

Capital Markets Foundation Release Notes

Innovation Release

Version 10.0

April 2017

This document applies to Apama Capital Markets Foundation Version 10.0 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2013-2017 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Table of Contents

What's New in Capital Markets Foundation 10.0	5
Miscellaneous enhancements and changes in 10.0.....	6
Deprecated features in 10.0.....	6
What's New in Capital Markets Foundation 9.12	7
Miscellaneous enhancements and changes in 9.12.....	8
What's New in Capital Markets Foundation 9.10	9
Miscellaneous enhancements and changes in 9.10.....	10
What's New in Capital Markets Foundation 9.9	11
Installation changes in 9.9.....	12
What's New in Capital Markets Foundation 5.3	13
Performance improvements due to extra parameter handling changes.....	14
MATLAB plug-in moved.....	14
Migrating to CMF 5.3.....	15
Miscellaneous enhancements and changes.....	15
What's New in Capital Markets Foundation 5.2	17
ATA now installed in APAMA_WORK directory.....	18
CMF installation script now sets up Apama Studio environment.....	18
CMF now uses channels internally.....	18
CMF can now use Universal Messaging.....	18
MDA protocol and library improvements.....	18
New helper events for creating DataViews.....	20
Features removed.....	20
What's New in Capital Markets Foundation 5.1.1	21
CMF 5.1.1 requires Apama 5.1.1.....	22
New risk firewall features.....	22
New position service framework features.....	24
New CMF position service code samples.....	25
New currency converter service.....	25
Deprecated bundle.....	26
What's New in Capital Markets Foundation 5.1	27
New risk firewall architecture.....	28
New utility interfaces.....	29
New session management interfaces.....	30
New market data subscribers.....	30
Machine readable news datastream added.....	32
Deprecated features.....	33

Removed features..... 33

1 What's New in Capital Markets Foundation 10.0

■ Miscellaneous enhancements and changes in 10.0	6
■ Deprecated features in 10.0	6

Apama Capital Markets Foundation 10.0 is the successor of Apama Capital Markets Foundation 9.12.

Apama Capital Markets Foundation 10.0 includes new features, enhancements, and changes as described in the following topics.

Miscellaneous enhancements and changes in 10.0

- If you need the QuantLib plug-in, you can now build it directly from the sources. This allows you to make full use of QuantLib. The QuantLib sources are now provided with a sample which can be found in the `APAMA_FOUNDATION_HOME/samples/QuantLib Sample` directory.

This change will also affect the ATA dependency on the Analytic bundle. If you are depending on functionality exposed by the Analytic bundle, you will have to manually add this bundle to your project. See also "Importing the ATA Project into Software AG Designer" in *Algorithmic Trading Accelerator*.

Deprecated features in 10.0

- The precompiled QuantLib plug-in library is deprecated and will be removed in a future release.
- Event Modeler is deprecated as of Apama version 9.12. Support for creating and deploying scenarios will be removed in a future release. It is recommended that you use EPL or queries to build new CMF applications.

2 What's New in Capital Markets Foundation 9.12

- Miscellaneous enhancements and changes in 9.12 8

Apama Capital Markets Foundation 9.12 is the successor of Apama Capital Markets Foundation 9.10. There is no version 9.11.

Apama Capital Markets Foundation 9.12 includes new features, enhancements, and changes as described in the following topics.

Miscellaneous enhancements and changes in 9.12

- The `DataViewManager` now supports the `decimal` field type.
- The following actions in each of the MDA subscriber factories are now static and no longer require an instance of the event to be called:

```
create()  
subscribe()  
subscribeCb()  
multipleSubscribe()  
multipleSubscribeCb()
```

The MDA samples have been updated to show how this can be achieved.

3 What's New in Capital Markets Foundation 9.10

- Miscellaneous enhancements and changes in 9.10 10

Apama Capital Markets Foundation 9.10 includes new features, enhancements, and changes as described in the following topics.

Miscellaneous enhancements and changes in 9.10

The Forward to Spot Converter (FTSC) now offers Orderbook as an available datastream connection type, along with Depth, BBA and Trade.

