important;
}
.colour-1 div.tab_body_single {
      border-left:         1px solid #72829C !important;
      border-top:          1px solid #72829C !important;
      border-right:        1px solid #72829C !important;
}

.colour-1 div.tab_body_selected {
      border-left:         1px solid #72829C;
      border-top:          1px solid #72829C;
}
.colour-1 div.tab_tail_selected,
.colour-1 div.tabs_tail_selected {
      border-top:          1px solid #72829C;
      border-right:        1px solid #72829C;
}
.colour-1                  {background-color:#72829C;}

.colour-1 .odd             {background-color:#C1D0DC;}
.colour-1 .odd .price      {background-color:#CDD7E0;}
.colour-1 .gridWidget,
.colour-1 .even {
      background-color: #D7DFE5;
}
.colour-1 .even .price     {background-color:#DDE2E6;}
.colour-1-light            {background-color:#72829C;}

.colour-1 .tileBackground  {background-color:#72829C;}
.colour-1 .tile            {background-color:#C1D0DC;}
.colour-1-dark             {background-color:#72829C;}
```

Commented out at the top of this code is a reminder of the five color values used in the 'color-1' scheme. The actual CSS definitions using those color values follow underneath them.

The CSS selectors used in these definitions are detailed in the figure below (note that for most of the selectors the color number prefix has been omitted):



Note that there are quite a few different CSS selectors for tabs:

1. A single tab by itself: `div.tab_body_single`
2. An unselected tab on a component with multiple tabs: `div.tab_body`.
3. The left-hand side of a selected tab on a component with multiple tabs: `div.tab_body_selected`.
4. The right hand side of a selected tab (containing the 'x' close icon): `div.tab_tail_selected` and `div.tabs_tail_selected`.

Changing the color of the tabs

Here we describe how to change the colors for both selected and unselected tabs for the color scheme `colour-1`

Changing the color of tabs that have been selected

In ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/styles/colors.css* find the colour-1 scheme:

```
/* Grey Blue (colour 1)
...
```

You should find this near the top of the file. Immediately under that line you will find a number of lines of CSS code prefixed with the `colour-1` CSS selector.

After the first few lines you will find the following declarations:

```
.colour-1 div.tab_body_single,
.colour-1 div.tab_body_selected,
.colour-1 div.tab_tail_selected,
.colour-1 div.tabs_tail_selected,
.colour-1 .gridHeaderContainer,
.colour-1 .TraderPanel_accountBoxArea {
      background-color:#72829C !important;
}
.colour-1 div.tab_body_single {
      border-left:         1px solid #72829C !important;
      border-top:          1px solid #72829C !important;
      border-right:        1px solid #72829C !important;
}
.colour-1 div.tab_body_selected {
      border-left:         1px solid #72829C;
      border-top:          1px solid #72829C;
}
.colour-1 div.tab_tail_selected,
.colour-1 div.tabs_tail_selected {
      border-top:          1px solid #72829C;
      border-right:        1px solid #72829C;
}
```

The first declaration, `background-color`, defines the background color of the selected tab. The other declarations define the width, style and color of the borders around the selected tab.

Change the color values in these declarations to `#e60`:

```
.colour-1 div.tab_body_single,
.colour-1 div.tab_body_selected,
.colour-1 div.tab_tail_selected,
.colour-1 div.tabs_tail_selected,
.colour-1 .gridHeaderContainer,
.colour-1 .TraderPanel_accountBoxArea {
      background-color:#e60 !important;
}
.colour-1 div.tab_body_single {
      border-left:         1px solid #e60 !important;
      border-top:          1px solid #e60 !important;
      border-right:        1px solid #e60 !important;
}
.colour-1 div.tab_body_selected {
      border-left:         1px solid #e60;
      border-top:          1px solid #e60;
}
.colour-1 div.tab_tail_selected,
.colour-1 div.tabs_tail_selected {
      border-top:          1px solid #e60;
      border-right:        1px solid #e60;
}
```

This CSS code sets the color of the selected tabs (`.tab_body_selected` is the CSS class name of the selected tabs) to the *shorthand* hex triplet color: `#e60`. This is a bright orange. You can play with putting different colors into the `background-color` property to get different colors.

Note: CSS 'shorthand' colors are an abbreviation of the usual six digit hex triplet colors. A shorthand hex triplet can be converted to a normal form by doubling up the digits. For example `#e60` is equivalent to `#ee6600`

Clear your cache and refresh your browser to see the changes that have been made to the color of the selected tabs and the bar underneath them. You will have to use the color chooser to select the first color (`colour-1`) on a component to see the effects of this change.

Changing the colors of unselected tabs

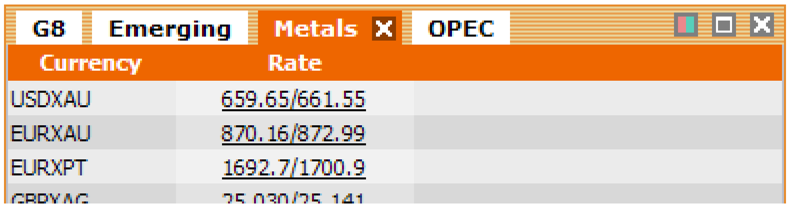
Now we want to change the color of the tabs that are *not* currently selected. To do this, look at the first few lines for color 1:

```
.colour-1 div.tab_body {
      background-color:#D7DFE5;
      border-left:         1px solid #D7DFE5;
      border-top:          1px solid #D7DFE5;
      border-right:        1px solid #D7DFE5;
      border-bottom:       1px solid #C1D0DC;
}
```

This CSS code sets the color of the unselected tabs (`.tab_body` is the CSS class name of the unselected tabs). We are going to set the unselected tabs (and their borders) to white by changing the colors in the code to the following:

```
.colour-1 div.tab_body {
      background-color: white;
      border-left:         1px solid white;
      border-top:          1px solid white;
      border-right:        1px solid white;
      border-bottom:       1px solid white;
}
```

Although in this case we have used the string `white` for the color white, you could just as easily have used the shorthand hex triplet `#fff`.
Your tabs should now look like this:



Currency	Rate
USDXAU	659.65/661.55
EURXAU	870.16/872.99
EURXPT	1692.7/1700.9
GBPXAG	25.030/25.141

Changing the border color of the tabs

If you want the tabs to stand out more you can add a border to them.

Let's have a look at the css declarations for the selected tab for `color-1` in the file ***$CT_INSTALL_DIR/*** *apps/webapps/caplintrader/applications/CaplinTrader/source/styles/colors.css* (as modified in the section Changing the color of the tabs 40):

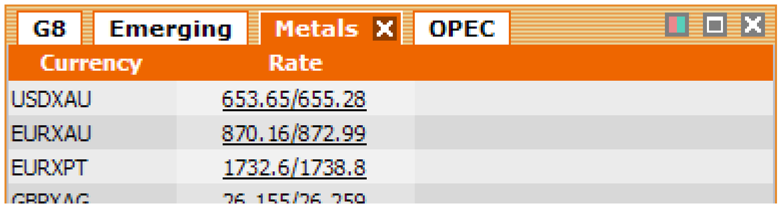
```
.colour-1 div.tab_body_single {
      border-left:         1px solid #e60 !important;
      border-top:          1px solid #e60 !important;
      border-right:        1px solid #e60 !important;
}
.colour-1 div.tab_body_selected {
      border-left:         1px solid #e60;
      border-top:          1px solid #e60;
}
.colour-1 div.tab_tail_selected,
.colour-1 div.tabs_tail_selected {
      border-top:          1px solid #e60;
      border-right:        1px solid #e60;
}
```

You can see here that the border declarations are set to the same color as the background, i.e. `#e60`. Let's change this so that the border is `white`:

```
.colour-1 div.tab_body_single {
      border-left:         1px solid white !important;
      border-top:          1px solid white !important;
      border-right:        1px solid white !important;
}
.colour-1 div.tab_body_selected {
      border-left:         1px solid white;
      border-top:          1px solid white;
}
.colour-1 div.tab_tail_selected,
.colour-1 div.tabs_tail_selected {
      border-top:          1px solid white;
      border-right:        1px solid white;
}
```

Now let's do the same for the unselected tabs. As the tabs are already white, we'll set the borders to be a contrasting bright orange (`#e60`): We need to change the first few lines for color 1 again

```
.colour-1 div.tab_body {
      background-color: white;
      border-left:         1px solid #e60;
      border-top:          1px solid #e60;
      border-right:        1px solid #e60;
      border-bottom:       1px solid white;
}
```



Currency	Rate
USDXAU	653.65/655.28
EURXAU	870.16/872.99
EURXPT	1732.6/1738.8

Component tabs with white borders on the selected tabs and and orange borders on the unselected tabs

Changing grid component colors

Now let's add in some orange shades into our grid panels. We want to keep the distinction between odd and even rows, so we'll use two different very light orange colors for each. Rows are numbered starting from zero, so the first row is even, followed by odd, and so on. To color the even rows, find the following declarations and change the colors to be the following:

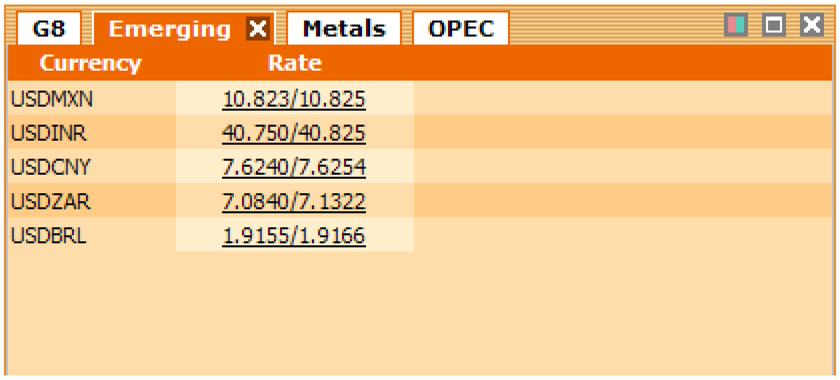
```
.colour-1 .gridWidget,
.colour-1 .even {
      background-color: #fda;
}
.colour-1 .even .price       {background-color:#fec;}
```

Note that we have also colored the background of the grid itself (`.gridWidget`) in the same color as the even rows. If you do not want tot do this you can split this selector out and give it a different color.

The odd rows are colored with the following:

```
.colour-1 .odd               {background-color:#fc8;}
.colour-1 .odd .price        {background-color:#fda;}
```

With these changes your grid rows should look like this:



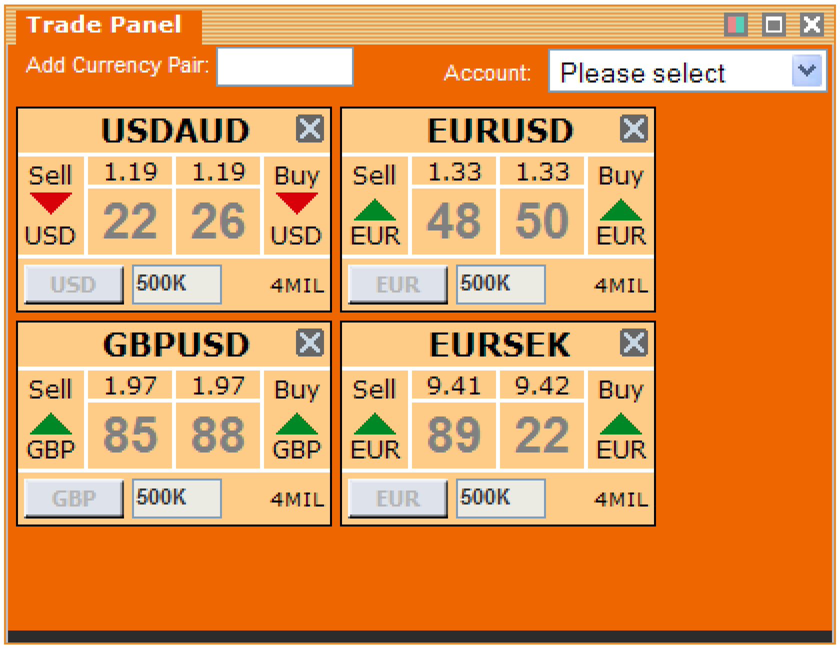
Currency	Rate
USDMXN	10.823/10.825
USDINR	40.750/40.825
USDCNY	7.6240/7.6254
USDZAR	7.0840/7.1322
USDBRL	1.9155/1.9166

Changing the trade panel colors

The Trade Panels use a different CSS class to define their colors, so change the following lines as suggested to set the Trader Panel background color to a light orange:

```
.colour-1 .tileBackground    {background-color:#e60;}
.colour-1 .tile              {background-color:#fc8;}
```

The Trade Panel should now sport the following color scheme:



Changing the loading panel color

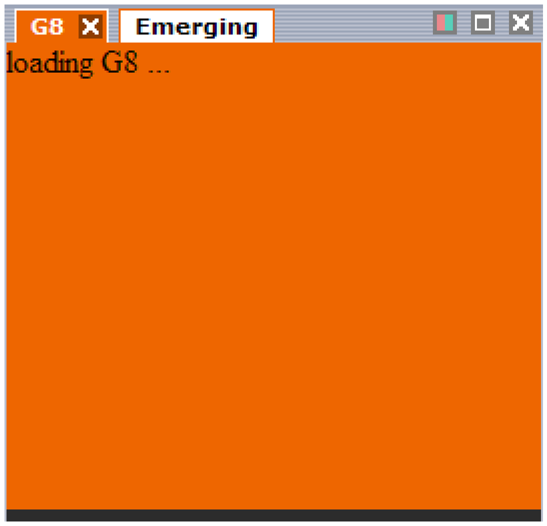
You may notice, at application startup, that there may be a delay before data has been loaded into the grid and trader panels. The color of the panels at this time can also be configured. To do this, find the line that reads:

```
.colour-1                    {background-color:#72829C;}
```

Change this to read:

```
.colour-1                    {background-color:#e60;}
```

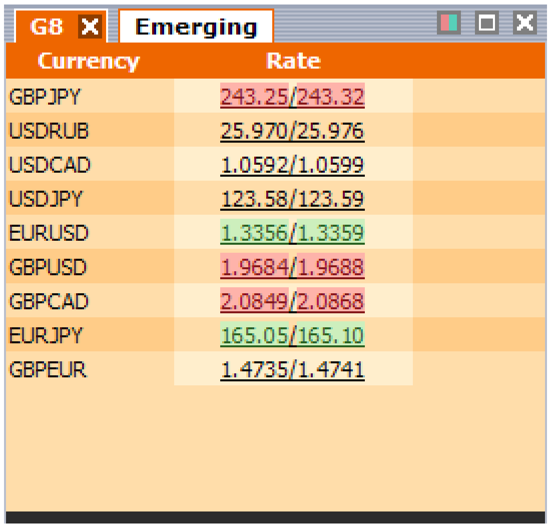
Loading panels that have been assigned `color-1` will now look like this:



loading G8 ...

Changing bottom border colors

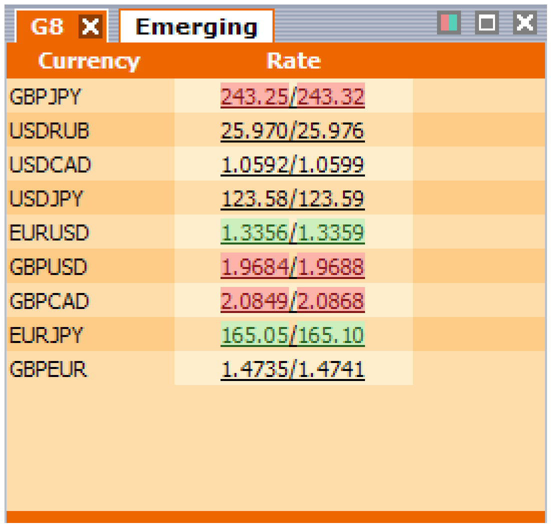
Notice that there is a border at the bottom of all the Panels which does not match the new orange scheme. In the following picture, the bottom border is dark gray, but it is different on each Panel as the color is set by the color chooser.



Currency	Rate
GBPJPY	243.25/243.32
USDRUB	25.970/25.976
USDCAD	1.0592/1.0599
USDJPY	123.58/123.59
EURUSD	1.3356/1.3359
GBPUSD	1.9684/1.9688
GBPCAD	2.0849/2.0868
EURJPY	165.05/165.10
GBPEUR	1.4735/1.4741

To change the bottom border color of the panels change the following CSS code to set it to the bright orange color:

```
.colour-1-dark       {background-color:#e60;}
```



Currency	Rate
GBPJPY	243.25/243.32
USDRUB	25.970/25.976
USDCAD	1.0592/1.0599
USDJPY	123.58/123.59
EURUSD	1.3356/1.3359
GBPUSD	1.9684/1.9688
GBPCAD	2.0849/2.0868
EURJPY	165.05/165.10
GBPEUR	1.4735/1.4741

To do the same for the Trade Panels, change the following line:

```
.colour-1-light        {background-color:#e60;}
```

Changing the number of color schemes available

If you have simply replaced one color scheme with another then you do not need to make the following changes. However, if you have added to the number of schemes, for example by creating a color scheme called `color-17` then you will need to do the following:

Open the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/build/xml/application.xml*.

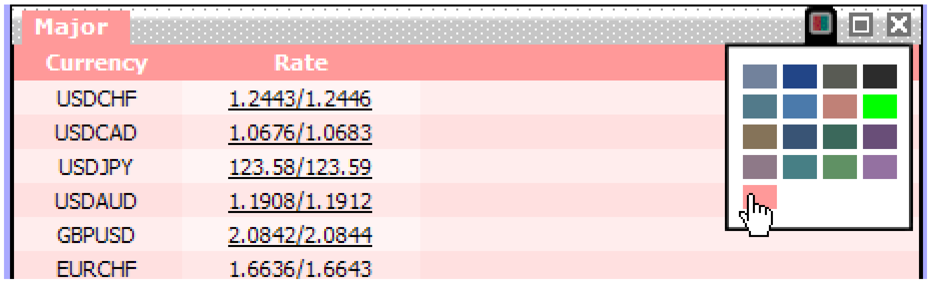
Scroll down and you will find a line that reads:

```
<property setter="setNumColours" value="16"/>
```

The number in the `value` attribute needs to be changed to reflect the actual number of color schemes available. If, for example, you have added one more color then you should set this value to 17.

