

CAPLIN

StreamLink For Flex 5.0

Quick Start Guide

April 2010

CONFIDENTIAL

Contents

1	Preface.....	1
1.1	What this document contains.....	1
	About Caplin document formats	1
1.2	Who should read this document.....	1
1.3	Related documents.....	2
1.4	Typographical conventions.....	2
1.5	Feedback.....	3
1.6	Acknowledgments.....	3
2	The StreamLink for Flex Evaluation Kit.....	4
2.1	About Adobe Flex.....	4
2.2	About Caplin StreamLink.....	4
2.3	About Caplin Xaqua.....	4
3	Architecture.....	5
3.1	The HTML Wrapper Page.....	6
4	Getting Started with StreamLink for Flex.....	7
4.1	Conventions used in this Guide.....	7
4.2	Prerequisites.....	7
4.3	Installing the SL4F Kit.....	9
4.4	Setting the Web Server Context.....	9
4.5	Creating the StreamLink for Flex project.....	10
4.6	Configuring the example application.....	15
4.7	Building the example application.....	16
4.8	Setting the launch configuration.....	17
4.9	Running the example application.....	20
4.10	What the example application does.....	21
5	Troubleshooting.....	22
5.1	Problem: nothing happens when I click Connect.....	22
5.2	Problem: I can't login to Liberator.....	23
5.3	Problem: I see a fatal SL4B error dialog when I try to load the page.....	23
5.4	Problem: I can login but when I request data nothing happens.....	23
6	Starting Liberator and the DemoSource.....	24
7	Further Information.....	25

8 **Glossary of terms and acronyms..... 26**

1 Preface

1.1 What this document contains

This document describes how to get started with Caplin's StreamLink for Flex (SL4F), version 5.0. It describes how to install the SL4F evaluation kit, and how to build and run an example Rich Internet Application using source code that is supplied with the kit.

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.

For the best reading experience

On the machine where your browser or PDF reader runs, install the following Microsoft Windows® fonts: Arial, Courier New, Times New Roman, Tahoma. You must have a suitable Microsoft license to use these fonts.

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 developers who want to use StreamLink for Flex to create real-time Rich Internet Applications that connect to Caplin Xaqua.

1.3 Related documents

◆ Caplin StreamLink for Flex API Reference

Reference documentation for the StreamLink for Flex API.

◆ Caplin StreamLink 5.0 Overview

A technical overview of Caplin StreamLink version 5.0.

◆ Caplin Xaqua Overview

A business and technical overview of Caplin Xaqua.

◆ Caplin StreamLink Configuration XML Reference

Describes the XML-based configuration of StreamLink 5.0 and above

◆ Caplin Liberator Administration Guide

Describes the Caplin Liberator server and its place within Caplin Xaqua. Explains how to install, configure, and manage the Liberator. Includes configuration reference information, and a list of Liberator's log and debug messages.

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.

Information about the applicability of a section is enclosed in a box like this.
For example: "This section only applies to version 1.3 of the product."

1.5 Feedback

Customer feedback can only improve the quality of our product documentation, and we would welcome any comments, criticisms or suggestions you may have regarding this document.

Visit our feedback web page at <https://support.caplin.com/documentfeedback/>.

1.6 Acknowledgments

Adobe Reader, *Adobe Flash*, *Adobe Flex*, and *ActionScript* are registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

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

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

JavaScript is a trademark of Sun Microsystems, Inc. in the U.S. or other countries.

2 The StreamLink for Flex Evaluation Kit

The StreamLink for Flex (SL4F) evaluation kit contains the source code, libraries, configuration files, and documentation that allow you to build an example Caplin Xaqua client application using Caplin StreamLink libraries and the Adobe Flex framework. A Caplin Xaqua client application that is built using Caplin StreamLink libraries and the Adobe Flex framework is known as a StreamLink for Flex application.

In this document you will learn how to:

- ◆ Install the SL4F evaluation kit on a Windows computer.
- ◆ Create a project and build an example application using Adobe Flex Builder™ 3. The source code for the example application is supplied with the kit, but Adobe Flex Builder is not supplied with the kit.
- ◆ Configure the example application to connect to Caplin Xaqua.
- ◆ Run the example application in a web browser.

2.1 About Adobe Flex

The following description of Adobe Flex® is taken from the Adobe Flex product description at <http://www.adobe.com/products/flex/overview/>.

Flex is a free, open source framework for building and maintaining web applications that deploy consistently on all major browsers, desktops, and operating systems. MXML, a declarative XML-based language, is used to describe user interface (UI) layout and behaviours, and ActionScript® 3, an object-oriented programming language, is used to create client logic. Flex also includes a component library comprising over 100 of the more commonly used UI elements for creating Rich Internet Applications (RIAs).

RIAs created with Flex can run in the browser using Adobe Flash® Player software or on the desktop on Adobe AIR™, a cross-operating system runtime. This enables Flex applications to run consistently across all major browsers and on the desktop.

2.2 About Caplin StreamLink

Caplin StreamLink is an API and underlying code library that allows Caplin Xaqua client applications to exchange real-time financial data and trade messages with Caplin Xaqua (see [About Caplin Xaqua](#)^[4]). StreamLink for Flex is an implementation of the Caplin StreamLink API that leverages the data distribution power of the Caplin Xaqua platform and the ease of Adobe Flex User Interface development, allowing banks to produce compelling real-time Rich Internet Applications.

For further information about Caplin StreamLink, see the **StreamLink Overview** document.

2.3 About Caplin Xaqua

Caplin Xaqua is a framework for building single-dealer platforms that enable banks to deliver multi-product trading direct to client desktops. Caplin Xaqua can be configured to connect to client applications that are implemented in one of several technologies, including StreamLink for Flex and Caplin Trader applications.

For further information about Caplin Xaqua, see the **Caplin Xaqua Overview**.

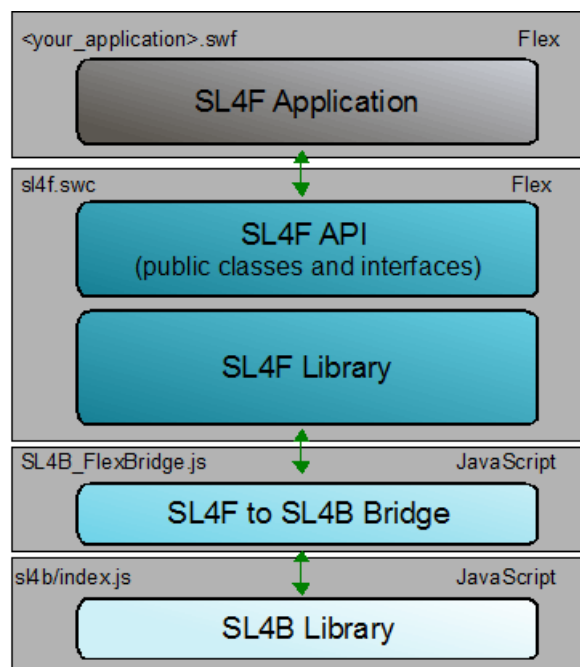
For further information about Caplin Trader, see the **Caplin Trader Overview**.

3 Architecture

A StreamLink for Flex application has four software components:

- ◆ The **client application**: A StreamLink for Flex application that uses the StreamLink for Flex API and library to access real-time data from Caplin Xaqua. The client application is typically written in ActionScript and MXML, and compiled into an *.swf* file that can be played by Adobe Flash Player. The *.swf* file can either be embedded in an HTML page for display in a browser, or run on the desktop as an Adobe AIR application.
- ◆ The **StreamLink for Flex library**: A Caplin Library that provides the API for accessing real-time data from Caplin Xaqua. StreamLink for Flex applications must include the StreamLink for Flex library in their build path in order to use StreamLink. The library is included in the kit as *SL4F.swc*.
- ◆ **StreamLink for Browsers (SL4B)**: StreamLink for Flex uses StreamLink for Browsers, Caplin's mature JavaScript library, to communicate with Caplin Xaqua. StreamLink for Browsers is not included in the StreamLink for Flex kit as it is usually hosted by Liberator. StreamLink for Flex applications are not required to communicate with SL4B directly, however you will need to configure SL4B to work with your StreamLink for Flex application. For this reason SL4B documentation is included in the *doc/SL4B_API_Documentation* folder of the kit.
- ◆ The **Flex to SL4B Bridge**: A JavaScript file that enables the StreamLink for Flex library to communicate with StreamLink for Browsers. This file is included in the kit as *SL4B_FlexBridge.js*.