The version number has been removed from the file `ACMEDashboardExtensions9.9.jar`. The new name is `ACMEDashboardExtensions.jar`. Due to this change, you may need to update your build scripts and automated tests.

4 What's New in Capital Markets Foundation 9.9

■ Installation changes in 9.9	12
-------------------------------------	----

Apama Capital Markets Foundation 9.9 is the successor of Apama Capital Markets Foundation 5.3.

Installation changes in 9.9

As of version 9.9, Apama Capital Markets Foundation is installed using the Software AG Installer, and official fixes are installed using the Software AG Update Manager. See the new *Installing Apama* guide for more information.

In the Software AG Installer, you can select the following options from the product selection tree:

- **Apama > Capital Markets Foundation.** Installs all the components of Apama Capital Markets Foundation, except those for the Software AG Designer integration.

Note: On UNIX, this is the only installation option. All other options listed here are only available on Windows.

- **Designer > Apama > Capital Markets Application Development.** Installs the Designer integration components of Apama Capital Markets Foundation.

You can quickly select all Apama and Apama Capital Markets Foundation components by clicking **Typical Installations** which is shown next to the product selection tree and then selecting **Apama**.

Due to the above-mentioned changes, all Windows start menu shortcuts have changed. The shortcuts for Apama Capital Markets Foundation are now available from **All Programs > Software AG > Tools > Apama Capital Markets *n.n.*** Note that **Software AG** is the default group name that you can change during the installation.

5 What's New in Capital Markets Foundation 5.3

■ Performance improvements due to extra parameter handling changes	14
■ MATLAB plug-in moved	14
■ Migrating to CMF 5.3	15
■ Miscellaneous enhancements and changes	15

Apama Platform for Capital Markets Release 5.3 adds the following new features and enhancements to the Capital Markets Foundation.

Performance improvements due to extra parameter handling changes

To significantly improve throughput and latency, the event protocol previously used to handle extra parameters has been redesigned to separate schema data from values. Values of extra parameters are now contained in the market data events, which removes the need for a second event.

For book based market data (`Orderbook`, `Depth`, `Quotebook` and `AggregatedBook`), the schema (type and description data) for any extra parameters is sent upon the initial connection. Each level or entry now has its own set of extra parameter values. This removes the need for extra parameters to be encoded into a top-level extra parameter value. Also, the schema can now be delta updated, which removes the need for the large snapshot events that were sent in previous releases.

Two new EPL interfaces provide access to extra parameters:

- `com.apama.md.client.EPSchemaInterface` — Each market data interface now provides the `getEPSchemaInterface()` action, which you can call to access the schema for any extra parameters.
- `com.apama.md.client.EPValuesInterface` — Each market data interface and each book entry interface now provides the `getEPValuesInterface()` action, which you can call to access the values of any extra parameters.

The `AggregatedBook` actions and interfaces are slightly different as they offer access to the extra parameters from the underlying datasources of the Aggregator/FTSC. They require the symbol and source id (known as the XEPKey) of the underlying datasource to access the values and schema.

See the `ApamaDoc` for details about using these interfaces.

For an example of using the new interfaces, see "Accessing extra parameters" in the *Capital Markets Foundation*.

MATLAB plug-in moved

The MATLAB plug-in has been moved from the Apama Capital Markets Foundation into the Apama 5.3. Documentation for using MATLAB products in a CMF application can now be found in *Developing Apama Applications*, in the section "Using MATLAB products in an application".

Migrating to CMF 5.3

Apama is now built with JDK 7. You can no longer use JDK 6 (or older) to develop Apama CMF applications nor can you use JRE 6 (or older) to deploy Apama CMF applications. While it should not be necessary to rebuild applications previously built with an older Java version, the recommendation is that you build your application with the newer JDK for two important reasons:

- To ensure that you detect compatibility problems before you want to deploy the application
- To take advantage of performance improvements in newer Java compiler versions

To develop, build, and test an Apama CMF 5.3 application, the recommendation is that you use Oracle JDK 8. The minimum you can use to build your application with Apama CMF 5.3 libraries is JDK 7.