```
<property setter="setNumColours" value="17"/>
```

Note: To execute the above XML change, you need to do the following:
Open a terminal shell (if you do not have one open already).
Type `cd $CT_INSTALL_DIR` followed by return.
Type `java -jar kits/CaplinTrader/webcentric_database_populator.jar apps/webapps/caplintrader/applications/CaplinTrader/build/xml`, followed by return.
This process will rebuild the user preferences database with the changes that you have just made.
Then clear the cache in your browser and refresh the page. You will have to login again, but after that click on the color chooser button and you will see the new color swatch in the top left position.



Currency	Rate
USDCHF	1.2443/1.2446
USDCAD	1.0676/1.0683
USDJPY	123.58/123.59
USDAUD	1.1908/1.1912
GBPUSD	2.0842/2.0844
EURCHF	1.6636/1.6643

Setting the same color scheme for all components

If you want to fix a permanent set of color schemes (i.e. one that cannot be changed via component's color choosers), you need to edit the chosen starting color schemes for the components in the main layout, and then remove the color chooser gadget. This will fix the color scheme and users will be unable to change it interactively.

To change the starting color schemes for the Caplin Trader Client GUI components you need to edit the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/build/xml/layouts/Default_FX_Layout.xml*. Open it and search for "`colour-`".

Each occurrence of this string assigns a color scheme to a particular window component, e.g. `colour-12` or `colour-4`. The simplest pattern is where all the window components have the same scheme and if you want to do this, you should set each color value to be `colour-1` (the reason for this is that dynamically created grids and trade panels start with the color set `colour-1`, see section Changing the chosen color scheme of dynamically created components 50).

If you save this file and refresh your browser, then you will find that all of the window components share the same color scheme, that of `colour-1` as defined in ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/styles/colors.css*

Note: To execute the above XML change, you need to do the following:
Open a terminal shell (if you do not have one open already).
Type `cd $CT_INSTALL_DIR` followed by return.
Type `java -jar kits/CaplinTrader/webcentric_database_populator.jar apps/webapps/caplintrader/applications/CaplinTrader/build/xml`, followed by return.
This process will rebuild the user preferences database with the changes that you have just made.
Then clear the cache in your browser and refresh the page. You will have to login again, but after that click on the color chooser button and you will see the new colors.

If you want to prevent the users from changing the color using the color chooser gadget, you need to remove the color chooser button. See the next section, Removing the color chooser buttons 49, for details on how to do this.

Removing the color chooser buttons

To stop users from attempting to use the color chooser, we should now remove the button that activates it from the component title bars. To do this you need to open up the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/build/xml/theme.xml*.

Search through the file for the text: `<Button button_id="colours"`

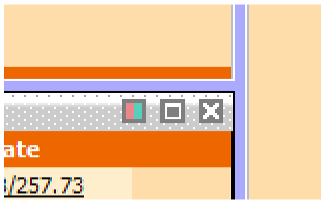
You will find this appears in two places; the first occurrence defines the button action when it is part of a title bar with only a single tab, and the second defines the button action when it is part of a title bar with two or more tabs.

You need to remove both of these declarations, so first delete the following text:

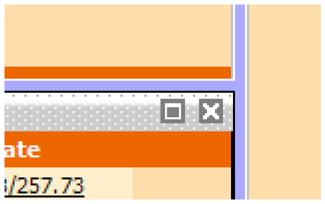
```
<Button width="16" height="19" img="%theme%/images/handle/buttons/colours.png"
      tooltip="Select color scheme for panel">
      <showPopup ref="." offset_left="-42"
             xref="Declarations/Popup[@id='Color_Selector']" defer="true"/>
</Button>
```

...and then delete the following text:

```
<Button button_id="colours" width="16" height="19"
      img="%theme%/images/handle/buttons/colours.png"
      tooltip="Select color scheme for panel">
      <showPopup ref="." offset_left="-42"
             xref="Declarations/Popup[@id='Color_Selector']" defer="true"/>
</Button>
```



Title bar before the color chooser has been removed



Title bar after the color chooser has been removed

Note: To execute the above XML change, you need to do the following:
Open a terminal shell (if you do not have one open already).
Type `cd $CT_INSTALL_DIR` followed by return.
Type `java -jar kits/CaplinTrader/webcentric_database_populator.jar apps/webapps/caplintrader/applications/CaplinTrader/build/xml`, followed by return.
This process will rebuild the user preferences database with the changes that you have just made.
Then clear the cache in your browser and refresh the page. You will have to login again, but after that you will see the color chooser buttons are gone.

Changing the chosen color scheme of dynamically created components

From the Caplin Trader Client menu (**Insert > Product Grid**) you are able to add a new grid component window to the layout. Once this component has been dragged and dropped into a particular location, it will have a color scheme assigned to it. The particular color scheme assigned depends upon the default scheme defined in the relevant file for that type of component.

The three component types are:

1. Grid:

   The default color scheme for a grid is defined in the file: ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/xml/components/container_grid_template.jsp*. Change `color-1` to the color of your choice.

2. Trade Panel:

   The default color scheme for a Trade Panel is defined in the file: ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/xml/components/fx_trade_panel.jsp*. Change `color-1` to the color of your choice.

3. Blotter:

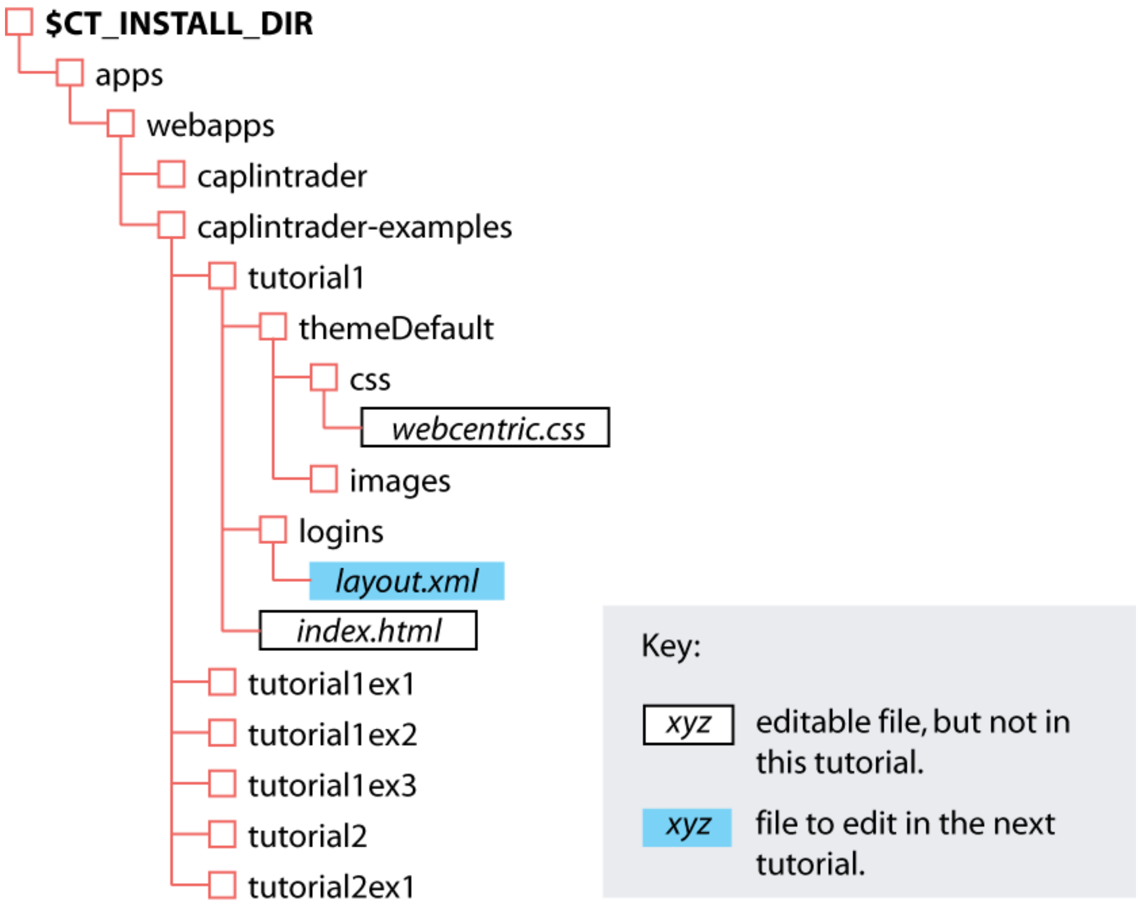
   The default color scheme for a blotter is defined in the file: ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/source/xml/components/fx_blotter.jsp*. Change `color-10` to the color of your choice.

4 Layout Concepts

You can change any of the Caplin Trader Client layouts interactively by dragging and dropping components, closing components or even adding new ones by using the menu options (such as Insert > Product Grid or
Insert > Trade Panel). You can also save these layouts by using the Layout > Save and Layout > Save As... menu options. However, changes to the default "Foreign Exchange" layout can only be saved to a new layout using the Layout > Save As... menu option. That is, you cannot permanently overwrite the default "Foreign Exchange" layout.

To change the default "Foreign Exchange" layout that is displayed when the Caplin Trader Client launches, or even define a layout for your own application, you will need to edit an XML file. A good XML editor with syntax completion will help you do this.

For the following discussion, let's look at where this file is located in your installation, and then we can look at what parts of these files may be customized. In your Caplin Trader installation, you will find the following directory structure. The file that you can modify (*layout.xml*) is shown with a light blue background in the following diagram:



In directory *tutorial1/logins* the XML file called *layout.xml* defines the layout of Caplin Trader Client's user interface. It contains the following XML code:

XML defining Caplin Trader Client application layout

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Application ... >
      <Declarations></Declarations>
      <GUI>
             <FrameItems>
             ... in here we define the layout of the application ...
             </FrameItems>
      </GUI>
</Application>
```

The XML tag `<GUI>` marks the beginning of the application's layout. You can put many XML tags in here, and even nest them when you want to arrange interface components in a specific way.

The GUI component represents the outermost container for the whole interface. Any child components of the GUI must be marked up within `<FrameItems></FrameItems>` tags.

The `<FrameItems></FrameItems>` tags serve to separate the definition of the GUI's child components from other markup that may be defined inside `<GUI></GUI>`.

Before we examine the markup in this file in more detail, let's look at some simple layout arrangements and the associated markup needed within the `<GUI>` tags.

4.1 The Panel

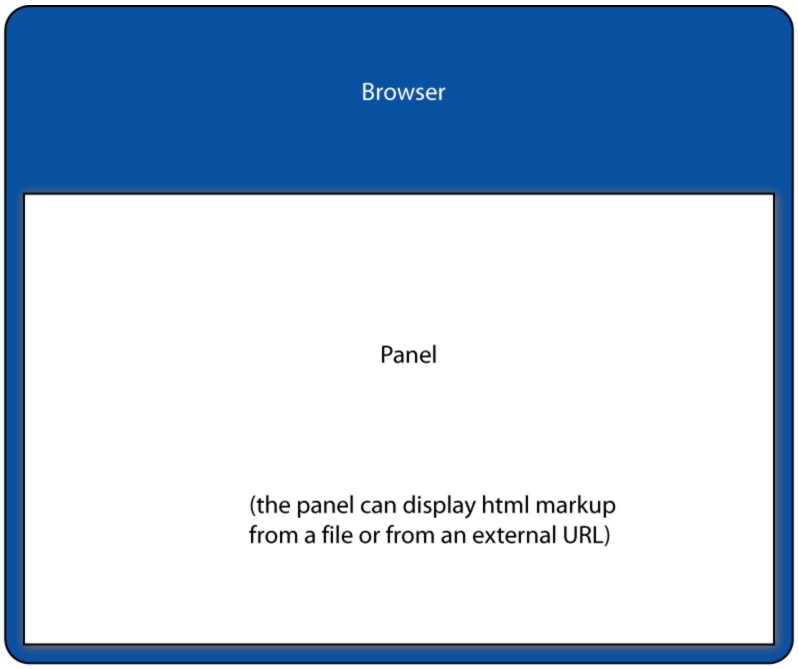
Although a typical Caplin Trader Client application will consist of a number of *nested* window components, the simplest GUI arrangement consists of a single **Panel**, which is a rectangular area into which HTML data can be displayed.

Note: You can use the Panel component whenever you wish to add some simple local HTML content to your layout. However, this component is *not* suitable for more complicated uses, such as:

1. dynamically changing content which makes use of JavaScript;
2. content which requires interaction with the rest of the application, such as responding to life-cycle events on the component itself (for example closing the component, resizing the component, and so on).

If you want to create components that are capable of doing more than the simple display of local HTML content, we recommend you read the document **CaplinTrader: Customizing the Content**.

This simplest possible configuration when displayed in a web browser looks like this:


Browser
Panel
(the panel can display html markup
from a file or from an external URL)

The XML markup that defines this panel is:

```
<GUI>
      <FrameItems>
             <Panel />
      </FrameItems>
</GUI>
```

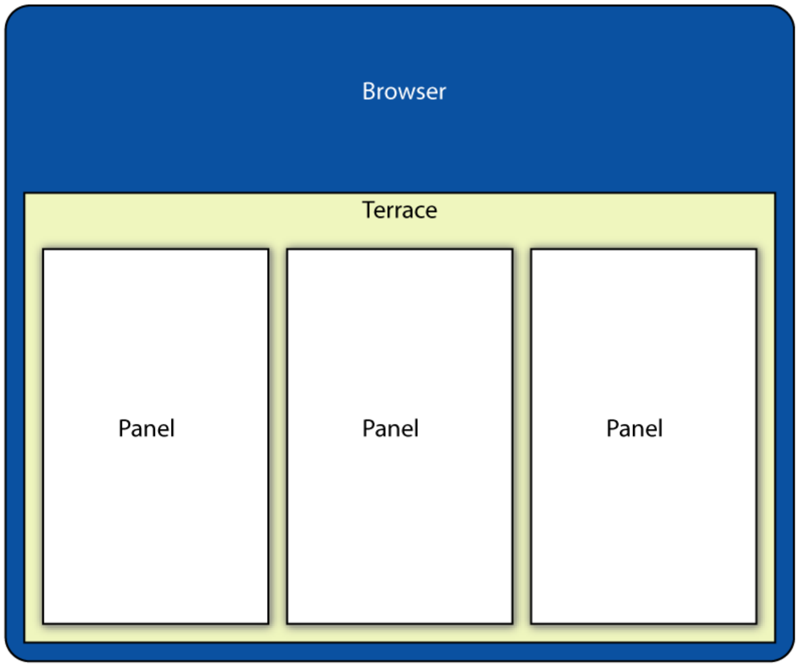
By wrapping Panels in other layout components, you can arrange a group of Panels (or other window elements) in a number of ways:

- Horizontally (a "Terrace")
- Vertically (a "Tower")
- Stacked on top of each other (a "Stack"), with tabs that allow the user to select which window is at the top of the stack

The following sections describe Terraces 55, Towers 56 and Stacks 57 in more detail.

