SYBASE®

User's Guide

Sybase Aleri Adapter for Thomson Reuters RMDS Marketfeed 2.2.0

DOCUMENT ID: DC01348-01-0220-01

LAST REVISED: April, 2011

countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Group Ltd.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Table of Contents

About This Guide

1. Purpose

This guide provides instructions for installing and configuring the Aleri Reuters Marketfeed Adapter for the Aleri Streaming Platform. The Aleri Reuters Marketfeed Adapter is referred to as the *Reuters Adapter* or *Adapter*.

2. Organization

This guide includes the following chapters:

General Product Information	Provides a brief overview of the features and functions of the Aleri Reuters Marketfeed Adapter.
Installing the Adapter	Explains how to install the Aleri Reuters Marketfeed Adapter and configure the user environment for it.
Configuring an Inbound Adapter	Describes how to configure the Reuters Adapter to receive Market Data from the Reuters Market Data Service (RMDS), for use by the Aleri Streaming Platform.
Configuring an Outbound Adapter	Describes how to configure the Reuters Adapter to publish data generated by the Aleri Streaming Platform to RMDS.
Administering Adapters	Includes procedures for modifying the Adapter Map files for performance or other reasons.

And, it includes the following appendices:

Command Usage	Provides a complete description of how to use the **rmds** command.
Environment Variables	Identifies all of the environment variables used by the Aleri Reuters Marketfeed Adapter.
Inbound Adapter XML Syntax	Describes in detail each of the elements that go into an Inbound Adapter map file and how they fit together.
Outbound Adapter XML Syntax,	Describes in detail each of the elements that go into an Outbound Adapter map file and how they fit together.
Logging Facilities	Describes the different logging mechanisms that can be used to monitor the Adapter and troubleshoot problems.

3. Related Documents

The following Aleri documents provide additional information related to the Aleri Reuters Marketfeed Adapter:

- *Aleri Reuters Marketfeed Adapter Release Notes*
- *Aleri Streaming Platform Authoring Guide*

Chapter 1. General Product Information

1.1. Introduction

The Aleri Reuters Marketfeed Adapter is a software interface between the Aleri Streaming Platform and the Reuters Market Data System (RMDS). It uses the Reuters Marketfeed message format. Aleri also has an adapter that uses Reuters' Open Message Model (OMM) format to interface with RMDS.

The Aleri Reuters Marketfeed Adapter can be configured as either an Inbound or an Outbound Adapter.

An Inbound Adapter is configured to subscribe to one or more Reuters Instrument Codes (RICs) on the RMDS to provide input to the Aleri Streaming Platform.

An Outbound Adapter can be configured to publish output from the Aleri Streaming Platform to the RMDS. This enables the Aleri Streaming Platform to use the speed and reliability of Reuters' infrastructure to deliver data.

Systems that require both inbound and outbound capabilities must run two Adapter instances.

1.2. Compatibility with the Aleri Streaming Platform

This release of the Aleri Reuters Marketfeed Adapter is fully compatible with release 3.1.5 of the Aleri Streaming Platform. The Adapter works with all of the latest enhancements to the Pub/Sub API included in that release.

It is also compatible with earlier releases of the Aleri Streaming Platform, but some of the Adapter's capabilities will not be supported by the Pub/Sub API in those earlier releases. Attempts to use these capabilities will result in warning messages about non-implemented features appearing in the Adapter log.

If you use this release of the Adapter with a release of the Aleri Streaming Platform prior to 3.1.5, you should set the Adapter debug level to DEBUG and check the Adapter log file for warning messages and any related debugging messages.

1.3. Supported Operating Systems

This release of the Aleri Reuters Marketfeed Adapter is supported on the following operating systems:

- Solaris™ 10 SPARC® update 5
- Solaris™ 10 x86 6/06 or higher
- Red Hat Linux® Enterprise AS4, version 4.5 or higher, 32-bit

Note:

Although the Adapter only runs on the Solaris™ and Linux® operating systems listed above, a running instance of the Adapter can work with Aleri Streaming Platform software running on Solaris™, Linux® or Windows®.

1.4. Reuters Requirements

For an Inbound Adapter, you need an RMDS market data connection that uses the Marketfeed protocol (not Reuters Open Message Model (OMM) protocol) and a working subscription for data on one or more financial instruments.

For an Outbound Adapter, you need a working connection with support for sending data to RMDS using the Marketfeed protocol.

Refer to the `doc/README` file to find out what version of each RMDS component is required.

Chapter 2. Installing the Marketfeed Adapter

2.1. Installing the Adapter Software

To install the Aleri Reuters Marketfeed Adapter on a Solaris™ or Linux® computer, follow this procedure.

1. Get the appropriate Aleri Reuters Marketfeed Adapter distribution file for your operating environment. Contact your Aleri sales representative for details.

   The name of the distribution file follows this format:

   `Aleri_ReutersAdapter_v`*`VERSION`*`_r`*`RELEASE_ENVIRONMENT`*`.tar.gz`

   `VERSION` is the Aleri Streaming Platform version number. *`RELEASE`* is the Reuters Adapter release number. *`ENVIRONMENT`* identifies the operating environment for this version of the Adapter, which will be one of the following:

i686	32-bit Linux® compiled using gcc
sun4_32_spro	32-bit SPARC® built with Sun's SPARCworks compiler
i86pc_32_spro	32-bit x86 (not SPARC®) built with Sun's SPARCworks compiler

   For example, the `Aleri_ReutersAdapter_v2.1.9_r12345_sun4_32_spro.tar.gz` distribution file contains release 12345 of the Adapter for Aleri Streaming Platform version 2.1.9 for Solaris™ SPARC® systems.

2. Copy the distribution file to the directory on your server that will be used as the Adapter's home directory.

   Any directory can be used, but Aleri recommends that you create an `adapters` directory parallel to the `$PLATFORM_HOME` directory with a `reuters` subdirectory underneath, and copy the distribution file to the `reuters` directory.

   If the `$PLATFORM_HOME` directory is `.../aleri/platform` (as recommended in the *Aleri Streaming Platform Installation Guide*), the new directory structure would look like this:

   ```
   .../aleri/
            platform/
                     (other subdirectories)
            adapters/
                     reuters/
   ```

3. Verify the distribution file using the MD5 value.

   - On a Sun server, execute the command

     ```
     /usr/bin/digest -a md5 ADAPTER_FILE
     ```

     Where *`ADAPTER_FILE`* is the name of the Adapter distribution file.

     For example, using the same filename used earlier:

     ```
     /usr/bin/digest -a md5 Aleri_ReutersAdapter_v2.1.9_r12345_sun4_32_spro.tar.gz
     ```

   - On a Linux® server, execute the command

```
md5sum ADAPTER_FILE
```

For example, using the name of a Linux distribution file:

```
md5sum Aleri_ReutersAdapter_v2.1.9_r12345_i686.tar.gz
```

4. Compare the output of this command with the verification string for this file, which is available on the Aleri download page.

   If they do not match, download the file and try again, or contact Aleri.

5. Unzip and untar the Adapter file.

   Using the filename from the above example:

   ```
   gunzip -c Aleri_ReutersAdapter_v2.1.9_r12345_i686.tar.gz | tar xvf -
   ```

   The system installs the Reuters Adapter binaries, configuration files, and other files below the directory containing the distribution file.

6. See the `MANIFEST` and `README.TXT` files under `$ALERI_REUTERS_HOME/doc/` for a complete listing of the files and subdirectories that comprise the distribution and verify that they were unpacked from the distribution file.

2.2. Enabling User Access

For each user account that will run the Aleri Reuters Marketfeed Adapter, do the following:

1. Make sure the user account has permission to execute the installed software.

2. Create an environment variable, `ALERI_REUTERS_HOME`, set to the full pathname of the directory in which you placed the Adapter distribution file.

   Add the following line to the `.profile` file in the user account's home directory:

   ```
   export ALERI_REUTERS_HOME=/home/aleri/aleri/adapter
   ```

   If you installed the Adapter somewhere else, substitute that path for `/home/aleri/aleri/adapter`. All references to that directory in this document will use this environment variable.

3. The Aleri Streaming Platform supports Pluggable Authentication Module (PAM), RSA, and Kerberos authentication. If your installation uses one of these authentication methods, make sure the user account is set up to work with that method of authentication.

2.3. Configuring Connections to Reuters

Before you begin configuring a Reuters connection for your Adapter,

- Create (or choose) a directory in which to store your site-specific configuration files.
- Create an environment variable, `MY_CONFIG`, set to the full pathname of that directory.
- Configure a Reuters connection for either an inbound (see Section 2.3.1, "Configuring an Inbound Connection from Reuters") or an outbound (see Section 2.3.2, "Configuring an Outbound Connection to Reuters") Adapter.

2.3.1. Configuring an Inbound Connection from Reuters

During the installation process a sample configuration file, `rfasub.cfg`, was placed in the `$ALERI_REUTERS_HOME/config` directory. This file, shown below, follows the Reuters format for configuration files.

```
# Port number of the p2ps (default 8101)
\Connections\Connection_SSLED\PortNumber           = 8101

# (should be the DACS name if DACS is enabled)
# username to connect with
\Connections\Connection_SSLED\UserName             = "triarch"

# p2ps host
\Connections\Connection_SSLED\ServerList           = "snake.aleri.com"

# Refer to RFA documentation for
# more advanced changes to the remaining
# entries
\Connections\Connection_SSLED\connectionType       = "SSLED"

\Adapters\SASS3_Adapter\requestQueueReadThreshold  = 1
\Adapters\SASS3_Adapter\mainLoopTimerInterval      = 200

\Adapters\SSLED_Adapter\masterFidFile              = "config/appendix_a"
\Adapters\SSLED_Adapter\enumTypeFile               = "config/enumtype.def"
\Adapters\SSLED_Adapter\downloadDataDict           = false

# Change the fileLoggerFilename appropriately for your setup
\Logger\AppLogger\windowsLoggerEnabled             = false
\Logger\AppLogger\fileLoggerEnabled                = true
\Logger\AppLogger\fileLoggerFilename               = "rfamf.{p}.log"

\Control\Entitlements\dacs_SbeEnabled              = false
\Control\Entitlements\dacs_CbeEnabled              = false

\Logger\ComponentLoggers\Connections\messageFile   = "config/messages/RFA6_Connections.mc"
\Logger\ComponentLoggers\Adapter\messageFile       = "config/messages/RFA6_Adapter.mc"
\Logger\ComponentLoggers\SessionCore\messageFile   = "config/messages/RFA6_SessionLayer.mc"
\Logger\ComponentLoggers\SSLED_Adapter\messageFile = "config/messages/RFA6_SSLED_Adapter.mc"


\Sessions\Session1\connectionList                   = "Connection_SSLED"
```

The following procedure describes how to modify this sample file for your site's RMDS connection. If you have multiple Adapters using multiple RMDS connections, you may need a separate, uniquely named, configuration file for each one. For a configuration file with a different name, you must either change the entry in the Inbound Adapter Map file or specify that filename using the `-F` option to the **rmds** command.

1. Obtain the following information from your system administrator:
   - the name of the server that receives Marketfeed data from RMDS
   - the port number on that machine to which your system connects
   - the username defined for your connection to Reuters
   - the name of each Reuters service to which you are subscribed
2. Make a copy of the sample configuration file in your `$MY_CONFIG` directory.

   ```
   cp $ALERI_REUTERS_HOME/config/rfasub.cfg $MY_CONFIG
   ```

3. Using your favorite editor, open the configuration file.

4. In the `\Connections\Connection_SSLED\PortNumber` line, replace the default port number, `8101`, with the number used by your Reuters connection if different.

5. In the `\Connections\Connection_SSLED\UserName` line, replace `triarch` with the username for your Reuters subscription (keep the surrounding quotation marks).

6. In the `\Connections\Connection_SSLED\ServerList` line, replace `gusto.aleri.com` with the name of your server that receives Marketfeed data from RMDS (keep the surrounding quotation marks). If your system has more than one server receiving data from RMDS, include all of their names in a comma separated list, in priority order.

7. In the `\Logger\AppLogger\fileLoggerFilename` line, you can, if you wish, change the name of the log file. The default filename specified here, `rfasub.{p}.log`, includes the string `{p}` which the Reuters library replaces with the UNIX Process ID when it creates the log file.

8. Save the modified file.

The other parameters in the configuration file also affect the functioning of the Aleri Reuters Marketfeed Adapter, and you may want to modify them as well. See the *RFA Configuration Guide* in `$ALERI_REUTERS_HOME/doc` for more details about configuring the RMDS connection, and Section E.2, “Reuters Logging” for more information about configuring logging.

2.3.2. Configuring an Outbound Connection to Reuters

During the installation process a sample configuration file, `rfapub.cfg`, was placed in the `$ALERI_REUTERS_HOME/config` directory. This file, shown below, follows the Reuters format for configuration files.

```
# installPrompt: port number for Source Distributor route
\Connections\Connection_SSLED_MP\ipcServerName      = "8105"

# Refer to RFA documentation for
# more advanced changes to the remaining
# entries
\Connections\Connection_SSLED_MP\connectionType     = "SSLED_MP"
\Connections\Connection_SSLED_MP\entitlementData    = false
\Sessions\Session1\connectionList                   = "Connection_SSLED_MP"

# Change the fileLoggerFilename appropriately for your setup
\Logger\AppLogger\windowsLoggerEnabled              = false
\Logger\AppLogger\fileLoggerEnabled                 = true
\Logger\AppLogger\fileLoggerFilename               = "rfamf.{p}.log"

\Control\Entitlements\dacs_SbeEnabled               = false
\Control\Entitlements\dacs_CbeEnabled               = false

\Logger\ComponentLoggers\Connections\messageFile      = "./config/messages/RFA6_Connections.mc"
\Logger\ComponentLoggers\Adapter\messageFile          = "./config/messages/RFA6_Adapter.mc"
\Logger\ComponentLoggers\SessionCore\messageFile      = "./config/messages/RFA6_SessionLayer.mc"
\Logger\ComponentLoggers\SSLED_Adapter\messageFile    = "./config/messages/RFA6_SSLED_Adapter.mc"
\Logger\ComponentLoggers\SSLED_MP_Adapter\messageFile = "./config/messages/RFA6_SSLED_MP_Adapter.mc"
```

The following procedure describes how to modify this sample file for your site's RMDS connection. If you have multiple Adapters using multiple RMDS connections, you may need a separate, uniquely named, configuration file for each one. For a configuration file with a different name, you must either change the entry in the Outbound Adapter Map file or specify that filename using the `-F` option to the **rmds** command.

1. Obtain the following information from your system administrator:

   - the port number at which the src_dist or RMDS infrastructure server will listen for updates from the Reuters Adapter
   - the name of the server that receives updates from the Aleri Streaming Platform

2. Make a copy of the sample configuration file in your `$MY_CONFIG` directory.

```
cp $ALERI_REUTERS_HOME/config/rfapub.cfg $MY_CONFIG
```

3. Using your favorite editor, open the configuration file.

4. In the `\Connections\Connection_SSLED_MP\ipcServerName` line, replace the default port number, `8105`, with the port number at which your src_dist will listen for updates from the Reuters Adapter if different.

5. In the `\Logger\AppLogger\fileLoggerFilename` line, you can, if you wish, change the name of the log file. The default filename specified here, `rfapub.{p}.log`) includes the string `{p}` which the Reuters library replaces with the UNIX Process ID when it creates the log file.

6. Save the modified file.

Chapter 3. Configuring an Inbound Marketfeed Adapter

When the Reuters Adapter is used to push data from the Reuters Market Data Service (RMDS) to the Aleri Streaming Platform it is called an Inbound Adapter. This chapter shows how to configure an Inbound Adapter.

3.1. Preparation

Before configuring an Inbound Adapter, you must decide what data you need and how you want to set up your system.

3.1.1. Data Decisions

The most important decisions involve how the incoming Reuters data fit into the data model. You must also decide whether you require Level1 or Level2 data. If you need Level1 data, proceed with the Marketfeed Adapter. If you need Level2 data, you should use the OMM Adapter instead.

