

# Reuters Bridge Specification

VERSION 1.0

- [Overview](#)
- [Package structure description](#)
  - [Package structure](#)
  - [License information](#)
- [Installation procedure](#)
- [Functionality description](#)
  - [Supported message types](#)
  - [Supported deal types](#)
  - [Main Use Case scenario](#)
  - [Mapping rules](#)
- [Configuration](#)
- [Reuters Bridge as a Windows service](#)
- [Logging](#)
- [Error diagnosis by means of command line](#)

## Overview

Reuters Bridge is an adaptor which is used for communication between financial information exchange protocols - Thomson Reuters Ticket Output Feed (TOF) Financial Information eXchange (FIX).

**Thomson Reuters Ticket Output Feed (hereinafter TOF)** - financial information exchange protocol which is distributed by [Thomson Reuters](#). TOF provides possibility for subscription and receiving data related to deals.

**Financial Information eXchange (hereinafter FIX)** - an open protocol which is a standard for exchanging financial data and making transactions in real time.

Reuters Bridge works as a service for receiving data *by* systems which support FIX protocol *from* systems which support TOF protocol, i.e. data is transmitted only in the direction from TOF server to FIX server. While working, Reuters Bridge acts as a client for both FIX server and TOF server.

## Package structure description

The Reuters Bridge program module is distributed as an archive. Reuters Bridge can work as a *console application* (ReutersBridge.exe --run console) or as a *Windows service*.

## Package structure

Name of the file/directory	Description
ReutersBridge.exe	Executable program file
log4cxx.dll	Logging library
log4cxx.properties	Logging configuration
config.cfg	Basic Reuters Bridge configuration file
fix.engine.properties	Basic FIX library configuration file
logs	Directory for FIX Antenna logs
backup	
doc	Supporting documentation
Reuters Bridge Specification.pdf	Reuters Bridge specification in English
Reuters Bridge Specification.pdf (in Russian)	Reuters Bridge specification in Russian
Reuters Bridge Mapping Table.pdf	Mapping table for converting fields from TOF format to FIX format
fixdic44.xml	FIX4.4 dictionary
rb-additional.xml	Extension of FIX4.4 dictionary with a set of custom tags

## License information

License information is kept in the 'engine.license' file which is provided separately from the installation package. Need to put the license file to the root Reuters Bridge directory ('./ReutersBridge/'). In order to update license, it is enough to replace the existing license file with new one.

**Note:** Reuters Bridge won't work without license or with an expired license. If your license has expired or was lost, contact [sales@btobits.com](mailto:sales@btobits.com) to obtain the new one.

## Installation procedure

Installation of the Reuters Bridge can be performed by following step-by-step instruction described below:

1. Create a directory on the local disk (e.g **c:\b2bits**)
2. Unpack the downloaded archive with Reuters Bridge to the created directory
3. Adjust configuration files for your current business needs
4. Open the command line and go to the application directory (e.g., **cd c:\b2bits\ReutersBridge**)
5. Use the following command to initiate application installation: **ReutersBridge.exe --run install**

## Functionality description

The Reuters Bridge program module performs the following main functions:

- supports FIX 4.4 protocol and TOF-protocol described in Reuters Bridge specification «Thomson Reuters Dealing - Datafeed TOF/IP & TOF /Serial User Guide» in version 4.7.10
- allows setting up and keeping a connection between TOF-protocol data source and FIX-protocol data source
- converts received TOF messages to FIX format
- sends converted messages from TOF server to FIX server

## Supported message types

Reuters Bridge supports the following message types:

### FIX -> Reuters Bridge:

- *Session Level Messages (Heartbeat (0), Logon (A), Logout (5), Reject (3), Resend Request (2), Sequence Reset (4), Test Request (1))*

### Reuters Bridge -> Reuters:

- *Data and Updates Request (332)*
- *Snapshot Request (333)*

### Reuters -> Reuters Bridge:

- *Record Response (340)*
- *Updates (316)*
- *Status Response (407)*

### Reuters Bridge -> FIX:

- *Session Level Messages (Heartbeat (0), Logon (A), Logout (5), Reject (3), Resend Request (2), Sequence Reset (4), Test Request (1))*
- *Trade Capture Report (MsgType = 'AE')*