4.2 The Terrace

The Terrace is used to group elements horizontally.



Three panels contained in a terrace

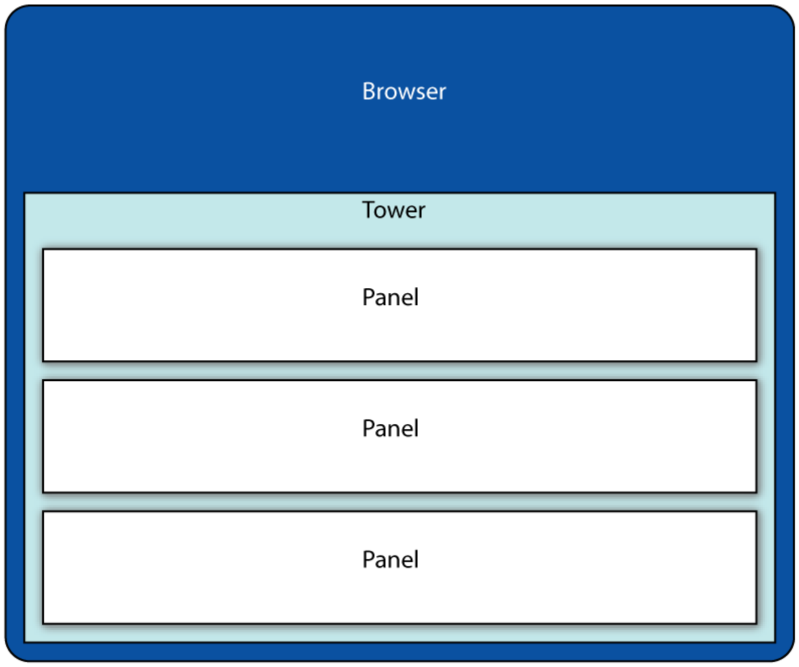
The XML markup that defines this Terrace is:

```
<GUI>
      <FrameItems>
             <Terrace>
                    <FrameItems>
                           <Panel />
                           <Panel />
                           <Panel />
                    </Frameitems>
             </Terrace>
      </Frameitems>
</GUI>
```

The `<Terrace>` tag must use the `<FrameItems>` tag to contain its visible window components. In the markup here, three Panels are contained within a Terrace and will be arranged horizontally. By default, each will be sized to fit one third of the width of the Terrace (note that size of the Panels in the figure above is for illustrative purposes only; in reality there is no spacing between the Panels and the inner edge of the Terrace).

4.3 The Tower

The Tower is used to arrange elements vertically.



Three panels contained in a tower

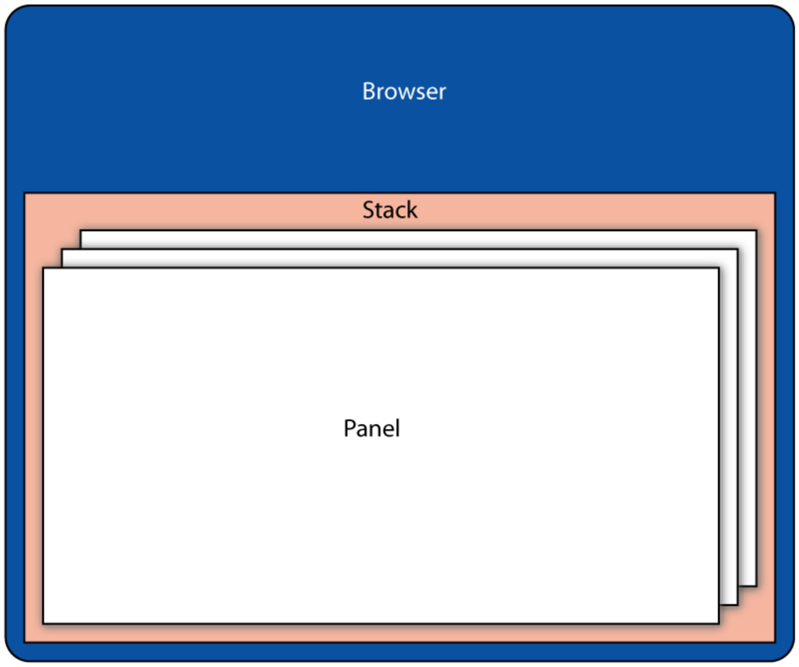
The XML markup that defines this Tower is:

```
<GUI>
      <FrameItems>
             <Tower>
                    <FrameItems>
                           <Panel />
                           <Panel />
                           <Panel />
                    </Frameitems>
             </Tower>
      </Frameitems>
</GUI>
```

The `<Tower>` tag must use the `<FrameItems>` tag to contain its visible window components. By default, the height of each Panel is scaled so that each is one third of the height of the Tower.

4.4 The Stack

The Stack (Panels layered on top of one another)



Three panels contained in a stack

The XML markup that defines this Stack is:

```
<GUI>
      <FrameItems>
             <Stack>
                    <FrameItems>
                           <Panel />
                           <Panel />
                           <Panel />
                    </Frameitems>
             </Stack>
      </Frameitems>
</GUI>
```

The `<Stack>` tag must use the `<FrameItems>` tag to contain its visible window components. Each Panel fills the whole height and width of the containing Stack. The Panels in the figure above are shown slightly smaller and are offset from each other for illustrative purposes only.

The Application Layout Stack

Whilst you don't *need* to use a Tower 56 or Terrace 55 in an application, you must include a special kind of stack in every application; this is called the *Application Layout Stack*.

The purpose of the Application Layout Stack is to serve as the main container for *draggable* window components. Typically windows outside of the Application Layout Stack will not be draggable.

Note: In more advanced setups, the layout of windows within the Application Layout Stack can be saved to a database. This is useful where the user has rearranged the layout of the windows within the Application Layout Stack and would like to preserve the new layout for when they next log in. Discussion of this feature is, however, beyond the scope of this document.

The way to setup an Application Layout Stack is simple, all you do is add the attribute `id="application-layout"` to the Stack tag. Here's the XML markup:

```
<Stack id="application-layout">
</Stack>
```

Typically you would use the `id` attribute when you intend to nest further Towers and Terraces within the Stack and allow the Panels they contain to be dragged and dropped in new arrangements.

4.5 Adding attributes to the Panel

The Panels you have seen so far use a number of default settings. You can change these defaults by adding attributes to the Panel tags.

Changing the Panel color

You can change the color of a Panel by adding the attribute `background="<color-value>"`**,** where `color-value` is a hexadecimal color code, in either the 6 digit form or the 3 digit CSS shorthand form.

A Panel with a red background would be defined as:

```
<Panel background="#f00" />
```

whereas a Panel with a blue background would be defined as:

```
<Panel background="#00f" />
```

Specifying the Panel size

As described in sections Terrace 55 and Tower 56, when you add a number of Panels to a Terrace or a Tower, they are sized proportionally so that each Panel takes up the same space as the other. However, you may want to explicitly define the size of a Panel. To do this add the attribute `width="<pixel-value>"`, or `height="<pixel-value>"` to prescribe a fixed pixel width or height.

Here are two examples.

This defines a Panel with a width of 100 pixels.

```
<Panel width="100" />
```

This defines a Panel with a height of 20 pixels.

```
<Panel height="20" />
```

Note: When panels are grouped in a layout component, then fixing the height or width of one of the Panels will leave the remaining height or width to be shared between the remaining Panels.

Example: Sharing width between the remaining Panels

```
<Terrace>
      <FrameItems>
             <Panel width="100" />
             <Panel />
             <Panel />
      </FrameItems>
</Terrace>
```

Note: if the browser is resized, the relative widths and heights of the Panels will be preserved. However, the widths and heights will now differ from those set in the XML file.

Adding content to a Panel

You can add HTML content to your Panels by referencing an external source file. You do this using the attribute `src="<file-location>"`.

For example:

```
<Panel src="content.html" />
```

Note: You can use the Panel component whenever you wish to add some simple local HTML content to your layout. However, this component is *not* suitable for more complicated uses, such as:

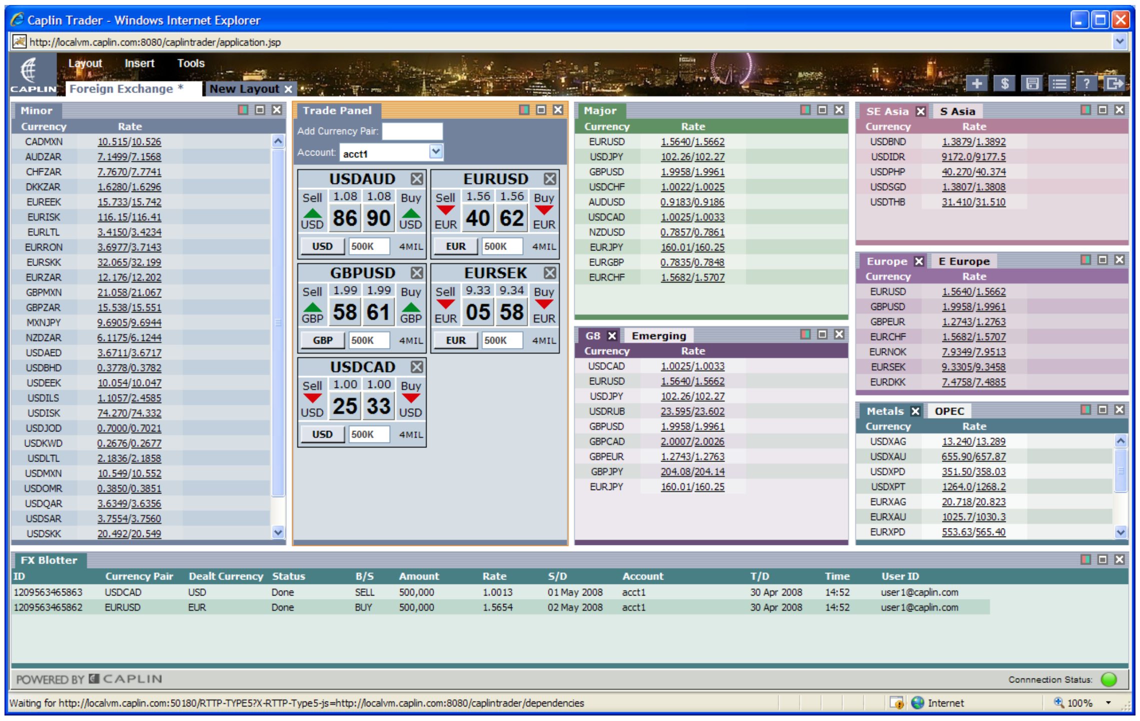
1. dynamically changing content which makes use of JavaScript;
2. content which requires interaction with the rest of the application, such as responding to life-cycle events on the component itself (for example closing the component, resizing the component, and so on).

If you want to create components that are capable of doing more than the simple display of local HTML content, we recommend you read the document **CaplinTrader: Customizing the Content**.

4.6 The Caplin Trader Client layout

The previous sections explained how set up simple layouts, how to add attributes to the Panels and how to set up the Application Layout Stack.

These concepts are used in the Caplin Trader Client Reference Implementation, as shown in the following picture and its accompanying schematic:


Caplin Trader - Windows Internet Explorer
http://localvm.caplin.com:8080/caplintrader/application.jsp
Layout
Insert
Tools
CAPLIN
Foreign Exchange *
New Layout
Trade Panel
Add Currency Pair:
Account: acct1
POWERED BY CAPLIN
Connnection Status:
Waiting for http://localvm.caplin.com:50180/RTTP-TYPE5?X-RTTP-Type5-js=http://localvm.caplin.com:8080/caplintrader/dependencies
Internet
100%

Minor

Currency	Rate
CADMXN	10.515/10.526
AUDZAR	7.1499/7.1568
CHFZAR	7.7670/7.7741
DKKZAR	1.6280/1.6296
EUREEK	15.733/15.742
EURISK	116.15/116.41
EURLTL	3.4150/3.4234
EURRON	3.6977/3.7143
EURSKK	32.065/32.199
EURZAR	12.176/12.202
GBPMXN	21.058/21.067
GBPZAR	15.538/15.551
MXNJPY	9.6905/9.6944
NZDZAR	6.1175/6.1244
USDAED	3.6711/3.6717
USDBHD	0.3778/0.3782
USDEEK	10.054/10.047
USDILS	1.1057/2.4585
USDISK	74.270/74.332
USDJOD	0.7000/0.7021
USDKWD	0.2676/0.2677
USDLTL	2.1836/2.1858
USDMXN	10.549/10.552
USDOMR	0.3850/0.3851
USDQAR	3.6349/3.6356
USDSAR	3.7554/3.7560
USDSKK	20.492/20.549

Trade Panel

Pair	Sell	Sell price	Buy price	Buy	Dealt	Amount	Limit
USDAUD	USD	1.08 86	1.08 90	USD	USD	500K	4MIL
EURUSD	EUR	1.56 40	1.56 62	EUR	EUR	500K	4MIL
GBPUSD	GBP	1.99 58	1.99 61	GBP	GBP	500K	4MIL
EURSEK	EUR	9.33 05	9.34 58	EUR	EUR	500K	4MIL
USDCAD	USD	1.00 25	1.00 33	USD	USD	500K	4MIL

Major

Currency	Rate
EURUSD	1.5640/1.5662
USDJPY	102.26/102.27
GBPUSD	1.9958/1.9961
USDCHF	1.0022/1.0025
AUDUSD	0.9183/0.9186
USDCAD	1.0025/1.0033
NZDUSD	0.7857/0.7861
EURJPY	160.01/160.25
EURGBP	0.7835/0.7848
EURCHF	1.5682/1.5707

G8 | Emerging

Currency	Rate
USDCAD	1.0025/1.0033
EURUSD	1.5640/1.5662
USDJPY	102.26/102.27
USDRUB	23.595/23.602
GBPUSD	1.9958/1.9961
GBPCAD	2.0007/2.0026
GBPEUR	1.2743/1.2763
GBPJPY	204.08/204.14
EURJPY	160.01/160.25

SE Asia | S Asia

Currency	Rate
USDBND	1.3879/1.3892
USDIDR	9172.0/9177.5
USDPHP	40.270/40.374
USDSGD	1.3807/1.3808
USDTHB	31.410/31.510

Europe | E Europe

Currency	Rate
EURUSD	1.5640/1.5662
GBPUSD	1.9958/1.9961
GBPEUR	1.2743/1.2763
EURCHF	1.5682/1.5707
EURNOK	7.9349/7.9513
EURSEK	9.3305/9.3458
EURDKK	7.4758/7.4885