To deploy an Apama CMF 5.3 application, the recommendation is that you use Oracle JRE 8, which is the version that Apama provides. Use of any JRE other than the one that Apama ships with is discouraged.

Miscellaneous enhancements and changes

The Apama Payload Plug-in has been deprecated so the CMF blocks Symbol Attribute v1.* and Order Manager v4.0 have also been deprecated. New Symbol Attribute v2.* and Order Manager v5.0 blocks have been created and should be used. They do not use the Payload Plug-in and instead use the inbuilt dictionary `toString()` and `parse()` methods. The deprecated blocks will be removed in a future release.

API reference documentation has been added for the Market Data Adapter Library to aid clients who are creating their own MDA compliant adapters. A link can be found to it from the main documentation start page.

6 What's New in Capital Markets Foundation 5.2

- ATA now installed in APAMA_WORK directory 18
- CMF installation script now sets up Apama Studio environment 18
- CMF now uses channels internally 18
- CMF can now use Universal Messaging 18
- MDA protocol and library improvements 18
- New helper events for creating DataViews 20
- Features removed 20

Apama Platform for Capital Markets Release 5.2 adds the following new features and enhancements to the Capital Markets Foundation.

ATA now installed in APAMA_WORK directory

On Windows systems, the Algorithmic Trading Accelerator is now installed and run from the `APAMA_WORK` directory.

The default location is `C:\Users\user_name\SoftwareAG\ApamaWork_5.2\ATA_5.2.0.0`.

CMF installation script now sets up Apama Studio environment

In previous releases, you were required to manually add the `APAMA_FOUNDATION_HOME` variable and the CMF bundles to Apama Studio. With CMF 5.2, the installation script does this automatically. Manual steps to set up Apama Studio for CMF development are no longer required.

CMF now uses channels internally

CMF performance has been enhanced by internal use of the Apama 5.2 channels feature.

MDA market data from adapters is now delivered directly to the subscribing context rather than through the main context, which improves both latency and throughput.

CMF can now use Universal Messaging

Universal Messaging (UM) can now be used as a message bus to deliver market data events to CMF components. Adapters and correlators can connect to UM realms to exchange Apama events. For details, see the following section in the Apama documentation: "Using Universal Messaging in Apama Applications" in *Deploying and Managing Apama Applications*.

MDA protocol and library improvements

The CMF performance has been enhanced by various MDA protocol and library improvements. The following throughput and latency improvements were recorded in laboratory conditions:

Throughput (orderbook updates/ second)	CMF 5.1.1	CMF 5.1.1	CMF 5.2	CMF 5.2
	1 context	4 contexts	1 context	4 contexts
Windows	3880	7140	6650 (+71%)	16400 (+130%)
Linux	4590	10160	6460 (+40%)	25580 (+151%)
Latency (orderbook milliseconds 99th percentile)	CMF 5.1.1	CMF 5.1.1	CMF 5.2	CMF 5.2
	Main context	Non- main context	Main context	Non- main context
Windows	0.82	0.87	0.60 (-27%)	0.60 (-31%)
Linux	0.81	0.85	0.60 (-26%)	0.60 (-29%)

The main/single context results show the performance improvements due to protocol and library improvements.

The non-main/multi-context results also show the performance and scaling improvements due to the use of channels.

These measurements were recorded on:

- Windows machine specifications: Intel® Xeon® CPU E3-1270 V2 @ 3.50GHz 8 Cores, 16Gb RAM, Windows 7 64-bit
- Linux machine specifications: Intel(R) Xeon(R) CPU E5-2690 @ 2.90GHz 32 Cores 64Gb RAM, RHEL 6 64-bit

New helper events for creating DataViews

The CMF now provides helper events to make it easier to create DataViews of `com.apama.utils.Params` events.

