

## Reuters TRIARCH 2000

# Triarch 2000 BestFeed Server Functional Specification

Date: 19 June 2001  
Version: 1.2 (Release)  
Author: Serge Kimura

Orchid Technology K.K.  
5F. Landic Toranomom II Bldg., 3-7-8 Toranomom, Minato-ku, Tokyo 105-0001 Japan

*This document contains information proprietary to Orchid Technology K.K. and may not be reproduced, disclosed or used in whole or part without express permission of Orchid Technology K.K.*

Orchid Technology K.K., by publishing this document, does not guarantee that any information contained herein is and will remain accurate or that use of the information will ensure correct and faultless operation of the relevant service or equipment. Orchid Technology K.K., its agents and employees shall not be held liable to or through any user of this document for any loss or damage whatsoever resulting from reliance on the information contained herein.

This document refers to the products and trademarks of manufacturers. Acknowledgement is made of all trademarks, registered trademarks and trading names that are referred to in the text.

This software application contains the Reuters SSL™. To use this software, a runtime licence is required from Reuters. Please consult your local Reuters contact.

© 2000, 2001 Orchid Technology K.K. All rights reserved.

Published by  
Orchid Technology K.K.

## Amendment History

Version	Date	Contents
1.0	26-Feb-01	<p>5.1. Dataset definition file format:</p> <ul style="list-style-type: none"> <li>- Added special value '\dataset_name' to RecordTemplate section to be able to publish the instrument's dataset name as a field in SSL record;</li> <li>- Added two new sections (Exceptions and ExceptionMap) to implement direct symbol conversion for instruments it is difficult to create parsing rules for.</li> </ul> <p>5.3. Server configuration file format:</p> <ul style="list-style-type: none"> <li>- Added BESTFEED_SSL_SNK_USER parameter to enable Sink side permissioning of the Server;</li> <li>- Added MASTER_FID_FILE parameter to specify location of fields dictionary file.</li> </ul> <p>6.4. BestFeed map collector:</p> <ul style="list-style-type: none"> <li>- Version of supported DACS files format changed from 4.2 to 4.4.</li> </ul> <p>7.1. Symbol conversion and fields mapping procedures:</p> <ul style="list-style-type: none"> <li>- Added specification for symbol conversion procedure in case of direct mapping of exception instrument codes.</li> </ul> <p>Appendix A. Dataset definition file example:</p> <ul style="list-style-type: none"> <li>- Modified to add an example of usage of direct symbol conversion (Exceptions and ExceptionMap sections).</li> </ul> <p>Appendix C. Server configuration file example:</p> <ul style="list-style-type: none"> <li>- Modified to add two new start-up parameters (BESTFEED_SSL_SNK_USER and MASTER_FID_FILE).</li> </ul>
1.1	31-May-01	<p>Whole document:</p> <ul style="list-style-type: none"> <li>- Changed source state string from "OK" to "UP".</li> </ul> <p>5.3. Server configuration file format:</p> <ul style="list-style-type: none"> <li>- Corrected default value for BESTFEED_SSL_SOURCE_NAME parameter from "No default" to "BESTFEED";</li> <li>- Added BESTFEED_ENABLE_DACs parameter to enable/disable DACs permissioning functionality;</li> <li>- Corrected name of BESTFEED_SSL_SNK_USER to BESTFEED_SSL_SINK_USER.</li> </ul>
1.2	19-June-01	<p>5.1. Dataset definition file format:</p> <ul style="list-style-type: none"> <li>- Removed Exceptions section, which was defining instrument codes for direct mapping. It is now being handled through separate file.</li> </ul> <p>5.2. Direct symbols mapping file format:</p> <ul style="list-style-type: none"> <li>- New section to specify format of the file that defines lists of exception instrument codes for direct symbol mapping;</li> <li>- As a result, numbering of following sections has been changed (5.2 to 5.3; 5.3 to 5.4).</li> </ul>

Version	Date	Contents
		5.4. Server configuration file format: <ul style="list-style-type: none"><li>- Added BESTFEED_DIRECT_SYMBOL_MAP_FILE_NAME parameter to specify file name for direct symbols mapping.</li></ul> 7.1. Symbols conversion and fields mapping procedures: <ul style="list-style-type: none"><li>- Modified to reflect usage of a separate file defining exception instrument codes for direct symbols mapping.</li></ul>

# Contents

<b>CONTENTS .....</b>	<b>5</b>
<b>1. GENERAL.....</b>	<b>6</b>
1.1 DESCRIPTION AND HARDWARE REQUIREMENTS .....	6
1.2 REFERENCES.....	6
<b>2 BESTFEED SERVER FUNCTIONAL REFERENCES .....</b>	<b>7</b>
2.1 BESTFEED SERVER FUNCTIONALITY .....	7
2.2 NOT-SUPPORTED DATA MESSAGE TYPES.....	8
<b>3 PUBLISHED DATA .....</b>	<b>9</b>
3.1 REUTERS SSL FID INITIALISATION VALUES BY FIELD TYPE.....	9
3.2 INDIVIDUAL RICS (PER DATASET).....	9
3.3 SYSTEM RICS .....	9
3.4 DACS-RELATED RICS.....	10
<b>4 TRIARCH RECORD TEMPLATES.....</b>	<b>11</b>
4.1 SERVER SYSTEM INFORMATION.....	11
4.2 DATASET SYSTEM INFORMATION .....	11
4.3 INDIVIDUAL INSTRUMENTS (PER DATASET) .....	12
<b>5 BESTFEED SERVER FILE FORMATS .....</b>	<b>13</b>
5.1 DATASET DEFINITION FILE FORMAT.....	13
5.2 DIRECT SYMBOLS MAPPING FILE FORMAT.....	17
5.3 DATA FEED STATUS FILE FORMAT .....	18
5.4 SERVER CONFIGURATION FILE FORMAT (START-UP PARAMETERS).....	19
<b>6 DACS SUPPORT.....</b>	<b>21</b>
6.1 PERMISSION DEFINITION PAGES .....	21
6.2 PAGE FORMAT .....	21
6.3 RECORD TEMPLATE .....	23
6.4 BESTFEED MAP COLLECTOR .....	24
<b>7 BESTFEED SERVER OPERATIONAL DESIGN.....</b>	<b>25</b>
7.1 SYMBOLS CONVERSION AND FIELDS MAPPING PROCEDURES.....	25
7.2 ACTIVE FEED SWITCHING PROCEDURES .....	26
7.3 TRIARCH-BASED LOAD BALANCING AND FAILOVER PROCEDURES.....	27
7.4 ITEM STATUS HANDLING PROCEDURES .....	28
<b>APPENDIX A – DATASET DEFINITION FILE EXAMPLE .....</b>	<b>29</b>
<b>APPENDIX B – DIRECT SYMBOLS MAPPING FILE EXAMPLE .....</b>	<b>33</b>
<b>APPENDIX C – FEED STATUS FILE EXAMPLE.....</b>	<b>34</b>
<b>APPENDIX D – SERVER CONFIGURATION FILE EXAMPLE .....</b>	<b>35</b>