Venue(s)	Decide which venue(s) are of interest (e.g., NYSE, NASDAQ, Toronto, ...)
RICs and FIDs	Determine what market data you need -- specifically, which Reuters Instrument Codes (RICs) you want the Adapter to provide to the Aleri Streaming Platform, and which Reuters Field IDs (FIDs) for these instruments you want to use.
Aleri Streams	The Aleri Reuters Adapter can furnish data to one or more data streams on the Aleri Streaming Platform. To use the Reuters Market Data provided by the Adapter, you must decide which existing data stream(s) will be mapped to the Adapter's data feed or define one or more new data streams.

3.1.2. Administrative Decisions

There are administrative decisions in addition to those related to the data model.

- *Session Name*

  This is an arbitrary string used to link the data model and the Adapter map file, but it must be used consistently.

- *Directories for logging and stream output*

  The Adapter writes its own log messages, and generates a separate set of Reuters log messages. You must specify where these log files should be written in the configuration.

- *Aleri user account*

  You must specify a valid Aleri Streaming Platform user account for the Adapter to use (unless you specified no authentication when you started the Aleri Streaming Platform.> See Section 2.2, “Enabling User Access” for details on setting up the selected account to use the Adapter.

3.1.3. Aleri Streaming Platform Information

You will need to know the following about the Aleri Streaming Platform instance to which you will be transmitting data.

- Does it run all Command and Control I/O through a Secure Socket Layer (i.e., was it started with the `-e` flag)?
- What type of authentication mechanism (Kerberos, PAM, RSA, or none) does it use?

3.1.4. Understanding the Inbound Adapter Map File

The map file configures the interface between the Aleri Reuters Marketfeed Adapter and the Aleri Streaming Platform. It specifies which Source Stream(s) will receive data from RMDS via the Adapter and it maps specific RMDS Field Identifiers (FIDs) to specific columns in that Source Stream.

Whether you create a new Source Stream to bring the RMDS data to the Aleri Streaming Platform, or map the incoming RMDS data to an existing stream, the most important part of creating the Inbound Adapter Map file is to work out this mapping. To do this well, it is best to start with an overview of the datatypes and data elements involved.

The Inbound Adapter Map file must accomplish three major tasks:

- Match incoming data elements to columns in one or more streams defined in the Aleri Streaming Platform configuration file.
- Match the RIC provided with each update from the Adapter with a row in the Aleri Streaming Platform configuration file.
- Ensure that each update from the Adapter can be converted into a record that provides a unique key for each stream being populated as defined by the stream's column definitions.

3.1.4.1. Aleri Streaming Platform Data Structures

See Section 3.2, “Getting Stream Information from the Data Model” for detailed information about the structure of an Aleri Streaming Platform data stream.

The most important structural aspects of the data stream are:

- Each data stream includes one or more data columns.
- Each column has a datatype. The Aleri Streaming Platform supports the following data types:

string	variable length character string
int32	32-bit integer
int64	64-bit integer
money	fixed-point number for which you can specify the number of digits of precision (four by default)
double	eight-byte floating-point number (IEEE double precision)
date	date/time (64-bit date and time field for 64-bit machines, and 32-bit field for 32-bit machines), represented as the number of seconds since the epoch (1970-01-01 00:00:00+00, or 1 January 1970 at midnight UTC)
timestamp	date/time in milliseconds (64-bits), represented as the number of milliseconds since the epoch (1970-01-01 00:00:00.000+00, or 1 January 1970 at midnight UTC)

- In most streams, each row has a unique key value. The Source Stream definition designates one or more columns as "key" columns.

3.1.4.2. Details of Incoming RMDS Data

When the Adapter subscribes to RMDS for a certain RIC, RMDS first sends an initial image containing all available market data for that RIC. After that, RMDS only sends out an update when any value(s) for a subscribed RIC change. Each update consists of the identifying RIC, with the Field Identifier (FID) and the new value for each change.

Each FID defined for RMDS has a datatype. The following section shows how Reuters datatypes are supported.

3.1.4.3. Mapping Market Data Fields

Each column in the target Aleri Streaming Platform stream must be mapped to a Reuters FID or a "pseudo-field" (see below for more information). For data fields, it's a matter of finding the appropriate FID for each column in the stream.

The datatype of the Aleri Streaming Platform column must be compatible with the datatype of the Reuters FID that feeds it. The following table gives possible matches between FID datatypes and Aleri Streaming Platform datatypes:

Reuters Data Type	**Aleri Streaming Platform Data Type**
INTEGER	int32 or int64
ALPHANUMERIC	string
ENUMERATED	int32
TIME, DATE	timestamp or date
PRICE	money or double
TIME_SECONDS	int64 or int32
BINARY	not supported

3.1.4.4. Mapping the Incoming RIC

The identifier of each incoming RMDS update is the Reuters Instrument Code (RIC). The RIC must be mapped to a column of datatype `string` in the stream (see below for details).

If the stream you want to map to does not have a suitable column, you must either add a column to the stream or map to a different stream.

3.1.4.5. Matching the Stream's Key

The Adapter Map file must configure the Adapter so that every update sent to the Aleri Streaming Platform stream includes a field or combination of fields conforming to the unique key defined for that stream. To make this more flexible, the Adapter configuration mechanism supports "pseudo-fields".

The market data updates that the Adapter receives from RMDS are mapped to columns in the Aleri Streaming Platform data stream using the `dataField` or `dateTimeField` element in the Map file (see below for details). RMDS also provides non-market data information: each update includes a RIC. In addition, the Adapter can be configured to add a sequence number to each update.

To make these data items available to the mapping process, the Map file mechanism supports the following six elements called "pseudo-fields".

Field	**Description**	**Datatype**
itemName	the RIC	string (required)
serviceName	the name of the service from which RMDS received the market data from this RIC	string (optional)
itemStale	the item state	int32 (optional)
sequenceNumber	a unique number, assigned sequentially by the Adapter to each incoming event (whether or not it is “interesting”, i.e., whether or not it causes an update)	int64 (optional)
FIDListField	shows the FID name and value for each updated value in the update	string (optional)
updateNumber	a unique number, assigned sequentially by the Adapter to each incoming update	int64 (optional)

The following example shows how these fields can be mapped to columns in the target stream.

3.2. Getting Stream Information from the Data Model

The first step in configuring the Inbound Adapter is to determine the source stream(s) on the Aleri Streaming Platform which will receive the RMDS Market Data. If the Aleri Streaming Platform data model does not already include one or more streams for this purpose, you must define a new stream (or streams) for use with the Reuters Adapter. If necessary, follow the procedure in the *Aleri Streaming Platform Authoring Guide* to define the new stream(s).

After you have chosen (or defined) the streams that will receive data from the Aleri Reuters Marketfeed Adapter, you must collect information about that stream from your data model file.

The Aleri Streaming Platform data model file contains one or more Stream Definitions. Each stream definition specifies a data stream that will be instantiated when the Aleri Streaming Platform is started up. The stream definition comprises:

- a unique ID for the stream
- a database store and/or output file for the stream data
- a list of the column(s) which will be used as the unique key value for each row in the data stream

Once you have decided which stream(s) will carry the RMDS data provided by the Reuters Adapter, you must get information from the stream definition in the data model file.

There is no standard for data model filenames. Two Aleri Streaming Platform installations may have completely different stream definitions. But, the definition of any stream includes the same basic set of components.

The following instructions refer to the example data model to show what components of the stream configuration you must identify in order to configure the Aleri Reuters Marketfeed Adapter.

Follow this procedure to gather the necessary information about the Reuters stream.

1. Open the data model to which the Adapter will be providing data. The example shown here is the `$ALERI_REUTERS_HOME/examples/example.model.xml` file supplied with the Aleri Reuters Marketfeed Adapter distribution.

```
<?xml version="1.0" encoding="UTF-8"?>

<Platform version="3.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Store id="Store1" index="hash" kind="memory"/>


  <SourceStream id="stream1" store="Store1" type="dynamic">
    <Column datatype="string" key="true" name="symbol"/>
    <Column datatype="string" key="false" name="service"/>
    <Column datatype="int32" key="false" name="seq"/>
    <Column datatype="int32" key="false" name="stale"/>
    <Column datatype="money" key="false" name="bid"/>
    <Column datatype="money" key="false" name="ask"/>
    <Column datatype="double" key="false" name="last"/>
    <Column datatype="int32" key="false" name="volume"/>
    <Column datatype="timestamp" key="false" name="when"/>
  </SourceStream>


  <SourceStream id="orderbookStream" store="Store1" type="dynamic">
    <Column datatype="string" key="true" name="symbol"/>
    <Column datatype="int32" key="false" name="PermissionCode"/>
    <Column datatype="int32" key="false" name="Stale"/>
    <Column datatype="int32" key="false" name="Currency"/>
    <Column datatype="date" key="false" name="TxnTimestamp"/>
    <Column datatype="int32" key="false" name="TxnTimestampTime"/>
    <Column datatype="money" key="false" name="AskPrice1"/>
    <Column datatype="money" key="false" name="AskSize1"/>
    <Column datatype="money" key="false" name="AskOrders1"/>
    <Column datatype="money" key="false" name="AskPrice2"/>
    <Column datatype="money" key="false" name="AskSize2"/>
    <Column datatype="money" key="false" name="AskOrders2"/>
```

```
    <Column datatype="money" key="false" name="AskPrice3"/>
    <Column datatype="money" key="false" name="AskSize3"/>
    <Column datatype="money" key="false" name="AskOrders3"/>
    <Column datatype="money" key="false" name="AskPrice4"/>
    <Column datatype="money" key="false" name="AskSize4"/>
    <Column datatype="money" key="false" name="AskOrders4"/>
    <Column datatype="money" key="false" name="AskPrice5"/>
    <Column datatype="money" key="false" name="AskSize5"/>
    <Column datatype="money" key="false" name="AskOrders5"/>
    <Column datatype="money" key="false" name="AskPrice6"/>
    <Column datatype="money" key="false" name="AskSize6"/>
    <Column datatype="money" key="false" name="AskOrders6"/>
    <Column datatype="money" key="false" name="AskPrice7"/>
    <Column datatype="money" key="false" name="AskSize7"/>
    <Column datatype="money" key="false" name="AskOrders7"/>
    <Column datatype="money" key="false" name="AskPrice8"/>
    <Column datatype="money" key="false" name="AskSize8"/>
    <Column datatype="money" key="false" name="AskOrders8"/>
    <Column datatype="money" key="false" name="AskPrice9"/>
    <Column datatype="money" key="false" name="AskSize9"/>
    <Column datatype="money" key="false" name="AskOrders9"/>
    <Column datatype="money" key="false" name="AskPrice10"/>
    <Column datatype="money" key="false" name="AskSize10"/>
    <Column datatype="money" key="false" name="AskOrders10"/>
    <Column datatype="money" key="false" name="BidPrice1"/>
    <Column datatype="money" key="false" name="BidSize1"/>
    <Column datatype="money" key="false" name="BidOrders1"/>
    <Column datatype="money" key="false" name="BidPrice2"/>
    <Column datatype="money" key="false" name="BidSize2"/>
    <Column datatype="money" key="false" name="BidOrders2"/>
    <Column datatype="money" key="false" name="BidPrice3"/>
    <Column datatype="money" key="false" name="BidSize3"/>
    <Column datatype="money" key="false" name="BidOrders3"/>
    <Column datatype="money" key="false" name="BidPrice4"/>
    <Column datatype="money" key="false" name="BidSize4"/>
    <Column datatype="money" key="false" name="BidOrders4"/>
    <Column datatype="money" key="false" name="BidPrice5"/>
    <Column datatype="money" key="false" name="BidSize5"/>
    <Column datatype="money" key="false" name="BidOrders5"/>
    <Column datatype="money" key="false" name="BidPrice6"/>
    <Column datatype="money" key="false" name="BidSize6"/>
    <Column datatype="money" key="false" name="BidOrders6"/>
    <Column datatype="money" key="false" name="BidPrice7"/>
    <Column datatype="money" key="false" name="BidSize7"/>
    <Column datatype="money" key="false" name="BidOrders7"/>
    <Column datatype="money" key="false" name="BidPrice8"/>
    <Column datatype="money" key="false" name="BidSize8"/>
    <Column datatype="money" key="false" name="BidOrders8"/>
    <Column datatype="money" key="false" name="BidPrice9"/>
    <Column datatype="money" key="false" name="BidSize9"/>
    <Column datatype="money" key="false" name="BidOrders9"/>
    <Column datatype="money" key="false" name="BidPrice10"/>
    <Column datatype="money" key="false" name="BidSize10"/>
    <Column datatype="money" key="false" name="BidOrders10"/>
  </SourceStream>


</Platform>
```

2. Find the name of the Source Stream. The opening `SourceStream` tag specifies the name of the stream as the value of the `id` attribute. The first Source Stream in this example is named “stream1”.

Note:

The stream used for subscription by the Aleri Reuters Marketfeed Adapter must always be a Source Stream.

3. Determine the key field(s). Examine each of the `Column` entries between the opening and closing `SourceStream` tags to see if the `key` attribute is set to “true”. In this example, “stream1”, has one key field: “symbol”.

4. Carefully note the number and order of the `Column` entries in the Source Stream definition.

   In the Inbound Adapter Map File, you will list the same set of data, in the same order.

3.3. Creating the Inbound Adapter Map File

This section describes how to create an Adapter Map file. All of the examples shown in the following procedure map updates from RMDS to the Source Stream defined in the `example.model.xml` file shown previously.

1. Using your favorite editor, open a new map file.

2. Enter the following as the first line of the file to specify that the Adapter Map File conforms to XML version 1.0.

```
<?xml version="1.0" encoding="UTF-8"?>
```

3. Enter the following document type declaration to specify that this is an Adapter Map File and that there is a separate file that it includes.

```
<!DOCTYPE adapter [
  <!ENTITY rmdsFields SYSTEM "rmds.sm.mf.xml">
]>
```

4. Add the opening and closing `adapter` tags. In the opening `adapter` tag, specify the name of the adapter. For example,

```
<adapter name="mySubscribeAdapter1">
</adapter>
```

5. After the opening `adapter` tag, add the `publication` element. Specify the name to be used in log messages for this adapter (see Section E.1, “Adapter Logging” for details) and any other attributes required to prescribe how the Adapter should deliver data to the Aleri Streaming Platform. For example,

```
      <publication name="RMDS Adapter - low latency" retryInterval="5" />
```

   This example also includes a *`retryInterval`* attribute with a value that tells the Adapter to wait five seconds before retrying if it fails to connect to the Aleri Streaming Platform.

6. After the `publication` element, add the opening and closing `streamMaps` tags to contain the `streamMap` elements that do the actual mapping between RMDS FIDs and columns of an Aleri Streaming Platform stream. Each `streamMap` maps to one and only one Aleri Streaming Platform stream.

```
        <streamMaps>
        </streamMaps>
```

   Since the *`streamMaps`* section can contain more than one *`streamMap`*, one instance of the Adapter can provide RMDS data to more than one Aleri Streaming Platform stream.

7. Enter a `streamMap` element for each Aleri Streaming Platform stream to which you wish to send RMDS data. For each streamMap,

   - Enter the opening `streamMap` tag specifying the name of the Aleri Streaming Platform stream to which the RMDS data will be sent as the value of the *`name`* attribute.

     You may also specify the *`flags`* attribute, which controls what happens to fields for which there is no new data in an update. By default, it is set to *`SHINE`* and fields for which there is no new data are untouched by the update and previous values shine through. But this is not always desirable. For example, for flattened order book data when there are ten bidders and three of them drop out, the absence of those FIDs in the update is new data. Shine-through would leave the old bids in place. In cases like this you should set the attribute *`flags`* to *`NO_SHINE`*.

   - Enter the closing `streamMap` tag.

   - Between the `streamMap` tags, you must add one mapping element for each column defined in the target

stream's definition. You can do this in the map file itself, or in a separate file that is included in the map file as an entity.

```
<streamMap name="stream1" flags="NO_SHINE">
&rmdsFields;
</streamMap>
```

8. After the `streamMaps` section, add the `rfa` element, including:

   - A `config` attribute that specifies the absolute path and filename of the Reuters configuration file.
   - A `sessionName` attribute that specifies a session name corresponding to the one used in the Reuters configuration file.

```
<rfa config="$ALERI_REUTERS_HOME/config/rfasub.cfg"
        sessionName="Session1" />
```

   The `rfa` element may also include attributes to modify the adapter's treatment of blanks (by default it converts them to zeroes). You may specify the value for the `blank` attribute, or you may specify values for each each datatype directly, using the `blankInt32`, `blankInt64`, `blankMoney`, `blankString`, `blankDate`, and `blankTimestamp` attributes. You should specify a value that does not conflict with any of the values you expect in your data. If you are using both inbound and outbound adapters, be sure to specify the same value for each attribute to both adapters.