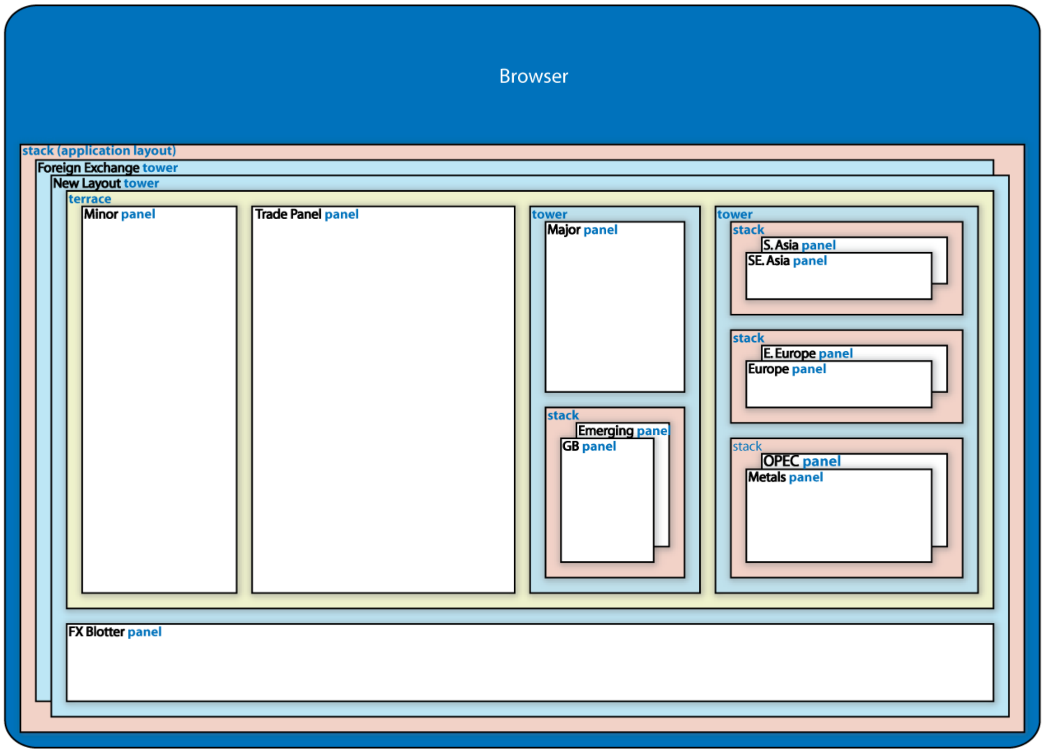
Metals | OPEC

Currency	Rate
USDXAG	13.240/13.289
USDXAU	655.90/657.87
USDXPD	351.50/358.03
USDXPT	1264.0/1268.2
EURXAG	20.718/20.823
EURXAU	1025.7/1030.3
EURXPD	553.63/565.40

FX Blotter

ID	Currency Pair	Dealt Currency	Status	B/S	Amount	Rate	S/D	Account	T/D	Time	User ID
1209563465863	USDCAD	USD	Done	SELL	500,000	1.0013	01 May 2008	acct1	30 Apr 2008	14:52	user1@caplin.com
1209563465862	EURUSD	EUR	Done	BUY	500,000	1.5654	02 May 2008	acct1	30 Apr 2008	14:52	user1@caplin.com

The layout of the Caplin Trader Reference Implementation



The layout schematic for the Caplin Trader screen shot above

You can see in this layout schematic that the Caplin Trader Client layout contains a number of nested components, where the main Application Layout Stack contains two Towers, the top-most of which contains a Terrace and a Panel ("FX Blotter"). The Terrace contains a number of components; two Panels ("Minor" and "Trade Panel") and two Towers.

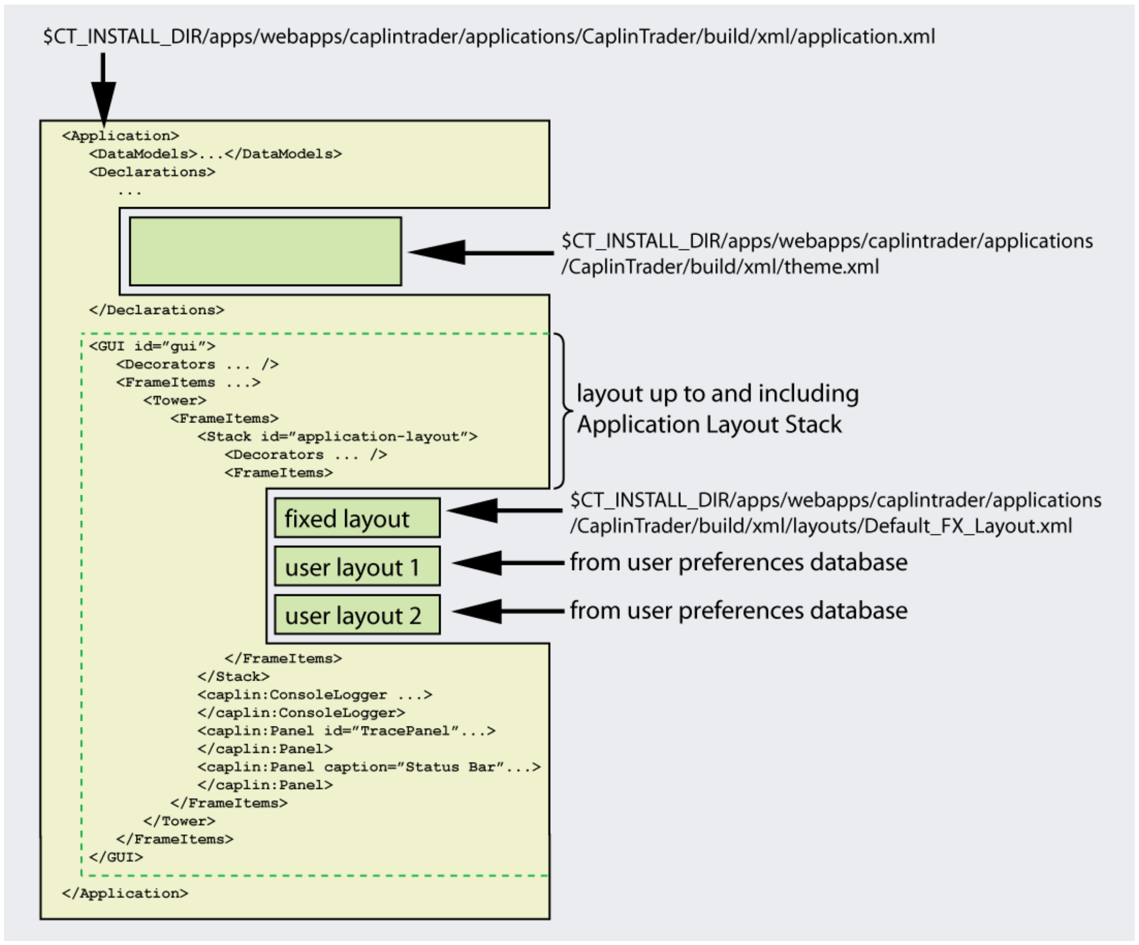
The first Tower contains a Panel ("Major") above a Stack. The second Tower contains three Stacks.

How the layout is loaded at runtime

Our first tutorial, Changing the Look and Feel of the Caplin Trader Client 7, addressed the color scheme of the Caplin Trader Client application.

In this section we will describe the extra complexity used in the demo's layout that facilitates a much greater level of end user customization whilst the Caplin Trader Client is running. The first thing to appreciate is that the menus in the demo allow the user to add more layouts to the Application Layout Stack whenever they wish. On each new layout, new grid widgets and Trade Panels can be added and dragged around to form a layout that the user desires. At any time during the user's session the new layouts can be saved to a database, so that next time they log in, they will be presented with their saved layout.

The figure below shows how the layout of the interface is determined when the application starts up.


$CT_INSTALL_DIR/apps/webapps/caplintrader/applications/CaplinTrader/build/xml/application.xml
$CT_INSTALL_DIR/apps/webapps/caplintrader/applications /CaplinTrader/build/xml/theme.xml
layout up to and including Application Layout Stack
$CT_INSTALL_DIR/apps/webapps/caplintrader/applications /CaplinTrader/build/xml/layouts/Default_FX_Layout.xml
fixed layout
user layout 1
from user preferences database
user layout 2
from user preferences database

```
<Application>
   <DataModels>...</DataModels>
   <Declarations>
      ...

   </Declarations>
   <GUI id="gui">
      <Decorators ... />
      <FrameItems ...>
         <Tower>
            <FrameItems>
               <Stack id="application-layout">
                  <Decorators ... />
                  <FrameItems>

                  </FrameItems>
               </Stack>
               <caplin:ConsoleLogger ...>
               </caplin:ConsoleLogger>
               <caplin:Panel id="TracePanel"...>
               </caplin:Panel>
               <caplin:Panel caption="Status Bar"...>
               </caplin:Panel>
            </FrameItems>
         </Tower>
      </FrameItems>
   </GUI>

</Application>
```

The outer part of the layout, which is common to all users, is contained in the file ***$CT_INSTALL_DIR/****apps/webapps/caplintrader/applications/CaplinTrader/build/xml/application.xml*. This file (specifically the section within the dotted green lines in the figure) defines the layout up to, and including, the Application Layout Stack. The content within `<FrameItems>` tags in the Application Layout Stack is imported at application startup from a number of different sources.

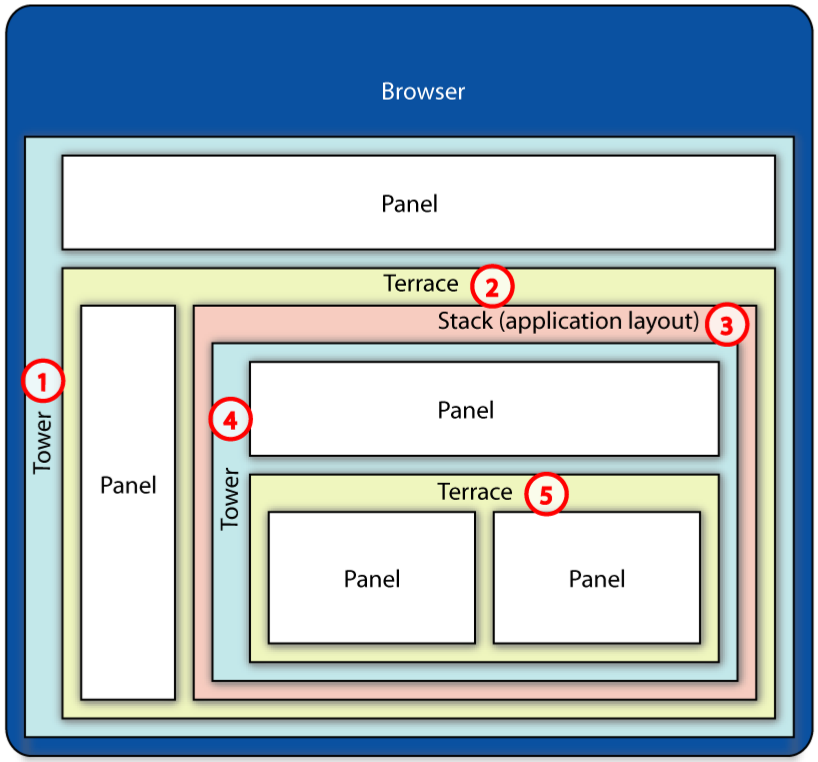
The first source is from ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/build/xml/layouts/Default_FX_Layout.xml*. This file makes up the grid widgets and trade blotters in the Foreign Exchange layout.

Further layouts, arranged alongside the Foreign Exchange layout, can be saved to, or loaded from, a database. In the figure we have shown two extra user layouts, 'user layout 1' and 'user layout 2', but the actual number can be more or less.

For all of the layouts there may be some shared characteristics that are always the same, such as what type of buttons to put on the title bars of the various window components. For example, in the Caplin Trader Client, three buttons (the color chooser , maximize button and close button) are present on all window components. These shared characteristics are defined in the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/build/xml/theme.xml*. This file is also imported into the file ***$CT_INSTALL_DIR****/apps/webapps/caplintrader/applications/CaplinTrader/build/xml/application.xml* at runtime.

5 Tutorial: Building a new layout

The most useful layouts are created by nesting layout components inside each other. For example, the following layout is created by nesting a Terrace (2) within a Tower (1), a Stack (3) within that Terrace, a second Tower (4) within that Stack and another Terrace (5) within that Tower.

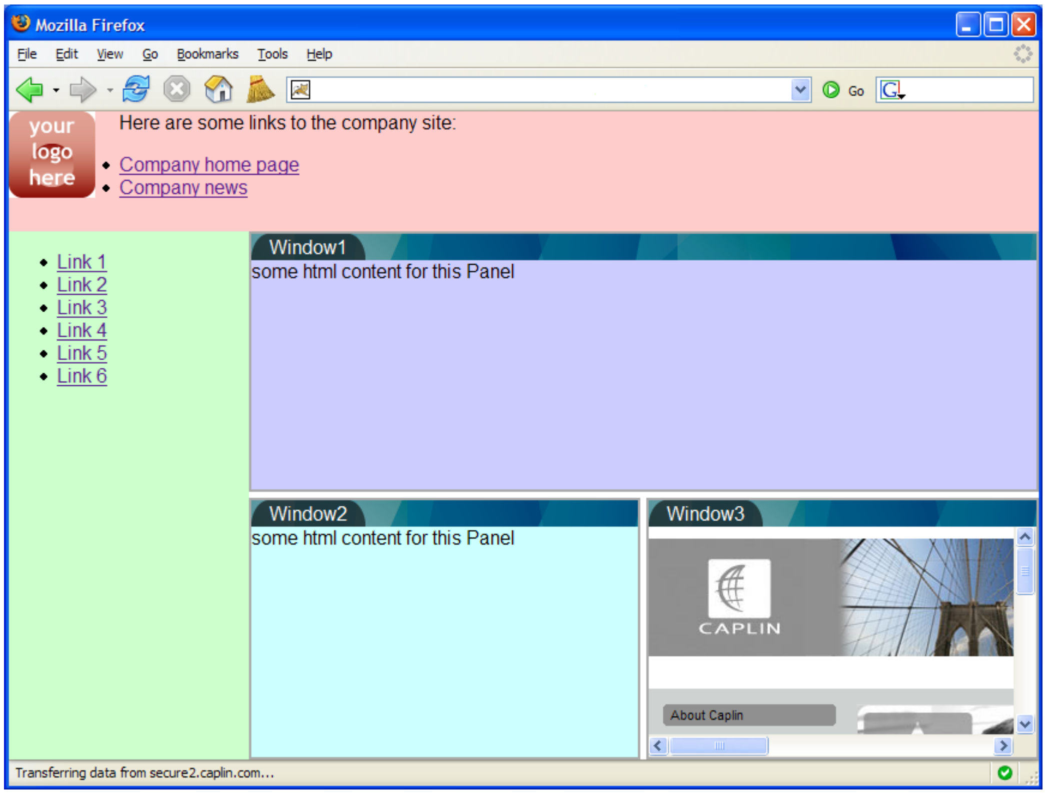


A complex layout using nested elements

The tutorial uses this layout since it follows a typical format of many web pages, having a title bar (the top Panel), a navigation bar (the left Panel), and a main content area (the Application Layout Stack).

- The top Panel will be used to display an HTML snippet that has a corporate logo and some dummy links to pages on a corporate website.
- The left side Panel will reference more HTML content that will contain a list of links as in a typical website menu.
- The middle three panels will be contained within the Application Layout Stack, and so this means that if we choose to drag and drop these Panels, we can do so anywhere within the borders of the Stack. The content of these Panels will be set to point to some external internet sources.

The picture below shows how this will look:



Here are some links to the company site:

- Company home page
- Company news

- Link 1
- Link 2
- Link 3
- Link 4
- Link 5
- Link 6

some html content for this Panel

some html content for this Panel

Transferring data from secure2.caplin.com...

Follow the next steps and you will see how easy it is to create this kind of layout.

5.1 Creating a nested layout

First go to the location on your web server where you have deployed your Caplin Trader installation. Since the directory where Caplin Trader is installed may not be the same in every case, we suggest you follow the procedure below to record this in an environment variable.

- Find the directory where the installed Caplin Trader software is located (which contains the directory *apps*).