The `com.apama.utils.SingleParamsDataView` event lets you create a `DataView` that contains the contents of a single `com.apama.utils.Params` event. It creates a two-column `DataView` that contains a row for each key/value pair from the supplied `Params` event. Actions defined in the `SingleParamsDataView` event let you update the contents of the `DataView`, as well as overwrite or clear the contents, or remove the `DataView`.

The `com.apama.utils.MultipleParamsDataView` event lets you create a `DataView` that contains the contents of multiple `com.apama.utils.Params` events that conform to a schema. The `MultipleParamsDataView` event creates a `DataView` with columns that match a `com.apama.utils.ParamsSchema` event. Each row represents a `Params` event that matches the `ParamsSchema` event. By default, every field in the schema is a key field but you can specify key fields to identify unique rows. You pass `Params` events to `MultipleParamsDataView` events to add, update or delete selected rows.

Features removed

The following, previously deprecated features have been removed:

- MDA Quote datastream — superseded by the Quotebook datasource, which was introduced in CMF 5.0.1.
- Legacy risk firewall — superseded by the CMF risk firewall, which was introduced in CMF 5.1.
- Legacy position services — superseded by the CMF position service framework, which was introduced in CMF 5.0.
- Legacy currency converter service — superseded by the CMF currency converter service, which was introduced in CMF 5.1.1.

7 What's New in Capital Markets Foundation 5.1.1

■ CMF 5.1.1 requires Apama 5.1.1	22
■ New risk firewall features	22
■ New position service framework features	24
■ New CMF position service code samples	25
■ New currency converter service	25
■ Deprecated bundle	26

Apama Platform for Capital Markets Release 5.1.1 adds the following new features and enhancements to the Capital Markets Foundation.

CMF 5.1.1 requires Apama 5.1.1

CMF release 5.1.1 requires Apama release 5.1.1. Be sure to install Apama 5.1.1 before you install the CMF.

New risk firewall features

The risk firewall introduced in CMF 5.1 provides the following enhancements in CMF 5.1.1:

- New default rule classes
 - Order operation ratio — Ensures that the number of amend order operations and/or cancel order operations in a specified time window do not exceed the allowed number of amend/cancel operations as defined by the specified ratio between amend/cancel order operations and new orders.
 - Order to trade ratio — Ensures that the number of order operations (new, amend, cancel) in a specified time window do not exceed the allowed number of operations as defined by the specified ratio between order operations and trades.
 - Reservation enforcer — Ensures that orders placed against reservation contracts do not exceed the minimum/maximum quantities set for reserved positions.

CMF samples for the default risk firewall rule classes have been updated to show the use of the new rules. See `Risk Firewall Rules Samples` in the `samples` directory of your CMF installation directory.

- DataViews are exposed
 - `RiskFirewallDataview` — For each risk firewall that has been created in this correlator, this `DataView` contains the name of the risk firewall, a string representation of the risk firewall parameters object, and whether or not the risk firewall is locked.
 - `FirewallRuleDefinitionDataview` — For all rules registered with all risk firewalls in this correlator, this `DataView` contains the name of the risk firewall, the name and description of the rule class, a string representation of the parameter schema for the rule class and the priority of the rule class.
 - `AllRuleClassInstance` — This `DataView` is exposed by the `MemoryStore` table that contains entries for all rule class instances for all risk firewalls in this correlator. This `DataView` contains the name of the risk firewall, the name of the rule class, the ID of the rule instance and a string representation of the rule instance's configuration.

- `OrderOperationCache` — This `DataView` is exposed by the `MemoryStore` table that contains entries for the order operation cache, which contains information for each failed order operation that has gone through any risk firewall in this correlator. This `DataView` contains extensive details for each failed order operation. The entries in the order operation cache, and hence in the `MemoryStore` table, are recovered upon correlator re-start and can be automatically purged according to the settings of the new risk firewall configuration parameters listed in the next item.
- New risk firewall configuration parameters to be set in the factory before creating the risk firewall:
 - `ORDER_OPERATION_CACHE_TIMEOUT` — Specifies the number of seconds that a row can be in the order operation cache before it is eligible to be purged.
 - `ORDER_OPERATION_CACHE_PURGE_INTERVAL` — Specifies the number of seconds between automatic purges of the cache.
 - `ORDER_OPERATION_CACHE_PURGE_TIME` — Specifies a time of day for automatic purging of the cache.
 - `ORDER_OPERATION_CACHE_MAX_ROWS` — Specifies the maximum number of rows that can be removed in one purge operation.
 - `ORDER_OPERATION_CACHE_PATH_DEFAULT` — Specifies the default file and path for the tables in the order operation store.
 - `ORDER_OPERATION_CACHE_PATH_KEY` — Specifies the configuration key value for setting the risk firewall default file and path for the tables in the order operation store.