9. Between the `rfa` element and the closing `adapter` tag, add the opening and closing `itemLists` tags. When entering the opening `itemLists` tag,

   - Specify the Reuters service from which the Adapter is receiving RMDS data as the value of the *`service`* attribute.
   - Specify the name of the Aleri Streaming Platform stream that is receiving the RMDS data as the value of the *`stream`* attribute.

```
<itemLists service="IDN_RDF" stream="stream1">
</itemLists>
```

   The `itemLists` tags will contain one or more pairs of opening and closing `itemList` tags.

10. Between the `itemLists` tags, add opening and closing `itemList` tags for each separate list of RICs to which the Adapter will subscribe.

11. Between the `itemList` tags, add an `item` element for each RIC you wish to add to the list. When entering the `item` element,

    - Specify a RIC to which the Adapter will subscribe as the value of the *`name`* attribute.
    - Optionally, specify the name of the queue you wish to use as the value of the *`rfaQueue`* attribute. Specifying an *`rfaQueue`* spawns a separate thread to do the processing.
    - Optionally, specify the name of the service you wish to use.

    For example:

```
<itemList>
        <item name="AAPL.O" rfaQueue="queue1" />
        <item name="CSCO.O" />
</itemList>
```

3.4. Running the Inbound Adapter

After you have configured the Adapter:

1. Make sure the Aleri Streaming Platform is up and running. Enter the following command on the machine where the Aleri Streaming Platform is installed:

```
ps -eaf | grep sp
```

   Or, if you are running the optimized production version of the Streaming Processor, enter the following command:

```
ps -eaf | grep sp-opt
```

2. If the Aleri Streaming Platform is running with RSA authentication, start the Adapter with the following command:

```
rmds -a inbound -f mapfile -p platform_host:port \
-c platform_username:platform_password -k private_rsa_key_file -d 7
```

3. If not, start the Adapter with the following command:

```
rmds -a inbound -e -f adapter_def -p platform_host:port \
 -c platform_username:platform_password -d 7
```

4. The Adapter starts up the subscription by first connecting to the Aleri Streaming Platform and then connecting to RMDS. Both connections must be operational for any data to flow.

If you plan to direct the Adapter's log output to stderr, as shown here, you may want to redirect stdout and stderr to a log file (e.g., append **`>& myrmdslog &`** to the command line shown above). For more information about the command line options, see Appendix A, *Command Usage* .

3.5. Testing the Adapter

If the Adapter is not working as expected, you can perform a quick sanity check by executing **rmds** as follows:

```
rmds -v
```

This command returns the version information. Make sure that:

- This *Reuters Marketfeed Adapter User's Guide* is for your version of the Adapter.
- The Aleri Streaming Platform to which you are connecting is compatible with your version of the Adapter. (Check this in Section 1.2, “Compatibility with the Aleri Streaming Platform”.)

There are three quick ways to see whether the Aleri Reuters Marketfeed Adapter is sending Reuters Market Data to the Aleri Streaming Platform:

- Use the Aleri Studio or the **sp_subscribe** command to check the output of the stream configured to receive Reuters data.
- Use the `tail` command on the redirected Adapter log file (specified in the Adapter Map file) or the Reuters subscriber log (specified in the configuration file `rfasub.cfg`) for activity.
- Run the **rmds** command with the `-d7` option to produce verbose output.

See Appendix A, *Command Usage* for more information about the `rmds` command.

Chapter 4. Configuring an Outbound Marketfeed Adapter

When the Reuters Adapter is used to push data from the Aleri Streaming Platform to RMDS (using RMDS as a message infrastructure) it is called an Outbound Adapter. This chapter shows how to configure an Outbound Adapter.

4.1. Preparation

Before configuring an Outbound Adapter, you must decide which data you wish to provide and how you want to set up your system.

4.1.1. Data Decisions

The first step is to identify the Aleri Streaming Platform data to be published, specifically, which columns from which streams to publish.

The Reuters Adapter can rearrange the columns from a stream in any order. Its output can also include constants. Additionally, the published output can include values from more than one stream.

When planning the output of the Reuters Outbound Adapter, keep the following issues in mind:

- For each stream whose data you want to publish, you must be able to specify a unique key in the Outbound Adapter map file. Since this Adapter sends data to RMDS, the unique identifier should be a RIC.
- Each data column you want to publish from any stream should map to a unique FID.
- Data from one column can be repeated in the published output, giving you a way to publish a DateTime value as separate Date and Time values.
- If the stream you are working with receives data about the same FID from more than one service, you can configure the Adapter to differentiate these data items by service and transmit each service's data separately.
- The first time the Reuters Adapter publishes to RMDS, it publishes values for all the columns for which it is configured. After that initial image, the Adapter only publishes updates for individual columns as these updates occur.

4.1.2. Administrative Decisions

There are administrative decisions in addition to those related to the data model.

- *Session Name*

  This is an arbitrary string used to link the data model and the Adapter map file, but it must be used consistently.

- *Directories for logging and stream output*

  The Adapter writes its own log messages and can generate a separate set of Reuters log messages. In the configuration, you must specify if and where these log files should be written.

- *Aleri user account*

  You must specify a valid user account for the Adapter to use. See Section 2.2, “Enabling User Access” for details on setting up the selected account to use the Adapter.

4.1.3. Aleri Streaming Platform Information

You will need to know the following about the Aleri Streaming Platform instance from which you will be receiving data.

- Does it run all Command and Control I/O through a Secure Socket Layer (i.e., was it started with the `-e` flag)?
- What type of authentication mechanism (Kerberos, PAM, RSA, or none) does it use?

4.1.4. Reuters Information

To enable the Aleri Reuters Marketfeed Adapter to publish to the RMDS, you will need the following information from Reuters:

- the name of the Reuters service on which the Adapter will transmit data
- up-to-date lists of valid Reuters Instrument Codes (RICs) and Field Identifiers (FID) used by RMDS
- the Product Permission Code assigned by Reuters

The Adapter does not work with the Reuters Data Access Control System (DACS). Hence the inclusion of the Product Permission Code in the previous list: to provide a mechanism for allowing access to the information you are transmitting on the RMDS.

A list of FIDs, `$ALERI_REUTERS_HOME/config/appendix_a`, has been supplied as part of the Reuters Adapter distribution. You can obtain the latest list and other information from your Reuters technical contact.

The datatype of the Aleri Streaming Platform column must be compatible with the Reuters FID datatype that feeds it. The following table shows possible matches between Aleri Streaming Platform and FID datatypes:

Aleri Streaming Platform Data Type	**Reuters Data Type**
INT32, INT64	INTEGER or PRICE
MONEY, DOUBLE	PRICE
STRING	ALPHANUMERIC
DATE, TIMESTAMP	DATE and/or TIME

4.2. Getting Stream Information from the Data Model

The first step in configuring the Outbound Adapter is determining which data elements from which streams on the Aleri Streaming Platform are to be published.

See Section 4.1.1, “Data Decisions” for more information.

After you have chosen (or defined) a data model containing the items for publication over RMDS via the Reuters Adapter, you must collect information from the stream(s) from which you wish to obtain the data to send to RMDS.

Each stream definition specifies a data stream that will be instantiated when the Aleri Streaming Platform is started up. The stream definition:

- specifies a unique ID for the stream
- identifies the column(s) which will be used as the unique key value for each row in the data stream

Once you have decided which stream(s) will provide the information to be sent to RMDS by the Reuters Adapter, you must get information from the stream definition in the data model file.

There is no standard for data model filenames. Two Aleri Streaming Platform installations may have completely different stream definitions. But, the definition of any stream includes the same basic set of components.

Follow this procedure to gather the necessary information from the data model.

1. Open the data model from which the Adapter will be obtaining data. The example shown here is the `$ALERI_REUTERS_HOME/examples/example.model.xml` file supplied with the Aleri Reuters Marketfeed Adapter distribution.

```
<?xml version="1.0" encoding="UTF-8"?>

<Platform version="3.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Store id="Store1" index="hash" kind="memory"/>


  <SourceStream id="stream1" store="Store1" type="dynamic">
    <Column datatype="string" key="true" name="symbol"/>
    <Column datatype="string" key="false" name="service"/>
    <Column datatype="int32" key="false" name="seq"/>
    <Column datatype="int32" key="false" name="stale"/>
    <Column datatype="money" key="false" name="bid"/>
    <Column datatype="money" key="false" name="ask"/>
    <Column datatype="double" key="false" name="last"/>
    <Column datatype="int32" key="false" name="volume"/>
    <Column datatype="timestamp" key="false" name="when"/>
  </SourceStream>


  <SourceStream id="orderbookStream" store="Store1" type="dynamic">
    <Column datatype="string" key="true" name="symbol"/>
    <Column datatype="int32" key="false" name="PermissionCode"/>
    <Column datatype="int32" key="false" name="Stale"/>
    <Column datatype="int32" key="false" name="Currency"/>
    <Column datatype="date" key="false" name="TxnTimestamp"/>
    <Column datatype="int32" key="false" name="TxnTimestampTime"/>
    <Column datatype="money" key="false" name="AskPrice1"/>
    <Column datatype="money" key="false" name="AskSize1"/>
    <Column datatype="money" key="false" name="AskOrders1"/>
    <Column datatype="money" key="false" name="AskPrice2"/>
    <Column datatype="money" key="false" name="AskSize2"/>
    <Column datatype="money" key="false" name="AskOrders2"/>
    <Column datatype="money" key="false" name="AskPrice3"/>
    <Column datatype="money" key="false" name="AskSize3"/>
    <Column datatype="money" key="false" name="AskOrders3"/>
    <Column datatype="money" key="false" name="AskPrice4"/>
    <Column datatype="money" key="false" name="AskSize4"/>
    <Column datatype="money" key="false" name="AskOrders4"/>
    <Column datatype="money" key="false" name="AskPrice5"/>
    <Column datatype="money" key="false" name="AskSize5"/>
    <Column datatype="money" key="false" name="AskOrders5"/>
    <Column datatype="money" key="false" name="AskPrice6"/>
    <Column datatype="money" key="false" name="AskSize6"/>
    <Column datatype="money" key="false" name="AskOrders6"/>
    <Column datatype="money" key="false" name="AskPrice7"/>
    <Column datatype="money" key="false" name="AskSize7"/>
    <Column datatype="money" key="false" name="AskOrders7"/>
    <Column datatype="money" key="false" name="AskPrice8"/>
    <Column datatype="money" key="false" name="AskSize8"/>
    <Column datatype="money" key="false" name="AskOrders8"/>
    <Column datatype="money" key="false" name="AskPrice9"/>
    <Column datatype="money" key="false" name="AskSize9"/>
    <Column datatype="money" key="false" name="AskOrders9"/>
    <Column datatype="money" key="false" name="AskPrice10"/>
    <Column datatype="money" key="false" name="AskSize10"/>
    <Column datatype="money" key="false" name="AskOrders10"/>
    <Column datatype="money" key="false" name="BidPrice1"/>
    <Column datatype="money" key="false" name="BidSize1"/>
    <Column datatype="money" key="false" name="BidOrders1"/>
    <Column datatype="money" key="false" name="BidPrice2"/>
    <Column datatype="money" key="false" name="BidSize2"/>
    <Column datatype="money" key="false" name="BidOrders2"/>
    <Column datatype="money" key="false" name="BidPrice3"/>
    <Column datatype="money" key="false" name="BidSize3"/>
    <Column datatype="money" key="false" name="BidOrders3"/>
    <Column datatype="money" key="false" name="BidPrice4"/>
    <Column datatype="money" key="false" name="BidSize4"/>
    <Column datatype="money" key="false" name="BidOrders4"/>
    <Column datatype="money" key="false" name="BidPrice5"/>
    <Column datatype="money" key="false" name="BidSize5"/>
    <Column datatype="money" key="false" name="BidOrders5"/>
    <Column datatype="money" key="false" name="BidPrice6"/>
```

```
    <Column datatype="money" key="false" name="BidSize6"/>
    <Column datatype="money" key="false" name="BidOrders6"/>
    <Column datatype="money" key="false" name="BidPrice7"/>
    <Column datatype="money" key="false" name="BidSize7"/>
    <Column datatype="money" key="false" name="BidOrders7"/>
    <Column datatype="money" key="false" name="BidPrice8"/>
    <Column datatype="money" key="false" name="BidSize8"/>
    <Column datatype="money" key="false" name="BidOrders8"/>
    <Column datatype="money" key="false" name="BidPrice9"/>
    <Column datatype="money" key="false" name="BidSize9"/>
    <Column datatype="money" key="false" name="BidOrders9"/>
    <Column datatype="money" key="false" name="BidPrice10"/>
    <Column datatype="money" key="false" name="BidSize10"/>
    <Column datatype="money" key="false" name="BidOrders10"/>
  </SourceStream>


</Platform>
```

2. From the definition of each stream defined in the model,

   a. Obtain the name of the stream from the *id* attribute in the opening tag of that stream.

   b. Verify that the key attribute is set to “true” for the column containing the RIC and note the column. In this example, both “stream1” and “orderbookStream” have the RIC in the column named “symbol” which is identified as a key field.

   c. Decide what data, if any, you want the Adapter to send to RMDS.

3. Carefully note which stream(s) contain data you wish to send to RMDS, and where in the stream definition it is located.

   In the Outbound Adapter Map File, you will reference each of the columns you want to publish.

4.3. Creating the Outbound Adapter Map File

This section describes how to create an Adapter Map file. All of the examples shown in the following procedure map updates from RMDS to the Source Stream defined in the example.model.xml file shown previously.

1. Using your favorite editor, open a new map file.

2. Enter the following as the first line of the file to specify that the Adapter Map File conforms to XML version 1.0.

```
<?xml version="1.0" encoding="UTF-8"?>
```

3. Add the opening and closing adapter tags.

```
<adapter>
</adapter>
```

4. Define the configuration of the Adapter's interface to RMDS by adding the rfa tag, with the following attributes:

   config — Specify the full pathname of the Reuters configuration file (see Section 2.3.2, “Configuring an Outbound Connection to Reuters”).

   fidFile — Specify the full pathname of the Reuters supplied file that lists all of the valid FIDs.

enumFile Specify the full pathname of the Reuters supplied file that lists each enumerated type along with the range of values it can take.

serviceName Specify the service name provided by Reuters for the Adapter to send data to RMDS.

sessionName Specify the `sessionName` value found in the Reuters configuration file, `rfasub.cfg`.

For example, using the files that were shipped with the Adapter distribution:

```
<rfa config="$ALERI_REUTERS_HOME/config/rfapub.cfg"
     fidFile="$ALERI_REUTERS_HOME/config/appendix_a"
     enumFile="$ALERI_REUTERS_HOME/config/enumtype.def"
     serviceName="IDN_RDF" sessionName="Session1" />
```

5. Add the `subscriptions` begin and end tags between the `rfa` element and the closing `adapter` tag. (The Adapter subscribes to the Aleri Streaming Platform to get the data to publish to the RMDS.)

```
<subscriptions>
</subscriptions>
```

6. Between the opening and closing `subscriptions` tags, add opening and closing `subscription` tags to define a subscription. Include the following attributes in the opening `subscription` tag:

name Specify a unique name for this subscription.

flags Set this parameter equal to “BASE” to obtain a complete set of initial values. This may be undesirable in situations such as recovery if there are a lot of unchanging values because getting those values would add latency to the other values. In these cases, set this parameter to “NO_BASE”.

```
<subscription name="subscription1" flags="BASE" >
</subscription>
```

Each subscription defined in the Outbound Adapter Map file must reference at least one Aleri Streaming Platform stream.

