CAPLIN MONITORING AND MANAGEMENT 4.4
Overview

April 2007

1   Preface

1.1   What this document contains

This document describes Caplin's Management and Monitoring solution and its place in the Caplin real-time data architecture.

1.2   Who should read this document

This document is intended for people who want to understand the benefits of Caplin's Management and Monitoring solution and to get an initial overview of how it works.

1.3   Typographical conventions

This document uses the following typographical conventions to identify particular elements within the text.

Type	Use
Arial Bold	Function names and methods.
	Other sections and chapters within this document.
Arial Italic	Parameter names and other variables.
Times Italic	File names, folders and directories.
Courier	Program output and code examples.
❖	Information bullet point
■	Instruction

1.4   Feedback

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

Please email your thoughts to documentation@caplin.com.

2   Introduction

The Caplin Platform contains full enterprise-wide monitoring and management capability. This feature allows a system administrator to monitor the activity and behaviour of the Caplin Platform ranging from a high-level view of the enterprise down to granular detail of every data item in the system.

Every application built with Caplin's DataSource SDK can be enabled for monitoring and management. This includes the Caplin Liberator, Caplin Transformer and all standard DataSource products such as DataSource for Reuters, DataSource for TIB and DataSource for Comstock.

The Caplin Enterprise Console is provided as a feature-rich cross-platform GUI which is used to monitor and manage any available feature that has been exposed to the monitoring subsystem by a DataSource application.

The Caplin Monitoring and Management solution is:

❖   Modular, flexible and scalable

❖   Easily extendable through API functions

❖   Controllable using an extensible cross-platform GUI framework

❖   Controllable through multiple APIs

2.1   Further reading

Please refer to these documents for more detailed information on different aspects of the monitoring and management system:

❖   Monitoring Getting Started Guide

❖   Liberator Administration Guide

❖   Customizing the Console Guide

❖   Monitoring JMX SDK JavaDoc

❖   Monitoring Socket Interface Specification

3   How does it work?

The Caplin monitoring and management solution exposes information about the application's state through standard attributes and allows management operations to be performed. Custom-built DataSources can easily extend this behaviour by adding new items for monitoring and management using standard DataSource API functions.

These capabilities are provided by a plug-in module which is loaded at runtime by the DataSource process, allowing the monitoring and management features to be presented using a variety of mechanisms and technologies. Currently there are two modules available, access to these is dependent on the server license issued.
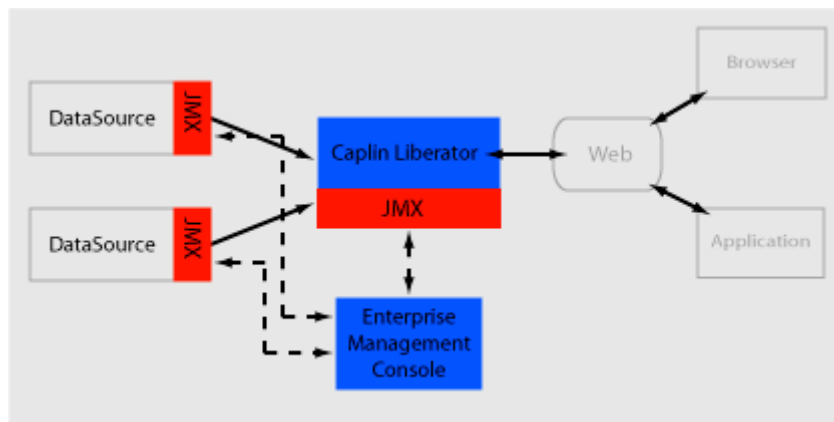

Figure 3-1: The Enterprise Management Console as part of the Caplin infrastructure

3.1   Monitoring modules

❖ JMX – Provides MBean-based monitoring and management for Java clients using a built-in JMX Server. This module allows any standard JSR160 JMX client to access the monitoring and management system. The Caplin Enterprise console uses the JMX monitoring plug-in and sample command-line applications are also provided.