The order operation cache can get very large. Be sure to either set parameters to enable automatic purging or manually remove rows from the order operation cache by invoking the `com.apama.firewall.RiskFirewall.clearObjection()` action, which is new in this release.

- New configuration parameters for the default order throttle rule class:
 - `REJECT_NEW_PARAMETER` — Indicates whether this risk firewall rule instance should reject `NewOrder` objects if the throttle limit has been reached.
 - `REJECT_AMEND_PARAMETER` — Indicates whether this risk firewall rule instance should reject `AmendOrder` objects if the throttle limit has been reached.
 - `REJECT_CANCEL_PARAMETER` — Indicates whether this risk firewall rule instance should reject `CancelOrder` objects if the throttle limit has been reached.
 - `REJECT_NEW_DEFAULT` — Default value for whether this risk firewall rule instance should reject `NewOrder` objects is the throttle limit has been reached. By default, this is `true`, which means that new order requests are rejected if the throttle limit has been reached.
 - `REJECT_AMEND_DEFAULT` — Default value for whether this risk firewall rule instance should reject `AmenOrder` objects if the throttle limit has been reached. By

default, this is `false`, which means that amend order requests are allowed even if the throttle limit has been reached.

- `REJECT_CANCEL_DEFAULT` — Default value for whether this risk firewall rule instance should reject `CancelOrder` objects if the throttle limit has been reached. By default, this is `false`, which means that cancel order requests are allowed even after the throttle limit has been reached.

Also, the Algorithmic Trading Accelerator has been updated to use the new risk firewall. It no longer uses the legacy risk firewall.

New position service framework features

The following position service framework enhancements have been added in 5.1.1:

- The new default tracker, `com.apama.position.tracker.RealizedPnLTracker` provides actual profit and loss values for executed orders normalized to a specified currency. Sample code for using a realized profit and loss tracker is in the `samples\Position Trackers Sample\objects` directory of your CMF installation directory. For information and instructions for using the new default tracker, see "Creating and subscribing to the realized profit and loss tracker" in *Capital Markets Foundation*.
- The new action, `com.apama.position.PSInterface.addSymbolSliceUpdateCallback()`, registers a user-defined callback action to be called every time that the position changes for a specified symbol slice in a specified subscription. You can define more than one update callback per subscription.
- New configuration parameters for all default trackers are provided in `com.apama.position.tracker.GenericPositionTrackerConstants`:
 - `TRACKER_CONFIG_TRACK_SEPARATE_CURRENCIES` — Indicates that you want to also track FX currency positions as separate currencies. For example, a symbol slice of `EUR/USD` and `GBP/USD` will also track the positions of `EUR`, `USD`, and `GBP` separately.
 - `TRACKER_CONFIG_CURRENCY_SEPARATOR` — Lets you specify the currency separator to use when splitting symbols. The default, `/`, is specified by the `TRACKER_CONFIG_DEFAULT_CURRENCY_SEPARATOR` parameter.
- The `com.apama.position.tracker.PSTrackerInterface` event provides two new actions for custom position trackers:
 - `subscribeSymbolSliceUpdates()`
 - `unsubscribeSymbolSliceUpdates()`

See the `ApamaDoc` for details.