7. Add the stream definition to the subscription.

a. Immediately before the closing `subscription` tag, insert the opening and closing `stream` tags. In the opening `stream` tag, include the *name* attribute set to the name of the stream.

b. To use a “constant” rather than a column in an Aleri Streaming Platform stream to specify your Reuters permission code, insert the `constant` tag immediately before the closing `stream` tag, including the following attributes:

name Specify the Reuters FID “PROD_PERM”.

value Specify the permission code issued by Reuters that certifies your permission to publish to RMDS.

c. Immediately following the opening `stream` tag, insert the `name` tag, with the attribute *column* set to the column before the column with the symbol or RIC in the data model (e.g., if the symbol or RIC was in the first column in the data model, set the value of *column* to “0”.

d. Immediately following the opening `name` tag, insert the `stale` tag, with the attribute *column* set to one less than the position of the value in the data model.

e. Between the `stale` and the `constant` tags, add a `field` tag for each data column in the stream that

you want to send to RMDS. Include the following attributes:

column Set this parameter to either the name of the column or the numeric position (one less than the position of the value in the data model).

name Specify the Reuters FID for this data.

For fields of datatype "double", you may also include the *precision* attribute, set to the number of digits you want after the decimal place in the value sent to RMDS.

For example,

```
<stream name="stream1" >
  <name column="0"/>
  <stale column="3" />
  <field column="4" name="BID" precision="5" />
  <field column="5" name="ASK" precision="0" />
  <field column="6" name="TRDPRC_1"/>
  <field column="7" name="ACVOL_1"/>
  <constant name="PROD_PERM" value="1"/>
</stream>
```

4.4. Running the Outbound Adapter

After you have configured the Adapter:

1. Make sure the Aleri Streaming Platform is up and running. Enter the following command on the machine where the Aleri Streaming Platform is installed:

```
ps -eaf | grep sp
```

2. Start the Adapter with the following command:

```
rmds -a outbound -f ADAPTER_DEF -p HOST:9100 -c USER:PASSWORD -d 7
```

The exact usage of the command depends on how you started your Aleri Streaming Platform. The Adapter must be invoked with compatible options. The command string shown invokes neither encryption nor authentication: either or both can be specified. For more information about the options to **rmds**, see Appendix A, *Command Usage*.

Note:

If you plan to direct the Adapter's log output to stderr, as shown here, you may want to redirect stderr to a log file (e.g., append **>& myrmdslog &** to the command line shown above).

4.5. Testing the Adapter

If the Adapter is not working as expected, you can perform a quick sanity check by executing **rmds** as follows:

```
rmds -v
```

This command returns the Adapter release number and the revision number of the source tree separated by an underscore character. Make sure that:

- Your version of the Reuters Adapter is covered by this version of the Aleri Reuters Marketfeed Adapter User's Guide. The release number and the revision number are on the title page (immediately following the cover page).
- Your version of the Adapter is compatible with your version of the Aleri Streaming Platform. (Check this in Section 1.2, “Compatibility with the Aleri Streaming Platform”.)

There are several ways to see whether the Aleri Reuters Marketfeed Adapter is publishing to RMDS:

- Use the **tail** command on the Adapter log file to which console output was redirected or any of the Reuters publisher log files (specified in `rfapub.cfg`) to look for activity.
- Use the **sp_subscribe** command to look at the outbound stream and verify that values are changing.
- Use RMDS tools to subscribe to RICs provided by the Outbound Adapter.
- Use an Inbound Adapter to subscribe to the Outbound Adapter.

Chapter 5. Administering Marketfeed Adapters

5.1. Specifying Multiple RICs

When configuring an Inbound Adapter, you will usually want to specify multiple RICs. There are several ways to do this:

- Specifying each individual RIC by entering the name directly into the Map file or using an XML ENTITY include file.
- Specifying a chain RIC from Reuters.
- Creating a Dynamic Watch List which employs the Aleri Streaming Platform to specify the list of RICs.
- Using a combination of the above options.

5.1.1. Entering Each Individual RIC

You can enter an *item* element declaration for each RIC you want in the *itemList* section of the Map file as shown in the following example.

```
<itemLists service="SSL_PUB" stream="stream1">
  <itemList>
    <item name="CSCO.O"/>
    <item name="K.N"/>
    <item name="KBN.N"/>
    <item name="KBR.N"/>
    <item name="ACAM.ARC"/>
    <item name="IBM.ARC"/>
  </itemList>
</itemLists>
```

But, it can become difficult to create and maintain your list of RICs this way if it is very large and/or changes frequently (e.g., all of the stocks traded on the NYSE).

All RICs for the same stream must use the same FID set. Since FIDs often vary by venue, you should use a different itemList and streamMap for each venue.

5.1.2. Using a Chain RIC

The second method uses a chain RIC rather than mapping each individual RIC. When you specify the name of a chain RIC, Reuters translates it to a list of individual RICs. Chain RICs usually contain all of the RICs from a single market or for a single index instrument such as the S&P 500 or the Russell 2000.

For example, to specify the chain RICs for the Dow Jones Index and the SIAC entities, you would add a *chainMap* section

```
        <streamMap name="chainMap" chain="1" >
                <itemName />                    <dataField name="REF_COUNT" />
                <dataField name="NEXT_LR" /> <dataField name="PREF_LINK" />
                <dataField name="LINK_1" /> <dataField name="LINK_2" />
                <dataField name="LINK_3" /> <dataField name="LINK_4" />
                <dataField name="LINK_5" /> <dataField name="LINK_6" />
                <dataField name="LINK_7" /> <dataField name="LINK_8" />
                <dataField name="LINK_9" /> <dataField name="LINK_10" />
                <dataField name="LINK_11" /> <dataField name="LINK_12" />
                <dataField name="LINK_13" /> <dataField name="LINK_14" />
                <dataField name="LONGNEXTLR" /> <dataField name="LONGPREVLR" />
                <dataField name="LONGLINK1" /> <dataField name="LONGLINK2" />
                <dataField name="LONGLINK3" /> <dataField name="LONGLINK4" />
                <dataField name="LONGLINK5" /> <dataField name="LONGLINK6" />
                <dataField name="LONGLINK7" /> <dataField name="LONGLINK8" />
                <dataField name="LONGLINK9" /> <dataField name="LONGLINK10" />
                <dataField name="LONGLINK11" /> <dataField name="LONGLINK12" />
                <dataField name="LONGLINK13" /> <dataField name="LONGLINK14" />
        </streamMap>
```

and enter their names in the *itemList* section.

```
<itemList stream="stream1" service="IDN_RDF" >
        <item name="0#.DJI" /> <!-- The Dow Jones Index -->
        <item name="0#SIAC" /> <!-- The entities of SIAC -->
</itemList>
```

For more details about chains, look at the example in `chain.example.map.xml` (which was put in the `$ALERI_REUTERS_HOME/examples` directory during Aleri Streaming Platform installation. For more information about Reuters chain RICs, refer to the *Reuters Venue Guide* for your chosen venue (which is available from Reuters).

5.1.3. Creating a Dynamic Watch List

The third method for specifying multiple RICs is to have the Adapter get its symbol list from the Aleri Streaming Platform. Creating a Dynamic Watch List is a bit more complex, but also more flexible. Chain RICs are limited to those defined by Reuters, but with this method you can specify your own customized list of RICs.

This method is also dynamic: when inserts or deletes occur on the stream configured using the following procedure, RMDS subscriptions to the appropriate RICs are started or stopped.

The following procedure assumes an already defined Source Stream (named MyInfoStream) to receive the data. And that you have used a manually edited list of RICs as described in Section 5.1.1, “Entering Each Individual RIC”

1. Define a stream on the Aleri Streaming Platform (e.g., *MyListStream*) which will publish to the Adapter the list of RICs to which you wish to subscribe. This stream must include the following three columns:

   symbol — Specifies a RIC symbol ticker (e.g., *CSCO.O*) to which the Adapter should subscribe.

   service — Specifies the RMDS service on which to subscribe to obtain data for that RIC.

   stream — Specifies the name of the stream (e.g., *MyInfoStream*) on which the Adapter will publish data for this RIC.

   It can also include an optional fourth column, rfaQueue.

   Refer to the *Aleri Streaming Platform Authoring Guide* for details on how to define a stream.

2. Define a second stream on the Aleri Streaming Platform (e.g., *MyInfoStream*) that will receive the data requested by the first stream.

3. Edit the Map file to include the Subscription.

```
<subscriptions>
  <subscription name="subscription1" flags="BASE" >
    <stream name="MyListStream" >
      <name column="3" /> <!-- symbol -->
      <field column="1" name="service"/>
      <field column="2" name="stream"/>
    </stream>
  </subscription>
</subscriptions>
```

4. Specify the set of RICs you want and send them to the first stream you created (e.g., *MyListStream*) to subscribe to them.

   a. Create a file with the same six columns that the stream expects in comma separated values (CSV) format. The columns are as follows:

      1. stream from which you are receiving data

2. opcode
3. placeholder (required to match the definition)
4. service
5. symbol
6. stream to which you are sending data

For example, open a new file, `RIClist.csv`, using your favorite editor and put in the following lines.

```
MyListStream,p,,IDN_RDF,MyInfoStream,CSCO.O
MyListStream,p,,IDN_RDF,MyInfoStream,K.N
MyListStream,p,,IDN_RDF,MyInfoStream,KBN.N
MyListStream,p,,IDN_RDF,MyInfoStream,KBN.R
MyListStream,p,,IDN_RDF,MyInfoStream,ACAM.ARC
MyListStream,p,,IDN_RDF,MyInfoStream,IBM.ARC
```

b. Send the data from the file to the Aleri Streaming Platform using the **sp_convert** and **sp_upload** commands. The following example assumes that you have installed all Aleri command line tools in the default directories and added those directories to your `PATH` variable. If you haven't, you must prepend the appropriate path to each command shown in this example.

For example, to send the file created in the previous step to an Aleri Streaming Platform running on port 11180 of your local server, enter the following command:

```
cat RIClist.csv | sp_convert -c user:password -d "," -p 11180 |\
 sp_upload -c user:password -p 11180
```

c. Start the Adapter.

```
rmds -f mapfile -d7 -c user:password -p 11180 >& logfile &
```

If the Adapter and the Aleri Streaming Platform are on different machines, you must enter the name of the host and the port number separated by `:` after the *`-p`* in the previous command.

5.2. Performance Tuning

Speed has at least two components: throughput and latency. In an inbound adapter, the Publication element has four attributes you can use to fine-tune performance.

`flushInterval` Specify an interval of time in microseconds (e.g., 5000 microseconds = 5 milliseconds) to wait while accumulating data. At the end of this interval, any accumulated events are sent to the Aleri Streaming Platform. Sending events less often allows more events to be placed into a message resulting in a communications overhead savings. Using a non-zero *`flushInterval`* makes the accumulation of events time-based.

`maxRecordsPerBlock` Specify the maximum number of accumulated events that the Adapter should send to the Aleri Streaming Platform at a time. When the number of accumulated events is larger than this value, the envelope or transaction is broken into fragments that are less than or equal to the specified value. For example, if accumulated event counts of over 1024 (which would immediately fill the Aleri Streaming Platform Gateway's inbound queue) are expected, you can set *`maxRecordsPerBlock`* to a value like *`500`* to

prevent this.

pendingLimit Specify a threshold for how many events must accumulate before being sent to the Aleri Streaming Platform. Setting this parameter to zero causes the Adapter to publish each event immediately when it happens (providing the lowest latency) but, at the expense of high network overhead (a TCP/IP packet for each update). If you set this parameter to a larger value, the Adapter will wait until that many events have accumulated, pack them efficiently in TCP/IP packets and send them to the Aleri Streaming Platform. This saves communication work but increases latency on both the Adapter and the Aleri Streaming Platform.

sendAsTransactions This parameter controls whether events are sent as an envelope or a transaction. You can specify this parameter on a per-stream basis.

Setting this parameter to **true** causes the Aleri Streaming Platform to treat a group of events as a single transaction. Transactions typically cause application-level workload savings, since the Aleri Streaming Platform will collapse multiple events to the same value (as determined by identical key columns) in a transaction to a single event. And, if the transaction contains a delete, more savings are achieved since updates prior to the delete can be discarded.

If you set this parameter to **false** and you're not in low-latency mode (*pendingLimit* and *flushInterval* both set to zero) then you use the *maxRecordsPerBlock* to control the size of the envelope. You'll get the communications overhead savings mentioned above, but not the transactional savings. This would be the preferred configuration for applications requiring each and every event, such as a market data compliance application.

As a general rule, for Quote-based applications, where only the most recent update matters, Transactions can be much more efficient. For Trades, where each and every event must be processed to compute a total volume, Envelopes should be used.

When both *flushInterval* and *pendingLimit* are used, no event will wait more than *flushInterval* before being sent and as long as *pendingLimit* (or more) events arrive they will be sent immediately. The Adapter waits *flushInterval* and, if any events have accumulated, it sends them. If *pendingLimit* or more events accumulate while the Adapter is sending the earlier events it sends the new events immediately (i.e., without waiting a *flushInterval*). If fewer than *pendingLimit* events accumulate while the Adapter is sending events it will pause and wait for *flushInterval* to elapse.

You can also use the *rfaQueue* attribute at the *itemLists*, *itemList*, or *item* element level. When specified, this attribute causes the element to be subscribed from Reuters on a named *rfaQueue*. Each *rfaQueue* is processed by its own thread within the Reuters Adapter. Spreading requests across multiple threads can reduce latency and improve overall adapter throughput at the cost of greater CPU usage.

Since all events (images and updates) for the same RIC come from Reuters on the same queue, the integrity of the order of arrival will be maintained for any individual RIC.

If you do not specify an *rfaQueue* for any of the elements, a single default queue (named "default") is used for all RICs.

5.3. Splitting Up An Adapter Map File

In some cases, it may be advantageous to put part of your Inbound or Outbound Adapter Map file in a separate file. For example, you might want to keep a Subscription configuration in a map file, but break out the list of RICs you want the Adapter to subscribe to.

The following procedures show how to put part of your Adapter mapping into a separate file, and how to reference this file from the main Map file.

5.3.1. Preparing the Subordinate Map File

Follow this procedure to create a “subordinate” file to hold part of the Map file configuration.

1. Go to the directory that contains the Map File.
2. Create a new file, with the extension `.xml`. It is not necessary to add a declaration of the XML version.
3. Insert the selected content from the Map file into the new file.

   The content you add depends on which part of the Map file you have decided to store separately. See the example below.
4. Add a comment to the new file, if you wish.
5. Save the file when you are done.

5.3.2. Modifying the Main Map File

Follow this procedure to modify the main Map file to reference the subordinate file(s).

1. Make sure the first line of the main Map file is:

```
<?xml version="1.0"?>
```

2. Add the following lines between the XML version declaration and the opening `adapter` tag:

```
  <!DOCTYPE adapter SYSTEM "adapter.dtd" [

    ]>
```

3. For each subordinate Map file,

   a. Add an entry like the following between the two lines just added.

```
    <!ENTITY SUBREF SYSTEM "SUBFILE">
```

   Where **SUBREF** is a string you will use to reference the subordinate file and **SUBFILE** is the path and filename of the subordinate file itself. Be sure to enclose the path and filename in quotation marks.

   b. Remove the content that you put in the subordinate Map file.

   c. Insert a string like the following to include the content from the subordinate Map file.

```
                    &SUBREF;
```

   Where **SUBREF** is the string you specified to reference the subordinate file.