The interaction between the four software components is shown in the diagram below, which illustrates how data flows between the components.

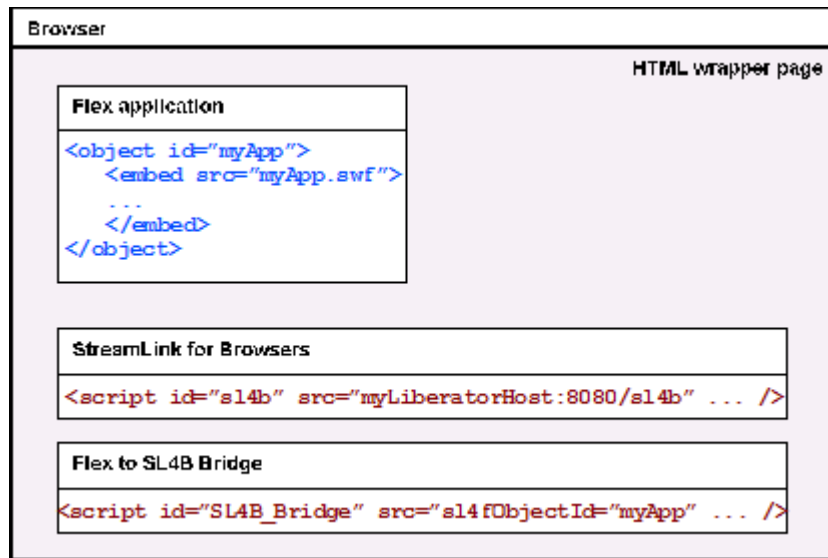


Architecture of a StreamLink for Flex application

3.1 The HTML Wrapper Page

A Flex project typically consists of ActionScript and MXML source files that compile to an SWF file, and an HTML wrapper page that embeds the SWF file in an `<object>` tag. The HTML wrapper page must also include the URL of the SL4B library and Adobe Flex to SL4B bridge inside `<script>` tags.

The diagram below shows how these components are specified in an HTML wrapper page.



Constructing the HTML Wrapper Page

The HTML wrapper page typically includes supplementary logic, for example JavaScript code that detects when a browser does not have Adobe Flash Player installed, prompting the user to visit a page to download the player.

4 Getting Started with StreamLink for Flex

The quickest way to get started with StreamLink for Flex is to build and run the example application supplied in the kit. Some configuration is also required, but this will help you understand how StreamLink for Flex works and how you can use the SL4F library in your own applications.

4.1 Conventions used in this Guide

This guide uses the following conventions for convenience:

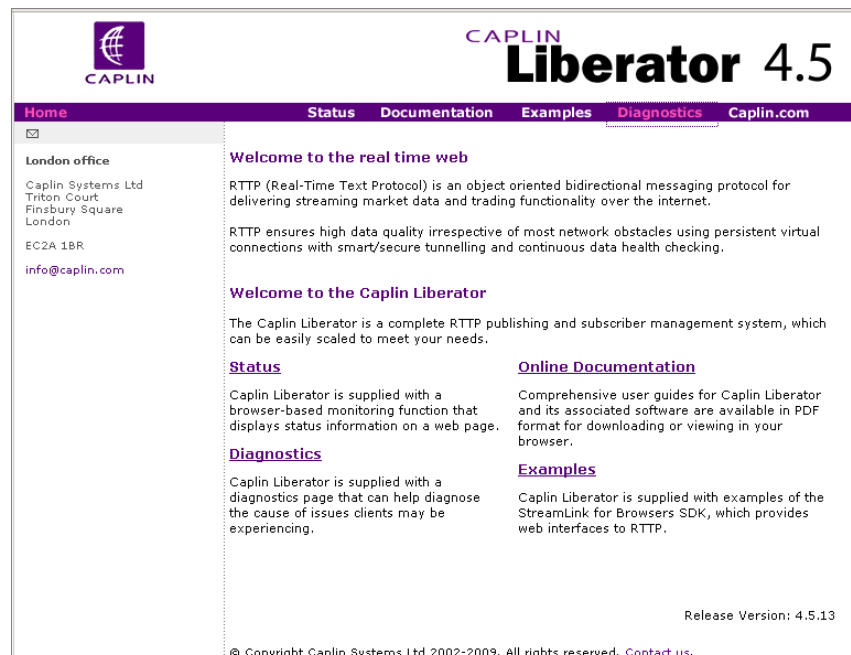
- ◆ The name of the server hosting Liberator is **liberatorserver** and the HTTP port is **8080**.
- ◆ The name of the server hosting the StreamLink for Flex application is **localhost** and the HTTP port is **8080**.
- ◆ Your domain is called **mydomain**. Due to cross-domain scripting restrictions, StreamLink for Flex can only connect to Liberators that are in the same domain as StreamLink for Flex.

When reading this guide, please substitute these names with your own server host names, port numbers, and domain name.

4.2 Prerequisites

The following steps must be completed before you start to use the StreamLink for Flex evaluation kit:

- ◆ Ensure that Caplin Liberator is running and that you have started the DemoSource that is included with Liberator (see [Starting Liberator and the DemoDataSource](#))^[24]. The example application requests objects that are provided to Liberator by the DemoSource. To confirm that Liberator and the DemoSource are set up correctly, you can run through these steps:
 - ◆ Open a browser and verify that the Liberator menu page is displayed when you visit `http://liberatorserver.mydomain.com:8080`.



- ◆ Click the “Status” link and verify that the status of the demosvc is OK or LIMITED.

Data Services			
Name:	demosvc	Name:	ExampleService
Status:	LIMITED	Status:	DOWN
Last Change:	Mar 12 15:43:11	Last Change:	Mar 12 15:42:54

- ◆ Click on the “Diagnostics” link, then the “Object browsing tool” link. In the “Enter Object:” box type /DEMO/MSFT and click the OK button.
- ◆ Verify that data updates are displayed on the page.

Request Objects				
Enter Object:	/DEMO/MSFT			
	OK Cancel			
Object Type	Record			
Parent Dir	/DEMO			
Type	211			
dFullName	Microsoft			
dTime	11:02	11:02	11:02	11:02
dLast	29.542	29.989	30.720	30.765
dNet.Chng	-0.298	0.149	0.880	0.925
dVolumeAcc	17431000	17428000	17416000	17395000
dBESTBid	30.077	30.588	30.442	
dBESTAsk	31.304	31.837	31.685	
SID	demosvc			

- ◆ Ensure that you have a web server running on the same machine on which you install the SL4F kit. StreamLink for Flex must make a connection to Liberator, which is not possible if you navigate to the HTML wrapper page using a “file:///” URL.
- ◆ Ensure that you have a StreamLink for Flex development environment. This guide assumes that you are using Adobe Flex Builder 3.0. Please note that menu options and user interface elements may differ slightly if you are using a different version of Flex Builder.

Tip See [Troubleshooting](#) if you cannot complete these steps.

4.3 Installing the SL4F Kit

Extract the contents of the StreamLink for Flex kit to the installation directory of your choice on your Windows machine. In this guide the installation directory is assumed to be *C:\SL4F*. The following files and directories will be installed:

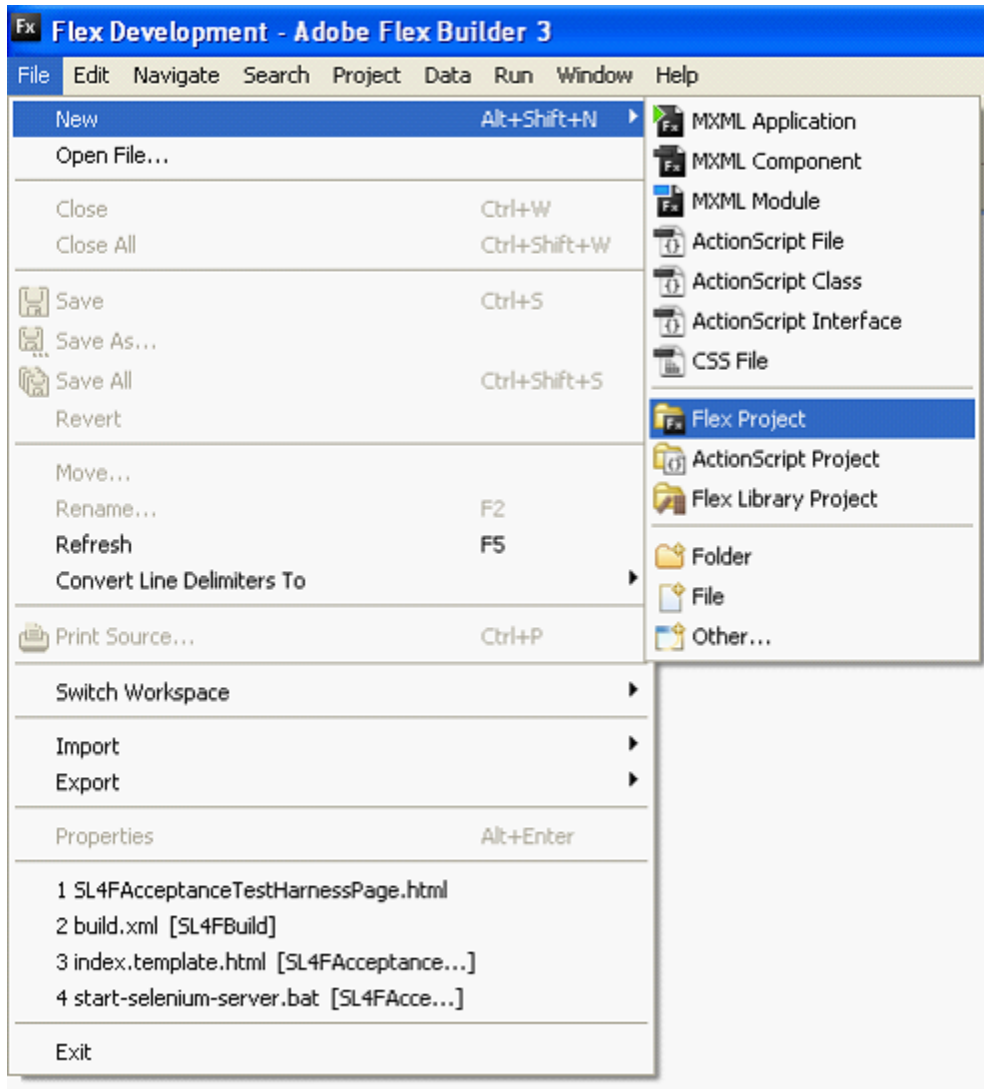
- ◆ *\SL4F.swc* – the StreamLink for Flex library
- ◆ *\SL4B_FlexBridge.js* – the Flex to StreamLink for Browsers bridge
- ◆ *\RELEASENOTE.txt* – the release note
- ◆ *\doc* – directory containing:
 - SL4B_API_Documentation* – the StreamLink for Browsers API reference.
 - SL4F_API_Documentation* – the StreamLink for Flex API reference.
 - StreamLinkOverview50* – a high level overview of Caplin StreamLink.
 - SL4FQuickStartGuide.pdf* – this document.
- ◆ *\example* – directory containing:
 - html-template-original* – the HTML wrapper page template and associated files
 - src* – directory containing:
 - com* – folder structure containing all ActionScript source files.
 - SL4FExamples.mxml* – the example application.

4.4 Setting the Web Server Context

A web server context must be set up that maps “/SL4F” to the installation directory “C:\SL4F”. This means that after you have installed the kit and set the web server context, you should be able to access the contents of the kit in a browser by navigating to <http://localhost.mydomain.com/SL4F>.

4.5 Creating the StreamLink for Flex project

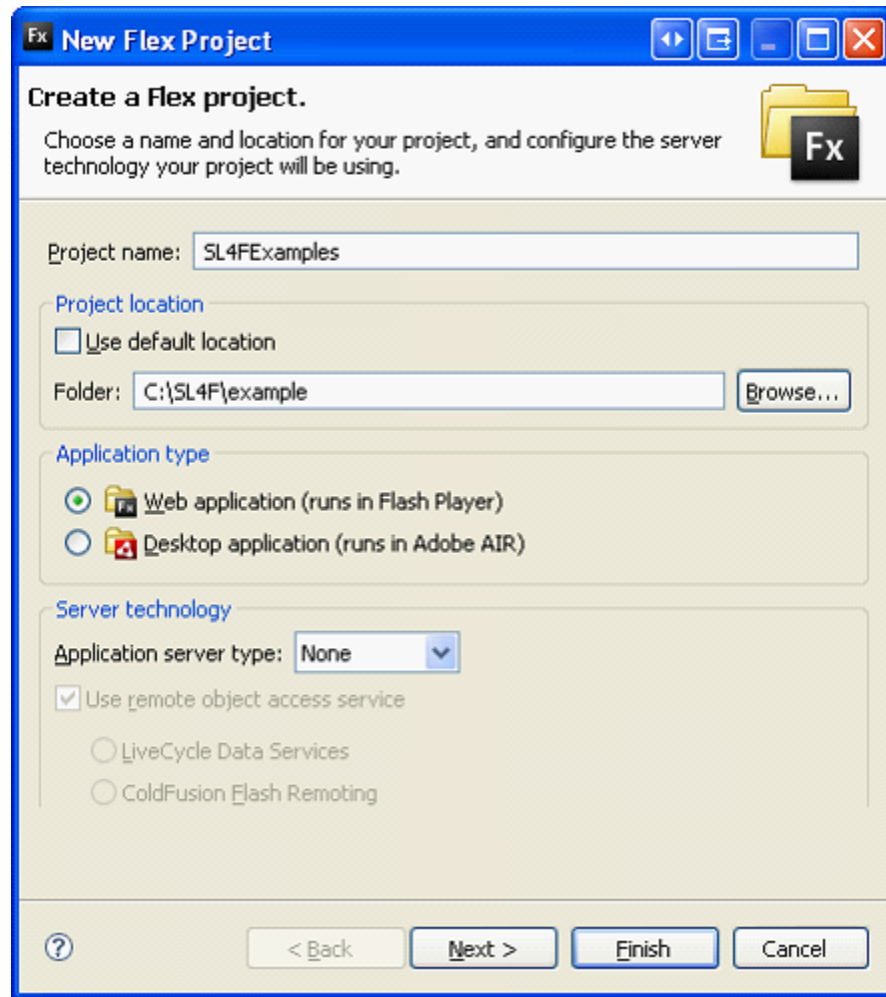
Open Adobe Flex Builder 3.0 and select File -> New -> Flex Project from the menu bar.



Creating a new project in Flex Builder

In the New Flex Project dialog that opens, complete the following steps:

- In the **Project Name** box, enter SL4FExamples.
- Uncheck **Use default location**.
- At the right hand side of the **Folder** box, browse to C:\SL4F\example and then click OK.

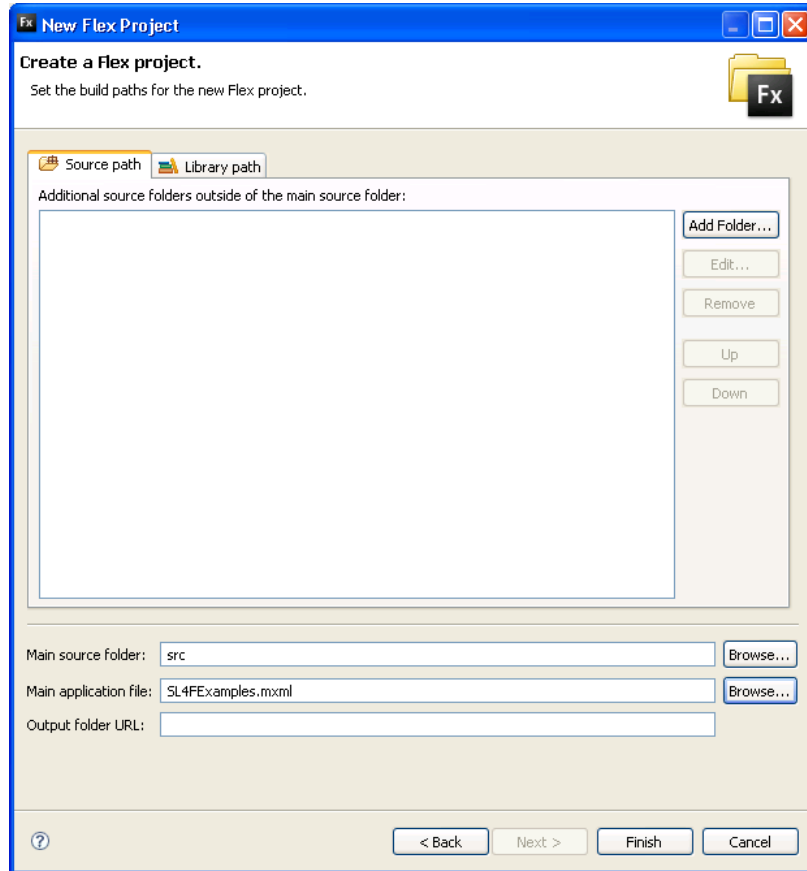


The New Flex Project dialog

- Click Next, and then click Next again in the dialog that opens (leaving the output folder on the default value).