# 1. General

## 1.1 Description and hardware requirements

BestFeed server is Reuters SSL4 source server that listens to several data feeds, delivering data for a given set of instruments and provides a single output service with data from the most preferred feed on Triarch network.

BestFeed server requires Sun Ultra SPARC workstation with Solaris 2.5.1 or higher.

## 1.2 References

Following documents were used when creating this specification:

- Triarch Programming Guide Reuters SSL 4.0 Reference Manual (Release June 1996).
- Triarch Programming Guide Reuters SSL 4.0 Developer's Guide (Release June 1996).
- Effective AWK Programming. A User's Guide to GNU Awk. Chapter: Regular Expressions (Edition 1.0.3 February 1997; [http://www.gnu.org/manual/gawk-3.0.3/html\\_chapter/gawk\\_toc.html](http://www.gnu.org/manual/gawk-3.0.3/html_chapter/gawk_toc.html)).
- Reuters Product Definition Pages. User Guide (version 1, 1996).

## 2 BestFeed Server Functional References

### 2.1 BestFeed Server Functionality

BestFeed server will perform following functions:

- Dataset management. The server will manage a collection of distinct datasets. For each dataset (defined by exchange and product type), the server will use a series of data feed sources that provide updates for the items belonging to a given dataset;
- Data feeds ranking. The various data sources feeding data for a given dataset will be ranked in order of their desirability and the state of each source ("UP", "SUSPECT", "DOWN"). Desirability of a data feed will be determined by logic separate from the server informing a judgment of the business users of the data. The current state and rank of each source feed within a dataset will be stored in a feed status file;
- Data record mapping. For data coming from a data feed with highest rank the server will map available fields to resulting record according to record template of a given dataset;
- Symbols mapping. The server will perform a mapping between feed-specific symbols available on the various data feeds supporting a given dataset and a standard symbol convention, providing a generic symbol interface for clients' requests. The rules for symbols mapping will be implemented using extended regular expression formulas;
- Data feed switching. For a given dataset, the server will deliver data from the feed with highest rank. In case when currently active feed's state is no longer "UP", the server will switch to the next highest ranked feed with "UP" state. In some cases, the server will change state of the feed itself, but an operator will normally manage feeds' states marking them as "SUSPECT" or "UP";
- Watchlist. Per dataset, the server will maintain a watchlist of active items in order to support a quick change in active data feed. The watchlist will contain the standard symbol, name of currently active feed, and for each data feed supporting a given dataset, feed-specific symbol and data cache with up-to-date IMAGE data from the feed;
- System status notifications. The server will maintain a special item to notify clients about changes in various parameters and states of the server and supported data feeds. For each configured dataset, the server will also publish a special item with various updated statistics related to the dataset;

- Triarch failover and load balancing. As a standard Triarch source server, the BestFeed server can be setup to run in multiple server configuration to allow Triarch failover mechanism to switch between servers and load balance requests across group of the servers;
- DACS support. Each given dataset will be configured as a separate subservice and assigned a numeric entitlement code. The server will publish these codes on Triarch using SSL locks mechanism. The server will also publish, as a series of special items, the map of its entitlement codes and subservices. A special sink application ("map collector") will be provided in order for the DACS administrator to be able to assign permissions to the subservices on the BestFeed service;
- Supported datasets. Datasets supported by the server will be defined in a special datasets configuration file;
- Logging. The server will maintain a date- and time-stamped log files with various start-up and operational messages. Every extraordinary condition will also be recorded in the server's log file. Once log file reaches it's maximum size, it will be renamed appending ".old" to original file name and new log file will be created;
- Triarch source name. It will be possible to set Triarch source name for the BestFeed server through configuration file.

## 2.2 Not-Supported Data Message Types

The BestFeed server will not support following data types:

- Chain instruments;
- TS1 historical data;
- Reuters News2000 data;
- Page-based data;
- SSL rename messages.

## 3 Published Data

### 3.1 Reuters SSL FID Initialisation Values by Field Type

Reuters SSL messages will be initialised to the following field values:

Field type	Initial value
INTEGER, PRICE	" +0" ("0" will be used in case of real zero value)
ALPHANUMERIC, ENUMERATED, DATE	"_" (one space)
TIME	"__:__" (two spaces, colon, two spaces)
TIME SECONDS	"__:__:__" (two spaces, colon, two spaces, colon, two spaces)

### 3.2 Individual RICs (per dataset)

Each dataset will be uniquely defined by product type and exchange. As a general rule, a dataset will be assigned prefix (defining the product type) and suffix (defining the exchange). These will be configured in datasets definition file. Although in some cases, it might be possible to omit suffix in certain datasets definitions as long as it does not introduce ambiguity across all defined datasets, the prefix is a mandatory part of individual RIC format.

The format of individual RICs will be:

"<product\_type\_prefix><instrument\_code>[<exchange\_suffix>]", where:

<product\_type\_prefix> is a prefix which defines a product type of the instrument;

<instrument\_code> is a standard code for the instrument;

<exchange\_suffix> is a suffix which defines an exchange where the instrument is traded.

The values for product type prefix and exchange suffix as well as the format of instrument code will be defined (per dataset) in dataset definition file using regular expression templates (see [5.1 Dataset Definition File Format](#)).

### 3.3 System RICs

The server will handle various status messages from supported data feeds as well as changes in it's own state by publishing updates on a special item. The RIC for this item will be defined by BESTFEED\_SERVER\_SYS\_RIC parameter in the server's configuration file.

Furthermore, for each configured dataset, the server will publish a special item with various statistical data related to the dataset. When set up to run in multi-servers configuration, each instance of the BestFeed server will publish unique items for its own datasets.