## Supported deal types

Below is the list of deal types, for which Reuters Bridge supports their mapping to FIX format and transmitting to FIX server:

1. FX Swap
2. FX Spot
3. FX Outright
4. FX FRA
5. FX Deposit
6. NDF FX Swap
7. NDF Outright

## Main Use Case scenario

Below is the **main Use Case scenario** of the Reuters Bridge program module:

1. Reuters Bridge establishes a session with FIX server
2. Reuters Bridge establishes a connection with TOF server
3. For each Dealing Server identifier specified in the configuration file Reuters Bridge sends a request to TOF server to receive deals
4. In case of successful processing, TOF server sends range of deals related to the requested Dealing Server identifier
5. Reuters Bridge constructs requests for deal data and related conversation for each deal from the range and sends them to TOF server

6. In case of successful requests processing, TOF server sends couple of messages which contain data about deal and related conversation
7. For each found "deal/conversation" pair, Reuters Bridge checks the deal type
8. If the deal type is supported by Reuters Bridge, it performs conversion of linked TOF messages which contain "deal/conversation" data to FIX message *Trade Capture Report (MsgType = 'AE')* according to the mapping rules (see the [Mapping rules](#) section for details)
9. If the conversion is successful, Reuters Bridge sends FIX message *Trade Capture Report (MsgType = 'AE')* to FIX server

#### Alternative scenarios:

##### Step 4, 6:

In case a request processing failed, TOF server sends *Status Response (407)* message which is processed by Reuters Bridge in the following way:

- If Status = 96 (Connection Restored), Reuters Bridge writes corresponding INFO record to the log
- If Status = 26 (Rejected, Try Again), Reuters Bridge writes corresponding WARNING record to the log and sends repeated request to TOF server
- else Reuters Bridge writes corresponding ERROR record to the log

##### Step 8:

In case the defined deal type is not supported by Reuters Bridge, it writes the corresponding record to the log, "deal/conversation" pair conversion is considered as completed.

##### Step 9:

In case conversion failed, Reuters Bridge writes the corresponding record to the log, "deal/conversation" pair conversion is considered as completed.

## Mapping rules

Detailed mapping rules are described in the [Reuters Bridge Mapping Table](#) article.

## Configuration

Common settings of the Reuters Bridge can be configured by means of the *config.cfg* configuration file. It allows you to configure common properties of the application, TOF server's parameters, FIX server's parameters and logging parameters.

The table below represents all available configurable properties:

Name	Type	Description
<i>Application section</i>		
App.Storage	Path	Path and name of the file where Reuters Bridge keeps the current state for the recovery procedure.
App.DayStartTime	HH:mm	Start time of the day. Defines the time of session establishment and start of data transferring.
App.DayEndTime	HH:mm	End time of the day. Defines the time of completion of data transferring and session closure until the next day.
<i>TOF protocol section</i>		
TOF.SID	String	Identifier of the TOF-server's client.
TOF.Server.Host	Host	IP-address or DNS-name of the TOF-server's host.
TOF.Server.Port	[1..65535]	TOF-server's port. <b>5003</b> – value by default.
TOF.Client.Port	[0..65535]	TOF-client's port. <b>0</b> – any free port; <b>5004</b> – value by default, for servers with a high level of security.
TOF.DealingServers	String	Dealing Server Identifier. Supports multiple values delimited by semicolon.
TOF.MaxSimultaneousRequests	Amount	Number of concurrent requests sent to the server. <b>0</b> – unlimited; <b>1</b> – value by default.