New CMF position service code samples

The new `Position Trackers Sample` in the `samples` directory of the CMF installation directory contains source code for the CMF default position trackers. You can use the source code as a base for customized trackers. The new sample includes code for

- Using the position service
- Using the default position trackers, including the new realized profit and loss default position tracker
- Tracking FX single currencies
- Using the new `com.apama.position.PSInterface.addSymbolSliceUpdateCallback()` action

New currency converter service

When you create a realized profit and loss default position tracker you must specify a currency converter. The tracker uses the currency converter to obtain values that are normalized to a base currency that you specify.

Currency converter configuration parameters let you specify the symbols whose values you need to normalize. If you specify for example, "USD/EUR" the currency converter will automatically obtain values for EUR to USD as well as for USD to EUR. Also, if there is not a direct exchange rate between two symbols, you can specify those symbols as a cross currency and the currency converter will automatically obtain information for a third symbol that it can use to convert between the two specified symbols.

The new APIs for creating and configuring a currency converter are in the `com.apama.ccyconverter` package and follow the same conventions used for the CMF APIs for session handlers, market data subscribers, and risk firewalls:

Event	Description
<code>CalculationExtension</code>	Defines the interface used by the currency converter to communicate and control calculation objects that supply values.
<code>CalculationExtensionFactory</code>	Creates a new instance of a <code>CalculationExtension</code> object. The created object is a default implementation. As needed, you should override the provided <code>CalculationExtension</code> actions with your own implementations.

Event	Description
<code>Consts</code>	Defines the constant values used to specify <code>CalculationExtension</code> configuration values.
<code>CurrencyConverter</code>	Defines the interface for using the currency converter.
<code>CurrencyConverterFactory</code>	Creates a new instance of a <code>CurrencyConverter</code> object.
<code>ErrorConsts</code>	Defines the constant values used for error handling in the currency converter.

The steps for using a currency converter are as follows:

1. Create a currency calculation extension.
2. Create a currency converter that uses the currency calculation extension you created.
3. Provide an interface to the currency converter you created. For example, when you want to use a realized profit and loss position tracker, you must specify a currency converter.

Sample code for using a currency converter is in the `samples\Currency Converter Sample` directory in your CMF installation directory. In addition, there are two sample currency calculation extensions in the `Currency Converter Sample\eventdefinitions` directory. Information and instructions for using a currency converter and a currency calculation extension are in the "Analytics" section of the *Capital Markets Foundation*.

Deprecated bundle

The `Currency Converter Datasource` bundle, which is in the `projects\analytics\datasources\currencyconvertor` directory of your CMF installation directory, is deprecated in this release. It will be removed in a future release.

8 What's New in Capital Markets Foundation 5.1

■ New risk firewall architecture	28
■ New utility interfaces	29
■ New session management interfaces	30
■ New market data subscribers	30
■ Machine readable news datastream added	32
■ Deprecated features	33
■ Removed features	33

Apama Platform for Capital Markets Release 5.1 adds new features and enhancements to the Capital Markets Foundation. The new features and changes are described in this document.

New risk firewall architecture

With Release 5.1, the Capital Markets Foundation supports a new architecture for its risk firewall. The risk firewall is a pre-trade risk firewall. It provides a mechanism for ensuring that orders being placed to a service comply with a set of rules/criteria.

The new architecture replaces the previous CMF risk firewall that was supported in previous releases. That implementation is now referred to as the legacy risk firewall.

Note: The legacy risk firewall provided in previous releases is deprecated and will be removed in a future release. You should stop using the legacy risk firewall and use only the new risk firewall provided with CMF 5.1.

The CMF 5.1 risk firewall is significantly faster and more scalable than the previous risk firewall. The new risk firewall architecture supports multiple contexts as well as multiple separate instances of a risk firewall in the same application. In addition, the developer interface for adding custom rules has been streamlined for ease of use.