The RIC format of these items will be:

"[<server\_prefix><dataset\_name>", where:

- <server\_prefix> is an optional prefix specifying instance of BestFeed server as defined by BESTFEED\_DATASET\_SYS\_RIC\_PREFIX parameter in the server's configuration file;
- <dataset\_name> is name of the dataset as configured in dataset definition file.

### 3.4 DACS-related RICs

The server will publish DACS-related information in a form of Permission Definition Pages, mapping datasets to their entitlement codes (see 6 DACS Support).

## 4 Triarch Record Templates

### 4.1 Server System Information

RIC\_FORMAT: Defined in the BestFeed server's configuration file  
 TEMPLATE\_NAME: BEST\_FEED\_SYSTEM\_STATUS  
 TEMPLATE\_NUMBER: -7000

FID	Acronym	Type	Length (max)	Description
1	PROD_PERM	INTEGER	5	Permission code
2	RDNDISPLAY	INTEGER	3	Display template number
3	DSPLY_NAME	ALPHANUMERIC	16	Set to "BestFeed Status"
5	TIMACT	TIME	5	Update time
17	ACTIV_DATE	DATE	11	Update date
264	BCAST_TEXT	ALPHANUMERIC	255	Latest system or error log message
1024	STORY_TIME	TIME_SECONDS	8	Time of latest log message
1027	STORY_DATE	DATE	11	Date of latest log message
1691	SPARE_VL1	INTEGER	15	Number of datasets
1692	SPARE_VL2	INTEGER	15	Total watchlist size

### 4.2 Dataset System Information

RIC\_FORMAT: .[<server>\_]<dataset>, where <server> is a prefix specifying instance of the server defined in server's configuration file; <dataset> is name of dataset defined in dataset definition file  
 TEMPLATE\_NAME: BEST\_FEED\_DATASET  
 TEMPLATE\_NUMBER: -7001

FID	Acronym	Type	Length (max)	Description
1	PROD_PERM	INTEGER	5	Permission code
2	RDNDISPLAY	INTEGER	3	Display template number
3	DSPLY_NAME	ALPHANUMERIC	16	Name of dataset
5	TIMACT	TIME	5	Update time
17	ACTIV_DATE	DATE	11	Update date
77	NUM_MOVES	INTEGER	15	Number of active data feed switches on this dataset
255	PROC_DATE	DATE	11	Date of last switch of active data feed
256	PROC_TIME	TIME	5	Time of last switch of active data feed
258	SEG_TEXT	ALPHANUMERIC	255	Active feed on this dataset
264	BCAST_TEXT	ALPHANUMERIC	255	Latest system or error log message for this dataset
1024	STORY_TIME	TIME_SECONDS	8	Time of latest log message
1027	STORY_DATE	DATE	11	Date of latest log message
1691	SPARE_VL1	INTEGER	15	Number of items in watchlist for this dataset

### 4.3 Individual Instruments (per dataset)

For each supported dataset, record template will be defined in dataset definition file, detailing all the fields to be published by the server, their availability and (if different from default) field number from each data feed supporting the dataset. Record templates for individual instruments may vary from dataset to dataset. However, as a rule, each record template will contain as a first field FID 1 (PROD\_PERM) with a numeric value of entitlement code of the dataset the instrument belongs to.

## 5 BestFeed Server File Formats

### 5.1 Dataset Definition File Format

Dataset definition file will define all different datasets supported by the server. For each dataset it will define name of the set, DACS entitlement code, product type prefix, standard instrument code, exchange suffix and the list of source feeds that provide data for items from a given dataset. For each data feed in that list, the file will define an output record template with fields mapping as well as rules for feed-specific symbol conversion. Dataset definition file makes an extensive use of regular expression formulas.

The following symbols are exceptions and not used as standard regular expression symbols:

- "." (dot), used in RIC format;
- "(" and ")" (opening and closing parentheses), used internally for lexemes matching.

Please refer to [Appendix A – Dataset definition file example](#).

Dataset definition file will have following format.

```

Dataset(dataset_name_1, DACS_code_1)
{
  <prefix: product_type_prefix>
  standard_instrument_code_template
  <suffix: exchange_suffix>

  RecordTemplate
  {
    `field_0`='fixed_value_0',
    `field_1`='fixed_value_1',
    `field_2`,
    ...
    `field_N`
  }

  DataFeed(SSL_source_name_1)
  {
    FieldMap
    {
      {`output_field_1`, `input_field_1`}
      ...
      {`output_field_N`, `input_field_N`}
    }
    Map(lexeme_1)
    {
      {`original_value_1`, `result_value_1`}
      ...
      {`original_value_N`, `result_value_N`}
    }
    ...
    Map(new_lexeme=lexeme_Y,lexeme_X,...)
    {
      {`original_value_1`, `result_value_1`}
      ...
      {`original_value_N`, `result_value_N`}
    }
    feed_specific_symbol_template
  }
  ...
  DataFeed(SSL_source_name_X)
  {
    .....
  }
}
.....
Dataset(dataset_name_X, DACS_code_X)
{
  .....
}

```

#### Sections definitions:

- Dataset defines a dataset;
- RecordTemplate defines record template to be published by the server for all items in current dataset;
- DataFeed groups a source feed-specific fields and symbols mappings;
- FieldMap defines mapping for the fields where it varies from default one in current data feed;
- Map defines a series of mapping rules for feed-specific symbol conversion in current data feed.