❖ sockmon – Provides socket-based monitoring and management using a simple command language.
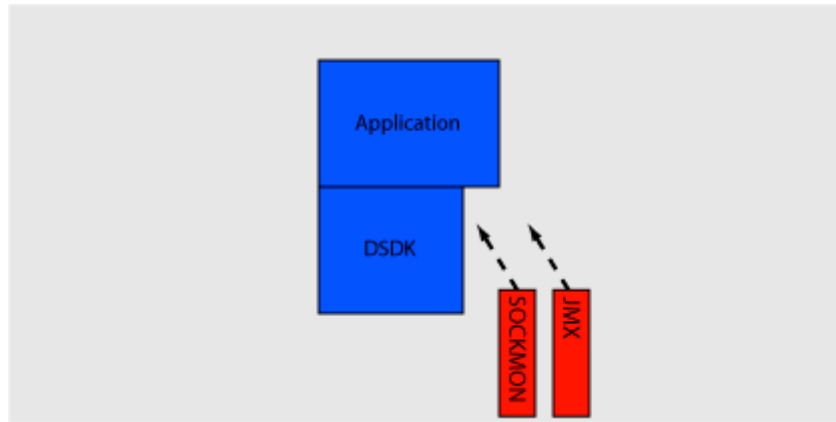
Figure 3-2: Alternative monitoring modules which can slot into an application

4   Monitoring vs Management

The Caplin Platform provides both monitoring and management functionality through the same interface. For brevity, where documentation refers to monitoring, management is also implied.

4.1   Monitoring

Monitoring is defined as viewing attributes of the currently running system. The majority of the items that are exposed by the platform fall under this category. They may be items that never change or items that change regularly. These attributes can usually have listeners registered that are notified of changes when they occur. Examples of monitoring attributes are:

❖   The current and peak counts of sessions logged-in to the Liberator

❖   A list of objects that a particular session was viewing

❖   The message count and rate for updates received by a DataSource

❖   The license detaicls for the system

4.2   Management

Management is defined as performing an operation to modify part of the currently running system. Management items are exposed as operations that take zero or more parameters and return a value, much like a function in a programming language. Examples of management operations:

❖   Reconnecting peers

❖   Re-validating user's permissions

❖   Ejecting session(s) that are currently logged-in to the Liberator

5   Key features

5.1   Datasource

Every DataSource exposes detailed information to the monitoring system as standard. Most counters can be reset and also publish calculated rates:

❖ System – Information related to the running process, e.g. software version, cpu / memory usage, process id, uptime etc.

❖ License – Details of the current license.

❖ Logging – Access to log messages written to various log files and control of logging levels and log file cycling.

❖ Peers – Information about configured and connected DataSource peers. Statistics for all messages sent and received. The ability to connect and disconnect and fail-over peers.

5.2   Liberator

In addition, the Caplin Liberator exposes the following features to the monitoring system:

❖ System – Aggregated information about total number of sessions, objects, data throughput etc. Information about any configured clusters.

❖ Users – Information about all connected users and their sessions, including latency and object subscription details. Ability to eject and revalidate users and cancel subscriptions.

❖ Objects – Details about all objects in the system – type, stale status, update time, subscription counts etc. Details of which sessions are viewing each object.

5.3   PriceMaster

The PriceMaster system uses the monitoring and management system to view and control the following features:

❖ Vendor Permissions and mappings – enables you to set up a sophisticated permissioning system defining which outputs are able to receive which objects.

❖ Page Construction – enables pages to be built with any combination of the objects being received by PriceMaster.

❖ Data Formatting.

5.4   Transformer and other DataSources

Additional monitoring features are currently in development for other Caplin products including Transformer and DataSource for Reuters. They will allow specific features of these products to be managed and monitored using the standard monitoring system.

5.5   Monitoring Object Naming

Each monitorable and/or manageable item is defined by a template that gives it a type name in a hierarchical namespace. This is prefixed by a server identifier for that process. For example, the logging template in the system *rttpd* would result in a template name of *rttpd.server.logging*.

Individual items are then identified by a primary key that uniquely identifies that item. For example, with the above template: *event_log* or *http_access_log*.

These are then used by the specific monitoring plug-in to create a suitable instance identifier. For example, in the JMX plugin, these two examples would create MBeans with the following ObjectNames:

rttpd.server.logging:name=event_log

rttpd.server.logging:name=http_access_log

Alternatively, in the sockmon plugin, they would be identified as follows:

rttpd.server.logging event_log

rttpd.server.logging access_log

This provides a consistent, navigable tree-like structure for the items. For example, in the JMX console's explorer view of the items they are arranged using a collapsible tree:
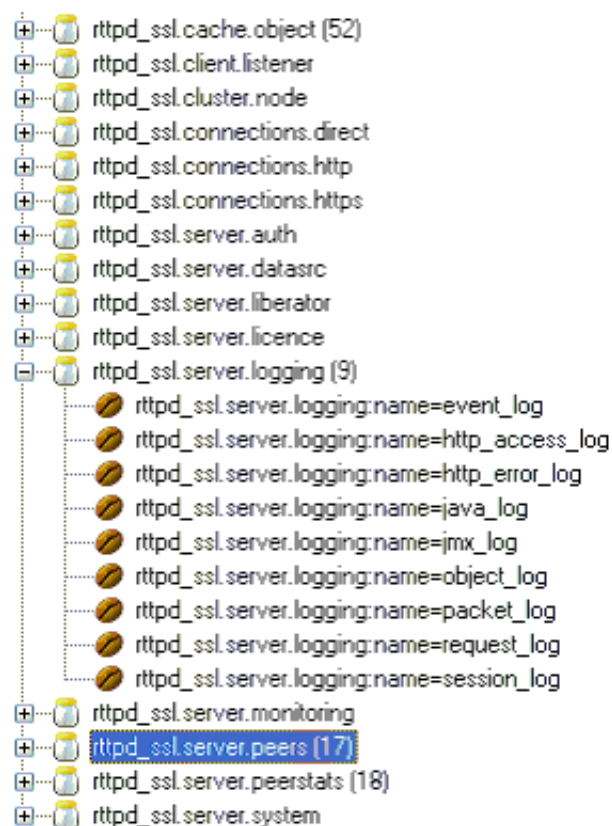
Figure 5-1: JMX console explorer example

## 5.6   Integrating with existing monitoring systems

Both of the standard monitoring modules can be used to integrate with existing monitoring systems.

The JMX module provides a standard JSR 160: JMX Remote API that can be accessed using existing JMX-aware applications such as alerting tools, graphical consoles etc.

The sockmon module can be accessed by applications that can make a socket connection such as script-based monitoring tools or proprietry applications. The interface is text command based.

6   Caplin Enterprise Console

The Caplin Enterprise Console is a fully-featured management console for the Caplin Platform.
It allows system administrators to easily monitor both the entire enterprise from a high-level view
and drill-down to individual servers and items within those servers.
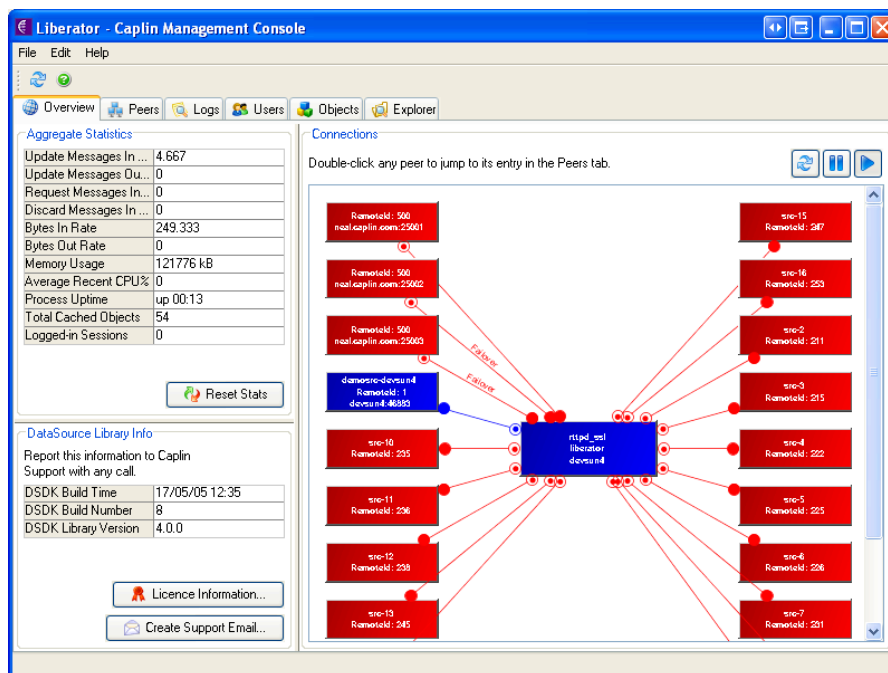
Figure 6-1: A high level view of the platform in the Caplin Management Console

The console connects to each server that is running the JMX monitoring plugin and provides a
graphical view of the system. It is based on an open framework that makes it easy to add new
tabs to a server's console to facilitate the monitoring and management of custom features that a
particular DataSource has exposed to the system.