  For example */home/CaplinTrader*

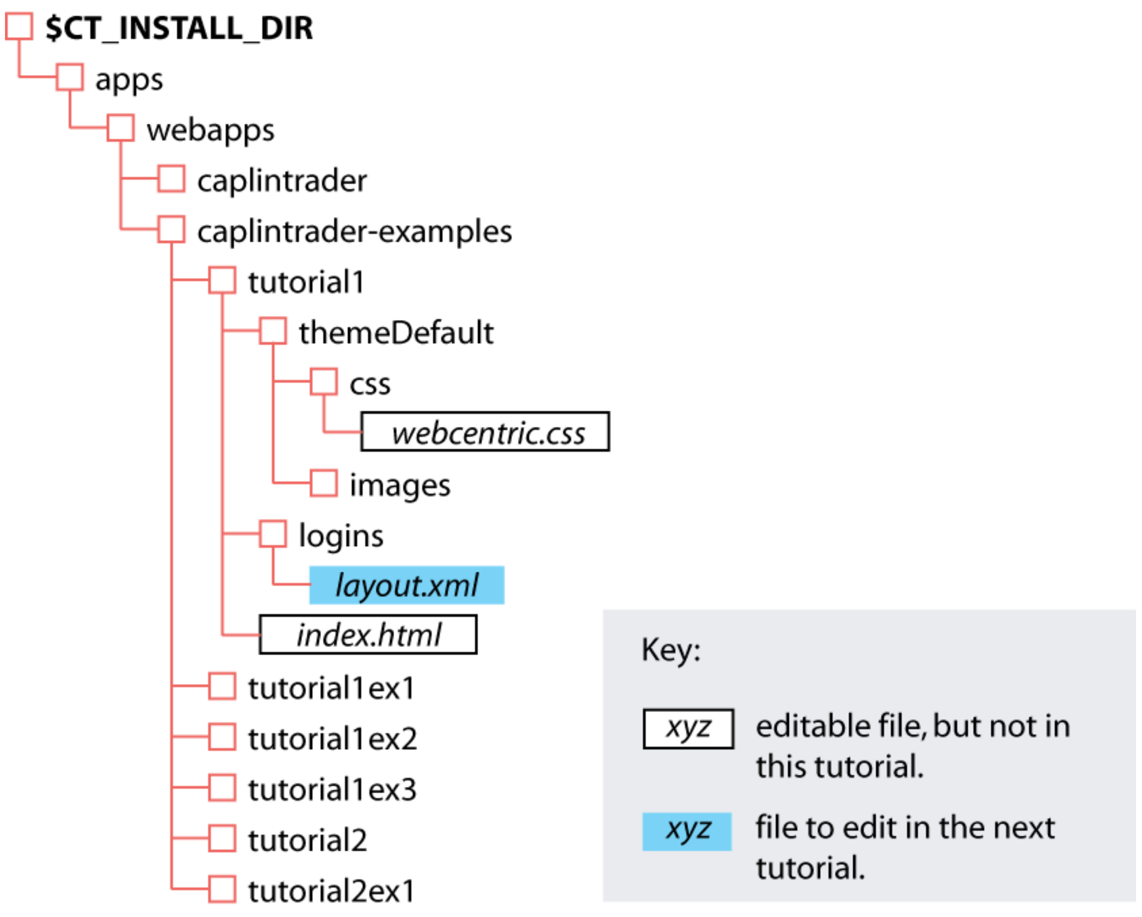
- Define an environment variable **`CT_INSTALL_DIR`** that resolves to your chosen installation directory.

  For example:

```
export CT_INSTALL_DIR=/home/CaplinTrader
```

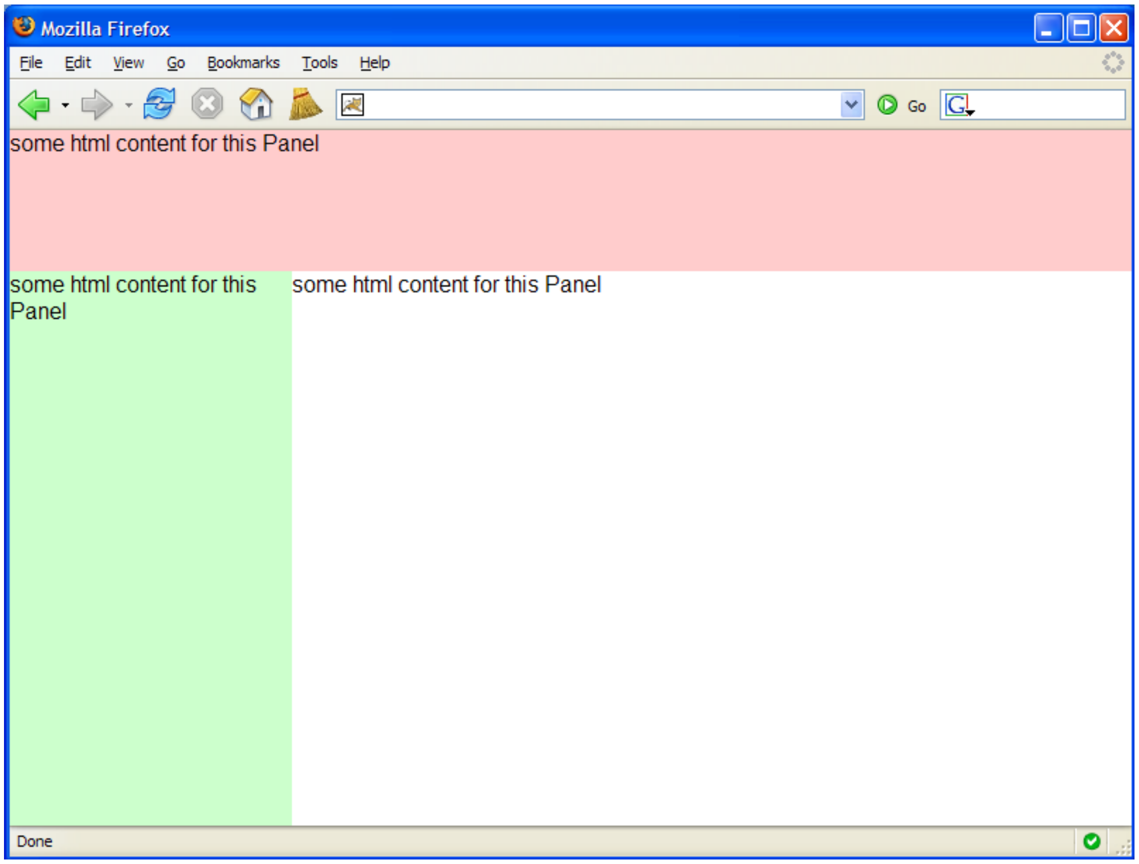
Note: In the rest of this tutorial the installation directory where the Caplin Trader evaluation software is located is referred to using the environment variable name ***$CT_INSTALL_DIR***

You will find the following folder structure:



In this tutorial you are going to start by working with the application defined in the folder tutorial1. To see what this looks like, launch the file tutorial1/index.html in your browser. The url for this will be something like
http://<your.domain.name>:8080/caplintrader-examples/tutorial1/index.html

The browser should display a page that looks like this:


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Go
some html content for this Panel
some html content for this Panel
some html content for this Panel
Done

The starting layout for this tutorial: title bar, navigation bar and main content area

In the *tutorial1/logins* directory you will find the file *layout.xml*

Open it up in your favorite XML editor.

The markup between the `<GUI>` tags should look like the following (note that tab indentation has been reduced here):

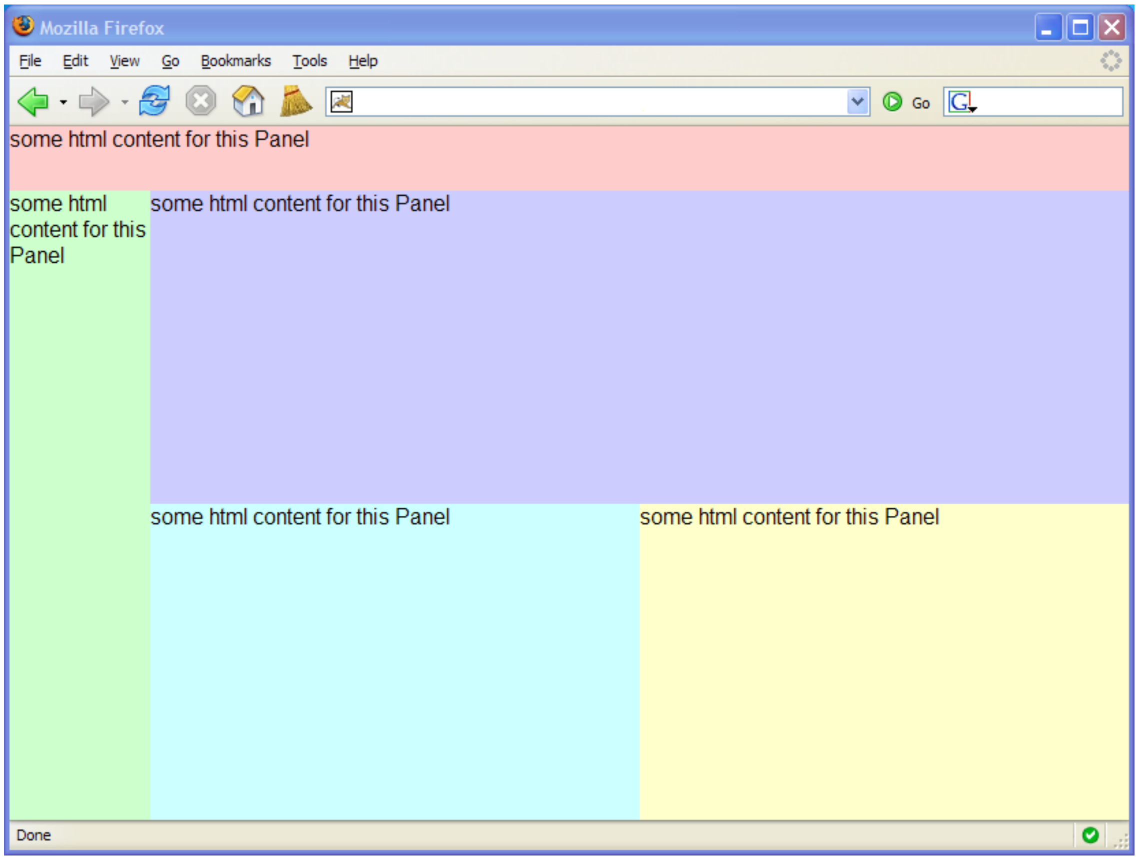
```
<GUI id="gui">
<FrameItems>
       <Tower>
       <FrameItems>
              <Panel height="100" background="#fcc" src="content.html"
                     fixed_size="true" />
              <Terrace>
              <FrameItems>
                     <Panel width="200" background="#cfc" src="content.html"
                            fixed_size="true" />
                     <Stack id="application-layout">
                     <FrameItems>
                            <Page id="1" page_type_id="1" layout_id="2"
                                   name="Foreign Exchange" path=""
                                   caption="Foreign Exchange">
                                   <Panel background="#fff" src="content.html" />
                            </Page>
                     </FrameItems>
                     </Stack>
              </FrameItems>
              </Terrace>
       </FrameItems>
       </Tower>
</FrameItems>
</GUI>
```

The three Panels in the markup correspond to the red title bar (`background="#fcc"`), the green navigation bar (`background="#cfc"`)and the main white content area (`background="#fff"`). We are going extend this simple layout by swapping the center Panel (in the Application Layout Stack) with three smaller Panels which, in the end, we will be able to resize, drag around and, in one of them, view external websites.

The first thing to do is to replace the inner Panel tag with code that defines a Tower with a Panel and a Terrace in it, the Terrace containing two further Panels:

```
<GUI id="gui">
<FrameItems>
  <Tower>
    <FrameItems>
      <Panel  height="100" background="#fcc" src="content.html"
              fixed_size="true" />
      <Terrace>
        <FrameItems>
          <Panel width="200" background="#cfc" src="content.html"
                 fixed_size="true" />
          <Stack id="application-layout">
            <FrameItems>
              <Page id="1" page_type_id="1" layout_id="2"
                    name="Foreign Exchange" path=""
                    caption="Foreign Exchange">
                <Tower>
                  <FrameItems>
                    <Panel background="#ccf" src="content.html">
                    </Panel>
                    <Terrace>
                       <FrameItems>
                        <Panel background="#cff" src="content.html">
                        </Panel>
                        <Panel background="#ffc" src="content.html">
                        </Panel>
                      </FrameItems>
                    </Terrace>
                  </FrameItems>
                </Tower>
              </Page>
            </FrameItems>
          </Stack>
        </FrameItems>
       </Terrace>
      <FrameItems>
  </Tower>
</FrameItems>
</GUI>
```

To see what this looks like, *clear the browser's cache* and refresh the web browser. The browser should display the following page:


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Go
some html content for this Panel
some html content for this Panel
some html content for this Panel
some html content for this Panel
some html content for this Panel
Done

Step 1: three central content Panels have been added

In case you do not see this, we have prepared a page for you so you can see what it should look like after this step: view the following URL on your installation:
http://<your.domain.name>:8080/caplintrader-examples/tutorial1step1/index.html

The corresponding code for the layout of this page can be seen at *tutorial1step1/logins/layout.xml*

However, for the rest of this tutorial, continue working with the file you have been editing. You can see working versions of the next steps of this application at:

- http://<your.domain.name>:8080/caplintrader-examples/tutorial1step2/index.html
- http://<your.domain.name>:8080/caplintrader-examples/tutorial1step3/index.html

5.2 Adding Decorators

One of the main features of the Caplin Trader Client application framework is the ability to have windows that can be dragged and dropped into new positions on-screen.

The way you can do this is by adding *Decorators* to your XML markup. Let's see how we do this.

A Decorator can be any addition to a layout or window component that adds some extra element that enhances its appearance or functionality. For example:

- A Border: a defined edge of specific width and color that surrounds the Panel.
- A Handle: a bar across the top of a Panel that can be grabbed to drag the Panel around.

Adding a border

Let's add a Border to the third Panel in our layout. Do this by adding the `<Decorators></Decorators>` tags to the Panel. Then, within those Decorator tags, add a Border component, with a width of 2 pixels. See the code below:

```
<Panel background="#ccf" src="content.html">
      <Decorators>
             <Border style="outer" border_width="2" />
      </Decorators>
</Panel>
```

Run this again (by clearing the browser cache and refreshing the web page) and you will see that the wide blue panel now has a border around it.

Note: The color of this border can be changed, but not in this file. We discuss color schemes in Changing the color scheme 15.