Other definitions:

<b>dataset_name_X</b>	name of the dataset, to be used as a reference in data feed status file, as well as special RIC for the dataset system information;
<b>DACS_code_X</b>	the dataset entitlement code for DACS permission;
<b>field_N</b>	Reuters mnemonic name or ID of the field to be included in a record published by the BestFeed server for the dataset;
<b>fixed_value_N</b>	used when the <b>field_N</b> must have a fixed value across all items in the dataset. If special value <i>'^active_server'</i> is specified, the server will update <b>field_N</b> with the name of currently active source feed for the dataset. If special value <i>'^dataset_name'</i> is specified, the server will update <b>field_N</b> with the name of the dataset as defined by <b>dataset_name_X</b> ;
<b>product_type_prefix</b>	prefix defining product type of the dataset;
<b>standard_instrument_code_template</b>	template defining standard code for instruments in the dataset. It is a set of patterns for regular expression matching. The format is: <lexeme_1: regular_expression_1> ... <lexeme_X: regular_expression_X>;
<b>exchange_suffix</b>	suffix defining exchange for the dataset;
<b>SSL_source_name_X</b>	SSL source name of the data feed providing data for items in the dataset;
<b>output_field_X</b>	Reuters mnemonic name or ID of the field which mapping is different from default one defined in RecordTemplate section;

<b>input_field_X</b>	ID of the field from the source data feed to be used to provide values for <b>output_field_X</b> . If 0 (zero) is specified, data for <b>output_field_X</b> is not available from the source feed;
<b>lexeme_X</b>	name of the lexeme to be used in feed-specific symbol mapping;
<b>new_lexeme</b>	lexeme created by concatenation of other lexemes to be used in special case mappings;
<b>original_value_X</b>	original value of the lexeme to be mapped. If "" ( <i>empty</i> ) is specified, the empty lexeme can also be matched. If special value <i>^others</i> is specified, the rule will match all unlisted values as well;
<b>result_value_X</b>	value to which <b>original_value_X</b> will be mapped for the lexeme mapping. If "" ( <i>empty</i> ) is specified, the lexeme will be mapped to empty. If special value <i>^error</i> is specified, the rule will generate a mapping error;
<b>feed_specific_symbol_template</b>	template defining feed-specific symbol format resulting from applying all regular expression matching <b>standard_instrument_code_template</b> and lexemes mapping. The format is: <lexeme_N> <lexeme_M> <new_lexeme> ... <: string_constant>

## 5.2 Direct Symbols Mapping File Format

Direct symbols mapping file will define for each dataset a list of standard instrument codes for which default symbol conversion procedures (as specified in dataset definition file) cannot be used and direct mapping is to be used instead.

Direct symbols mapping file will have following format.

```
DatasetExceptions(dataset_name_1=SSL_source_name_1,SSL_source_name_2,...SSL_source_name_N)
{
    { 'exception_ric_1'='source_1_ric_1', 'source_2_ric_1',... 'source_N_ric_1' }
    { 'exception_ric_2'='source_1_ric_2', 'source_2_ric_2',... 'source_N_ric_2' }
    ...
    { 'exception_ric_M'='source_1_ric_M', 'source_2_ric_M',... 'source_N_ric_M' }
}
.....
DatasetExceptions(dataset_name_X=SSL_source_name_1,SSL_source_name_2,...SSL_source_name_N)
{
    .....
}
```

Where:

<b>dataset_name_X</b>	name of the dataset as defined in dataset definitions file (see <a href="#">5.1 Dataset Definition File Format</a> );
<b>SSL_source_name_N</b>	SSL source name of the data feed providing data for items in the dataset. The specified SSL source names must exactly match source names defined for the dataset in dataset definitions file;
<b>exception_ric_M</b>	standard instrument code for which direct symbol mapping is defined;
<b>source_N_ric_M</b>	value to which <b>exception_ric_M</b> will be converted for direct symbol mapping on the source defined by <b>SSL_source_name_N</b> . Number of such values must be the same as number of SSL source names defined for a given dataset.

Please refer to [Appendix B – Direct symbols mapping file example](#).

### 5.3 Data Feed Status File Format

Data feed status file will comprise a list of datasets and the current state of each feed supporting the dataset. The file will have following format.

```
Dataset(dataset_name_1)
{
  SSL_source_name_1: status_1
  ...
  SSL_source_name_N: status_N
}
...
Dataset(dataset_name_X)
{
  SSL_source_name_1: status_1
  ...
  SSL_source_name_N: status_N
}
```

Where:

<b>dataset_name_X</b>	name of the dataset as defined in dataset definition file (see <a href="#">5.1 Dataset Definition File Format</a> );
<b>SSL_source_name_N</b>	SSL source name of the data feed providing data for items in the dataset;
<b>status_N</b>	current status of the data feed. It can be "UP" (if data feed is considered to be a reliable source of data) or "SUSPECT" (if there is a problem with the feed).

For each dataset, data feeds are listed in order of their desirability (i.e. the feed with the highest rank goes first).

Please refer to [Appendix C – Feed status file example](#).

## 5.4 Server Configuration File Format (Start-Up Parameters)

The BestFeed configuration file will have format of UNIX shell start-up script with all configuration parameters defined as exported environmental variables. The file will have following format.

```
parameter_name_1 = parameter_value_1
export parameter_name_1
...
parameter_name_N = parameter_value_N
export parameter_name_N
```

Where:

**parameter\_name\_N**            name of the configuration parameter;  
**parameter\_value\_N**        start-up value of the parameter.

Following parameters can be used to configure the BestFeed server.

Parameter name	Definitions	Default value
BESTFEED_SSL_SOURCE_NAME	SSL service name	"BESTFEED"
BESTFEED_ENABLE_DACS	DACS flag: if set to TRUE the server will create and publish DACS Access Locks for each non-system item	"TRUE"
BESTFEED_MAX_CACHE_SIZE	The maximum number of items supported on SSL Source Distributor. This parameter must be in range of [100, 20000]	2000
BESTFEED_SSL_SNK_USER	The value of this parameter can be used to convey permission information of the server's Sink part	"" (empty)
BESTFEED_DATASET_DESCR_FILE_NAME	Dataset definition file name	"bestfeed.rul"
BESTFEED_DIRECT_SYMBOL_MAP_FILE_NAME	Direct symbols mapping file name	"bestfeed.direct"
BESTFEED_FEED_STATUS_FILE_NAME	Feed status file name	"bestfeed.status"
BESTFEED_LOG_FILE_NAME	Log file name	"bestfeed.log"
BESTFEED_SSL_LOG_FILE_NAME	SSL messages log file name	"bestfeed_ssl.log"
BESTFEED_MAX_LOG_FILE_SIZE	Maximum size of the log file (in bytes). When the file reaches it's maximum size, it will be moved to *.old (overwriting if it existed).	256 * 1024 (256 Kb)
BESTFEED_SERVER_SYS_RIC	Server system status RIC. The server will publish various status and statistical information on this item.	".STATUS"
BESTFEED_SERVER_SYS_ITEM_PROD_PERM	Server system item permission code	0
BESTFEED_SERVER_SYS_ITEM_DISPL_TEMPL_NO	Server system item display template number	0
BESTFEED_DATASET_SYS_RIC_PREFIX	Dataset system status RIC prefix.	""
BESTFEED_DATASET_SYS_ITEM_PROD_PERM	Dataset system item permission code	0