5.3.3. Example

The following Map File, `subInclude.map.xml`, configures the Inbound Adapter to reference two subordinate files, each containing a list of RICs for the Adapter to subscribe to.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE adapter [
  <!ENTITY RIClist1 SYSTEM "RIClist1.sm.mf.xml">
  <!ENTITY RIClist2 SYSTEM "RIClist2.sm.mf.xml">
  <!ENTITY rmdsFields SYSTEM "rmds.sm.mf.xml">
]>
<adapter>
```

```
  <publication name="RMDS Adapter" retryInterval="5"
     sendAsTransactions="0" flushInterval="1000" intraSubscribeDelay="100"/>
  <streamMaps>
    <streamMap name="stream1">
          &rmdsFields;
    </streamMap>
  </streamMaps>
  <rfa config="$ALERI_REUTERS_HOME/config/rmdsmf.cfg"
       sessionName="Inbound" />

  <itemLists>
    &RIClist1;
    &RIClist2;
  </itemLists>
</adapter>
```

The first file, `RIClist1.sm.mf.xml`, contains the following entries:

```
<!-- This fragment is meant to be included in an itemLists section.-->
<!-- These are FX RICs -->
    <itemList service="IDN_RDF" stream="stream1">
      <item name="GRMN.O"/>
      <item name="INTC.O"/>
      <item name="KLAC.O"/>
      <item name="XLNX.O"/>
      <item name="YHOO.O"/>
    </itemList>
```

The second file, `RIClist2.sm.mf.xml`, contains the following entries:

```
<!-- This fragment is meant to be included in an itemLists section.-->
<!-- These are FX RICs -->
    <itemList service="IDN_RDF" stream="stream1">
      <item name="AUD="/>
      <item name="CAD="/>
      <item name="DKKTN="/>
      <item name="GBPSW="/>
      <item name="GBPTN="/>
      <item name="JPYSN="/>
      <item name="JPYSW="/>
      <item name="JPYTN="/>
      <item name="HKD="/>
      <item name="SGDSW="/>
      <item name="ZAR="/>
      <item name="ZARSN="/>
    </itemList>
```

Appendix A. Command Usage

Name

rmds — The Aleri Reuters Marketfeed Adapter adapts data from the Reuters Market Data System (RMDS) to the Aleri Streaming Platform and vice versa.

Synopsis

`rmds` -f *`mapFile`* -p *`[host:]port`* [*`OPTION`* ...]

Description

rmds can operate as either a data source or sink (i.e., as either an Inbound or Outbound Adapter). An Inbound Adapter passes data from RMDS in to the Aleri Streaming Platform. An Outbound Adapter passes data from the Aleri Streaming Platform out to RMDS. A single Adapter instance cannot operate both ways. If you wish to have an Inbound Adapter and an Outbound Adapter, you must run two separate Adapter instances.

The metadata describing the connection has several parts, including a Map file, configuration file and possibly a configuration stream resident on a running instance of the Aleri Streaming Platform.

Only limited Level2 data is available via RMDS Marketfeed. For full orderbook depth, use the Reuters OMM Adapter (**rmdsomm**).

The **rmds** command runs the debug version of the RMDS Marketfeed Adapter and the **rmds-opt** command runs the optimized version. The optimized version is suitable for production use.

The process runs as a daemon, getting its configuration from a mapFile. It handles SIGHUP; so you can enter **`kill -s SIGHUP <pid>`** on Linux or **`kill -s HUP <pid>`** on Solaris (where <pid> is the process id of the rmds daemon, which can be obtained using the **ps** command) to shut the adapter down gracefully. Using the `KILL` signal rather than the `HUP` signal may prevent a complete clean up of system resources.

There are three directories underneath the directory where the Adapter was installed containing additional information: `doc`, `examples` and `config`. The `doc` directory contains Reuters README files describing various configuration options. The `examples` directory contains several example map files demonstrating many features. The `config` directory contains example RMDS configuration files. Minimally, the RMDS config file must be modified with your site's specific information. Typically the map file will also be modified to match the Aleri Streaming Platform application.

Required Arguments

`-f` *`mapFile`*	Specify the map file containing the metadata required to map the market data to/from RMDS.
`-p` *`[hostname:]port`*	Specify the *`port`*, and optionally the *`hostname`*, of the command and control interface within a running instance of the Aleri Streaming Platform. If you do not specify a hostname, the default is localhost. This instance of the Aleri Streaming Platform will be used to receive RMDS Marketfeed data.

Options

`-a` *`in`*`|`*`out`*	Specifies whether the RMDS Marketfeed Adapter instance is passing data in to the Aleri Streaming Platform or receiving data passed out from it. The possible values are **`in`** and **`out`**. Since the default value is **`in`**, this option is typically omitted when subscribing to Market Data. For backwards compatibility, "subscribe" (in) and "publish" (out) are still allowed, but deprecated.
`-c` *`user[:password]`*	If you are using an authentication method that requires credentials (i.e., Kerberos, PAM, or RSA), this option passes those authentication creden-

	tials to the Aleri Streaming Platform. If the Aleri Streaming Platform successfully authenticates with these credentials, the connection is maintained, otherwise the Aleri Streaming Platform will immediately close the connection.
`-d` *`debugLevel`*	Set the debug level. The valid range is 0-7, with 0 being minimal and 7 being verbose. By default it is set to 4.
`-e`	Negotiate encrypted OpenSSL sockets for all communication with the Aleri Streaming Platform, which must be started in encrypted mode when using this option.
`-F` *`configFile`*	Specify the RMDS Configuration file, overriding the configuration file specified in the map file.
`-g` *`gatewayHost`*	Specify the Streaming Processor gateway host explicitly.
`-G`	Use Kerberos authentication. This option is required when the Streaming Processor was started with the `-V gssapi` option.
`-h`	Print a short help message describing the syntax of this command.
`-H` *`[hostname:]port`*	Specify the *`port`*, and optionally the *`hostname`*, of the command and control interface for the hot spare instance of the Aleri Streaming Platform. If you do not specify a hostname, the default is localhost.
`-k` *`privateRSAKeyFile`*	Perform authentication using the RSA private key file mechanism instead of password authentication. The *`privateRSAKeyFile`* must specify the absolute path filename of the private RSA key file. With this option enabled, the user name must be specified with the `-c` option, but the password is not required. In addition, the Aleri Streaming Platform must be started with the `-k` option.
`-l` *`0|1|2|3`*	Specify where log messages get sent. Use **`0`** for no log messages, **`1`** to send to stderr only (the default), **`2`** to send to syslog only, and **`3`** to send to both stderr and syslog.
`-P` *`precision`*	Specify the number of displayed decimal places in output for values of type DOUBLE (the default is 3). So a variable of type DOUBLE, with an actual value of 27.123456, is displayed as 27.123 by default. If you set **`-P 2`**, it would be displayed as 27.12. If you set **`-P 6`**, it would be displayed as 27.123456.
`-r` *`subscribeRetryInterval`*	Specify how many seconds to wait (default is 300) between attempts to resubscribe to a RIC. (If a subscription to a RIC is marked CLOSED or CLOSEDRECOVER, you must resubscribe to that RIC for data to flow.) To disable resubscribe attempts, specify **`0`** as the value. Periodically resubscribing can compensate for a temporary condition where the source was not ready for subscribers. Each unsuccessful resubscribe attempt will generate a failure event which may result in a status update marking the item stale.
`-s` *`streamName`*	Specify the stream to be used when running in discovery mode. This option is used by the connector start mechanism and specifies the single stream whose mapped columns were discovered.
`-v`	Print the version of the RMDS Marketfeed Adapter and exit.
`-w` *`retrySeconds`*	Specify how many seconds to wait between retries when connecting to the Aleri Streaming Platform. The default is 5. Specifying **`0`** means do not retry; try only once.
`-x` *`optName`*	Specify various extra settings; use **`-x help`** to see a list of possible values.

`-z publishCount`	Terminate after passing *`publishCount`* values to the Aleri Streaming Platform. By default this is 0, which means never terminate.

Environment Variables

Several environment variables also influence the behavior of the RMDS Marketfeed Adapter.

`ALERI_REUTERS_HOME`	Specify the path to the Adapter's distribution directory.
`ALERI_RMDS_SUBSCRIBE_DEBUG_LEVEL`	Set to 7 to log the subscribed values when at log level 7 (-d 7). Other values are reserved for future use.
`ALERI_RMDS_SUBSCRIBE_DEBUG_SYMBOLS`	Specify which symbols (RICs) should be logged. Set this variable to one or more RICs, separated by spaces. This acts as a filter for ALERI_RMDS_SUBSCRIBE_DEBUG_LEVEL.
`ALERI_RMDS_PUBLISH_DEBUG_LEVEL`	Set to 7 to log the subscribed values when at log level 7 (-d 7). Other values are reserved for future use.
`ALERI_RMDS_PUBLISH_DEBUG_SYMBOLS`	Specify which symbols (RICs) should be logged. Set this variable to one or more RICs, separated by spaces. This acts as a filter for ALERI_RMDS_PUBLISH_DEBUG_LEVEL.

For a description of these and additional environment variables, enter one of the following commands:

```
rmds -a in -x help
rmds -a out -x help
```

Examples

A typical invocation to subscribe to RMDS and deliver Market Data to the Aleri Streaming Platform is:

```
rmds -c user:passwd -f myMap.xml -p host:port -d 7 >& rmds.in.log &
```

To send data from the Aleri Streaming Platform to RMDS, a typical invocation would be:

```
rmds -a out -c user:passwd -f myMap.xml -p host:spPort -d 7  >& rmds.out.log &
```

See Also

rmdsomm(1)

Bugs

See the *Aleri RMDS Marketfeed Adapter Release Notes* for known issues.

The `doc` directory contains Reuters `README` files describing various known issues with the Reuters API.

Copyright

Appendix B. Environment Variables

The Inbound and Outbound Adapters use the following environment variables.

Environment Variable	Used By	Description
ALERI_ACCUMULATOR_DELAY	Inbound	Expert: Delay connection to the Aleri Streaming Platform (seconds).
ALERI_DISABLE_REPORT_ENCODING_NULL	Outbound	Stop warning about blank to null conversions (bool) [false].
ALERI_FLUSH_INTERVAL	Inbound	Override the publication flushInterval (microseconds).
ALERI_INTRASUBSCRIBE_DELAY	Inbound	Override the Map attribute (milliseconds).
ALERI_LOG_CONFIG_EVENTS	Both	Set log level (1-7) for config event processing [-1]
ALERI_MARKETFEED_DUMP	Outbound	Set the log level (0-7) at which to dump raw Reuters messages to the log.
ALERI_MAX_RECORDS_PER_BLOCK	Inbound	Override the publication maxRecordsPerBlock (count).
ALERI_PENDING_LIMIT	Inbound	Override the publication pendingLimit.
ALERI_RETRY_INTERVAL	Both	Override the publication retryInterval.
ALERI_REUTERS_HOME	Both	Specify the installation directory.
ALERI_RMDS_DISPATCH	Both	Expert: Dispatch RFA every N milliseconds [10,000].
ALERI_RMDS_EVENT_TRACE	Both	Expert: Enables RFA event tracing every N event (int).
ALERI_RMDS_PUBLISH_BUFSIZE	Outbound	Override the buffer size.
ALERI_RMDS_PUBLISH_DEBUG_LEVEL	Outbound	Set to 7 to see values [not in -opt].
ALERI_RMDS_PUBLISH_DEBUG_SYMBOLS	Outbound	Contains a space-delimited list of symbols for use when default behavior of not showing symbols is overridden. If it's not set, then all symbols are used.
ALERI_RMDS_SUBSCRIBE_DEBUG_LEVEL	Inbound	Set to 7 to see values [not in -opt].
ALERI_RMDS_SUBSCRIBE_DEBUG_SYMBOLS	Inbound	Contains a space-delimited list of symbols for above. If it's not set, all symbols are used.
ALERI_RMDS_SUBSCRIBE_SYMBOL_FORMAT	Inbound	Specify symbol list format: 0 for multi-line or 1 for single-line .
ALERI_SEND_AS_TRANSACTIONS	Inbound	Override the Map attribute.
ALERI_SHOW_FIELD_INFO	Inbound	Show FID, column, spColumn and stream name [false].
ALERI_SHOW_SP_EVENT_DATA	Outbound	Set log level (1-7) for events from the Aleri Streaming Platform [-1].

Appendix C. Inbound Marketfeed Adapter Map File XML Syntax

C.1. Summary

The following listing shows the structure of an Inbound Adapter Map File. Each line of this summary lists one element of the map file structure. This appendix documents each of these elements (they are arranged in alphabetical order).

```
adapter                             (required, limit one)
  |----publication                  (required, limit one)
  |----streamMaps                   (required, limit one)
  |       '----streamMap            (required)
  |         |----itemName           (required, limit one)
  |         |----serviceName        (optional)
  |         |----sequenceNumber     (optional)
  |         |----itemStale          (optional)
  |         |----dataField          (required)
  |         |----updateNumber          (required)
  |         |----dateTimeField      (optional)
  |         |----FIDListField       (optional)
  |         '----nullField          (optional)
  |----recordTypeMap                (optional)
  |       '----recordType           (optional)
  |----rfa                          (required, limit one)
  '----itemLists                    (required, limit one)
          '----itemList             (required)
                 '----item          (optional)
```

C.2. `adapter`

```
adapter
```

Summary

This is the root element of the map file. All configuration sections should be nested between `<adapter>` and `</adapter>`.

Children

The following child elements are defined for `adapter`. All of these elements must be present, and in the order specified.

Name	Requirement
publication	Exactly one required
streamMaps	Exactly one required
recordTypeMap	Optional
rfa	Exactly one required
itemLists	Exactly one required

Attributes

Name	Description	Requirement
name	A string that uniquely identifies this Adapter (included in log entries)	optional

Notes

None

C.3. dataField

```
adapter
  '----streamMaps
         '----streamMap
                '----dataField
```

Summary

In the streamMap definition, maps one column from a Source Stream to a Reuters Field ID (FID).

Children

None

Attributes

Name	Description	Requirement
name	The Reuters FID that identifies the data item that will appear in this column of the Source Stream	required
key	true or false, depending on whether this column is part of the Source Stream's unique key.	See Note

Notes

- Each element in the streamMap section of the Inbound Adapter map file must represent a column in the Row Definition of the target Source Stream. (The order of the streamMap elements must mirror the order of the columns in the RowDef.) If the column in the RowDef is a data item (Bid, Ask, etc.), the corresponding streamMap entry should be a dataField element whose name attribute identifies a specific FID. Any time RMDS publishes an update tagged with that FID, the Adapter sends it to the Aleri Streaming Platform Source Stream as a value in the corresponding row.
- The key attribute must be used to set the value to true. If this column is not part of the stream's key, this attribute can be omitted.

Example

```
<streamMap name="stream1">
        <itemName key="true"/>
        <FIDListField />
        <!-- serviceName / -->
        <sequenceNumber />
        <itemStale/>
        <dataField name="BID"/>
        <dataField name="ASK"/>
        <dataField name="TRDPRC_1"/>
        <dataField name="ACVOL_1"/>
        <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
</streamMap>
```

The example shown above maps columns 5-8 of the Aleri Streaming Platform Source Stream stream1 to the Reuters FIDs BID, ASK, TRDPRC_1 and ACVOL_1.

C.4. dateTimeField

```
adapter
  '----streamMaps
         '----streamMap
                '----dateTimeField
```

Summary

In a streamMap, maps a Reuters date or time FID (or one of each) to a date and/or timestamp column in an Aleri Streaming Platform Source Stream.

Children

None

Attributes

Name	Value	Requirement
dateName	The FID of the date value provided by RMDS	see Note
timeName	The FID of the time value provided by RMDS	see Note

Notes

- The most commonly used datatype for date/time information in Aleri Streaming Platform data streams is dateTime, which combines both date and time. In most cases, however, the updates provided by RMDS and brought in to the Aleri Streaming Platform by the Reuters Adapter will have separate FIDs for date and time.

  The Map file provides the dateTimeField element, which provides separate attributes for date and time, to remedy this discrepancy. This allows you to map two FIDs (one for date, one for time) to the same column in the Source Stream definition.

- One of these three attributes must be used. If dateTime is used, it must be used alone. dateName and timeName can be used either separately or together.

- The value for each FID must match one listed in the FID list referenced in the Reuters-side configuration file (The FID list provided with the Adapter is named appendix_a). This file is referenced in the configuration file rfasub.cfg.

Example

```
<streamMap name="stream1">
        <itemName key="true"/>
        <FIDListField />
        <!-- serviceName / -->
        <sequenceNumber />
        <itemStale/>
        <dataField name="BID"/>
        <dataField name="ASK"/>
        <dataField name="TRDPRC_1"/>
        <dataField name="ACVOL_1"/>
        <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
</streamMap>
```

This example maps the TIMACT and ACTIV_DATE FIDs together to the ninth column of the Aleri Streaming Platform Source Stream stream1.

C.5. `FIDListField`

```
adapter
  '----streamMaps
         '----streamMap
                '----FIDListField
```

Summary

In a streamMap, creates a single string which maps all of the Reuters FIDs with their values for an event to the Aleri Streaming Platform Source Stream. This is useful when testing and debugging because it enables you to see all of the FIDs that an event returns.

Children

None

Attributes

Name	Value	Requirement
Name	a string that will appear in any Adapter-related log entries	optional

Example