The risk firewall lets applications define a set of rules that orders must be validated against. Evaluated orders are either approved and forwarded to an order management system, or rejected and sent back to the order sender that originated the order. After an approved order is processed by an order management system, an order update passes back through the risk firewall and is processed by each rule class registered with the risk firewall. This enables the risk firewall to enforce limits. The risk firewall provides default rule classes that enforce limits on individual orders for:

- Order price
- Order quantity
- Order throttle (number of orders per second)
- Order value (quantity multiplied by price)

Also, the risk firewall provides default rule classes that enforce cumulative limits across multiple orders for:

- Client credit limit
- Open/pending position limit

The new interfaces for implementing one or more uniquely named risk firewalls in your application are:

- In the `com.apama.firewall` package:
 - `RiskFirewallFactory` creates risk firewall instances and provides connections to existing risk firewalls.

- `RiskFirewall` is either a local risk firewall or a connection to a remote (in another context or monitor) risk firewall. Connected risk firewalls do not support all `RiskFirewall` actions.
- `RuleClassFactory` creates the rule class interface for defining customized rule classes that can be registered with a risk firewall.
- `RuleClass` defines the interface that the risk firewall uses to communicate with its registered rule classes.
- `OrderSender` sends orders to the risk firewall and handles order updates after the risk firewall has processed them.
- `OrderReceiver` receives approved orders from the risk firewall, and can forward them to order management systems, receive order updates back from the OMS, and forward those back through the risk firewall to the order senders.
- In the `com.apama.firewall.rules` package:
 - `ClientCreditLimitRiskFirewallRule`
 - `OrderPriceLimitRiskFirewallRule`
 - `OrderQuantityLimitRiskFirewallRule`
 - `OrderThrottleLimitRiskFirewallRule`
 - `OrderValueLimitRiskFirewallRule`
 - `PositionLimitRiskFirewallRule`

Note: By default, approval from the risk firewall requires an order to match at least one instance of every registered rule class. This is different from the legacy risk firewall, which approves orders only if there are no rules that match the order.

Information and instructions for using the new risk firewall architecture are in the "Risk Firewall" section of the *Capital Markets Foundation*.

New utility interfaces

This release introduces two utility interfaces that you can use with the new session interfaces, new subscribers, and new risk firewall.

- `com.apama.utils.Params`
This object provides actions for adding, obtaining and removing parameters.
- `com.apama.utils.Error`
This object provides actions for setting and obtaining information associated with an error.

New session management interfaces

A session provides the interface between your CMF application and a source of data, such as an external adapter. In this CMF release, you can use sessions only with market data architecture (MDA) subscribers. It is expected that future CMF releases will extend the use of sessions to other CMF components.

The event interfaces for implementing session management have been streamlined for ease of use. Common tasks, such as connection to an external datasource, can be accomplished in a single line of code.

The interfaces for using sessions are:

- `com.apama.session.SessionHandlerFactory`
- `com.apama.session.SessionHandler`

The factory object lets you create a session handler and optionally connect to a session. The session handler lets you manage the session interaction between your application and the external datasource. You use it to connect, disconnect, add parameters, add callbacks to be executed upon successful subscription, and override default error behavior for sessions. Information about how to use sessions in the "Session Management" section of the *Capital Markets Foundation*.

You can use the new utility interfaces, `com.apama.utils.Params` and `com.apama.utils.Error`, with the session management interfaces.

New market data subscribers

In a CMF application, to obtain market data, you now set up one or more market data subscribers. CMF provides subscribers for the following types of market data:

- BBA
- Depth
- News
- Orderbook
- Quotebook
- Trade

The event interfaces for subscribing to market data have been streamlined for ease of use. Common tasks, such as subscription to an external datasource, can be accomplished in a single line of code.

The following table lists the new interfaces that you use to subscribe to market data. ApamaDoc provides details about the fields in each interface.