Adding a Handle

To drag a panel around, so add a *Handle* to it using the `<Handle>` tag. Note that we must also add a *caption* to the containing Panel, so that it knows what 'title' to put on the Handle:

```
<Panel background="#ccf" src="content.html" caption="Window1">
      <Decorators>
             <Border style="outer" border_width="2" />
             <Handle  handle_height="22"  drag_action="SNAP_FRAMEITEM"
                    drop_target="SNAP_FRAMEITEM" />
      </Decorators>
</Panel>
```

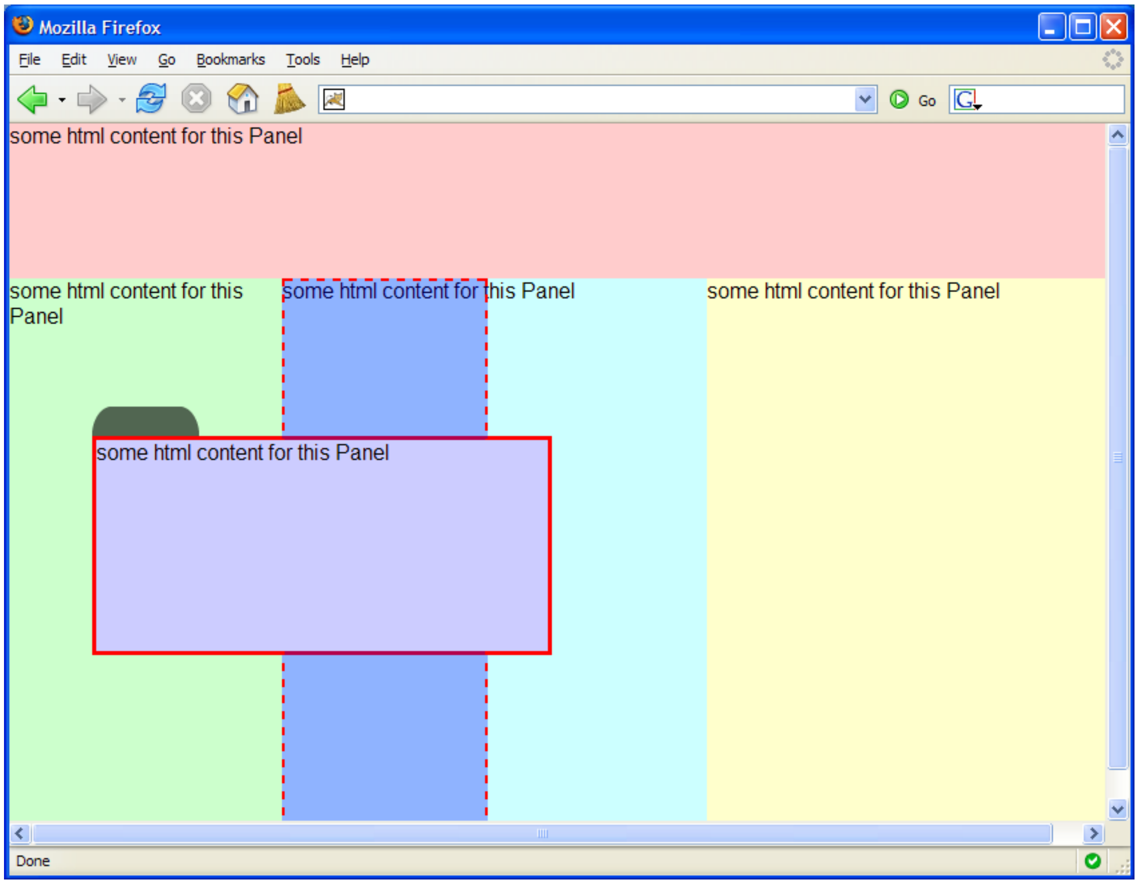
Notice how the Handle has a `handle_height` attribute which is again a pixel value. The second attribute, `drag_action` allows you to define whether the Panel can be dragged, or not. This attribute should be set to `"SNAP_FRAMEITEM"`. Finally the last attribute, `drop_action`, allows you to specify that a window in the process of being dragged, can be dropped on the Handle of another window. When you do this, the two windows are layered on top of one another in a Stack arrangement, and each Panel can be accessed via the Tabs on the newly created Stack. The value for this attribute is also `"SNAP_FRAMEITEM"`.

Also notice that we have added the attribute `caption="Window1"` to the Panel tag. This is necessary because, every time you add a Handle to a Panel it must have a title to put in the handle. Typically this title reflects the content of the Panel in question. Here we have called it "Window1" since we have not put any specific content in the Panel yet.

If you refresh the web page with this new change (remember to clear the cache), you can see that the Panel in question now has a 'title bar' (the Handle). Hold down the mouse button on the title bar and you can drag the Panel around. While dragging the Panel, you will notice that the remaining Panels fill the space where "Window1" was located.

The styling that has been used to give the Handle its specific look and feel has been predefined in the files *webapps/tutorial1/themeDefault/css/webcentric.css* and *webcentric-ie6.css*.
There is no need to edit these files now, if you want to see how to change this styling, see Advanced tutorial: changing the appearance of your new layout 81.

In the Layout Concepts 51 section we discussed how the Application Layout Stack defines the limits of the drag and drop action for Panels contained within it. You can see this in action here when you try to drag the Panel to the left or to the top. The furthest it can be dragged is to the limits of the Application Layout Stack.


Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
Go
some html content for this Panel
some html content for this Panel
some html content for this Panel
some html content for this Panel
some html content for this Panel
Done

Dragging the top window to the left of the Application Layout Stack

Making the other Panels draggable

Let's make the other two Panels within the Application Layout Stack draggable too. The simplest way of doing this is to copy and paste the `<Decorators></Decorators>` tags and their contents into the other two Panels. However, since those Decorator declarations would be identical, it's better to declare the Decorators *only once* and reference them from the three Panels.

To do this, start by copying the `<Decorators></Decorators>` tags and their contents and delete them from inside the Panel. This should leave you with:

```
<Panel background="#ccf" src="content.html" caption="Window1">
</Panel>
```

Now paste the Decorators declaration in between the `<Declarations></Declarations>` near the top of the file. Since you're going to *reference* this Decorators declaration, you need to give it a unique id (`id="decorator1"` in this case).

Your Declarations section will now look like:

```
<Declarations>
      <Decorators id="decorator1">
             <Border style="outer" border_width="2"/>
             <Handle  handle_height="22"  drag_action="SNAP_FRAMEITEM"
                   drop_target="SNAP_FRAMEITEM" />
      </Decorators>
</Declarations>
```

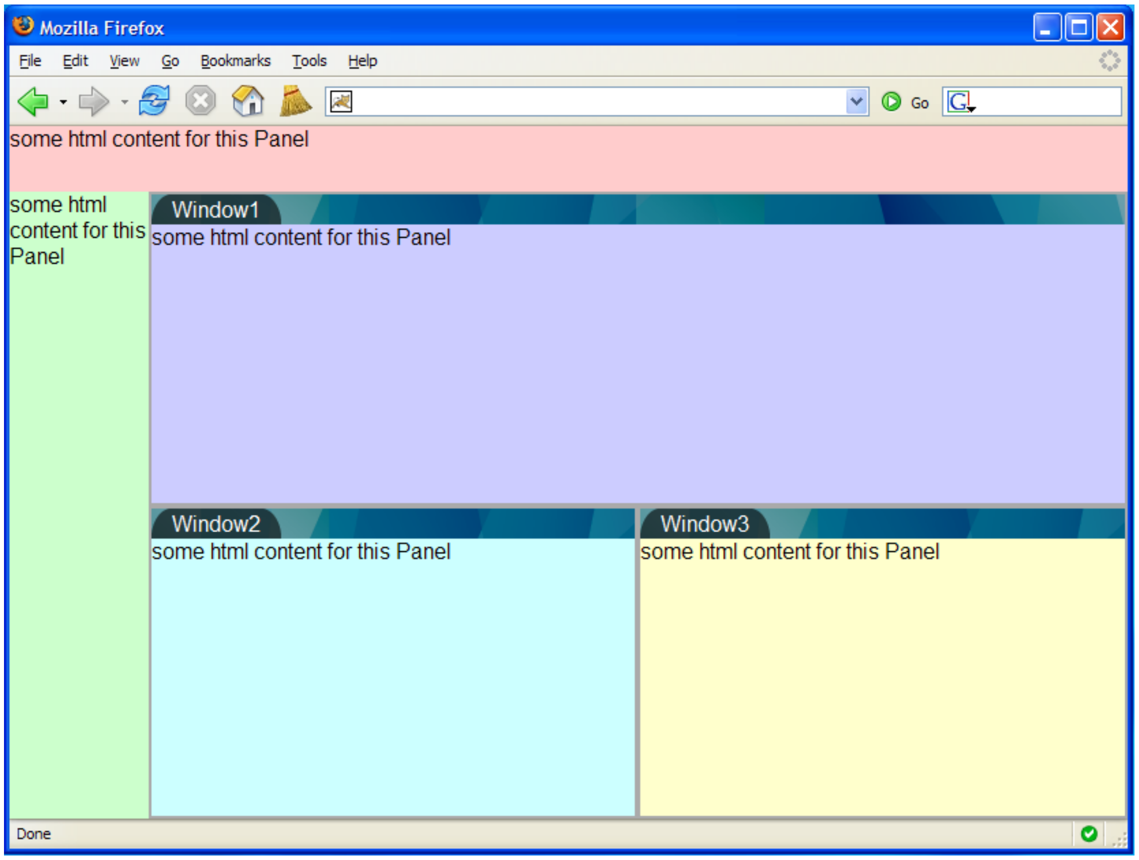
Finally, all that remains is to reference `decorator1` from inside the three Panels in the Application Layout Stack. So, write the following inside the first Panel in the Application Layout Stack:

```
<Panel background="#ccf" src="content.html" caption="Window1">
      <Decorators xref="Declarations/Decorators[@id='decorator1']" />
</Panel>
```

Now do the same for the other two Panels, but don't forget that both of those Panels will need a `caption` attribute within their Handles. They may look like this:

```
<Panel background="#cff" src="content.html" caption="Window2">
      <Decorators xref="Declarations/Decorators[@id='decorator1']" />
</Panel>
<Panel background="#ffc" src="content.html" caption="Window3">
      <Decorators xref="Declarations/Decorators[@id='decorator1']" />
</Panel>
```

Run the application again in your browser (by clearing the browser cache and refreshing the web page) and you will see the following:



some html content for this Panel

some html content for this Panel

some html content for this Panel

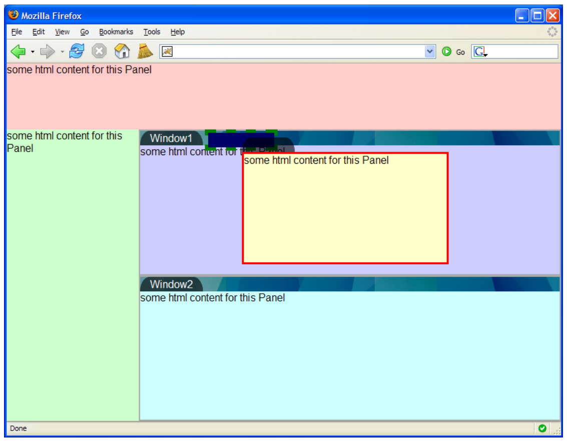
some html content for this Panel

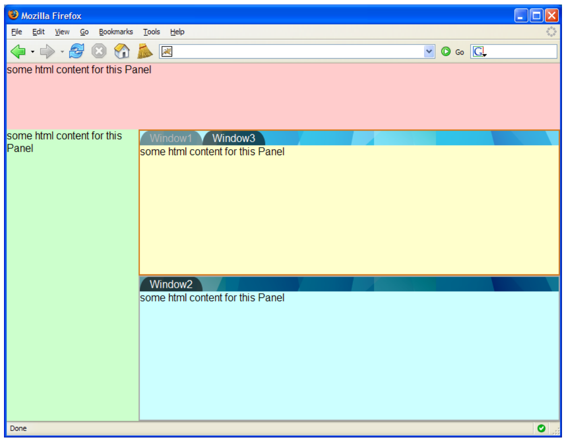
some html content for this Panel

Handles (title bars) have been added to each of the three central content Panels

You will be able to drag any of the three central Panels around on the screen. However, at this stage you will only be able to *drop* them in certain locations:

- on the borders of the Application Layout Stack (the main content area).
- on the handle (title bar) of another window. When you do this, the two windows are layered on top of each other and become a Stack. You can then toggle between the two (or more) layered windows via the tabs.


Mozilla Firefox
some html content for this Panel
some html content for this Panel
Window1
some html content for this Panel
Window2
some html content for this Panel
Done


Mozilla Firefox
some html content for this Panel
some html content for this Panel
Window1
Window3
some html content for this Panel
Window2
some html content for this Panel
Done

(1) Dragging the bottom right window over the Handle (title bar) of the top window

(2) After being dropped, it is converted into a window with two tabs.

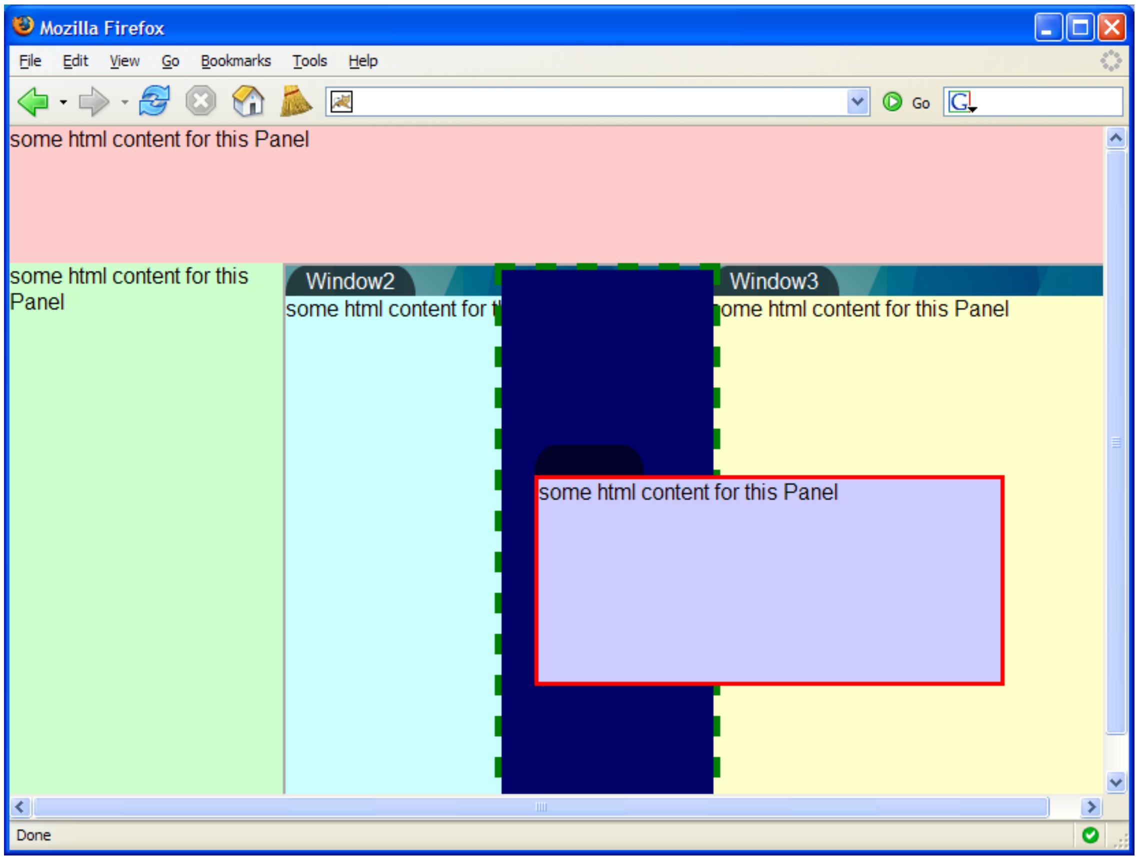
More about dragging and dropping Panels

At the moment if you drop a window anywhere else, it will just stay where you dropped it. It would be better if we could drop any one of the Panels in between the others. This would give much more flexibility to the user's interaction with the Application.

The way you can do this is by adding the attribute `drop_target="SNAP_FRAMEITEM"` to the Panels in the Application Layout Stack, as follows:

```
<Tower>
<FrameItems>
   <Panel background="#ccf" src="content.html" caption="Window1"
          drop_target="SNAP_FRAMEITEM">
      <Decorators xref="Declarations/Decorators[@id='decorator1']"/>
   </Panel>
   <Terrace>
      <FrameItems>
         <Panel background="#cff" src="content.html" caption="Window2"
                drop_target="SNAP_FRAMEITEM">
            <Decorators xref="Declarations/Decorators[@id='decorator1']"/>
         </Panel>
         <Panel background="#ffc" src="content.html" caption="Window3"
                drop_target="SNAP_FRAMEITEM">
            <Decorators xref="Declarations/Decorators[@id='decorator1']"/>
         </Panel>
      </FrameItems>
   </Terrace>
</FrameItems>
</Tower>
```

Clear the cache, reload the web page, and now drag one of the panels. You will see many more areas shown where you can drop the panel:


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Go
some html content for this Panel
some html content for this Panel
Window2
some html content for t
Window3
ome html content for this Panel
some html content for this Panel
Done

The dotted rectangle marks where Panels can be dropped; you can now drop a Panel in between other windows.