Parameter name	Definitions	Default value
BESTFEED_DATASET_SYS_ITEM_DISPL_TEMPL_NO	Dataset system item display template number	0
BESTFEED_DACS_INFO_START_PAGE	DACS permission first page number.	0
BESTFEED_COMM_FAULT_PROCESS	Server's reaction on SSL Sink Distributor channel dismount. "INFORM" – stale all items and wait for recovery; "DISMOUNT" – disconnect from SSL Source Distributor	"INFORM"
BESTFEED_INIT_TIMEOUT	Server initialisation timeout. This parameter defines the time interval (in seconds) the server will wait after mounting SSL Sink Distributor channel before activating the service. This interval must be long enough to receive all initial SSL service state events, because the services that become UP after initialisation will not be activated until the server receives USR1 signal. The value must be in the range of [5, 900]	15
BESTFEED_CLOSE_LINGER_TIMEOUT	Close linger timeout. This parameter defines the time interval (in seconds) during which the data streams related to an instrument will remain open after instrument was closed. The value must be in the range of [0, 1800]	240
MASTER_FID_FILE	Fields dictionary file. The server will search for the file in several locations in following order: <ol style="list-style-type: none"> <li>1) file specified by this parameter;</li> <li>2) appendix_a file in the server's local directory;</li> <li>3) appendix_a file in /var/Triarch directory</li> </ol>	"appendix_a"

Please refer to [Appendix D – Server configuration file example](#).

## 6 DACS Support

### 6.1 Permission Definition Pages

The permission definition pages are a series of IDN large pages that contain details of the BestFeed server's datasets and their corresponding entitlement codes (permissionable entities). These pages are designed for use by computer system and not for visual display. They provide information required to build and operate local data entitlements systems. These pages will be used by the Reuters entitlements system, the Data Access and Control System (DACS).

These pages provide a structured means to:

- identify datasets supported by the BestFeed server and permissionable entities used for them;
- identify time applicability for the datasets.

These pages do not contain pricing information.

### 6.2 Page Format

Permission definition pages will have the following RIC structure:

**"PERMISSION<page\_sequence>"**

Where <page\_sequence> is a three-digit number allowing for series of pages, starting at number defined by BESTFEED DACS\_INFO\_START\_PAGE parameter in the server's configuration file.

All pages use the 25x80 page template; field list #82 (PAGE\_25X80). The data contained within the fields uses the ASCII character set. All the row fields are padded on the right with spaces, for example, if the dataset name is defined as a 64 characters field within the 80 characters page row field, then the name "TSE\_Stocks" will have 54 trailing spaces. Row one (i.e. FID #315) of all pages is always reserved for the page title.

Each row starts with a non-space character in the first position of the row field. List of definitions may also extend over more than one page. Empty lines or lines with a space character in the first position should be ignored.

Row 2 through to row 25 (FID #316-339) each contains permission definition for a single dataset in the format:

"<dataset\_name><space><entitlement\_code><space><revision\_date>"

Where:

<dataset\_name> is a sixty-four characters field containing name of a dataset as defined in dataset definition file.

<entitlement\_code> is a six characters field containing an entitlement code (permissionable entity) for the given dataset as defined in dataset definition file.

<revision\_date> is an eight characters field containing the date when the dataset entitlement information was last updated. The field is in the ISO standard date format: YYYYMMDD, e.g. 20010214 for 14 February 2001.

First page code for permission definition pages is "PERMISSION<first\_page>", where <first\_page> is a three-digit number defined in the server's configuration file to identify first page number in series of permission definition pages published by the instance of the server.

In multiple-server configurations, each instance of the BestFeed server must be configured properly to guarantee that it's own range of series of permission definition pages is not overlapping with those from other instances. This can be done by setting BESTFEED\_DACS\_INFO\_START\_PAGE parameter in each next instance's configuration file to a value that is greater than possible last page number in previous instance's permission pages range (see 5.4 Server Configuration File Format (Start-Up Parameters)).

Example: Each permission definition page can hold information about up to 24 datasets, so if a particular instance of the server has 26 different datasets configured and it's first page number is set to "000", it will publish two permission definition pages ("PERMISSION000" and "PERMISSION001"). Therefore, the next instance of the server must be configured to have first page number set to "002" to avoid overlapping pages' codes.

## 6.3 Record Template

TEMPLATE\_NAME: PAGE\_25X80

TEMPLATE\_NUMBER: 82

FID	Acronym	Type	Length (max)	Description
1	PROD_PERM	INTEGER	5	Set to 0
2	RDNDISPLAY	INTEGER	3	Set to 0
259	RECORDTYPE	INTEGER	3	Set to 228 (product definition pages)
315	ROW80_1	ALPHANUMERIC	80	Page title always set to "DATASET PERMISSIONS DATA"
316	ROW80_2	ALPHANUMERIC	80	Character text string corresponding to a row of data on a page
317	ROW80_3	ALPHANUMERIC	80	
318	ROW80_4	ALPHANUMERIC	80	
319	ROW80_5	ALPHANUMERIC	80	
320	ROW80_6	ALPHANUMERIC	80	
321	ROW80_7	ALPHANUMERIC	80	
322	ROW80_8	ALPHANUMERIC	80	
323	ROW80_9	ALPHANUMERIC	80	
324	ROW80_10	ALPHANUMERIC	80	
325	ROW80_11	ALPHANUMERIC	80	
326	ROW80_12	ALPHANUMERIC	80	
327	ROW80_13	ALPHANUMERIC	80	
328	ROW80_14	ALPHANUMERIC	80	
329	ROW80_15	ALPHANUMERIC	80	
330	ROW80_16	ALPHANUMERIC	80	
331	ROW80_17	ALPHANUMERIC	80	
332	ROW80_18	ALPHANUMERIC	80	
333	ROW80_19	ALPHANUMERIC	80	
334	ROW80_20	ALPHANUMERIC	80	
335	ROW80_21	ALPHANUMERIC	80	
336	ROW80_22	ALPHANUMERIC	80	
337	ROW80_23	ALPHANUMERIC	80	
338	ROW80_24	ALPHANUMERIC	80	
339	ROW80_25	ALPHANUMERIC	80	

## 6.4 BestFeed Map Collector

A special sink application ("map collector") is provided to allow the DACS administrator to assign permissions to the subservices for the BestFeed service. The map collector will:

- a) request the map items (Permission Definition Pages) from the BestFeed server;
- b) determine if there were any changes since the previous map was received;
- c) convert any revised information into a file that can be loaded into the DACS database.