Interface	Description
com.apama.md.BBASubscriberFactory com.apama.md.BBASubscriber	For each type of market data, there is a factory object and a subscriber object. The factory object for each type of market data provides the same actions. Likewise, the subscriber object for each type of market data provides the same actions.
com.apama.md.DepthSubscriberFactory com.apama.md.DepthSubscriber	For each type of market data, there is a factory object and a subscriber object. The factory object for each type of market data provides the same actions. Likewise, the subscriber object for each type of market data provides the same actions.
com.apama.md.NewsSubscriberFactory com.apama.md.NewsSubscriber	For each type of market data, there is a factory object and a subscriber object. The factory object for each type of market data provides the same actions. Likewise, the subscriber object for each type of market data provides the same actions.
com.apama.md.OrderbookSubscriberFactory com.apama.md.OrderbookSubscriber	Each factory object creates its eponymous subscriber object. In addition, each factory object provides the most common actions you would execute on the associated subscriber object. Many of these interfaces provide pairs of actions in which one takes a user-defined callback and one does not, for example, <code>TradeSubscriber.subscribe()</code> and <code>TradeSubscriber.subscribeCb()</code>
com.apama.md.QuotebookSubscriberFactory com.apama.md.QuotebookSubscriber	Each factory object creates its eponymous subscriber object. In addition, each factory object provides the most common actions you would execute on the associated subscriber object. Many of these interfaces provide pairs of actions in which one takes a user-defined callback and one does not, for example, <code>TradeSubscriber.subscribe()</code> and <code>TradeSubscriber.subscribeCb()</code>
com.apama.md.TradeSubscriberFactory com.apama.md.TradeSubscriber	Each factory object creates its eponymous subscriber object. In addition, each factory object provides the most common actions you would execute on the associated subscriber object. Many of these interfaces provide pairs of actions in which one takes a user-defined callback and one does not, for example, <code>TradeSubscriber.subscribe()</code> and <code>TradeSubscriber.subscribeCb()</code>
com.apama.md.HandlerErrorConstants	This object provides the possible values for the <code>errorType</code> parameter in a

Interface	Description
	<code>com.apama.utils.Error</code> event of an error callback of a subscriber. They also provide the possible keys for the <code>extraParams</code> dictionary.

You should use the new market data subscriber interfaces, in conjunction with the new session management interfaces, in place of the low-level market data interfaces, such as `com.apama.session.SessionManagerFactory`. Also, you can use the new utility interfaces, `com.apama.utils.Params` and `com.apama.utils.Error`, with the session management interfaces.

Documentation for using market data subscribers is in the "Market Data Management" section of the *Capital Markets Foundation*.

Machine readable news datastream added

With this release, you can use CMF's market data architecture to subscribe to news stories from common sources. News subscribers use the new session interfaces provided in this release, and behave similarly to the subscribers (also new in this release) provided for BBA, Depth, Orderbook, Quotebook, and Trade events.

When working with data from news sources you can listen for news stories that are new or updated, or have been deleted. In addition, a news event has a field that contains a dictionary of extra parameters. This ensures that you can process all information associated with a news story regardless of the source. The adapter for the news source defines any extra parameters that might be provided with news stories.

The following event interfaces are provided for working with news. For details, see "Subscribing to news" in the *Capital Markets Foundation* as well as the *ApamaDoc*.

- `com.apama.md.N` - Basic news event
- `com.apama.md.ND` - News story deletion event
- `com.apama.md.NS` - Schema of metadata that describes the extra parameters for news stories sent from the connected datasource
- `com.apama.md.NSI` - Metadata for a single extra parameter
- `com.apama.md.NSD` - Event for updating the extra parameters schema
- `com.apama.md.client.CurrentNewsInterface` - Interface for reading news stories
- `com.apama.md.WrappedN` - First event the adapter sends to your application upon subscription

Deprecated features

The CMF risk firewall provided in previous releases is deprecated and will be removed in a future release. You should stop using the legacy risk firewall and use only the new risk firewall provided with CMF 5.1.

The Quote datastream is deprecated in CMF 5.1 and will be removed in a future release. It is superseded by the Quotebook datasource, which was introduced in CMF 5.0.1.

As documented in the previous CMF release, the legacy position service is deprecated and will be removed in a future release. If you have not already done so, you should stop using the legacy position service and use only the new position service framework.

Removed features

The following features, which were deprecated in previous releases, have been removed:

- `ATOMIC_DELTA` transfer protocol, which was in the new Market Data Architecture. Use the compound delta mode instead.
- The C Math Library correlator plug-in. Use the math functions provided natively in EPL.