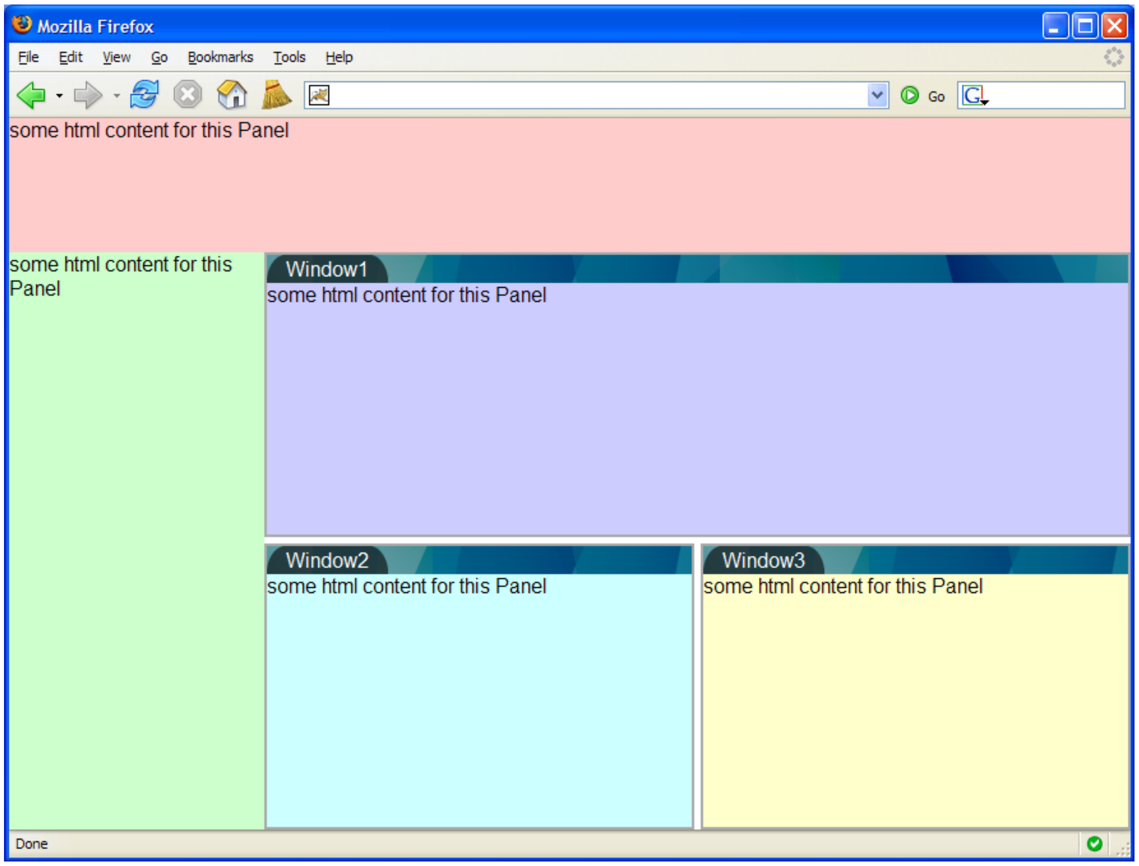
Resizing windows by adding splitters

We will now add the ability to dynamically resize the windows inside the Application Layout Stack. To do this, we need to add *Splitters.* These are vertical or horizontal 'bars' between windows. You can drag them to resize the windows that are bordered by a splitter.

To add a splitter to a Tower or Terrace, simply add the attribute `splitters="true"` to the Tower or Terrace tag. For example we will add this attribute to the Tower and Terrace *within* the Application Layout Stack.

```
<Tower splitters="true">
<FrameItems>
   <Panel background="#ccf" src="content.html" caption="Window1"
      drop_target="SNAP_FRAMEITEM">
      <Decorators xref="Declarations/Decorators[@id='decorator1']"/>
   </Panel>
   <Terrace splitters="true">
      <FrameItems>
         <Panel background="#cff" src="content.html" caption="Window2"
                drop_target="SNAP_FRAMEITEM">
            <Decorators xref="Declarations/Decorators[@id='decorator1']"/>
         </Panel>
         <Panel background="#ffc" src="content.html" caption="Window3"
                drop_target="SNAP_FRAMEITEM">
            <Decorators xref="Declarations/Decorators[@id='decorator1']"/>
         </Panel>
      </FrameItems>
   </Terrace>
</FrameItems>
</Tower>
```

Clear the cache, refresh your webpage, and you should notice two white bars appear between the internal windows, as shown in the following picture. You can drag these bars around to resize the various windows.



some html content for this Panel

some html content for this Panel

some html content for this Panel

some html content for this Panel

some html content for this Panel

Step 2: splitters have been added so that windows can be resized

If you are having trouble getting to this stage, you can see how the results should look at:
http://<your.domain.name>:8080/caplintrader-examples/tutorial1step2/index.html

The corresponding code can be seen at *tutorial1step2/logins/layout.xml*

5.3 Adding local content (logo and site map)

Now let's add some content to make this a useful application.

The very top Panel (the first one in the XML file) is a great place for a corporate logo and some links to your corporate website. You can put any local HTML markup in the HTML file referenced by a Panel, so let's do that. Change the `src="..."` attribute in the top Panel to refer to *titlebar.html*:

```
<Panel height="100" background="#fcc" src="titlebar.html" />
```

The left-most Panel (the second one in the XML markup) is the usual place where a site menu might go. Change the `src` reference to point to *menu.html*:

```
<Panel width="200" background="#cfc" src="menu.html" />
```

Clear the cache, reload the application, and you will see a logo, some dummy links in the top Panel, and a menu of links in the sidebar Panel. The simple HTML code for both of these is all in the two HTML files *titlebar.html* and *menu.html*. Feel free to open them in an editor and modify them as you wish.

Note: You can use the Panel component whenever you wish to add some simple local HTML content to your layout. However, this component is *not* suitable for more complicated uses, such as:

1. dynamically changing content which makes use of JavaScript;
2. content which requires interaction with the rest of the application, such as responding to life-cycle events on the component itself (for example closing the component, resizing the component, and so on).

If you want to create components that are capable of doing more than the simple display of local HTML content, we recommend you read the document **CaplinTrader: Customizing the Content**.

5.4 Adding external content

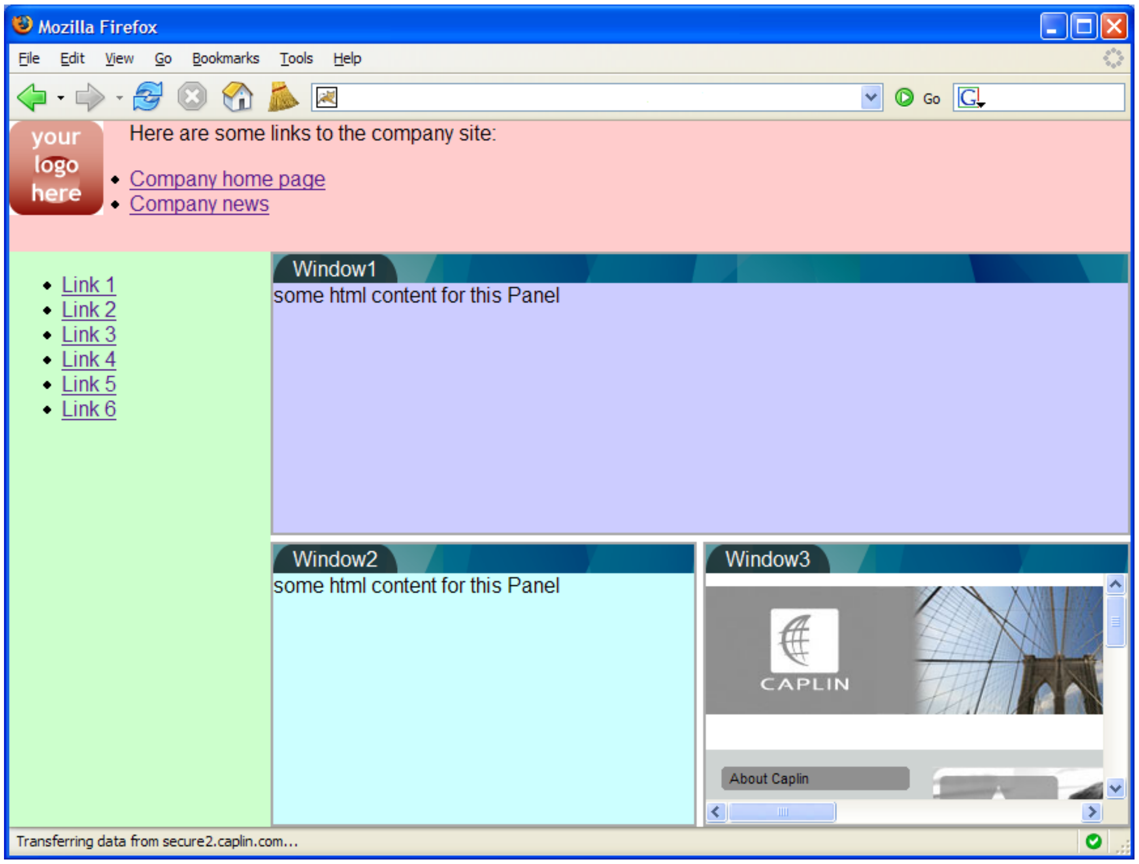
Now let's add some *external* content in the bottom right window. To do that, we must change the Panel to a **Frame**. Whilst Panels can only access local content, a Frame can access external content.

Change the last Panel in the XML markup to be a Frame, and remove the `background="#ffc"` attribute, since setting the background has no effect on an external website.

Set the `src="..."` attribute to point to http://www.caplin.com :

```
<Frame src="http://www.caplin.com" caption="Window3"
       drop_target="SNAP_FRAMEITEM">
   <Decorators xref="Declarations/Decorators[@id='decorator1']"/>
</Frame>
```

Your application should now look like this:



Here are some links to the company site:

- Company home page
- Company news

- Link 1
- Link 2
- Link 3
- Link 4
- Link 5
- Link 6

some html content for this Panel

some html content for this Panel

Transferring data from secure2.caplin.com...

Step 3: we have added external content via the Frame tag

If you had trouble getting to this final stage, you can see how it should look at: http://<your.domain.name>:8080/caplintrader-examples/tutorial1step3/index.html The corresponding code can be seen at *tutorial1step3/logins/layout.xml*

This concludes the layout tutorial. Feel free to play with the layout markup and attributes as discussed to change the way this application behaves and the internal or external content that is loaded.

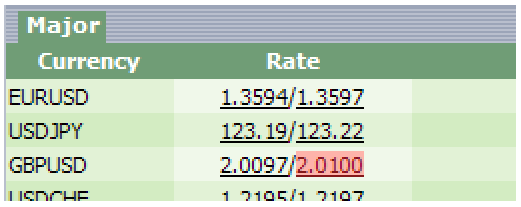
In the next advanced tutorial Changing the appearance of your new layout 81 we move on to modifying the look and feel of the tabs in the title bars.

6 Advanced tutorial: changing the appearance of your new layout

In this tutorial we will explore changing the appearance of the alternative Caplin Trader Client layout built in the tutorial Building a new layout 64, by modifying the CSS file that defines the *visual styles* for the layout. In particular we will examine how to change the shape and color of the tabs by referencing a new set of image files used to render them.

6.1 Introduction

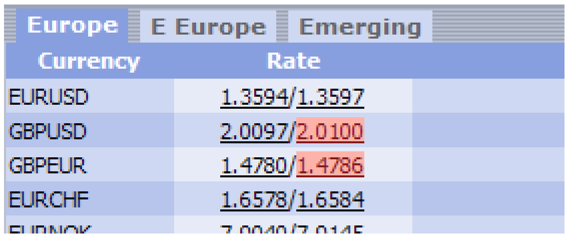
Most window components in Caplin Trader Client have 'tabs' which tell the user what the component relates to. For example, in the picture below, the tab is the element marked "Major" (the component contains foreign exchange currency rates for major currency pairs).



Major

Currency	Rate
EURUSD	1.3594/1.3597
USDJPY	123.19/123.22
GBPUSD	2.0097/2.0100
USDCHF	1.2195/1.2197

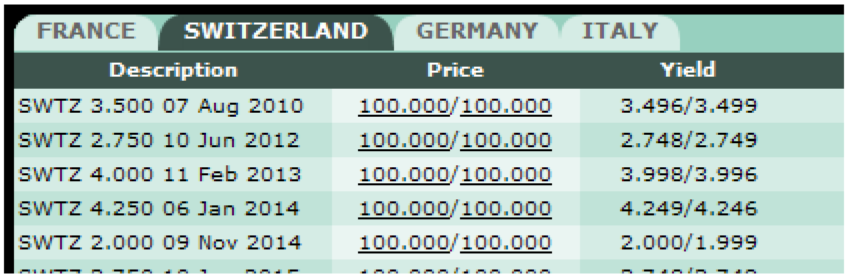
If the window consists of several layered elements in a Stack, then you will see a number of tabs above the window area; one for each element in the Stack. Only one of the tabs can be selected at any one time, and the element displayed corresponds to that tab. In the picture below, there are three tabs, and the component shown corresponds to the selected tab, "Europe".



Europe | E Europe | Emerging

Currency	Rate
EURUSD	1.3594/1.3597
GBPUSD	2.0097/2.0100
GBPEUR	1.4780/1.4786
EURCHF	1.6578/1.6584
EURNOK	7.0040/7.0145

Tabs can be given different visual characteristics by changing the images used to render them. For example you could change them so they look like this:



FRANCE | SWITZERLAND | GERMANY | ITALY

Description	Price	Yield
SWTZ 3.500 07 Aug 2010	100.000/100.000	3.496/3.499
SWTZ 2.750 10 Jun 2012	100.000/100.000	2.748/2.749
SWTZ 4.000 11 Feb 2013	100.000/100.000	3.998/3.996
SWTZ 4.250 06 Jan 2014	100.000/100.000	4.249/4.246
SWTZ 2.000 09 Nov 2014	100.000/100.000	2.000/1.999
[illegible]	[illegible]	[illegible]

6.2 Changing the Tabs in your Caplin Trader Client GUI

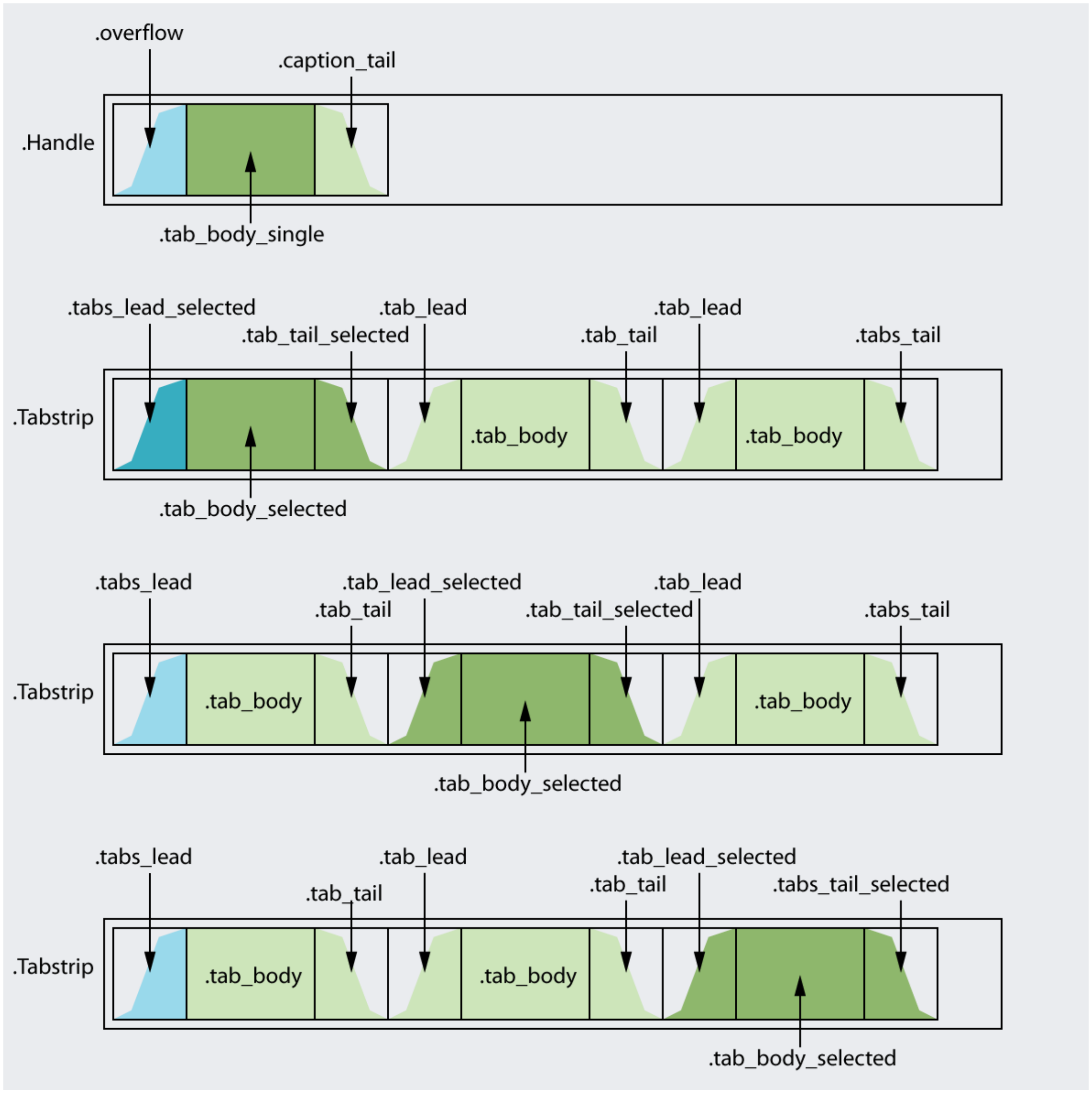
In this section you will change the rounded corners on the layout that you've built up and replace them with triangular corners. To do this you will need to change some CSS declarations. You will need to change two groups of declarations, corresponding to two different interface situations:

- when there is only one Tab at the top of a Panel, the tab is always contained in a Handle.
- when there is more than one tab at the top of a Panel, the tabs are always contained in a Tabstrip.