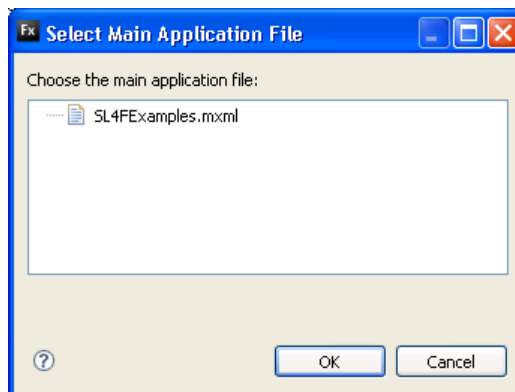
In the next dialog that opens you configure the Flex build paths for your project. The first task is to set the main application file:

- Click the **Source path** tab.



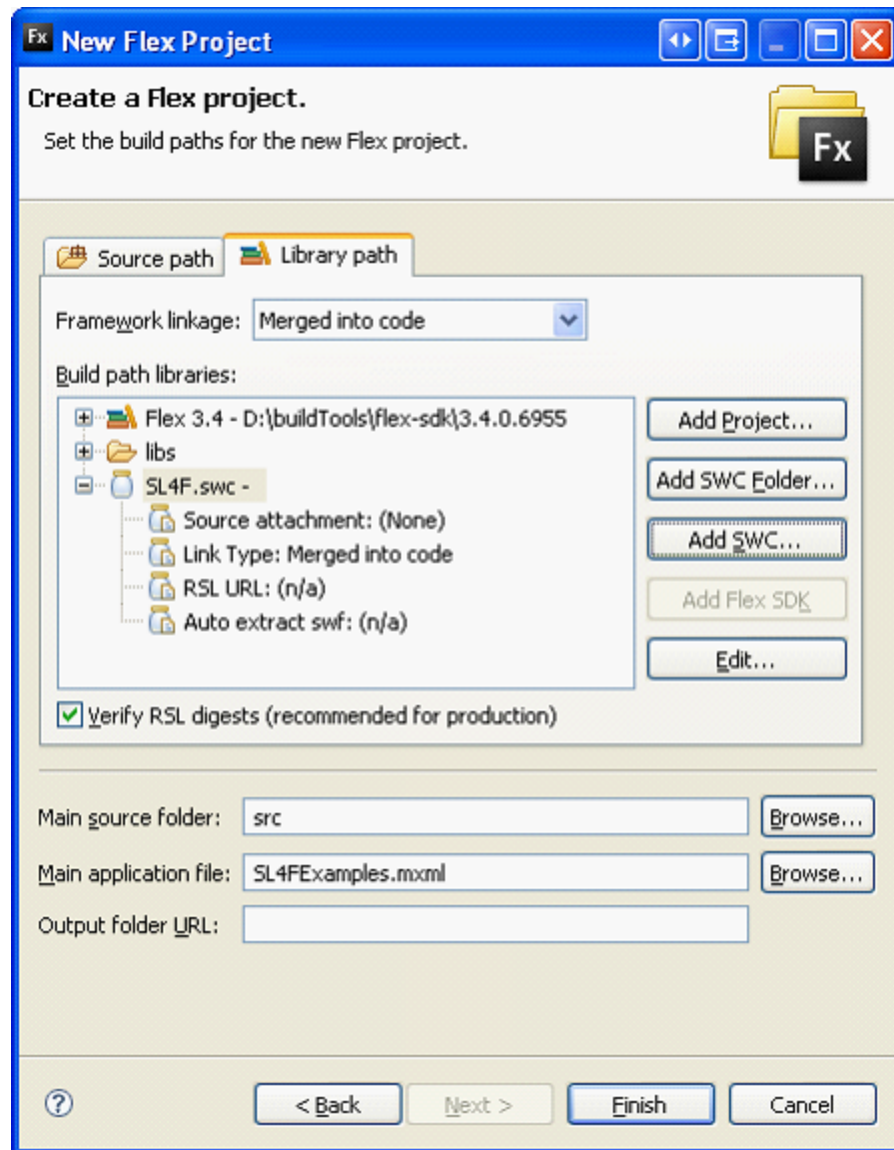
The Source path tab

- At the right hand side of the **Main application file** box, click Browse..., select *SL4FExamples.mxml* in the dialog that opens, and then click OK.



You will now add the StreamLink for Flex library to the build path:

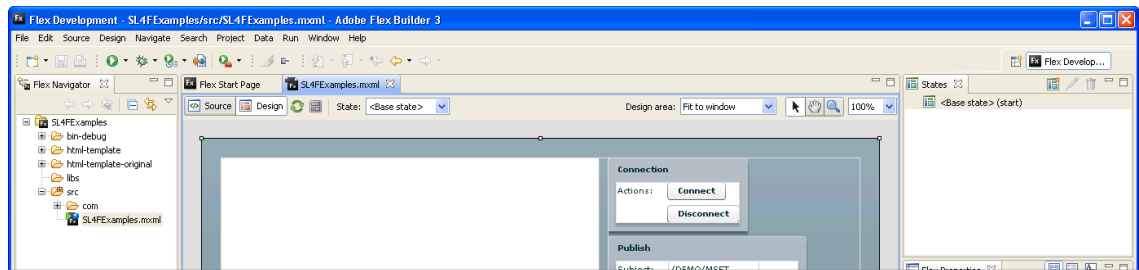
- Click the **Library path** tab.
- Click Add SWC...
- Click Browse...
- Navigate to `C:\SL4F\SL4F.swc`, click Open and then click OK.



The Library path tab

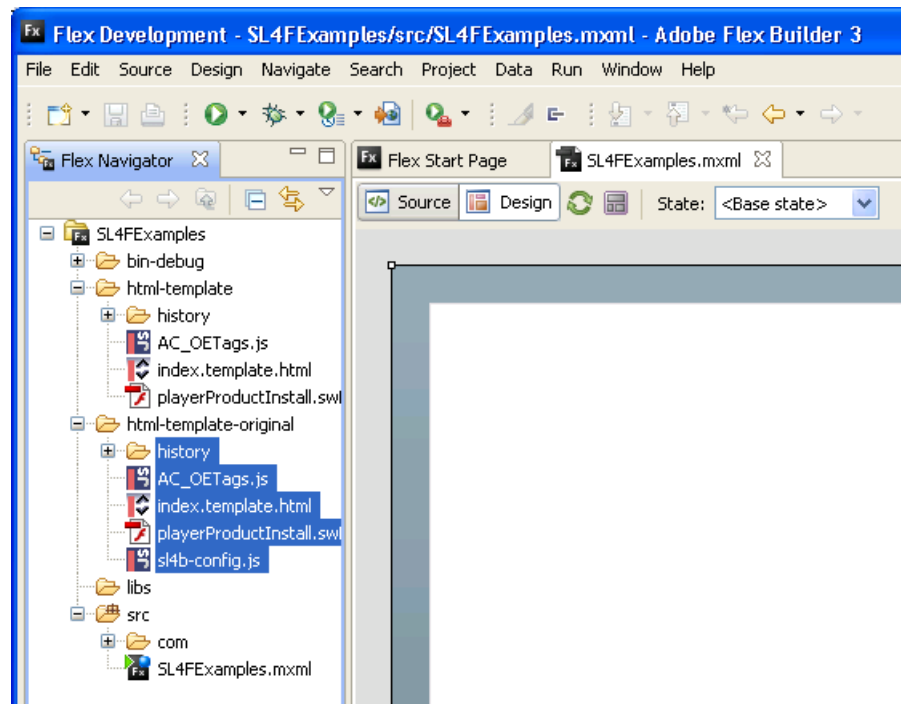
- Click Finish to complete the project set up.

Flex Builder 3.0 will display the example application in the middle of the project window (in the *SL4FExamples.mxml* project tab).



Project setup complete

- Copy the complete contents of the *html-template-original* folder (as provided in the kit) to the *html-template* folder (generated by Flex Builder).



4.6 Configuring the example application

Before building the example application, you must change the default configuration so that the application can connect to your Liberator.

Tip: To open a file in Flex Builder 3.0, double click the file in the File Navigator.

