CAPLIN

Caplin Platform 6.0

Deployment Framework Overview

December 2012

Contents

1 Preface

1.1 What this document contains

This document gives an overview of the Caplin Platform Deployment Framework. It also explains the concept of Caplin Platform blades.

About Caplin document formats

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

1.2 Who should read this document

This document is intended for developers who want to create a trading system based on Caplin Platform components.

Before reading this document, you should be familiar with the concepts and terms that are introduced in the document **Caplin Platform Overview**.

1.3 Related documents

- **Caplin Platform: Overview**

  A business and technical overview of the Caplin Platform.
- **Best Practices For Deploying The Caplin Platform**

  Provides recommendations for deploying the Caplin Platform in a typical live environment, and discusses failover scenarios for achieving high service availability.
- **Caplin Platform: How To Use The Deployment Framework**

  Explains how to install and use the Caplin Platform Deployment Framework, and describes how Caplin Platform blades are deployed to the Framework to create a working Caplin Platform system.
- **How To Create C And Lua Platform Blades**

  Explains in detail how to create C and Lua blades for the Caplin Platform.
- **How To Create A Platform Java Blade**

  Explains in detail how to create new Java®-based Caplin Platform Blades using the Caplin Integration Suite Toolkit.

1.4 Typographical conventions

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

Type	***Uses***
Glossary Term	Items that appear in the “Glossary of terms and acronyms”
XYZ Product Overview	Document name
♦	Information bullet point

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

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

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

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

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

Apple Mac is a registered trademark of Apple Inc.

2 What is the Caplin Platform Deployment Framework?

The Caplin Platform Deployment Framework provides a framework for creating a fully functional web trading platform based on the Caplin Platform. The framework is supplied as a kit that is unpacked on the server or servers on which the Caplin Platform components will be deployed.

Using this framework has the following benefits:

- By deploying Caplin Platform components to the Framework, you can easily create a fully functional trading platform.
- You can use the same Deployment Framework to:
  - develop and test custom Caplin Platform components,
  - create a 'back end' system to test newly developed client trading applications,
  - create a production system when development and testing is complete.
- When a Caplin Platform component is updated, it is easy to replace the old component with the new component.
- The Platform supports blades, re-usable software modules containing the code and resources needed to implement a working feature of a trading system. When a new feature is required, it is easy to deploy the blade that provides the feature. For more detail about blades, see section 3.
- By running simple scripts, you can easily:
  - activate and deactivate Platform features,
  - start and stop the system.
- The Framework's well defined directory structure is organized in a modular hierarchy.

  This follows the best practices established by Caplin for managing the files and directories in trading systems that are based on the Caplin Platform. For example, it is easy to locate the configuration files that you need to edit, and those files all use the same configuration syntax.

2.1 Hardware and operating system platforms supported

The Deployment Framework can be used on the following hardware and operating system platforms:

- Linux® EL5 64 bit
- Windows® XP 32 bit (requires Cygwin – see http://www.cygwin.com/)
- Apple Mac® 64 bit

2.2 Deploying components

You can deploy two kinds of component to the Deployment Framework:

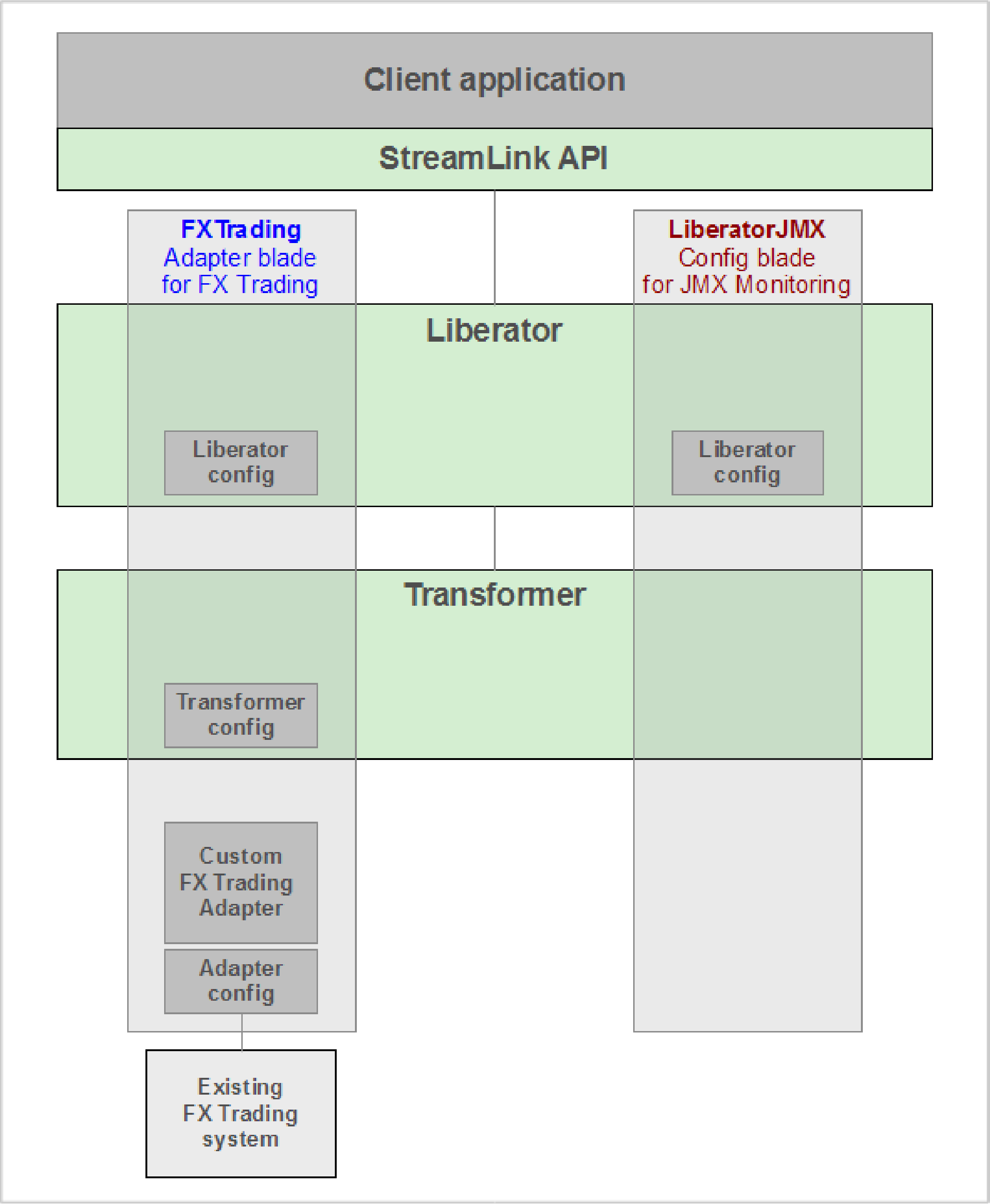
- The Caplin Platform's **core components** – **Caplin Liberator** and **Caplin Transformer**.
- **Caplin Platform blades** that provide the features of a system running on the Caplin Platform. Blades for the Caplin Platform can be supplied by Caplin Systems or you can create your own.

  For more about blades, see section 3.

For information about suitable architectures for deploying Caplin Platform components, please refer to the document **Best Practices For Deploying The Caplin Platform**.

3 Caplin Platform blades

A Caplin Platform blade is a re-usable software module containing the code and resources needed to implement a working feature of a trading system that uses the Caplin Platform. Examples of such features are FX trading and JMX monitoring, as shown in the following diagram.



Simple example of a Caplin Platform with two blades deployed

Each blade is a self contained set of files. There are three types of blade:

- **Config blade**

  This type of blade consists solely of configuration for the Platform's core components (Liberator and/or Transformer).

- **Adapter blade**

  This type of blade connects to, and supplies data to, a Liberator or Transformer. It consists of an **Integration Adapter** (an executable binary file), **DataSource** configuration, and core component configuration. Integration adapters can be written in Java or C.