```
<streamMap name="stream1">
        <itemName key="true"/>
        <FIDListField />
        <!-- serviceName / -->
        <sequenceNumber />
        <itemStale/>
        <dataField name="BID"/>
        <dataField name="ASK"/>
        <dataField name="TRDPRC_1"/>
        <dataField name="ACVOL_1"/>
        <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
</streamMap>
```

In this example, the second column of the Source Stream is identified as the one that will carry the FIDList string of any update from the Adapter.

C.6. item

```
adapter
  '----itemLists
         '----itemList
                '----item
```

Summary

This is the element used to identify a RIC to which the Reuters Adapter will subscribe.

Children

None

Attributes

Name	Value	Requirement
name	a RIC to which the Adapter will subscribe	required
rfaQueue	a name for the rfaQueue, which, if provided, will replace the default rfaQueue name and cause a separate thread to be used for this queue	optional
service	the name of a Reuters Service that will provide incoming data through RMDS	optional if already specified in the parent itemList or itemLists element, otherwise required
stream	the Source Stream on which updates for this RIC will be brought to the Aleri Streaming Platform	optional if already specified in the parent itemList or itemLists element, otherwise required

Notes

- The value for the `name` attribute must match one listed in the `appendix_a` file referenced in the Reuters-side configuration file (`rfasub.cfg` is the name of the file provided with the Adapter).
- If you specify a stream name here, updates for this RIC will be brought in to the Aleri Streaming Platform on that stream. If you do not specify a stream here, the stream specified at the `itemList` level will be used.
- The stream you specify must match one of the streamMaps defined elsewhere in the map file by the value of the `streamMap`'s `name` attribute.

Example

```
<itemLists service="SSL_PUB" stream="stream1">
        <itemList  service="IDN_RDF" >
                <item name="EUR=" />
                <item name="EURJPY=" stream="stream6" />
        </itemList>
</itemLists>
```

These two `item` elements subscribe the Adapter to the RICs `EUR` and `EURJPY`. The `EUR` updates will be sent to the stream `stream1` which was set in the itemLists element. The `EURJPY` updates will be sent to the stream `stream6`, since the `item` level `stream` attribute overrides the `itemLists` level attribute.

C.7. itemList

```
adapter
  '----itemLists
         '----itemList
```

Summary

Encloses a list of one or more instances of the item element.

Children

The following child element is defined for itemList.

Name	Requirement
item	0 or more required

Attributes

Name	Value	Requirement
rfaQueue	a name for the rfaQueue, if provided, it will replace the default rfaQueue name and cause a separate thread to be used for this queue	optional
service	the name of a Reuters Service that will provide incoming data through RMDS	optional if already specified in the parent itemLists element or in all child item elements, otherwise required
stream	the name of an Aleri Streaming Platform Source Stream that will receive updates on the RICs specified in this list of items.	optional if already specified in the parent itemLists element or in all child item elements, otherwise required

Notes

- You configure the Adapter to push updates for every item in this section to that stream (although this specification can be overridden at the item level) by specifying a stream name for this element.
- The Adapter supports more than one itemList element under itemLists; this allows you to configure one instance of the Adapter to direct updates from two or more groups of RICs to different Aleri Streaming Platform source streams.
- The stream you specify must match one of the streamMaps defined elsewhere in the map file (by the value of the streamMap's name attribute.
- Use of the rfaQueue attribute controls scalability (see Section 5.2, “Performance Tuning”).

Example

```
<itemLists service="SSL_PUB" stream="stream1">
        <itemList  service="IDN_RDF" >
                <item name="EUR=" />
                <item name="EURJPY=" stream="stream6" />
        </itemList>
</itemLists>
```

This itemList element sets the service attribute to IDN_RDF, overriding the SSL_PUB service attribute defined in the parent itemLists element.

C.8. itemLists

```
adapter
  '----itemLists
```

Summary

Encloses the Map file section for one or more instances of itemList.

Children

The following child elements are defined for itemLists.

Name	Requirement
itemList	One required, two or more supported

Notes

Each itemList instance in this section is a list of one or more of the RICs to which the Adapter will subscribe.

Attributes

Name	Value	Requirement
name	a string that will appear in any Adapter-related log entries	optional
rfaQueue	a name for the rfaQueue, if provided, it will replace the default rfaQueue name and cause a separate thread to be used for this queue	optional
service	the name of a Reuters Service that will provide incoming data through RMDS	optional if specified in the child itemLists and/or item elements so that all child item elements either specify or inherit it, otherwise required
stream	The name of an Aleri Streaming Platform Source Stream that will receive updates on the RICs specified in the item lists in this section. (A default that can be overridden at the item level.	optional if specified in the child itemLists and/or item elements so that all child item elements either specify or inherit it, otherwise required

Get the value for service from your Reuters administrator.

Example

```
<itemLists service="SSL_PUB" stream="stream1">
        <itemList  service="IDN_RDF" >
                <item name="EUR=" />
                <item name="EURJPY=" stream="stream6" />
        </itemList>
</itemLists>
```

This itemLists element sets the service attribute to SSL_PUB and the stream attribute to stream1. These attributes will either be inherited or overridden at the itemList and/or item level.

C.9. itemName

```
adapter
  '----streamMaps
         '----streamMap
                '----itemName
```

Summary

In the streamMap definition, itemName identifies the row in the Aleri Streaming Platform Source Stream that will carry the RIC from the RMDS update.

Parent

streamMap

Children

None

Attributes

Name	Value	Requirement
key	true or false, depending on whether or not this column is part of the Source Stream's unique key	See Note 3

Notes

1. The itemName element should be inserted in the streamMap to correspond with the column in the RowDef that carries the RIC or Symbol. If this column is part of the Source Stream's key, the key attribute should be set to true.
2. This element is one of the "pseudo-fields" that specify data items that are not part of the data feed coming directly from RMDS.
3. You do not need to use the key attribute. It is present for backwards compatibility.

Example

```
<streamMap name="stream1">
        <itemName key="true"/>
        <FIDListField />
        <!-- serviceName / -->
        <sequenceNumber />
        <itemStale/>
        <dataField name="BID"/>
        <dataField name="ASK"/>
        <dataField name="TRDPRC_1"/>
        <dataField name="ACVOL_1"/>
        <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
</streamMap>
```

In this example, the first column of the Source Stream is identified as the one that will carry the RIC value of any update from the Adapter. It is also identified as part of the stream's key.

C.10. `itemStale`

```
adapter
  '----streamMaps
         '----streamMap
                '----itemStale
```

Summary

In the `streamMap` definition, identifies a column in the Aleri Streaming Platform Source Stream that carries a flag indicating whether incoming RMDS data has gone stale.

Children

None

Attributes

Name	Value	Requirement
name	a string that will appear in any Adapter-related log entries	required

Notes

Use this element in the streamMap if one of the columns in the Source Stream is a "stale" flag.

RMDS itself does not supply a "stale" flag with regular Market Data, although it may pass along such a flag if it is provided by another service you are subscribing to via RMDS. If this element is used in the streamMap, the Adapter will send out an update value of "1" if it receives a "stale" flag from RMDS or stops receiving any data from RMDS.

Example

```
<streamMap name="stream1">
        <itemName key="true"/>
        <FIDListField />
        <!-- serviceName / -->
        <sequenceNumber />
        <itemStale/>
        <dataField name="BID"/>
        <dataField name="ASK"/>
        <dataField name="TRDPRC_1"/>
        <dataField name="ACVOL_1"/>
        <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
</streamMap>
```

In this example, the fourth column of the Source Stream is identified as the one that will be updated if the Adapter receives a "stale" notification or stops receiving data from RMDS.

C.11. `nullField`

```
adapter
  '----streamMaps
         '----streamMap
                '----nullField
```

Summary

In a streamMap, it acts as a placeholder that always delivers a NULL value to the Aleri Streaming Platform Source Stream. This lets you add extra fields to a Source Stream to get the configuration you want.

Children

None

Attributes

Name	Value	Requirement
name	a string that will appear in any Adapter-related log entries	optional

Notes

When experimenting with a model, you can use a `nullField` to temporarily stop feeding data into one column of the stream. In this case, you can simply keep the name of the `dataField` that you are temporarily replacing, as in the following example.

Example

```
<streamMap name="stream1">
        <itemName key="true"/>
        <FIDListField />
        <!-- serviceName / -->
        <sequenceNumber />
        <itemStale/>
        <dataField name="BID"/>
        <dataField name="ASK"/>
        <nullField name="TRDPRC_1"/>
        <dataField name="ACVOL_1"/>
        <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
</streamMap>
```

In this example, the seventh column of the Source Stream is identified as a placeholder that will receive a null value in each update from the Adapter. It includes the name of the `dataField` that it replaces for debugging purposes.

C.12. publication

```
adapter
  '----publication>
```

Summary

Specifies basic operating parameters for this instance of the Adapter.

Children

None

Attributes

Name	Value	Requirement
flushInterval	Specify the number of microseconds the Adapter will allow events to accumulate before sending them to the Aleri Streaming Platform. A non-zero flushInterval makes the accumulation of events time-based.	optional
intraSubscribeDelay	Specify the number of milliseconds the Adapter will pause between subscription requests.	optional
maxRecordsPerBlock	Specify the maximum number of accumulated events that the Adapter should send to the Aleri Streaming Platform at a time. This reduces the size of each transaction or envelope fragment when there is a large number of accumulated events. For example, if 140 events have accumulated and `maxRecordsPerBlock` is set to *`50`*, the Adapter will send the envelope or transaction as three fragments.	optional
name	Specify a string that will identify the Adapter instance in log file entries.	required
pendingLimit	Specify the number of events that may accumulate before the Adapter sends them in to the Aleri Streaming Platform. Using a pendingLimit makes the accumulation of events count-based.	optional
retryInterval	Specify the number of seconds the Adapter will try to connect to RMDS before shutting down.	required
sendAsTransactions	Set to *`true`* to treat a group of updates as a single transaction or *`false`* to treat them as separate transactions within an envelope.	optional

Notes

You can optimize the Adapter's performance using the `pendingLimit` and `flushInterval` attributes, along with the `maxRecordsPerBlock` and `sendAsTransaction` attributes from the Pub/Sub interface that the Adapter uses to communicate with the Aleri Streaming Platform. See Section 5.2, “Performance Tuning” for details.

Some venues send initial images as multi-part messages, which can produce large datasets. The `intraSubscribeDelay` attribute provides the ability to pace these subscriptions and prevent the Adapter from being overwhelmed by initial images. The default value is zero, which is suitable for short RIC lists. When it is set to a non-zero value, the Adapter will pause for that many milliseconds between subscription requests. The suggested value is ten (10).

Example

```
<publication name="RMDS Adapter - low latency" retryInterval="5"
        flushInterval="0" pendingLimit="0" sendAsTransactions="0" />
```

C.13. `recordType`

```
adapter
  '----recordTypeMap
         '----recordType
```

Summary

Maps an Aleri Streaming Platform Source Stream to pre-defined set of FIDs.

Children

None

Attributes

Name	Value	Requirement
number	The ID of a recordType defined in Reuters configuration	required
stream	The name of an Aleri Streaming Platform Source Stream to which this record will be mapped	required

Notes

- The pre-defined Record specified by `recordType`, must match all the columns in the stream's definition.
- Otherwise these columns must be explicitly mapped in a `streamMap` configuration.

Example

```
<recordTypeMap>
  <recordType number="123" stream="eqInput"/>
</recordTypeMap>
```

This example maps a set of FIDs pre-defined as record "123" to the Source Stream `eqInput`.

C.14. recordTypeMap

```
adapter
  '----recordTypeMap
```

Summary

Encloses the Map file section for one or more instances of recordType.

Children

The following child elements are defined for recordTypeMap.

Name	Requirement
recordType	Zero or more supported

Attributes

None

Notes

- A stream must have either a recordTypeMap or a streamMap; not both.
- The pre-defined Record must match all the columns in the stream's definition to use the implicit mapping provided by recordTypeMap.
- Otherwise these columns must be explicitly mapped in a streamMap configuration.

Example

```
<recordTypeMap>
  <recordType number="123" stream="eqInput"/>
</recordTypeMap>
```

This example maps a set of FIDs pre-defined as record "123" to the Source Stream eqInput.

C.15. rfa

```
adapter
  '----rfa
```

Summary

Links the subscriber map file to the Reuters-side configuration file.

Children

None

Attributes

Name	Value	Requirement
config	The absolute path and filename of the Reuters-side configuration file for subscription (the sample file supplied with the Adapter is at `$ALERI_REUTERS_HOME/config/rfasub.cfg`.	required
configDatabaseName	must be set to RFA	required
enumFile	The full pathname of the Reuters supplied file that lists each enumerated type along with the range of values it can take.	required
fidFile	The full pathname of the Reuters supplied file that lists all of the valid FIDs.	required
sessionName	A reference to a session name defined in the Reuters-side configuration file for subscription.	required
blank	specify a marker to use for blanks	optional
blankInt32	specify a marker to use for blank Int32 fields	optional
blankInt64	specify a marker to use for blank Int64 fields	optional
blankMoney	specify a marker to use for blank Money fields	optional
blankString	specify a marker to use for blank String fields	optional
blankDate	specify a marker to use for blank Date fields	optional
blankTimestamp	specify a marker to use for blank Timestamp fields	optional

Notes

None

Example

```
        <rfa config="$ALERI_REUTERS_HOME/config/rfasub.cfg"
                sessionName="Session1" />
```

This example points the Reuters Adapter to the Reuters-side configuration in the file `rfasub.cfg`. The list line in this configuration file is:

```
  \Sessions\Session1\connectionList                    = "Connection_SSLED"
```

This line defines a session name that is referenced by other lines in the configuration file. When the map file references a session name in the `sessionName` attribute, it links the Adapter to the Reuters-side configuration parameters identified by that name.

C.16. sequenceNumber

```
adapter
  '----streamMaps
         '----streamMap
                '----sequenceNumber
```

Summary

In the `streamMap` definition, maps a column in the Aleri Streaming Platform Source Stream that will be populated by a unique number generated by the Adapter, not provided as part of the data from RMDS.

Children

None

Attributes

Name	Value	Requirement
`key`	`true` or `false`, depending on whether this column is part of the Source Stream's unique key	See Note
name	a string that will appear in log entries	optional

Notes

- The Adapter maintains a separate counter for each RIC to which it is subscribed. Each time it receives an update for a RIC, it increments its counter for that RIC. This number is the one sent to the Source Stream column mapped by the `sequenceNumber` element.
- Many Source Stream definitions include a column specification like the following:

  ```
  <Column datatype="int32" name="Id"/>
  ```

  This line specifies a unique ID for the Source Stream. The `sequenceNumber` pseudo-field is a good match for this column in the Inbound Adapter map file.
- The `key` attribute must be used to set the value to `true`. If this column is not part of the stream's key, this attribute can be omitted.

Example

```
<streamMap name="stream1">
        <itemName key="true"/>
        <FIDListField />
        <!-- serviceName / -->
        <sequenceNumber />
        <itemStale/>
        <dataField name="BID"/>
        <dataField name="ASK"/>
        <dataField name="TRDPRC_1"/>
        <dataField name="ACVOL_1"/>
        <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
</streamMap>
```

In this example, the third column of the Source Stream is mapped to the sequence number provided by the Adapter. This column is also identified as part of the Source Stream's unique key.

C.17. serviceName

```
adapter
  '----streamMaps
         '----streamMap
                '----serviceName
```

Summary

In the streamMap definition, maps a column in the Aleri Streaming Platform Source Stream to the Service identifier that the Adapter provides as part of the "envelope" for each update. This string could, potentially, be used to provide namespace scope for a RIC that was provided by two different services to which you subscribed.

Parent

streamMap

Children

None

Attributes

Name	Value	Requirement
key	true or false, depending on whether this column is part of the Source Stream's unique key.	See Note

Note

The key attribute must be used to set the value to true. If this column is not part of the stream's key, this attribute can be omitted.

Example

```
<streamMap name="stream1">
        <itemName key="true"/>
        <FIDListField />
        <!-- serviceName / -->
        <sequenceNumber />
        <itemStale/>
        <dataField name="BID"/>
        <dataField name="ASK"/>
        <dataField name="TRDPRC_1"/>
        <dataField name="ACVOL_1"/>
        <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
</streamMap>
```

In this example, no column of the Source Stream is mapped to the service name provided by the Adapter because it is commented out.

C.18. streamMap