- Open the file `\html-template\index.template.html`.
- Scroll to the bottom of the file. You will find two `<script>` tags; one that includes StreamLink for Browsers and one that includes the Flex to SL4B bridge file.
- In the `<script id="sl4b"...>` line, change the `src` attribute so that it points to your Liberator. For example `src=http://liberatorserver.mydomain.com:8080/sl4b/`.
- You may want to make other changes to the `<script id="sl4b"...>` line, if for example you want to use KeyMaster to authenticate your session. For more information, please see the **Configuring SL4B** section of the **StreamLink for Browsers API reference**.
- There should be no need to change the `<script id="SL4B_FlexBridge"...>` line, however you may want to change the `src` attribute if the relative path to the `SL4B_FlexBridge.js` file is not correct.
- Save any changes you make.

Now that the `index.template.html` file is correct you will also need to edit the StreamLink configuration file.

- Open the file `\html-template\sl4b-config.js`.
- Change the `commondomain` attribute to your common domain, for example `"mydomain.com"`.
- Change the `serverurl` attribute to the address of your Liberator, for example `"http://liberatorserver.mydomain.com:8080"`.
- Save any changes you make.

Finally, check that the credentials used to login to Liberator are valid.

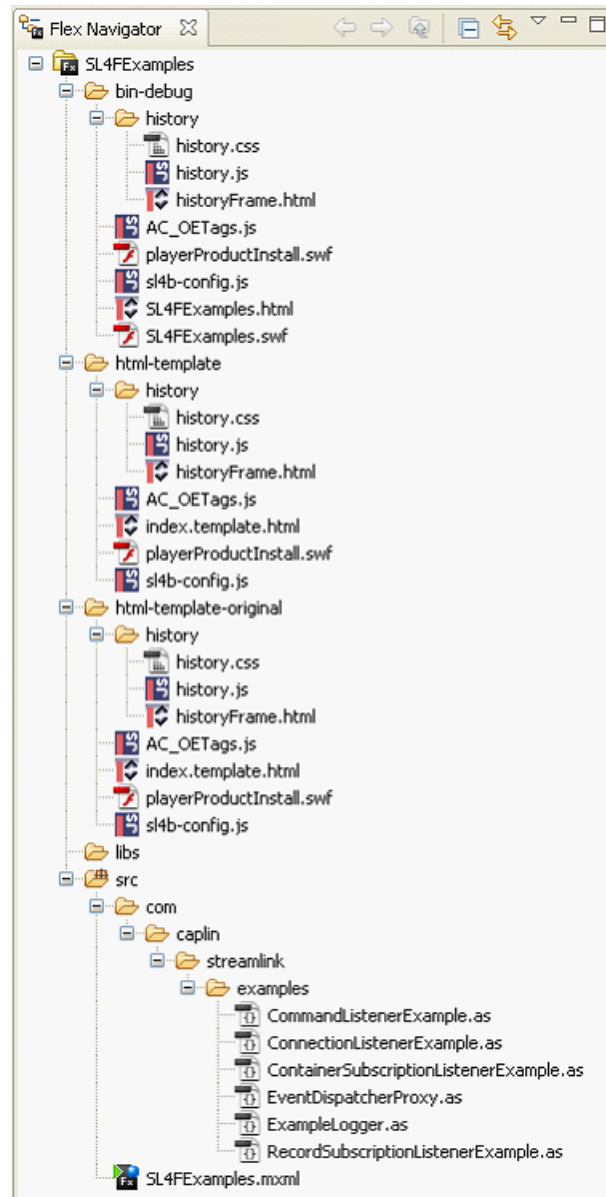
- Open the file `\src\SL4FExamples.mxml`.
- Click the **Source** tab and find this line:

```
var credentialsProvider:ICredentialsProvider =  
    new PasswordCredentialsProvider("admin", "admin");
```
- If necessary, edit the two parameters to provide credentials that will be accepted by Liberator. The first parameter is the username and the second is the password. You may need to consult your Liberator administrator to find out which credentials are appropriate.
- Save any changes you make.

4.7 Building the example application

- Select Clean... on the Project menu to build the project. The *bin-debug* folder will be populated with files.
- Ensure there were no compilation problems.

The build process is now complete and your workspace should look like this:



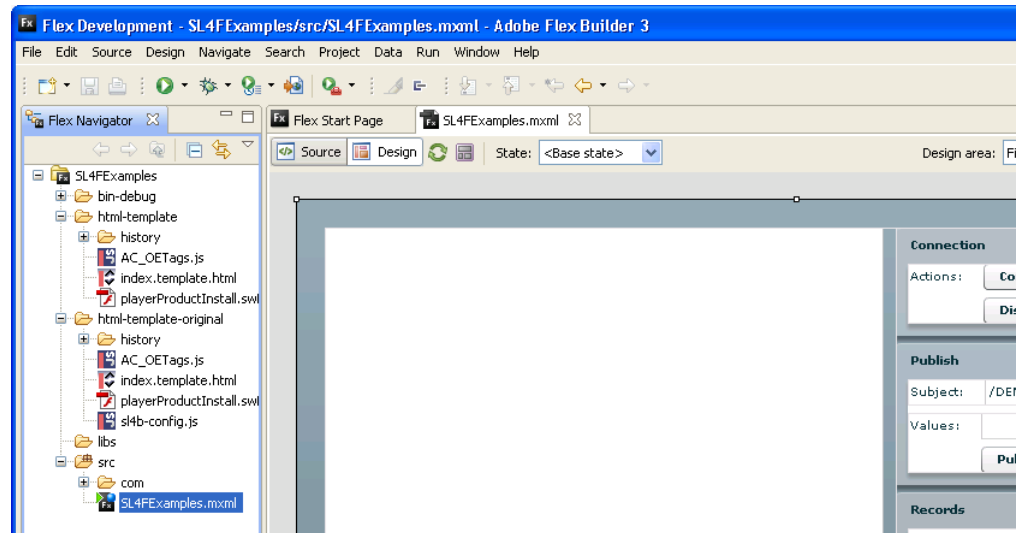
The example project directory structure

4.8 Setting the launch configuration

By default Flex Builder will launch the example application using the local path to the HTML wrapper file. This will cause the browser to display the page using a `file://` URL. In order for StreamLink for Flex to work, the page must be served by a web server and accessed using an `http://` URL.

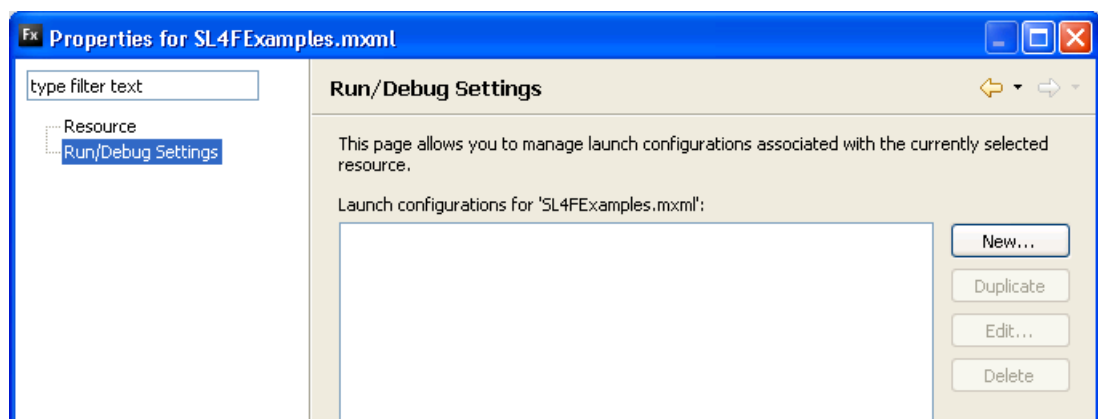
You can change the URL that Flex Builder uses by following these steps.

- Right click on the file `\src\SL4FExamples.mxml` and select Properties.