- **Service blade**

  This type of blade contains a module written in C, Java^{TM}, or Lua that is to be loaded into one of the core components; for example, a permissioning auth (authorization) module that is loaded into Liberator.

Sometimes the implementation of a particular feature requires more than one blade, but a blade will never provide more than one feature. You can switch a feature on and off by activating and deactivating the blade or blades that provide it.

The Caplin Platform Deployment Framework is supplied with several built-in configuration-only blades; for a list of these, see the document **Caplin Platform: How To Use The Deployment Framework**. The Framework also provides configuration that supports failover components; for details of how this is set up, refer to the same document.

3.1 Why use blades?

Blades give you several benefits:

- Liberator and Transformer can load the configuration they need for a feature from the blade that provides that feature. There is no need to edit monolithic configuration files for these core components.
- Configuration is easy to find as it is always in the same location in each blade.
- It is easy to add new features to your system simply by deploying the relevant blades.
- Deploying a blade can be as simple as copying a kit to the Deployment Framework and running a supplied script.
- You can activate and or deactivate features by running simple scripts.

3.2 Creating custom blades

You can create your own custom blades, typically, Integration Adapter blades. For information on how to do this in Java using the **Caplin Integration Suite**, see the document **Caplin Integration Suite: How To Create A Platform Java Blade**. Alternatively, to create a custom blade in C or Lua, see the document **Caplin Platform: How To Create C And Lua Blades**.

4 Glossary of terms and acronyms

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

Term	Definition
Adapter blade	A **blade** for the **Caplin Platform** that consists of an **Integration Adapter** and associated configuration.
Blade	A re-usable software module containing the code and resources needed to implement a business feature. In this document the term blade is short for **Caplin Platform blade**.
Blade toolkit	A set of commands to create, build and export **Caplin Platform blades**.
Caplin Integration Suite (CIS)	A set of APIs and tools for creating adapters that integrate the **Caplin Platform** with external systems. It includes the **blade toolkit**.
Caplin Liberator	A financial internet hub that delivers data and messages in real time to and from subscribers over any network.
Caplin Platform	An integrated suite of software that supports the services and distribution capabilities needed for web trading. It consists of Caplin Liberator, Caplin Transformer, Caplin KeyMaster, Caplin Director, and Caplin Management Console. For more information, see the **Caplin Platform Overview**.
Caplin Platform blade	A **blade** designed for use with the **Caplin Platform**. A Caplin Platform blade can be an **Adapter blade**, **Config blade**, or **Service blade**.
Caplin Platform Deployment Framework	A configuration and deployment environment for the **Caplin Platform** that supports **Caplin Platform blades**.
Caplin Transformer	An event-driven, real-time data transformation engine optimized for web trading services. These services are implemented in Transformer Modules.
Client application	In the context of the **Caplin Platform**, a client application is any application that uses the **StreamLink API** to communicate with **Caplin Liberator**.
Config blade	A **blade** for the **Caplin Platform** that enables a feature through configuration.

Term	Definition
Core component	A Caplin Platform component that is supplied with the **Caplin Platform Deployment Framework**, but is not a blade. The core components are **Caplin Liberator** and **Caplin Transformer**.
DataSource	DataSource is the messaging infrastructure used by the **Caplin Platform** and **Integration Adapters**.
Deployment Framework	In this document, this term is short for the **Caplin Platform Deployment Framework**.
Framework	In this document, this term is short for the **Caplin Platform Deployment Framework**.
Integration Adapter	A server application that allows an external system to communicate with the **Caplin Platform**. An Integration Adapter is a **DataSource application** and is created using the **Caplin Integration Suite**.
Service blade	A blade for the **Caplin Platform** that includes a **Transformer module** or a **Liberator Auth module**.
StreamLink API	An API that allows a client application to communicate with a **Caplin Liberator**. There are StreamLink APIs for various technologies; for example, Java, JavaScript, .NET and Silverlight applications, and Objective-C running on iOS.

Contact Us

Caplin Systems Ltd

Cutlers Court

115 Houndsditch

London EC3A 7BR

Telephone: +44 20 7826 9600

www.caplin.com

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.