```
adapter
  '----streamMaps
         '----streamMap
```

Summary

The opening and closing `streamMap` tags enclose the section of the Subscriber Map file that defines the mappings between the columns of an Aleri Streaming Platform Source Stream and the RMDS FIDs being subscribed to by the Adapter.

Children

The following child elements are defined for `streamMap`. These child elements can occur in any order, but for a specific `streamMap`, the order of the child elements must mirror the order of the columns of the Source Stream (as defined in the data model). This is how the Adapter is configured to deliver RMDS updates to the appropriate rows in the Source Stream.

Name	Requirement
`dataField`	One required, two or more supported
`dateTimeField`	Zero or more supported
`itemName`	One required, two or more supported
`itemStale`	Zero or one supported
`sequenceNumber`	Zero or one supported
`serviceName`	Zero or one supported

Attributes

Name	Value	Requirement
`name`	References the Source Stream to which the RMDS updates will be mapped. Must match the name of a Source Stream defined in the Aleri Streaming Platform data model.	required
`opcode`	Defines the operation the Adapter will perform when sending updates to the Source Stream. Possible values are `insert` and `upsert`. The `insert` operation adds new updates to the end of the Source Stream. The `upsert` operation replaces an existing Source Stream entry if its key matches the entry's key; if not, the update is added.	optional (default value is `upsert`)

Example

```
<streamMaps>
        <streamMap name="stream1">
                <itemName key="true"/>
                <FIDListField />
                <!-- serviceName / -->
                <sequenceNumber />
                <itemStale/>
                <dataField name="BID"/>
                <dataField name="ASK"/>
                <dataField name="TRDPRC_1"/>
                <dataField name="ACVOL_1"/>
                <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
        </streamMap>
</streamMaps>
```

This example maps a set of the Adapter's updates to an Aleri Streaming Platform Source Stream named `stream1`. All updates going to this Source Stream will be added using the `upsert` mode.

The RICs whose updates will be sent to this Source Stream are specified in an `itemList` elsewhere in the map file that also references `stream1`.

C.19. streamMaps

```
adapter
  '----streamMaps
```

Summary

Encloses the Map file section for one or more instances of streamMap.

Parent

adapter

Children

The following child elements are defined for streamMaps.

Name	Requirement
streamMap	one required, two or more supported

Attributes

None

Notes

- Each streamMap instance in this section maps incoming FIDs from the Reuters adapter to columns in an Aleri Streaming Platform Source Stream.
- A stream must have a streamMap or a recordTypeMap; not both.

Example

```
<streamMaps>
        <streamMap name="stream1">
                <itemName key="true"/>
                <FIDListField />
                <!-- serviceName / -->
                <sequenceNumber />
                <itemStale/>
                <dataField name="BID"/>
                <dataField name="ASK"/>
                <dataField name="TRDPRC_1"/>
                <dataField name="ACVOL_1"/>
                <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
        </streamMap>
</streamMaps>
```

C.20. updateNumber

```
adapter
  '----streamMaps
         '----streamMap
                '----updateNumber
```

Summary

In the streamMap definition, maps a column in the Aleri Streaming Platform Source Stream that will be populated by a unique number generated by the Adapter, not provided as part of the data from RMDS.

Children

None

Attributes

Name	Value	Requirement
key	true or false, depending on whether this column is part of the Source Stream's unique key	See Note
name	a string that will appear in log entries	optional

Notes

- The Adapter maintains a separate counter for each RIC to which it is subscribed. Each time it receives an update for a RIC, it increments its counter for that RIC. This number is the one sent to the Source Stream column mapped by the updateNumber element.
- Many Source Stream definitions include a column specification like the following:

  ```
  <Column datatype="int32" name="Id"/>
  ```

  This line specifies a unique ID for the Source Stream. The updateNumber pseudo-field is a good match for this column in the Inbound Adapter map file.
- The key attribute must be used to set the value to true. If this column is not part of the stream's key, this attribute can be omitted.

Example

```
<streamMap name="stream1">
        <itemName key="true"/>
        <FIDListField />
        <!-- serviceName / -->
        <updateNumber />
        <itemStale/>
        <dataField name="BID"/>
        <dataField name="ASK"/>
        <dataField name="TRDPRC_1"/>
        <dataField name="ACVOL_1"/>
        <dateTimeField timeName="TIMACT" dateName="ACTIV_DATE"/>
</streamMap>
```

In this example, the third column of the Source Stream is mapped to the update number provided by the Adapter. This column is also identified as part of the Source Stream's unique key.

Appendix D. Outbound Marketfeed Adapter Map File XML Syntax

D.1. Summary

The following listing shows the structure of an Outbound Adapter Map File. Each line of this summary lists one element of the map file structure. This appendix documents each of these elements, in alphabetical order.

```
adapter                                     (required, limit one)
  |----rfa                                  (required, limit one)
  '----subscriptions                        (required, limit one)
         '----subscription                  (required)
                '----stream                 (required)
                       |----name            (required, limit one)
                       |----service         (optional)
                       |      '----enum     (required)
                       |----stale           (optional)
                       |----field           (required)
                       '----constant        (optional)
```

D.2. adapter

```
adapter
```

Summary

This is the root element of the map file. All configuration sections should be nested between <adapter> and </adapter>.

Children

The following child elements are defined for adapter. All of these elements must be present, and in the order specified.

Name	Requirement
rfa	Exactly one required
subscriptions	Exactly one required

Attributes

None

Notes

None

D.3. constant

```
adapter
  '----subscriptions
         '----subscription
                '----stream
                       '----constant
```

Summary

Defines a data item with a constant value that will be published to RMDS by the Adapter.

Children

None

Attributes

Name	Value	Requirement
name	The name associated with this data item in the data image published by the Adapter.	Required
value	The value of this constant (always the same whenever this data item is published to the Aleri Streaming Platform)	Required

Notes

On startup, the Adapter publishes a complete image to RMDS, containing all data items defined in the Map file. After that, the Adapter only publishes updated values for data items when they change, unless the Aleri Streaming Platform goes stale and then recovers. This means that the value for *constant* will only be published when a complete image is published.

Example

```
  <stream name="stream1" >
          <name column="0"/>
          <stale column="3" name="ACVOL_1"/>

          <field column="1" name="DSPLY_NAME" />
          <field column="4" name="BID" precision="47" />
          <field column="5" name="ASK" precision="0" />
          <field column="6" name="TRDPRC_1"/>
          <field column="7" name="ACVOL_1"/>
          <constant name="PROD_PERM" value="1"/>
  </stream>
```

This example defines a constant called *PROD_PERM*, with the constant value "1", to be published with data values from the stream *stream1* under the publication name *subscription1*.

D.4. enum

```
adapter
  '----subscriptions
         '----subscription
                '----stream
                       '----service
                              '----enum
```

Summary

Maps the value of the Aleri Streaming Platform stream's *service* column to a unique string that will be prepended to the *name* element of an update published to RMDS by the adapter. If the Aleri Streaming Platform stream from which you are publishing handles data items for the same symbol from different sources (the "Ask" price for IBM from NASDAQ and from S&P, for example), you can use the *service* and *enum* attributes in the Outbound Adapter Map file to configure the Adapter to differentiate between updates of the same value for the same symbol from different sources.

Parent

service

Children

None

Attributes

Name	Value	Requirement
value	A possible value for the data stream column specified by the *service* element.	Required
prefix	The string prepended to the value of the *name* element when it publishes updates received from the Aleri Streaming Platform with the *service* value that matches *prefix*.	Required

Notes

The *service* element in the Outbound Adapter Map File must contain one *enum* element for each possible value in the source column. See Section D.8, “service” for more details.

Example

```
<service column="2" delim="_">
     <enum value="RDF"  prefix="R"/>
     <enum value="ISFS" prefix="I"/>
 </service>
```

Within a service definition, each enum element specifies a particular service. Based on this value, the published RICs are renamed to indicate the provider of the data. Assuming RIC.X is the RIC found in the name column. If the value in column 2 is RDF, the RIC will become "R_RIC.X". If the value in column 2 is ISFS, the RIC will become "I_RIC.X". If neither is true, no value will be published.

D.5. `field`

```
adapter
  '----subscriptions
         '----subscription
                '----stream
                       '----field
```

Summary

In a `stream` definition in an Outbound Adapter Map File, it specifies a column from an Aleri Streaming Platform source stream or config stream to be published to RMDS.

Parent

`stream`

Children

None

Attributes

Name	Value	Requirement
`column`	A number that represents the position of the source column in the stream being published from (the first column in the stream has the number "0")	Required
`name`	The FID that identifies this data value when published to RMDS	Required
`precision`	An integer that specifies the total number of digits after the decimal point in the published value (e.g., 1.23 has a `precision` of *2*)	optional

Notes

- The value of the *`name`* attribute may be modified to indicate the source of the data item if you have defined the `service` and `enum` elements in this `stream` definition.
- The *`precision`* attribute should only be included for columns of datatype *`double`*.

Example

```
<stream name="stream1" >
        <name column="0"/>
        <stale column="3" name="ACVOL_1"/>

        <field column="1" name="DSPLY_NAME" />
        <field column="4" name="BID" precision="47" />
        <field column="5" name="ASK" precision="0" />
        <field column="6" name="TRDPRC_1"/>
        <field column="7" name="ACVOL_1"/>
        <constant name="PROD_PERM" value="1"/>
</stream>
```

In this example, the Adapter is configured to publish updates from the fourth, fifth, sixth and seventh columns of the Aleri Streaming Platform stream named *`stream1`* as data items named *`BID`*, *`ASK`*, *`TRDPRC_1`* and *`ACVOL_1`*, respectively.

D.6. name

```
adapter
   '----subscriptions
          '----subscription
                 '----stream
                        '----name
```

Summary

In a `stream` definition in an Outbound Adapter Map File, specifies the column from the source Aleri Streaming Platform stream that provides the value that the Adapter will use as the unique identifier of each update.

Parent

`stream`

Children

None

Attributes

Name	Value	Requirement
`column`	A number that represents the position of the column in the Aleri Streaming Platform stream that carries the stream's unique identifier (the first column in the stream has the number "0")	either Column or Column Name
`column name`	The name of the column in the Aleri Streaming Platform stream that carries the stream's unique identifier	either Column or Column Name

Notes

- The publishing Adapter uses RMDS as a simple message bus; the published updates do not have to conform to Reuters protocols. This means that the column specified by this element does not have to be a Reuters RIC, but it must follow Reuters RIC syntax.
- If the source stream's unique key is a composition of two or more columns, you can use the *`name`* element in combination with one or more instances of the *`service`* element to configure the Adapter to publish updates with completely unique names.

Example

```
<stream name="stream1" >
        <name column="0"/>
        <stale column="3" name="ACVOL_1"/>

        <field column="1" name="DSPLY_NAME" key="true"/>
        <field column="4" name="BID" precision="47" />
        <field column="5" name="ASK" precision="0" />
        <field column="6" name="TRDPRC_1"/>
        <field column="7" name="ACVOL_1"/>
        <constant name="PROD_PERM" value="1"/>
</stream>
```

This example identifies the first column of *`stream1`* as its unique identifier or "key" column.

D.7. `rfa`

```
adapter
  '----rfa
```

Summary

Provides information for configuring the Reuters side of the Adapter, including an explicit reference to the Reuters-side configuration file.

Parent

`adapter`

Children

None

Attributes

Name	Value	Requirement
serviceName	Defines a Service Name that will be included in the header of every update sent out by the Reuters Adapter	optional
config	The absolute path and filename of the Reuters-side configuration file for publication (the sample file supplied with the Adapter is at `$ALERI_REUTERS_HOME/config/rfapub.cfg`).	required
sessionName	A reference to a session named defined in the Reuters-side configuration file for publication.	required
configDatabaseName	A reference to the Reuters database name	optional

Notes

None

Example

```
<rfa serviceName="IDN_RDF"
              config="$ALERI_REUTERS_HOME/config/rfapub.cfg"
              sessionName="Session1" configDatabaseName="RFA" />
```

This example points the Reuters Adapter to the Reuters-side configuration in the file `rfapub.cfg`. The first four uncommented lines in this configuration file are:

```
\Connections\Connection_SSLED_MP\ipcServerName      = "8105"
\Connections\Connection_SSLED_MP\connectionType     = "SSLED_MP"
\Connections\Connection_SSLED_MP\entitlementData    = false
\Sessions\Session1\connectionList                   = "Connection_SSLED_MP"
```

The last of these lines implicitly defines a session name that is defined as the *`sessionName`* in the Map file. The other three lines from `rfapub.cfg` key on this session name. This is how the value for *`sessionName`* ties this `publication` section of the Map file to a configuration set in the `.cfg` file.

When the Adapter publishes using this configuration, each update is identified with the serviceName "ALERI".

D.8. service

```
adapter
   '----subscriptions
          '----subscription
                 '----stream
                        '----service
                               '----enum
```

Summary

In a stream definition in an Outbound Adapter Map File, identifies a column from the source stream that is another component of the stream's unique key.

Parent

stream

Children

None

Attributes

Name	Value	Requirement
column	A number that represents the position of the column with the secondary key value (the first column in the stream has the number "0")	Required
delim	specifies a character to use as the separator between a name and a prefix	optional

Notes

The *service* element in the Outbound Adapter Map File must contain one *enum* element for each possible value in the source column. See Section D.8, "service" for more details.

Example

```
<service column="2" delim="_">
     <enum value="RDF"  prefix="R"/>
     <enum value="ISFS" prefix="I"/>
 </service>
```

This section configures the Adapter to test the value of the second column of every update from the Aleri Streaming Platform stream (the value of the *name* attribute of the stream element).

If the value is *"RDF"*, the Adapter adds the prefix *"R"* followed by the specified delim value to the name of the published update (the value of the *name* attribute of the publication element).

If the value is *"ISFS"*, the Adapter adds the prefix *"I"* to the name of the published update.

D.9. stale

```
adapter
  '----subscriptions
         '----subscription
                '----stream
                       '----stale
```

Summary

In a stream definition in an Outbound Adapter Map File, identifies a column from the source Aleri Streaming Platform stream whose value will change from "0" to "1" if the stream goes stale (if, for example, one of the stream's data sources is no longer being updated).

Parent

stream

Children

None

Attributes

Name	Value	Requirement
column	a number that represents the position of the column in the Aleri Streaming Platform stream's "stale" column (the first column in the stream has the number "0")	required
name	a string that identifies the stale column so that it may be mapped to a FID (published)	optional

Notes

None

Example

```
    <stream name="stream1" >
            <name column="0"/>
            <stale column="3" name="ACVOL_1"/>

            <field column="1" name="DSPLY_NAME" />
            <field column="4" name="BID" precision="47" />
            <field column="5" name="ASK" precision="0" />
            <field column="6" name="TRDPRC_1"/>
            <field column="7" name="ACVOL_1"/>
            <constant name="PROD_PERM" value="1"/>
    </stream>
```

This example identifies the third column of *stream1* as its "stale" column. If the "stale" column is specified, the column value will be published and the RIC will be marked stale.

D.10. `stream`

```
adapter
  '----subscriptions
         '----subscription
                '----stream
```

Summary

In a `publication` section in an Outbound Adapter Map File, identifies the Aleri Streaming Platform stream from which the Adapter will get the data it publishes to RMDS. The sub-elements in this section identify the columns in that stream that will be published.

Parent

`subscription`

Children

Name	Requirement
`name`	One
`service`	Optional
`stale`	Optional
`field`	One or more
`constant`	Optional

Attributes

Name	Value	Requirement
`exitOn-StreamExit`	This is a boolean attribute. When true, **rmds** terminates if the stream exits, the Aleri Streaming Platform exits, or the connection is lost.	Optional
`finalizer`	This string specifies an action to take if `heartbeat` milliseconds go by without an event being published to the Aleri Streaming Platform.	optional
`heartbeat`	This integer specifies how many milliseconds to wait without an event being published to the Aleri Streaming Platform before executing the `finalizer` action .	optional
`name`	The name of the stream from which the Adapter receives the data it publishes on RMDS	Required
`templateNumber`	A Reuters template set up in the RMDS configuration	Optional

Notes

- The value of the *`name`* attribute must be defined in the Aleri Streaming Platform data model.
- Any stream in the Aleri Streaming Platform data model can map to only one `stream` section in the map file.
- The `templateNumber` must be a unique identifier of the `stream` for which it is defined.

Example