The map collector will support map files format for DACS version 4.4 and will create three files containing the following information:

- The current map;
- The prior map;
- The changes between the prior map and the current map.

Except when the map is initially loaded, the important details for the DACS administrator are the changes from the last file. Thus the file loaded into the DACS database should be a file of changes, not the complete file. (Naturally, at the time of initial loading the whole map is loaded as changes.)

[Not complete. To be defined]

## 7 BestFeed Server Operational Design

### 7.1 Symbols Conversion and Fields Mapping Procedures

The server will operate as follows for a given dataset. For each item requested of the server, the symbol conversion rules (from dataset definition file) will be applied. Product type prefix and exchange suffix will be used to determine dataset to which the item belongs to. If the dataset has a list of exception instruments' codes for direct symbol conversion, the server will use exception-mapping rules defined for each data feed (from direct symbols mapping file) to construct feed-specific symbols for a given dataset. Otherwise, the server will try to match instrument code using product type prefix, standard instrument code template and exchange suffix from corresponding dataset definition. If the matching is successful and all lexemes are defined, the server will use data feed-specific mapping rules to convert values of certain lexemes. It will then use feed specific symbol template to construct a new symbol from original and/or converted lexemes. The process described above will be repeated for each data feed defined for a given dataset. Feed-specific symbols of the item will then be requested from each data feed supporting the dataset. However, if the symbol conversion rules cause different items (from different datasets) to be converted to the same feed-specific symbol, the server will report error and reject all further requests for the items.

Upon receiving item's data from a source feed, the server will use fields mapping rules to create a record for the item. First, the server will create a default record as defined by record template for each dataset in dataset definition file. As a default, the same field IDs will be used to retrieve values from source data and populate resulting record (e.g. if dataset's default record template has field with ID 6 or TRDPRC\_1, then the value of the field with ID 6 from source data will be used as value for resulting field ID 6). However, for each data feed there can be defined several feed-specific fields mapping rules. If such rules are defined for current source feed, the server will use them to overwrite default values in resulting record (e.g. if data feed-specific field mapping rule has field with ID 6 or TRDPRC\_1 mapped to field ID 22, then the value of the field with ID 22 from source data will be used as value for resulting field ID 6). If a certain field in this feed-specific mapping rules maps to zero value ('0'), it means that the source feed doesn't provide data for the field, and such a field will be set to initialisation value according to it's type (see [3.1 Reuters SSL FID Initialisation Values by Field Type](#)). This is done to indicate that the field is not available from a given data feed.

## 7.2 Active Feed Switching Procedures

When a problem is discovered with a feed, the internal state of the feed will be changed from "UP" to "SUSPECT" or "DOWN". This will be done either automatically (when the server handles a Triarch "Service Down" global status message for the feed) or manually (by an operator). If that feed was the active feed for a dataset, the server will switch its active feed to the most desirable feed with "UP" status. If that feed was not the active feed, the change will simply be recorded and may affect a feed choice later, but it will not affect the active feed immediately.

If the active feed becomes "SUSPECT" or "DOWN" and there is not any other feed with "UP" status for a given dataset, the server will mark all items from the dataset as "STALE" by sending SSL\_ET\_ITEM\_STATUS\_STALE messages.

When a problem with a "SUSPECT" (or "DOWN") data feed is resolved, the state of the feed will be changed to "UP". If that feed is more desirable than the currently active feed, the server will switch the active feed to the feed that was just recovered. Otherwise, the active feed will stay the same and the state change will be recorded and may affect a choice of active feed later.

The server will respond to a special signal (KILL signal **SIGUSR1**) by re-reading the feed status file and switching feeds for datasets. It will do so in two cases (for each dataset):

- 1) The currently active feed does no longer have "UP" state. In this case the server will switch to the feed with highest rank in "UP" state;
- 2) A feed with higher rank, which had been marked "SUSPECT" or "DOWN" before, has now its state changed to "UP". In this case, the server will switch to this newly recovered feed.

Whenever the active feed has changed for a given dataset, the server will deliver IMAGE messages from local cache of the new active feed, followed by subsequent updates for all items belonging to the dataset.

As a rule, a data feed will be marked "SUSPECT" by an operator editing feed status file and sending a pre-defined signal to the server. A data feed will be marked "DOWN" automatically by the server upon receiving Triarch global status message "Service Down" for that feed. The server will not re-read feed status file in this case, but will use feed status information stored in its memory instead. A data feed will be marked "UP" by an operator editing feed status file and sending a signal to the server. If a data feed was previously marked "DOWN" after "Service Down" message, the server, upon receiving Triarch global status message "Service Up", will not automatically mark the feed "UP".

Instead, it will wait for a signal from an operator to re-read feed status file and process it as described above. This is done to prevent the situations when a feed, which was automatically marked "DOWN" by the server, has its status changed to "SUSPECT" by an operator in feed status file and at the same time Triarch global status message "Service Up" is received for this feed.

An operator will be able to detect changes in data feeds' status by subscribing to a system RIC published by the server (see [3.3 System RICs](#)).

### 7.3 Triarch-based Load Balancing and Failover Procedures

The BestFeed server as a standard Triarch source server can be set up to run in multiple-server configuration. This is done by assigning the same Triarch source name to multiple instances of the BestFeed server running on separate machines. This will be transparent to end-users who will only see one BestFeed service by that name. Such configuration provides load balancing as well as failover functionality on Triarch network. Triarch Source Distributor will generate load factor for each instance of BestFeed server using internal algorithm. The SSL will regard all servers of the BestFeed service having a normalized load level less than or equal to a default threshold value as equal for load balancing purposes. That is, the SSL will route an Open Request for a new item to the server within this subset having the least number of outstanding requests for new items. If all the servers within this subset become blocked (which means that the server's "maximum number of outstanding requests for new items" limit has been reached), the SSL will stop sending Open Requests for new items to the service until one of these servers unblocks. If no servers have a normalized load level less than or equal to a default threshold value, an Open Request for a new item will be sent to the absolutely least loaded server. Once this server becomes blocked, the SSL will stop sending Open Requests for new items to the entire service until the least loaded server unblocks.