6.1   DataSource

Every DataSource will have the following common views (tabs) available by default:

❖ Overview – A view of the DataSource and its connected peers including configurable aggregate statistics, version details and license information.

❖ Peers – Information on all configured and connected peers with a detail view available showing message counts and rates. The ability to connect / disconnect and fail-over any connections.

❖ Logs – Details of all the available log files, their messages and control of logging levels and log cycling.

❖ Explorer – A powerful tree-based view of the entire monitoring system. This allows control over every aspect of any features that have been published to the monitoring subsystem.

6.2   Liberator

The Caplin Liberator includes the above views and customizes the Overview to add details specific to Liberator such as total users and objects. These additional views are also provided:

❖ Users – A searchable view of all the users and sessions in the Liberator, with a detail view that shows information about latency, bandwidth, subscriptions, throttling etc.

❖ Objects – Detail on all the objects in the Liberator including information about their update rate, type, status and sessions subscribed.
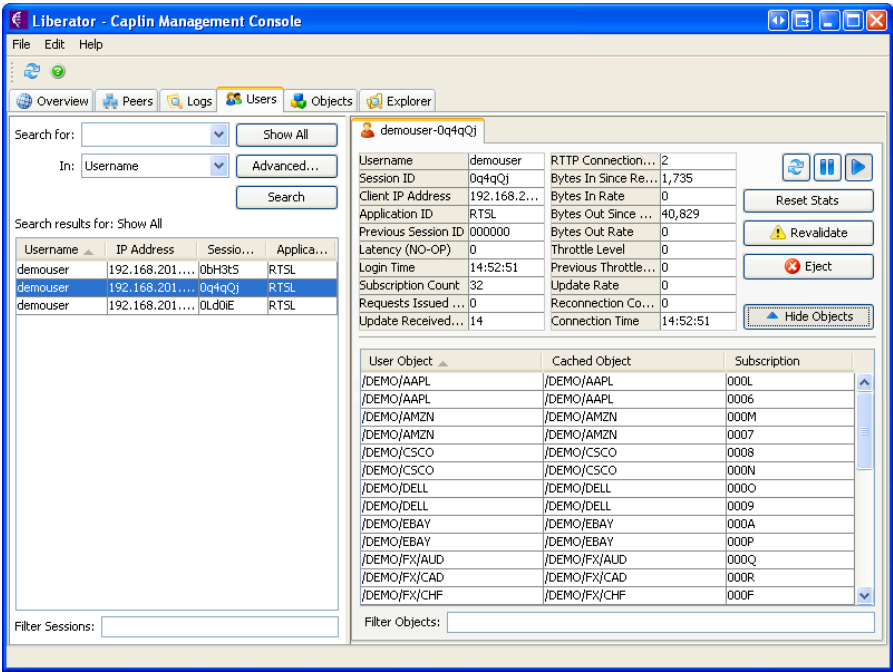
6.3   PriceMaster

The PriceMaster console includes the following views:

❖ Overview - diagrammatically shows the input and output feeds connected to the PriceMaster. Clicking on the boxes representing input and output feeds in the diagram displays data about the connection. You can also click on PriceMaster itself to view connection information specific to PriceMaster.

❖ Permissions Editor - enables you to set up a sophisticated permissioning system defining which outputs are able to receive which objects. As a user you are able to search for objects being received by PriceMaster and then add them to your permissioning list, setting the permissions as appropriate against each of the outputs to which you are connected.

❖ Page Construction - enables pages to be built with any combination of the objects being received by PriceMaster. Objects are added to your page template simply by being dragged from the record pool. You can create columns in your template to represent the fields of the records which will be displayed on the page.

❖ Administration - allows the current state of the cache to be viewed including individual record and page preview using real data. Additionally, objects can be deleted from the cache or they can be re-published to selected outputs.

The console includes a complete on-line help guide that provides detailed documentation on the available features.

6.4   Configuration

Each server console type (e.g. Liberator, Transformer, PriceMaster or a custom DataSource) is defined using an XML configuration file. This configuration file specifies which tabs are available in that console and their order and layout in the display.

A simple API is provided to allow new custom views to be created in Java containing components for viewing and controlling monitoring and management items. These could either be items within Liberator and Caplin's pre-built DataSources or a custom application developed with the DataSource SDK.

CAPLIN

Contact Us

Triton Court
14 Finsbury Square
London EC2A 1BR
UK
Telephone: +44 20 7826 9600
Fax: +44 20 7826 9610

www.caplin.com

info@caplin.com