Right click file and select Properties

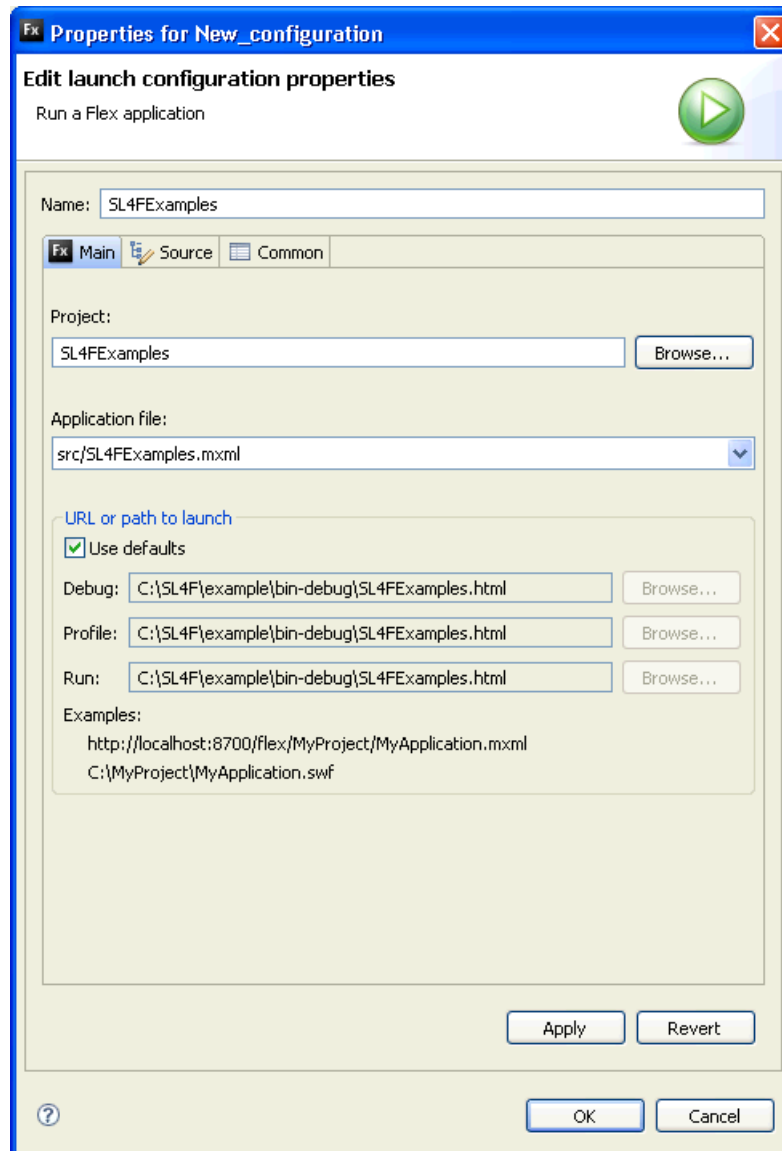
- Click on Run/Debug Settings in the left pane.



Run/Debug settings

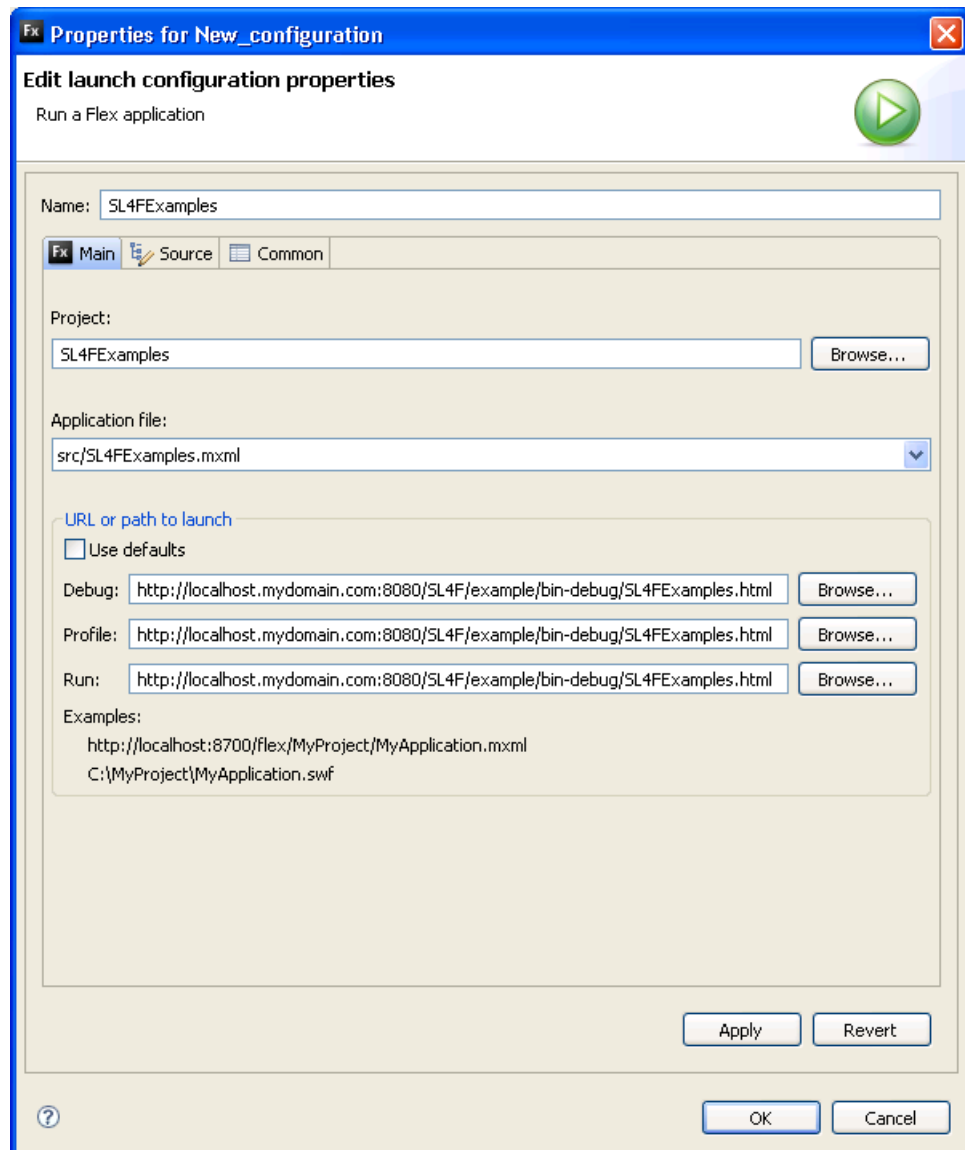
- Click New..., select Flex Application, and then click OK.

- Type SL4FExamples in the Name: box of the dialog that opens.



Edit launch properties

- In the [URL or path to launch](#) area, uncheck Use defaults.
- In the Debug, Profile and Run boxes, type the URL of the *bin-debug/SL4FExamples.html* file as it would be served by your local web server. For example, if your web server has a context that maps *C:\SL4F* to */SL4F*, then the values in the text boxes would look like this:



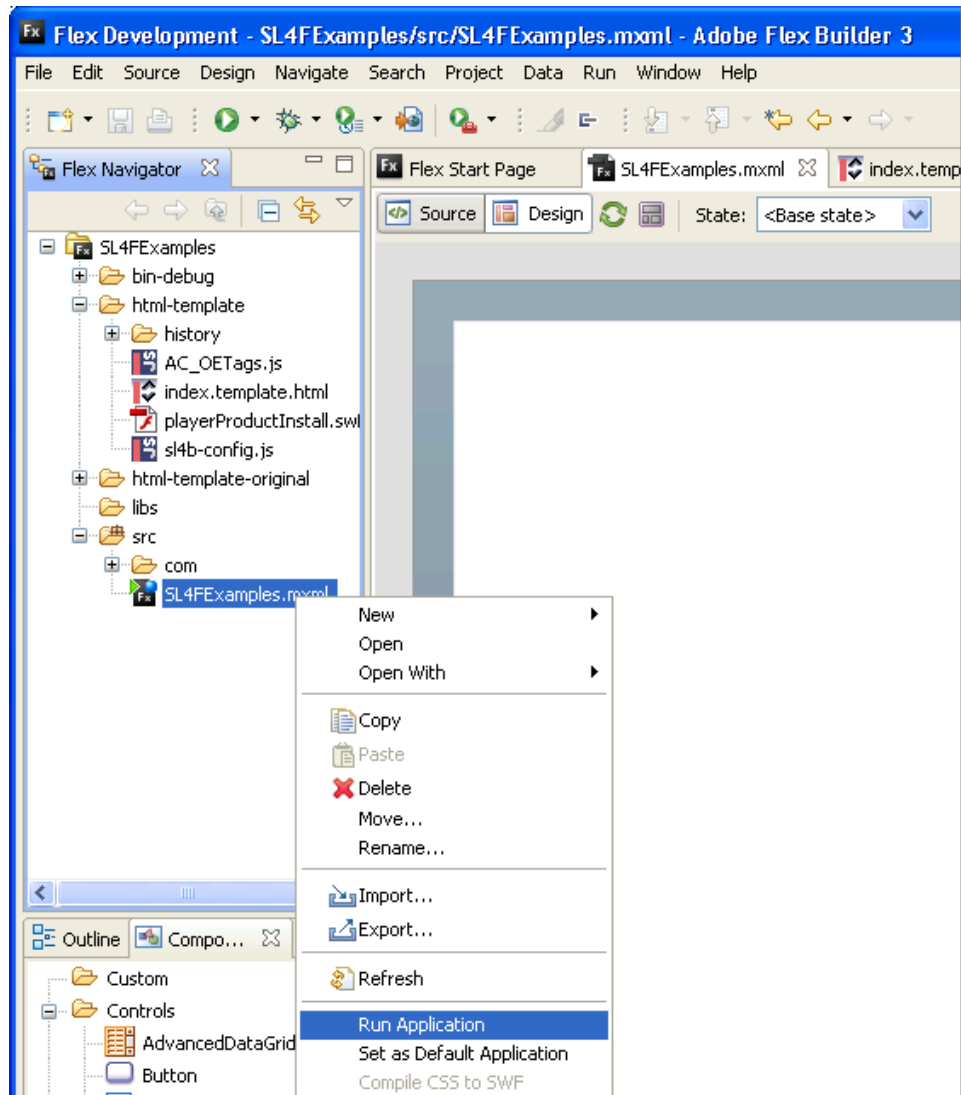
The launch configuration dialog

- Click OK to apply the changes (the launch configuration dialog will close).
- Click OK to close the **Run/Debug Settings** dialog.

4.9 Running the example application

You can run the example application from Flex Builder.

- Right click `\src\SL4FExamples.mxml` in Flex Navigator and select Run Application.



Right click and select Run Application

The example StreamLink for Flex application will open in a web browser.

4.10 What the example application does

The example application has a large log window that displays log messages from StreamLink for Flex, and several control panels to the right of the log window that allow you to request and publish data from Caplin Xaqua.

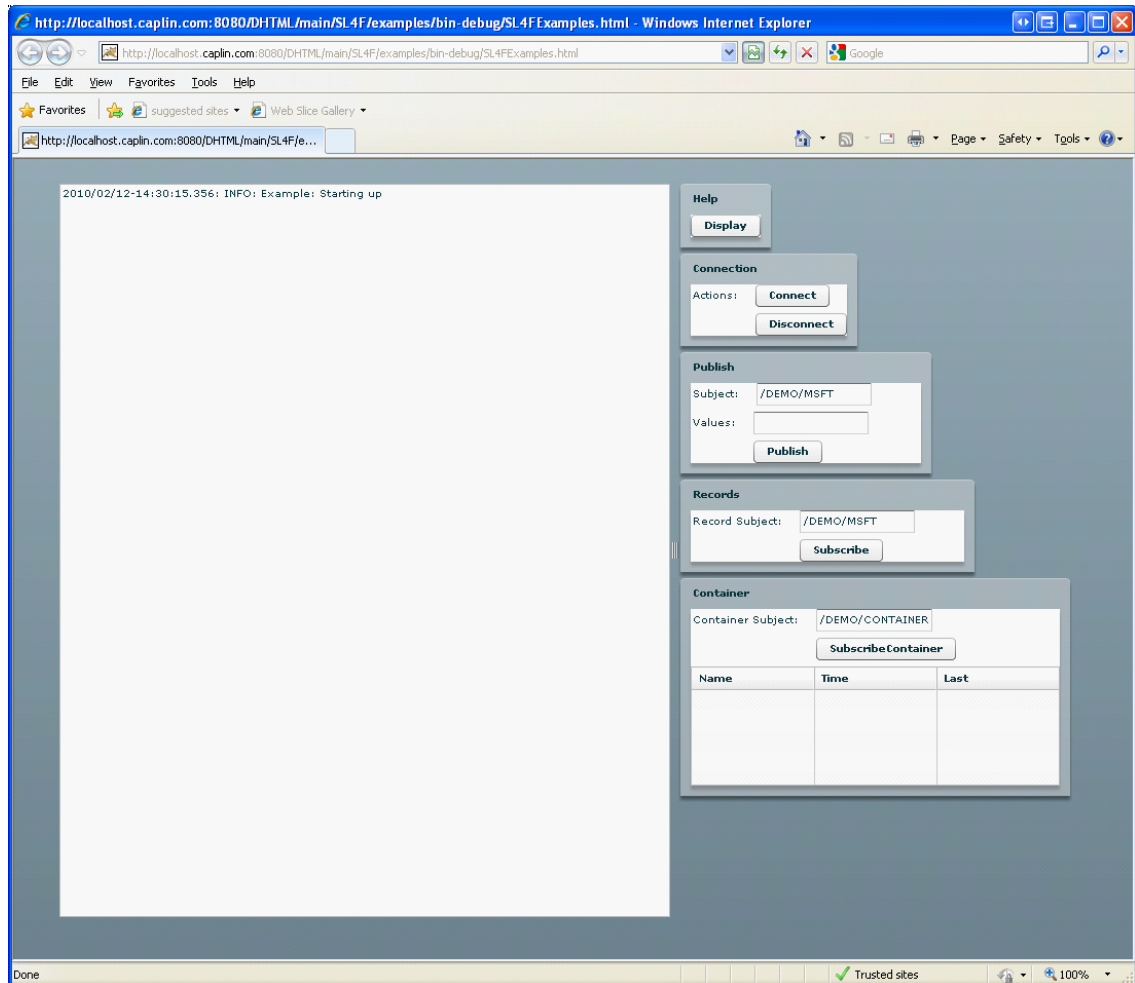


Fig 9. The example application

- For further information on how to use the application, click the Display button in the Help panel.

5 Troubleshooting

The following sections describe common problems that you may encounter while running the example application and explain how to resolve them.

5.1 Problem: nothing happens when I click Connect

This will happen if either StreamLink for Browsers or the Flex to SL4B bridge have not loaded.

- Check that the *html-template\index.template.html* contains the `<script id="sl4b">` tag and that the `src` element points to the location of the *SL4B index.js* file.
- Ensure that the Liberator that serves SL4B is running.
- Check that the *html-template\index.template.html* contains the `<script id="SL4B_FlexBridge">` tag and that the `src` element points to the *SL4B_FlexBridge.js* file.

In Firefox you can use the Net panel of the Firebug extension to see if the relevant files have been loaded. For example if the Liberator that serves SL4B is down then Firebug will show this output:

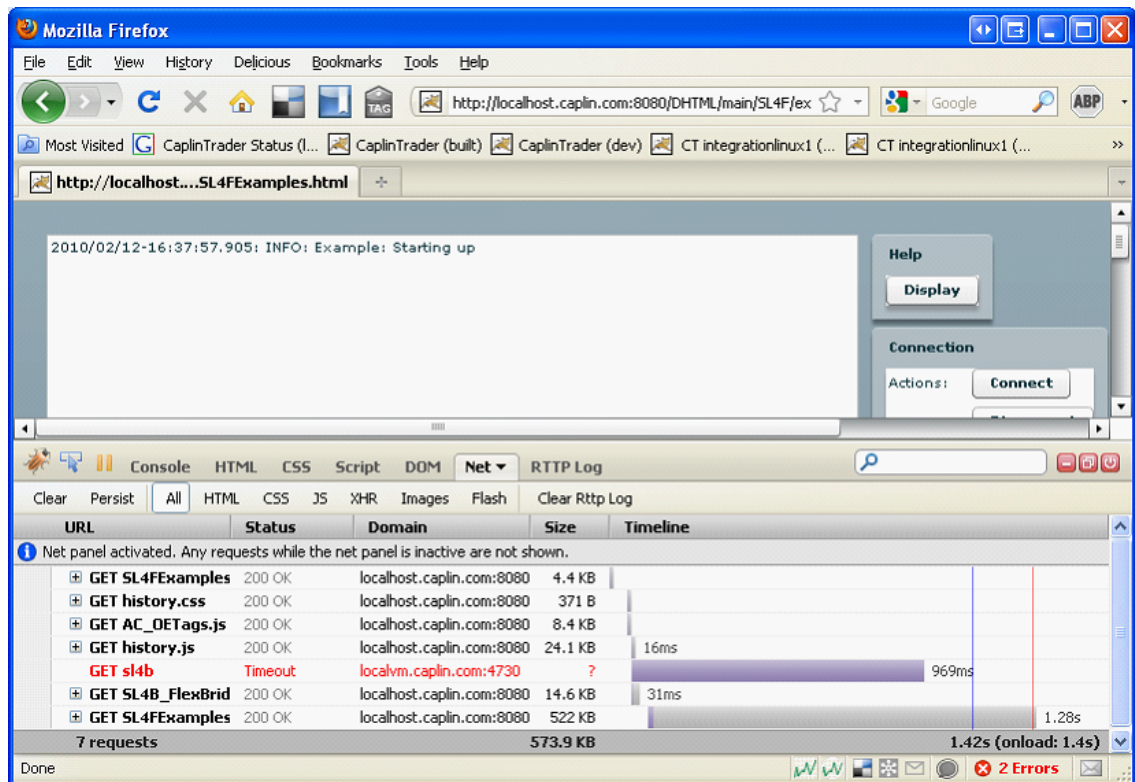


Fig 10. Failure to load SL4B

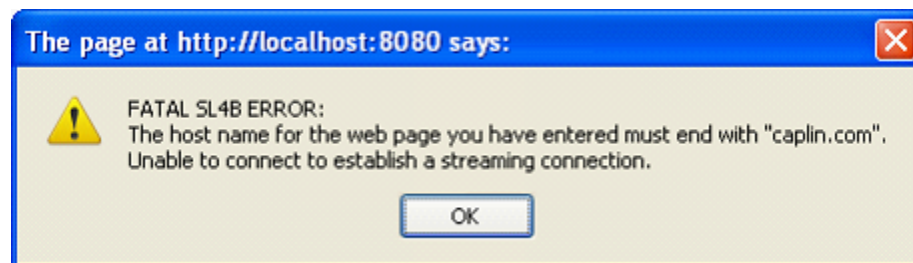
5.2 Problem: I can't login to Liberator

This occurs if the user credentials in the file *SL4FExamples.mxml* are rejected or if the example application is trying to connect to the wrong Liberator (see [Configuring the example application](#)^[15]).

- In the file *html-template\sl4b-config.js* ensure that the value of the `serverurl` attribute is correctly set to the Liberator you want to connect to.
- In the file *example\src\SL4FExamples.mxml* ensure that the `PasswordCredentialsProvider` is using the correct username and password.

5.3 Problem: I see a fatal SL4B error dialog when I try to load the page

This error is displayed if the common domain is incorrectly configured.



- Ensure that the machine that StreamLink for Flex is running on and the machine hosting Liberator are in the same domain.
- In the *html-template\sl4b-config.js* file ensure that the `commondomain` attribute is set correctly.
- When accessing the example page in your browser, ensure that you enter the full server name including domain in the address bar. For example use this:

```
http://localhost.mydomain.com:8080/SL4F/examples/bin-debug/SL4FExamples.html
```

not this:

```
http://localhost:8080/SL4F/examples/bin-debug/SL4FExamples.html
```

5.4 Problem: I can login but when I request data nothing happens

This will happen if the DemoSource that provides example data is not connected to the Liberator.

- Verify that you have started the DemoSource that is provided with Liberator.
- View the Liberator status page and ensure that the DemoSource is connected to Liberator.

6 Starting Liberator and the DemoSource

The following instructions apply if Liberator is installed as a standalone kit on a Linux machine.

- ◆ Navigate to the Liberator installation directory.
- ◆ To start Liberator, run the command `etc/rtttd start`
- ◆ To start the DemoSource, run the command `etc/demosrc start`

For complete instructions on starting Liberator and any supplied DataSources, refer to the documentation in your Liberator installation kit.

Tip: The Liberator supplied with the evaluation version of Caplin Trader is not configured to run the DemoSource. If you want to build and run the example SL4F application, you are advised to install a standalone version of Liberator.

7 Further Information

If further information is required, please see the following documents:

- ◆ **StreamLink Overview**
- ◆ **StreamLink for Flex API Reference**
- ◆ **StreamLink Configuration XML Reference**

8 Glossary of terms and acronyms

This section contains a glossary of terms, abbreviations, and acronyms relating to the StreamLink for Flex product.

Term	Definition
Adobe Flash	A multimedia platform from Adobe Systems Inc. for adding animation and interactivity to web pages.
Adobe Flash Player	A software player from Adobe Systems Inc. for viewing animations and movies. The Adobe Flash player can run Adobe Flex and Adobe Flash applications, and is available as a plugin for web browsers.
Adobe Flex	A software development kit and integrated development environment from Adobe Systems Inc. for creating Rich Internet Applications based on the Adobe Flash platform. Flex uses ActionScript code and XML-based user interface descriptions (MXML), and compiles them into binary Flash files (SWF files) that run in the Adobe Flash Player.
Adobe Flex Builder	A integrated development environment (IDE) from Adobe Systems Inc. for developing applications that use the Adobe Flex framework.
Caplin Liberator	Caplin Liberator is a real-time financial internet hub (server) that delivers trade messages and market data to and from subscribers over any network.
Caplin Trader	A Rich Internet Application framework for constructing Caplin Xaqua client applications for browser-based trading. It includes a comprehensive set of trading GUI components.
Caplin Xaqua	A framework for building single-dealer platforms that enables banks to deliver multi-product trading direct to client desktops.
Caplin Xaqua client	<p>A client desktop application that interfaces with Caplin Xaqua to deliver multi-product trading to end users. The application can be implemented in any technology that is supported by Caplin Xaqua; for example Ajax, Microsoft .NET, Microsoft Silverlight™, Adobe Flex, and Java™.</p> <p>Both the StreamLink for Flex library and the Caplin Trader framework can be used to create browser-based Caplin Xaqua client applications.</p>
Flex to SL4B bridge	A JavaScript file that enables the StreamLink for Flex library to communicate with StreamLink for Browsers .
JavaScript	A scripting language used in web browsers.
KeyMaster	<p>Caplin KeyMaster is software that integrates Caplin Xaqua with any existing web-based authentication system, so that end users or web applications do not need to explicitly log in to Caplin Liberator in addition to their normal login procedure. It implements a secure method of user authentication by means of a user credentials token that is digitally signed using public key encryption.</p> <p>KeyMaster is usually integrated with a single sign-on system.</p>
Liberator	Another name for Caplin Liberator .

Term	Definition
Rich Internet Application	A web application, such as an application implemented using StreamLink for Flex , that has the features and functions of a desktop application, but which does not need to be installed on the client platform. Rich Internet Applications typically run in web browsers.
RTTP	<u>Real Time Text Protocol</u> . Caplin's protocol for streaming real-time financial data from Caplin Liberator servers to client applications, and for transmitting trade messages between clients and Liberator in both directions.
Single sign-on	A user authentication process in which a user supplies just one set of user credentials (such as a user name and password). The user can then access multiple applications and systems without being prompted for credentials again.
SL4B	<u>StreamLink for (4) Browsers</u>
SL4F	<u>StreamLink for (4) Flex</u>
StreamLink	A Caplin library that connects client applications to Liberator using the RTTP protocol.
StreamLink for Browsers	StreamLink for Browsers is a JavaScript implementation of StreamLink that runs in web browsers. It allows browser-based applications to communicate with Caplin Liberator.
StreamLink for Flex	A Caplin library that allows Adobe Flex applications to access real-time data from Caplin Liberator . StreamLink for Flex uses StreamLink for Browsers to communicate with Liberator.
StreamLink for Flex application	A Caplin Xaqua client application that is built using StreamLink for Flex libraries and the Adobe Flex framework.
User credentials	Information used to authenticate a user; for example, a user name and password.

Contact Us

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

The information contained in this publication is subject to UK, US and international copyright laws and treaties and all rights are reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the written authorization of an Officer of Caplin Systems Limited.

Various Caplin technologies described in this document are the subject of patent applications. All trademarks, company names, logos and service marks/names ("Marks") displayed in this publication are the property of Caplin or other third parties and may be registered trademarks. You are not permitted to use any Mark without the prior written consent of Caplin or the owner of that Mark.

This publication is provided "as is" without warranty of any kind, either express or implied, including, but not limited to, warranties of merchantability, fitness for a particular purpose, or non-infringement.

This publication could include technical inaccuracies or typographical errors and is subject to change without notice. Changes are periodically added to the information herein; these changes will be incorporated in new editions of this publication.

Caplin Systems Limited may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

This publication may contain links to third-party web sites; Caplin Systems Limited is not responsible for the content of such sites.