For Triarch infrastructure to be able to switch over from failed BestFeed server to a working one, the failed server needs to be disconnected from Triarch as soon as it detects failure (e.g. loss of Sink Distributor connection, internal error, etc.). After completing recovery, the failed BestFeed server will have to re-connect to Triarch to be included into multi-server configuration. As mentioned above, the whole process will be transparent to end-users and will be handled by Triarch infrastructure itself.

The behaviour of the server on failure is defined by [BESTFEED\\_COMM\\_FAULT\\_PROCESS](#) parameter in the server's configuration file. The value of this parameter must be set to "DISMOUNT" to enable failover procedure described above (see [5.4 Server Configuration File Format \(Start-Up Parameters\)](#)).

## 7.4 Item Status Handling Procedures

If an item becomes "STALE" on the active feed of a dataset, the server will forward the item's "STALE" status to SSL.

If an item requested from a feed supporting the dataset is closed on that feed, the server will process it as follows:

- if the feed was the active feed for the dataset, the server will close the item on SSL (by sending SSL\_ET\_ITEM\_STATUS\_CLOSED message);
- if the feed was not the active feed for the dataset, the server will mark the item as closed on that feed, and, if the feed becomes active later, will close it on SSL during active feed change procedure.

Generally speaking, the current status of an item will always be defined by the status of corresponding feed-specific item on the active feed.

## Appendix A – Dataset definition file example

```

;-----
; TSE Equities
; RIC format: "S.xxxxx.T", where xxxxx is company code
; Prefix: S
; Suffix: T
; PE: 1
;-----
Dataset(TSE_Stocks , 1)
{
    ; Standard code template
    <prefix : S.>
    <company : [0-9]{4}>
    <spare : [0-35-9]?>
    <dot : .>
    <suffix : T>

    ; Default record template for the dataset
    RecordTemplate
    {
        'RDNDISPLAY'='96',
        'RECORDTYPE'='113',
        'ROW64_1'='\active_server', ; SSL source name of active server
        'ROW64_2'='\dataset_name', ; name of this dataset
        'TRDPRC_1',
        'ACVOL_1',
        'BID',
        'BID_1',
        'ASK',
        'ASK_1'
    }

    ; IDN_SELECTFEED-specific FIDs mapping and symbol conversion rules
    DataFeed(IDN_SELECTFEED)
    {
        FieldMap ; defines exceptions to a default RecordTemplate
        {
            {'BID_1', 0}
            {'ASK_1', 0}
        }

        <company >
        <spare >
        < :.>
        <suffix >
    }

    ; TSE_ITA-specific FIDs mapping and symbol conversion rules
    DataFeed(TSE_ITA)
    {
        FieldMap ; defines exceptions to a default RecordTemplate
        {
            {'BID' , 0}
            {'BID_1', 0}
            {'ASK' , 0}
            {'ASK_1', 0}
        }

        < :IA>
        <company >
        < :. >
        <suffix >
    }
}

```

(continued on the next page)

(continued from previous page)

```

; TSE64K-specific FIDs mapping and symbol conversion rules
DataFeed(TSE64K)
{
    ; No exceptions to a default RecordTemplate

    Map(spare)
    {
        {', '0'}
    }

    < :T >
    < :0000>
    <company >
    <spare >
    < :0 >
}

}
; End of TSE_Stocks dataset

;-----
; Futures
; RIC format: "F.iiimmy.x", where iii is underlying index;
; mmy is delivery month/year (MMY); x is exchange
; Prefix: F
; PE: 2
;-----
Dataset(Futures , 2)
{
    ; Standard code template
    <prefix : F.>
    <index : [A-Z]{3}>
    <month : 01|02|03|04|05|06|07|08|09|10|11|12>
    <year : [0-9]>
    <dot : .>
    <market : T|OS>

    ; Default record template for the dataset
    RecordTemplate
    {
        'RDNDISPLAY'='98',
        'RECORDTYPE'='34',
        'ROW64_1'='\active_server', ; SSL source name of active server
        'ROW64_2'='\dataset_name', ; name of this dataset
        'TRDPRC_1',
        'NETCHNG_1',
        'ACVOL_1',
        'BID',
        'ASK',
        'CONTR_MNTH',
        'OPINT_1',
        'RT_YIELD_1',
        'YLD_NETCHG',
        'BID_YIELD',
        'ASK_YIELD'
    }

    ; IDN_SELECTFEED-specific FIDs mapping and symbol conversion rules
    DataFeed(IDN_SELECTFEED)
    {
        ; No exceptions to a default RecordTemplate

```

(continued on the next page)

(continued from previous page)

```

Map(month)
{
    {'01', 'F'}
    {'02', 'G'}
    {'03', 'H'}
    {'04', 'J'}
    {'05', 'K'}
    {'06', 'M'}
    {'07', 'N'}
    {'08', 'O'}
    {'09', 'U'}
    {'10', 'V'}
    {'11', 'X'}
    {'12', 'Z'}
}

<index >
<month >
<year >
}

; TSE_FLEX-specific FIDs mapping and symbol conversion rules
DataFeed(TSE_FLEX)
{
    FieldMap ; defines exceptions to a default RecordTemplate
    {
        {'RT_YIELD_1', 0 }
        {'YLD_NETCHG', 0 }
        {'BID' , 436}
        {'ASK' , 441}
        {'BID_YIELD' , 22 }
        {'ASK_YIELD' , 25 }
    }

    Map(year)
    {
        {'0', '5'}
        {'1', '6'}
        {'2', '7'}
        {'3', '8'}
        {'4', '9'}
        {'5', '0'}
        {'6', '1'}
        {'7', '2'}
        {'8', '3'}
        {'9', '4'}
    }

    Map(market)
    {
        {'T', '0'}
        {'OS', '1'}
    }

    Map(index)
    {
        {'JNI', '18'} ; Nikkei 225
        {'JNW', '16'} ; Nikkei 300
        {'JTI', '05'} ; TOPIX
        {'JTB', '03'} ; TB
        {'JGB', '01'} ; JGB long-term
        {'JMB', '04'} ; JGB mid-term
        {'\others', '\error'}
    }
}