```
<stream name="stream1">
  <name column="0"/>
  <field column="4" name="TRDPRC_1"/>
  <field column="9" name="BID" precision="5"/>
</stream>
```

This example configures the Aleri Streaming Platform to publish data from a stream named `stream1`.

D.11. subscription

```
adapter
  '----subscriptions
         '----subscription
```

Summary

The `subscription` section contains one or more instances of the `stream` element; enabling you to configure the Adapter to receive data from one or more streams.

The Outbound Adapter Map file can contain two or more `subscription` sections. At runtime, the publishing mechanism for each `subscription` section is instantiated on a separate thread, which provides scalability.

Parent

`subscriptions`

Children

The following child elements are defined for `subscription`.

Name	Requirement
`stream`	One or more

Attributes

Name	Value	Requirement
`name`	A name for this subscription. This name appears in the data updates published on RMDS by the Adapter and in log file entries.	Required

Example

```
<subscriptions>
  <subscription name="subscription1" >
    <stream name="stream1" >
      <name column="0"/>
      <field column="4" name="BID"/>
      <field column="5" name="ASK"/>
      <field column="6" name="TRDPRC_1"/>
      <field column="7" name="ACVOL_1"/>
      <constant name="PROD_PERM" value="1"/>
    </stream>
  </subscription>
</subscriptions>
```

This example configures the Adapter to publish some columns from the Aleri Streaming Platform stream *`stream1`* using the name *`subscription1`*.

D.12. subscriptions

```
adapter
  '----subscriptions
```

Summary

Encloses the Map file section for one or more instances of subscription.

Parent

adapter

Children

The following child elements are defined for subscriptions.

Name	Requirement
subscription	One or more

Attributes

None

Notes

Each subscription instance in this section defines one set of data that the Adapter will publish to RMDS.

Example

See Section D.11, "subscription" for an example.

Appendix E. Logging Facilities

A complete implementation of the Aleri Reuters Adapter supports two different logging mechanisms that can be used to check the Adapter's performance and to diagnose problems:

- The Adapter's own logging mechanism
- Reuters-side logging facilities

This appendix describes how to work with each of these logging resources and provides some sample error messages and possible causes.

You can configure the Adapter to write log messages to stderr, syslog, or both.

E.1. Adapter Logging

The Reuters Adapter supports the same flags for logging as the Aleri Streaming Platform executables, specifically:

`-d` sets the debug level (0=emergency messages only, 7=all messages)

`-l` tells the adapter to write log messages to stderr, syslog, both, or neither

See Appendix A, *Command Usage* for specific values for these flags.

Note:

The Adapter puts out "Emergency" level messages only when it is shutting down.

If you use the `-l` flag to direct Adapter log messages to stderr, you may also want to redirect `stderr` to a file.

Section E.3, “Log messages and meanings” lists some sample Adapter log messages along with a possible explanation for each one.

The *`name`* attribute of the *`publication`* element in the Inbound Adapter map file specifies a descriptive text string that will be logged to help identify how the Adapter was configured.

For example, lines 3-6 of `subexample.xml` specify the "publication" element for a subscribing instance of the Reuters Adapter, as follows:

```
  <publication
    name="RMDS Adapter exp"
    retryInterval="5"
  />
```

As the Adapter connects with and interacts with the Aleri Streaming Platform, this configuration causes the Adapter to write log messages such as the following:

```
 (0.123) @1 INFO: Configuring publication with name RMDS Adapter exp
```

The first two fields are the timestamp (in seconds since startup) and the thread number, respectively. These fields are not included in the sample log messages later in this appendix.

The base time for the timestamp, along with other information, is written to the log file on startup as shown in the following example. To convert the timestamp to a date and time, simply add the number of seconds to the base time.

```
(63359098041.768) @1 NOTICE:Base time is 10/08/08-17:27:21

(0.001) @1 NOTICE:install/i86pc_64_spro/bin/rmdsomm version:
1.0.3a-alpha_r18674M
 -a sub -c cimtest:-- -d 7
```

```
-f /home/aleri/support/1.0.3/ReutersOMMAdapter/marketprice.map.xml
 -l 1 -p tigris:12192 -P 1
(0.001) @1 NOTICE:pid=28649
(0.001) @1 DEBUG:Using ALERI_RMDSOMM_SUBSCRIBE_DEBUG_LEVEL=7
```

E.1.1. Page Data and Partial Page Updates

Some Reuters data comes as pages which use Marketfeed Partial format. Each page consists of multiple lines; initially sent as a snapshot. Page data is supported without any special configuration. The following extract from an Adapter log file shows the delivery of the initial page image (which is highlighted).

```
(27.729) @6 INFO:Publishing VOD.mGBPd 21 of 21 on stream1 as UPSERT
        _ITEM_NAME_ STRING: VOD.mGBPd
     _SERVICE_NAME_ STRING: IDN_RDF
   _SEQUENCE_NUMBER_ INT32: 1
        _ITEM_STALE_ INT32: 0
             ROW80_1 STRING: VOD.mGBPd               SI Quote Publication
             ROW80_2 STRING:
             ROW80_3 STRING: DATE:03/07/2008                                                Time:11:09
             ROW80_4 STRING:
             ROW80_5 STRING: Time   Venue        SI  Bid Size Bid Price Ask Price Ask Size Status
             ROW80_6 STRING: ====   =====        ==  ======== ========= ========= ======== ======
             ROW80_7 STRING: 110937 GSILGB2XXXX GSIL         1  150.9000  150.9500        1 OPEN
             ROW80_8 STRING: 070021 SBILGB2LXXX CITI                                      OPEN
             ROW80_9 STRING: 110909 CSFBGB2LXXX CSFB       329  150.7000  151.1500      329 OPEN
            ROW80_10 STRING: 110942 DEUTGB22ZEQ DBBL       528  150.6500  151.2000      527 OPEN
            ROW80_11 STRING: 110946 ABNAGB22XXX ABNV    483306  150.9000  150.9500   483306 OPEN
            ROW80_12 STRING: 110936 UBSWGB2LEQU UBSI         1  149.7682  152.1325        1 OPEN
            ROW80_13 STRING: 110828 SBUKGB21XXX CITI     20600  150.9000  151.0000    20600 OPEN
            ROW80_14 STRING: 110937 SLIIGB2LXXX LEHM      3750  150.9000  150.9500       15 OPEN
            ROW80_15 STRING:
            ROW80_16 STRING:
            ROW80_17 STRING:
(27.730) @6 DEBUG:Immediate flush for low latency; opcode=p
```

Each line of the page has its own FID to facilitate line-oriented deltas to the page. The Adapter parses the partial page updates from Reuters and produces strings like the ones shown highlighted in the following extract from an Adapter log file.

```
(49.934) @6 DEBUG:Processing update for VOD.mGBPd from service IDN_RDF
(49.934) @6 INFO:Publishing VOD.mGBPd 4 of 21 on stream1 as UPSERT
             _ITEM_NAME_ STRING: VOD.mGBPd
       _SEQUENCE_NUMBER_ INT32: 2
                ROW80_3 STRING: off:78 size:2 value:10
               ROW80_11 STRING: off:2 size:3 value:101
(49.934) @6 DEBUG:Immediate flush for low latency; opcode=p
(50.315) @6 DEBUG:Processing update for VOD.mGBPd from service IDN_RDF
(50.315) @6 INFO:Publishing VOD.mGBPd 3 of 21 on stream1 as UPSERT
             _ITEM_NAME_ STRING: VOD.mGBPd
       _SEQUENCE_NUMBER_ INT32: 3
               ROW80_11 STRING: off:5 size:1 value:7
(50.315) @6 DEBUG:Immediate flush for low latency; opcode=p
```

The first update in the example is to write the 2-character string "10" at an offset of 78 characters in the line of the page which contains the data from the ROW80_3 FID. The second update in the example is to write the 3-character string "101" at an offset of 2 characters in the line of the page which contains the data from the ROW80_11 FID. The third update in the example is to write the 1-character string "7" at an offset of 5 characters in the line of the page which contains the data from the ROW80_11 FID. Thus, updates for page data are very concise.

E.1.2. Modifying Log Entry Format

You can modify the default format of log entries in two ways.

- You can configure your system to log messages that show what values flow to the Aleri Streaming Platform on a single line rather than the default multi-line format by setting the environment variable `ALERI_RMDS_SUBSCRIBE_SYMBOL_FORMAT` to `1`. When messages are written to a log file, this can make it easier to scan for specific items.
- You can specify the number of decimal places displayed on output for double type variables using the *`-P`* option to **rmds**.

By default, log messages that show what values flow to the Aleri Streaming Platform are written in multi-line format as shown in the following example.

```
(38079.526) @2 INFO:Publishing VOD.mGBPd 3 of 9 on stream1 as UPSERT
      _ITEM_NAME_ STRING: VOD.mGBPd
_SEQUENCE_NUMBER_ INT32: 953
          ROW80_7 STRING: off:53 size:2 value:45
```

If you set the environment variable `ALERI_RMDS_SUBSCRIBE_SYMBOL_FORMAT` to `1` these messages are written are written in single line format as shown in the following example.

```
(17.794) @5 DEBUG:stream1 p values: _ITEM_NAME_=VOD.mGBPd _SEQUENCE_NUMBER_=2
ROW 80_3=off:78 size:2 value:20
```

In the preceding examples, the single line message contains a `p` rather than the opcode `UPSERT` found in the multi-line message. Single line messages use the following abbreviations for the opcodes:

i	insert
d	delete
u	update
p	upsert
D	safedelete

The *-P* option can alter the display of variables that are declared to be of the double datatype, as `ask` and `last` are in the following example. This only affects the display of the variables; not the contents.

```
<RowDefinition id="marketfeed_RowDef">
   <Column name="symbol"  datatype="string" />
   <Column name="service" datatype="string" />
   <Column name="seq"     datatype="int32"  />
   <Column name="stale"   datatype="int32"  />
   <Column name="bid"     datatype="money" />
   <Column name="ask"     datatype="double" />
   <Column name="last"    datatype="double" />
   <Column name="volume"  datatype="int32"  />
   <Column name="when"    datatype="timestamp" />
</RowDefinition>
```

If you accept the default precision, variables of type double (e.g., ASK in the following example) are written with three digits to the right of the decimal.

```
(5.089) @5 INFO:Publishing EURJPY= 7 of 9 on stream1 as UPSERT
(5.090) @5 DEBUG:stream1 p values: _ITEM_NAME_=EURJPY=
_SEQUENCE_NUMBER_=1 _ITEM_STALE_=0 BID=137.4800 ASK=137.530 ACVOL_1=0
ACTIV_DATE+TIMACT=2008-10-06T21:07:00.000 (1223327220000)
```

If you specify the option *-P 7* when enter the **rmds** command, variables of type double (e.g., ASK in the following example) are written with seven digits to the right of the decimal. Variables of other types are not affected.

```
(4.913) @5 INFO:Publishing EURJPY= 7 of 9 on stream1 as UPSERT
(4.913) @5 DEBUG:stream1 p values: _ITEM_NAME_=EURJPY=
_SEQUENCE_NUMBER_=1 _ITEM_STALE_=0 BID=137.5200 ASK=137.5700000
ACVOL_1=0 ACTIV_DATE+TIMACT=2008-10-06T20:55:00.000 (1223326500000)
```

E.2. Reuters Logging

The Adapter's interface to RMDS can also be configured to write to a logging facility. In the Reuters-side configuration file (`rfasub.cfg` and `rfapub.cfg` are the ones provided with the Adapter), you can turn logging on or off and specify a path and filename of the log file. The Reuters interface also supports a set of "message files".

The Reuters-side configuration file contains a set of configuration entries for the Reuters "Logger" facility.

```
\Logger\AppLogger\fileLoggerEnabled                = true
\Logger\AppLogger\fileLoggerFilename               = "rfasub.{p}.log"
```

This configuration set turns on Reuters logging for the Reuters Adapter. The log messages will be written to the `rfasub.`*`PID`*`.log` file, where *`PID`* is the Adapter's process ID.

- The first line in this set, `\Logger\AppLogger\windowsLoggerEnabled = false`, pertains to a Windows® logging facility that is not supported for the Reuters Adapter.
- These example lines are from `rfasub.cfg`, the file that configures an Adapter that subscribes to RMDS. The configuration file for publication, `rfapub.cfg`, contains the same configuration lines (except that the value for *`fileLoggerFilename`* is *`rfapub.{p}.log`*.

The same file contains configuration entries for Component Loggers, as follows:

```
\Logger\ComponentLoggers\Connections\messageFile   =\
   "config/messages/RFA6_Connections.mc"
\Logger\ComponentLoggers\Adapter\messageFile       =\
   "config/messages/RFA6_Adapter.mc"
\Logger\ComponentLoggers\SessionCore\messageFile   =\
   "config/messages/RFA6_SessionLayer.mc"
\Logger\ComponentLoggers\SSLED_Adapter\messageFile =\
   "config/messages/RFA6_SSLED_Adapter.mc"
```

E.3. Log messages and meanings

The following entries are examples of messages you might see in the Adapter log file. The actual format and working of the log messages, as well as the nature of the events logged and the log levels associated with these events, may change in subsequent releases of the Aleri Reuters Marketfeed Adapter.

Message:	`NOTICE:Item BARC.VX is closed: No Quality of Service is available to process subscription, timeout expired`
Cause:	The value for the Reuters username in the Reuters config file is incorrect (case matters) or the Reuters Service name in the Map file is incorrect.
Message:	`DEBUG: Immediate flush for low latency`
Cause:	Data received from RMDS being sent to the Aleri Streaming Platform immediately.
Message:	`NOTICE:XMLRPC ERROR-116: The connection to the server could not be established. Please make sure the server is up, and check the specified host name/port, user name/password, and encryption settings. If a host name is specified, make sure that it can be resolved through a DNS lookup. (5.092) @1 INFO:Could not connect to SP; (tigris:12190 cimtest) will retry in 5 seconds.`
Cause:	Cannot connect to the server running the Aleri Streaming Platform.
Message:	`Ignoring market data event because no significant fields updated`
Cause:	The Adapter received data from Reuters, but none of the fields were of interest to the Aleri Streaming Platform stream, so no data was sent.

Message: `ERROR: Error publishing: PUBLICATION ERROR-442: The send method of this publication object failed.`

Cause: Connection to the Aleri Streaming Platform unsuccessful during a message transmission.

Message: `ERROR:Mismatch between platform stream (9 columns) and adapter (31 columns for stream: stream1)`

Cause: The number of columns defined in the Adapter did not match the number of columns in the stream.

Message: `WARNING: Aleri Streaming Platform down, dropping all subscriptions`

followed by multiple iterations of a message like the following:
`DEBUG: Unsubscribing item: EUR= service: IDN_RDF`

Cause: Lost connection to the Aleri Streaming Platform. Stopping subscriptions to RMDS data since the Adapter has nowhere to put it.

Message: `WARNING: Discarding data rec'd after unsubscribe`

Cause: Before the Adapter shut off the subscription, more data arrived. This data is discarded because there is no connection to the Aleri Streaming Platform.

Message: `DEBUG: Processing update for EUR= from service IDN_RDF`

Cause: An update for RIC "EUR=" on service named "IDN_RDF" has arrived.

Message: `WARNING: Aleri Streaming Platform down, dropping all subscriptions`

followed by numerous repetitions of the following:
`DEBUG: Unsubscribing item: EUR= service: IDN_RDF`

Cause: Lost connection to the Aleri Streaming Platform. Stopping subscriptions to RMDS data since the Adapter has nowhere to put it.

Message: `WARNING: Discarding data rec'd after unsubscribe`

Cause: Before the Adapter shut off the subscription, more data arrived. This data is discarded because there is no connection to the Aleri Streaming Platform.

Message: `EMERGENCY: Fatal Error at line 0, column 0 of config file: An exception occurred! Type:RuntimeException, Message:The primary document entity could not be opened. Id=/home/aleri/adapter/trunk/src/ReutersAdapter/xxsubexample.xml`

Cause: Specified config file is unavailable

Message: `EMERGENCY: Fatal Error at line 0, column 0 of config file: An exception occurred! Type:RuntimeException, Message:The primary document entity could not be opened. Id=/home/aleri/adapter/trunk/src/ReutersAdapter/xxsubexample.xml`

Cause: The specified config file is unavailable.