TOF.Session.Reconnect.MinTimeout	Seconds	The minimum value of the timeout between attempts to reconnect to the server. <b>5</b> – value by default.
TOF.Session.Reconnect.MaxTimeout	Seconds	The maximum value of the timeout between attempts to reconnect to the server. <b>60</b> – value by default.
TOF.Session.Reconnect.TimeoutIncrement	Seconds	Value on which timeout increases after each failed connection attempt (until reaches the value <i>TOF.Session.Reconnect.MaxTimeout</i> ). <b>5</b> – value by default.
TOF.Message.Tickets.MinNumber	Int, > 0	Minimum number of the deal message used by server. <b>1</b> – value by default. <b>Note:</b> see TOF protocol documentation for rules of messages numbering.
TOF.Message.Tickets.MaxNumber	Int, > 0	Maximum number of the deal message used by server. <b>999999</b> – value by default. <b>Note:</b> see TOF protocol documentation for rules of messages numbering.
TOF.Debug.MessagesLog.Enable	[true false]	Enables/disables saving incoming TOF messages to separate file on the disk.
TOF.Debug.MessagesLog.Dir	Path	Path to the directory for saving incoming TOF messages.
<i>FIX protocol section</i>		
FIX.Properties.File	Path	Path and name of the FIX library configuration file ( <i>fix.engine.properties</i> ).
FIX.Sender.ID	String	Identifier of the FIX-server's client.
FIX.Target.ID	String	FIX-server identifier.
FIX.Version	[4.0, 4.1, 4.2, 4.3, 4.4, 5.0]	Used FIX protocol version. <b>4.4</b> – recommended value.
FIX.Target.Host	Host	IP-address or DNS-name of the FIX-server's host.
FIX.Target.Port	[1..65535]	FIX server port. <b>9106</b> – value by default.
FIX.HeartBeatInterval	Seconds	Heartbeat interval for FIX protocol. <b>30</b> – value by default.
FIX.Session.Reconnect.MinTimeout	Seconds	The minimum value of the timeout between attempts to reconnect to the server. <b>5</b> – value by default.
FIX.Session.Reconnect.MaxTimeout	Seconds	The maximum value of the timeout between attempts to reconnect to the server. <b>60</b> – value by default.
FIX.Session.Reconnect.TimeoutIncrement	Seconds	Value on which timeout increases after each failed connection attempt (until reaches the value <i>FIX.Session.Reconnect.MaxTimeout</i> ). <b>5</b> – value by default.
FIX.Session.StatusCheckTimeout	Seconds	Interval for checking status of the FIX session. <b>30</b> – value by default.
<i>Logging section</i>		
RBLog.log4cxx.Config	Path	Path to the logging configuration file.

FIX level setting can be configured by means of the **fix.engine.properties** file.

## Reuters Bridge as a Windows service

Generally, for work with Windows services, the following commands are used:

```

rem Service Installation

ReutersBridge --name <Service name> --config <path to cfg file> --run install

rem Service Uninstallation

ReutersBridge --name <Service name> --run uninstall

```

If the `--name` parameter is not specified, "ReutersBridge" will be used.

If the `--config` parameter is not specified, "<path to the executable file>/config.cfg" will be used.

**Note:** If the application works as a service, paths in configuration files must be absolute.

## Logging

All incoming and outgoing messages can be logged for further analysis and troubleshooting.

- `Tof-messages.in` – logging incoming TOF messages in text file;
- `SenderCompID-TargetCompID.in` – logging incoming messages from FIX session in text file;
- `SenderCompID-TargetCompID.out` – logging outgoing messages from FIX session in text file;

Logging is configured by means of the `config.cfg` and `log4cxx.properties` configuration files. Refer to the [How to configure logging and logs rotation in Reuters Bridge](#) article for detailed instructions.

## Error diagnosis by means of command line

Generally, error diagnosis is performed by means of application logs. However, sometimes (e.g. for the cases which can take place before logging subsystem initialization is done) it can be useful to run the application in the command line mode.

To run the application in command line mode, you need to:

1. Open the Windows command line and go to the application directory (e.g., `cd c:\b2bits\ReutersBridge`)
2. Run the application using the following command: `ReutersBridge.exe --run console`

Below is the list of console parameters:

Parameter	Arguments	Description
--run	console	Application launch in command line mode.
	install	Application service installation.
	uninstall	Application service uninstallation.
--config	Path	Allows to use specified configuration file.
--xml-storage	Path	Allows to use specified state file.
--dbg-no-fix		Debug parameter - disables transmitting messages to FIX server.
--dbg-tof-file	Path	Allows to replay messages from the log file (see <code>TOF.Debug.MessagesLog.Enable</code> and <code>TOF.Debug.MessagesLog.Dir</code> configuration properties). Connection to TOF server is not established.
--dbg-tof-rate	Number of messages in second	Defines the number of messages to be replayed in one second. Is used together with <code>dbg-tof-file</code> parameter. <i>Default value – 1.</i>
--dbg-tof-cyclic		Defines the mode of cyclic logs replay. Is used with <code>dbg-tof-file</code> parameter. <i>Default value – disabled.</i>