When you drop a window on to the top of another window with only one tab, the Handle that contains it is converted into a Tabstrip. This is important to understand, beacause we need to make two separate sets of declarations for each scenario. We need to define the CSS tab components for when they are contained in a Handle (only one tab), and we need to define the tab components for when they are contained in a Tabstrip (more than one tab).

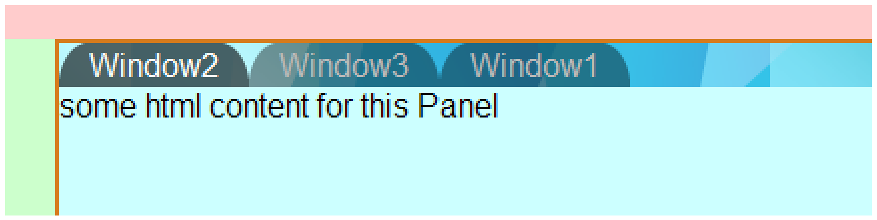
CSS classes for defining tabs

First, let's look at the CSS classes that we will need to modify in order to do this:



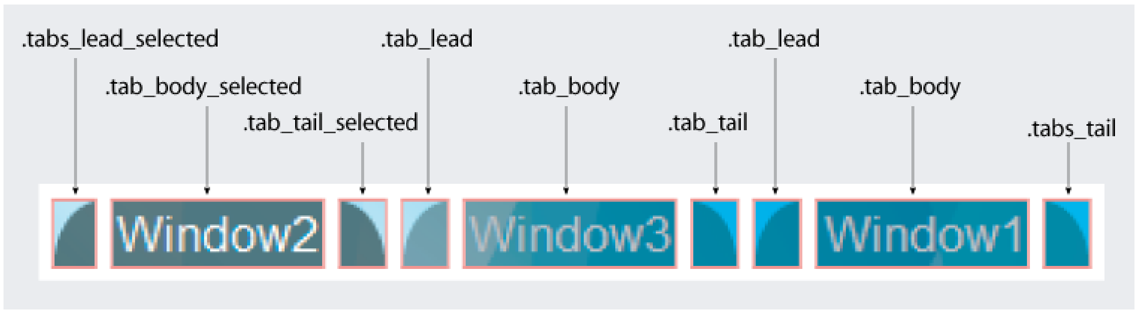
The CSS class names used to specify the different tab areas

Currently the tabs in your application look like this:



some html content for this Panel

The following diagram shows the individual images comprising the tabs, and the corresponding CSS classes that define how to render them.



Changing the appearance of the single tab in a handle

We will now change the CSS declarations for the situation where there is only one tab at the top of a Panel. Let's look in *webapps\caplintrader-examples\tutorial2\themeDefault\css\webcentric.css* to see where these files have been defined.
Search for:

```
.Handle div.overflow
```

In the code following you can see the following CSS class declarations for the beginning (`.Handle div.overflow`), the middle (`.Handle div.tab_body_single`) and the end (`.Handle div.caption_tail`):

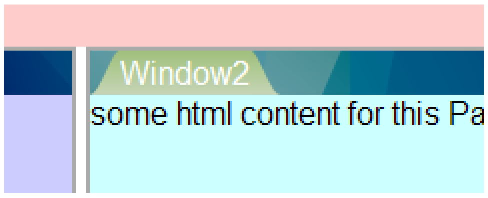
```
.Handle div.overflow,
.Handle_drag div.overflow {
      width:15px;
      background:url(../images/cornerLeft_alpha60.png) no-repeat left top;
/*    background:url(../images/greenTriangleLeftSelected.png) no-repeat left top; */
}

.Handle div.tab_body_single,
.Handle_drag div.tab_body_single {
      background:url(../images/tabFill_alpha60.png) repeat-x right top;
/*    background:url(../images/greenTriangleFillSelected.png) repeat-x left top; */
}

.Handle div.caption_tail,
.Handle_drag div.caption_tail {
      width:15px;
      background:url(../images/cornerRight_alpha60.png) no-repeat left top;
/*    background:url(../images/greenTriangleRightSelected.png) no-repeat left top;*/
}
```

In each of these CSS class declarations the property of interest is `background`. Notice that in each declaration there is a second commented out version of `background`. We can change the image paths for the `background` property to point to different images that we would like to use. For convenience in this tutorial, we will simply uncomment the line that is commented out, and comment out the previous line.

So, change the code above so that it reads:

```
.Handle div.overflow,
.Handle_drag div.overflow {
      width:15px;
/*    background:url(../images/cornerLeft_alpha60.png) no-repeat left top; */
    background:url(../images/greenTriangleLeftSelected.png) no-repeat left top;
}

.Handle div.tab_body_single,
.Handle_drag div.tab_body_single {
/*    background:url(../images/tabFill_alpha60.png) repeat-x right top; */
    background:url(../images/greenTriangleFillSelected.png) repeat-x left top;
}

.Handle div.caption_tail,
.Handle_drag div.caption_tail {
      width:15px;
/*    background:url(../images/cornerRight_alpha60.png) no-repeat left top;  */
    background:url(../images/greenTriangleRightSelected.png) no-repeat left top;
}
```

If you are using Firefox, Opera or Internet Explorer v7 as your browser, your tabs should look like this:



some html content for this Pa

However, if you are using Internet Explorer v6 (IE 6), you will need to do some further changes because IE 6 does not handle transparency in some image formats correctly. You will need to change the `filter` property. To complete these further changes, open up the file *webcentric-ie6.css* in your editor (from the same folder).

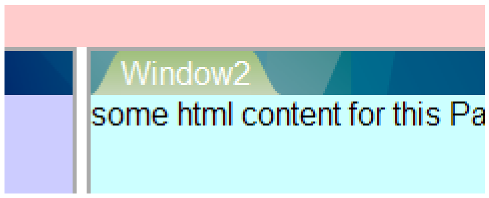
The first *three* declarations will look like this:

```
.Handle div.overflow,
.Handle_drag div.overflow {
       background:none;
       filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/cornerLeft_alpha60.png',
               sizingMethod='crop');
/*     filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/greenTriangleLeftSelected.png',
               sizingMethod='crop');*/
}

.Handle div.tab_body_single,
.Handle_drag div.tab_body_single {
       background:none;
       filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/tabFill_alpha60.png',
               sizingMethod='scale');
/*     filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/greenTriangleFillSelected.png',
               sizingMethod='scale');*/
}

.Handle div.caption_tail,
.Handle_drag div.caption_tail {
       background:none;
       filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/cornerRight_alpha60.png',
               sizingMethod='crop');
/*     filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/greenTriangleRightSelected.png',
               sizingMethod='crop');*/
}
```

Notice that for your convenience the commented out code (split on to three lines for each declaration) should be uncommented and the previous block for the `filter` property should be commented out. After you have done this, the CSS declarations should look like this:

```
.Handle div.overflow,
.Handle_drag div.overflow {
       background:none;
/*     filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/cornerLeft_alpha60.png',
               sizingMethod='crop');*/
       filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/greenTriangleLeftSelected.png',
               sizingMethod='crop');
}

.Handle div.tab_body_single,
.Handle_drag div.tab_body_single {
       background:none;
/*     filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/tabFill_alpha60.png',
               sizingMethod='scale');*/
       filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/greenTriangleFillSelected.png',
               sizingMethod='scale');
}

.Handle div.caption_tail,
.Handle_drag div.caption_tail {
       background:none;
/*     filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/cornerRight_alpha60.png',
               sizingMethod='crop');*/
       filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
               src='themeDefault/images/greenTriangleRightSelected.png',
               sizingMethod='crop');
}
```

If you are using Internet Explorer 6 (IE 6) then you can now clear the cache and refresh the browser and the tabs in IE 6 will also look like this:



Changing the appearance of the tabs in a tabstrip

That changes the look of the tabs for the case when there is only one tab on the title bar for any window, but what about when there is more than one? You get more than one tab on a title bar when another window is dragged and dropped onto the Handle of another window. If you try this now you can see that the tabs still look like the old version, that is to say they are charcoal gray with rounded corners rather than triangular.

In this case we need to change the images used for the tabs when there is more than one tab on a Tabstrip.

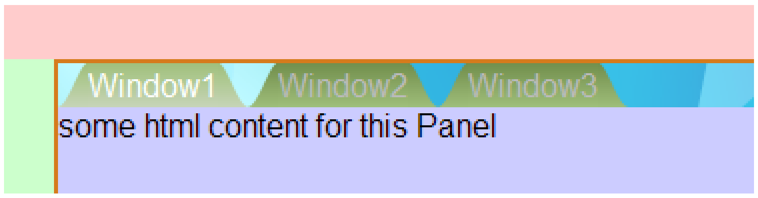
For our example here, we have chosen to have the same shape tabs as the green ones you have just seen, but in a darker green color. So to change the graphics for these, we need to change a few lines under the code that we've just changed. Going back to *webcentric.css*, and from the fourth declaration onwards, uncomment the remaining commented out background properties, and comment out the ones that were previously active:

```
.tab_lead_selected,
.tabs_lead_selected {
      width:15px;
/*    background:url(../images/cornerLeft_alpha60.png) no-repeat left top;  */
    background:url(../images/greenTriangleLeftSelected.png) no-repeat left top;
}

.tab_body_selected {
/*    background:url(../images/tabFill_alpha60.png) repeat-x right top; */
    background:url(../images/greenTriangleFillSelected.png) repeat-x left top;
}

.tab_tail_selected,
.tabs_tail_selected {
      width:15px;
/*    background:url(../images/cornerRight_alpha60.png) no-repeat left top;  */
    background:url(../images/greenTriangleRightSelected.png) no-repeat left top;
}

.tab_lead,
.tabs_lead {
      width:15px;
/*    background:url(../images/cornerLeft_alpha40.png) no-repeat left top;  */
    background:url(../images/greenTriangleLeftNotSelected.png) no-repeat left top;
}

.tab_body {
/*    background:url(../images/tabFill_alpha40.png) repeat-x right bottom; */
    background:url(../images/greenTriangleFillNotSelected.png) repeat-x left top;
}

.tab_tail,
.tabs_tail {
      width:15px;
/*    background:url(../images/cornerRight_alpha40.png) no-repeat left top;  */
    background:url(../images/greenTriangleRightNotSelected.png) no-repeat left top;
}
```

Once again though, you will need to change the corresponding declarations in the file *webcentric-ie6.css* to get it to work in Internet Explorer 6. So, if you change the remaining seven declarations, your code looks like this:

```
.tab_lead_selected,
.tabs_lead_selected {
      background:none;
/*    filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/cornerLeft_alpha60.png',
             sizingMethod='crop');*/
      filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/greenTriangleLeftSelected.png',
             sizingMethod='crop');
}
.tab_body_selected {
      background:none;
/*    filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/tabFill_alpha60.png',
             sizingMethod='scale');*/
      filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/greenTriangleFillSelected.png',
             sizingMethod='scale');
}
.tab_tail_selected,
.tabs_tail_selected {
      background:none;
/*    filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/cornerRight_alpha60.png',
             sizingMethod='crop');*/
      filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/greenTriangleRightSelected.png',
             sizingMethod='crop');
}


.tab_lead,
.tabs_lead {
      background:none;
/*    filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/cornerLeft_alpha40.png',
             sizingMethod='crop');*/
      filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/greenTriangleLeftNotSelected.png',
             sizingMethod='crop');
}
.tab_body {
      background:none;
/*    filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/tabFill_alpha40.png',
             sizingMethod='scale');*/
      filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/greenTriangleFillNotSelected.png',
             sizingMethod='scale');
}
.tab_tail,
.tabs_tail {
      background:none;
/*    filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/cornerRight_alpha40.png',
             sizingMethod='crop');*/
      filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
             src='themeDefault/images/greenTriangleRightNotSelected.png',
             sizingMethod='crop');
}
```

Clear your cache, refresh your browser and drag and drop two of the windows on to the title bar of the remaining window. You will now see a tab group like this (when the "Window1" tab is selected):



This brings us to the end of this tutorial which should leave you with a better understanding of how tabs are styled and how to change them to a style of your own choice.

7 Glossary of terms and acronyms

This section contains a glossary of terms and acronyms relating to the appearance of Caplin Trader Client.

Term	Definition
Application Layout Stack	A special type of **Stack**, only one of which is allowed in the interface. The Application Layout Stack is used to define the area within which windows can be rearranged dynamically (via drag and drop actions). Rearranged window layouts within the Application Layout Stack may, under some configurations, be saved to a database. This allows the layout to be recalled next time the user logs in.
Caption	The displayed name of a **Panel** or **Frame**.
CSS	Cascading Style Sheet, CSS files are used to customize the look and feel of web pages and applications. They are used to customize the look and feel of the Caplin Trader Client.
Frame	A rectangular area in the Caplin Trader user interface in which external HTML content may be displayed.
Handle	A rectangular area above a **Panel** or **Frame** which allows the user to drag the Panel or Frame. The Handle may also contain one tab where a caption for the Panel or Frame can be displayed.
Multiple Document Interface (MDI)	The type of interface used in many Microsoft Windows® applications, where you can open a number of documents simultaneously in the application. For example, consider Microsoft Word, where you can open many Word documents simultaneously.
Splitter	A draggable horizontal or vertical bar that separates window components in Caplin Trader. Dragging a Splitter allows the user to resize the window components that have the Splitter as a boundary.
Stack	A Caplin Trader window component that acts as a container for other components and arranges them one layer on top of another. A number of Tabs are provided to allow the user to switch between the layers. Also see The Stack 57.
Tab	A graphical element, located vertically above the top of a window component, that allows the user to switch to another layered window component in a **Stack**.
Tabstrip	A rectangular area above a **Panel** or **Frame** which contains a number of **Tabs**.
Terrace	A Caplin Trader window component that acts as a container for other components and arranges them in a horizontal row. Also see The Terrace 55.
Tower	A Caplin Trader window component that acts as a container for other components and arranges them in a vertical column. Also see The Tower 56.
Panel	A rectangular area in the Caplin Trader user interface in which internal HTML content may be loaded. Also see The Panel 53.
XML	Extensible Markup Language, XML files are used to specify the visual layout of the Caplin Trader Application.

Index

CAPLIN THE REAL TIME WEB COMPANY

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