```

(continued on the next page)

(continued from previous page)

```
<      :T16>
<year      >
<month     >
<      :00 >
<index     >
<market    >
}
}
; End of Futures dataset
```

## Appendix B – Direct symbols mapping file example

```
-----  
; Exception instruments codes for direct conversion  
-----  
  
; TSE Equities  
DatasetExceptions(TSE_Stocks = IDN_SELECTFEED, TSE_ITA, TSE64K)  
{  
    ; Direct symbol conversion  
    ; Input      = IDN      TSE_ITA      TSE64K  
    {'S.MSFT.T' = 'MSFT.T', 'IAMSFT.T', 'TMSFT0000'}  
    {'S.IBM.T'  = 'IBM.T',  'IAIBM.T',  'T_IBM0000'}  
    {'S.SUN.T'  = 'SUN.T',  'IASUN.T',  'T_SUN0000'}  
}  
  
; No exception instruments codes for Futures dataset  
; No need to define empty section  
;DatasetExceptions(Futures = IDN_SELECTFEED, TSE_FLEX)  
;{  
;}
```

## Appendix C – Feed status file example

```
Dataset(TSE_Stocks)
{
    TSE64K      : UP
    IDN_SELECTFEED : SUSPECT
    TSE_ITA     : UP
}

Dataset(Futures)
{
    TSE_FLEX      : UP
    IDN_SELECTFEED : UP
}
```

## Appendix D – Server configuration file example

```
#!/sbin/sh
# BESTFEED SSL4 source server startup script
# SERVER VERSION: 1.0
# PLATFORM: Solaris 2.6

# SSL Source Name (default: BESTFEED)
BESTFEED_SSL_SOURCE_NAME=BESTFEED
export BESTFEED_SSL_SOURCE_NAME

# DACS enabling flag (default: TRUE)
# If TRUE the server will create and publish
# a DACS Access Lock for each non-system item.
BESTFEED_ENABLE_DACs=TRUE
export BESTFEED_ENABLE_DACs

# Cache Size (default: 2000)
# The maximum number of interests supported (per open interest the server will have a
# number of data streams requested from different sources).
# This parameter must be in the range of [100, 20000].
BESTFEED_MAX_CACHE_SIZE=2000
export BESTFEED_MAX_CACHE_SIZE

# SSL Sink User Name (default: '')
# The value of this parameter will be passed to sslSnkMount() call
# to convey permissioning information. For detailed description
# please refer to the Reuters SSL 4.0 Reference Manual.
BESTFEED_SSL_SINK_USER=bestfeed
export BESTFEED_SSL_SINK_USER

# Dataset Definition File Name (default: bestfeed.rul)
BESTFEED_DATASET_DESCR_FILE_NAME= bestfeed.rul
export BESTFEED_DATASET_DESCR_FILE_NAME

# Direct Symbols Mapping File Name (default: bestfeed.direct)
BESTFEED_DIRECT_SYMBOL_MAP_FILE_NAME= bestfeed.direct
export BESTFEED_DIRECT_SYMBOL_MAP_FILE_NAME

# Feed Status File Name (default: bestfeed.status)
BESTFEED_FEED_STATUS_FILE_NAME= bestfeed.status
export BESTFEED_FEED_STATUS_FILE_NAME

# Log File Name (default: bestfeed.log)
BESTFEED_LOG_FILE_NAME=bestfeed.log
export BESTFEED_LOG_FILE_NAME

# SSL Messages Log File Name (default: bestfeed_ssl.log)
BESTFEED_SSL_LOG_FILE_NAME=bestfeed_ssl.log
export BESTFEED_SSL_LOG_FILE_NAME

# Maximum Size of the Log File (default: 256 Kb)
# When the file reaches maximum size, it will be moved to *.old
# If *.old already exists it will be overwritten.
BESTFEED_MAX_LOG_FILE_SIZE=500000
export BESTFEED_MAX_LOG_FILE_SIZE

# Server System Status RIC (default: .STATUS)
# The server will publish various statistical information on this item.
BESTFEED_SERVER_SYS_RIC=.STATUS
export BESTFEED_SERVER_SYS_RIC

# Server System Item Permission Code (default: 0)
BESTFEED_SERVER_SYS_ITEM_PROD_PERM=0
export BESTFEED_SERVER_SYS_ITEM_PROD_PERM
```

(continued on the next page)

(continued from previous page)

```

# Server System Item Display Template Number (default: 0)
BESTFEED_SERVER_SYS_ITEM_DISPL_TEMPL_NO=0
export BESTFEED_SERVER_SYS_ITEM_DISPL_TEMPL_NO

# Dataset System Status RIC Prefix (default: .)
# For each configured dataset, the server will publish a special item with
# various statistical data and the RIC format of these items will be as follows:
# <dataset_sys_ric_prefix><dataset_name> where:
# <dataset_sys_ric_prefix> is the value of this parameter, and
# <dataset_name> is the name of the dataset as configured in the definition file.
BESTFEED_DATASET_SYS_RIC_PREFIX=.
export BESTFEED_DATASET_SYS_RIC_PREFIX

# Dataset System Item Permission Code (default: 0)
BESTFEED_DATASET_SYS_ITEM_PROD_PERM=0
export BESTFEED_DATASET_SYS_ITEM_PROD_PERM

# Dataset System Item Display Template Number (default: 0)
BESTFEED_DATASET_SYS_ITEM_DISPL_TEMPL_NO=0
export BESTFEED_DATASET_SYS_ITEM_DISPL_TEMPL_NO

# DACS Information Start Page Number (default: 0)
# The server will publish DACS information in special items
# with the RIC format as follows:
# PERMISSION<page_sequence> where <page_sequence> is a three digit
# decimal number starting from the value of this parameter.
# For example if this parameter is set to 3 then the first DACS information item
# will be available as 'PERMISSION003', next will be 'PERMISSION004' and so on.
BESTFEED_DACS_INFO_START_PAGE=0
export BESTFEED_DACS_INFO_START_PAGE

# Server's Reaction on SSL Sink Channel Dismount (default: INFORM)
# INFORM - stale all instruments and wait;
# DISMOUNT - dismount the SSL Source Channel.
BESTFEED_COMM_FAULT_PROCESS=DISMOUNT
export BESTFEED_COMM_FAULT_PROCESS

# Server Initialisation Timeout (default: 15)
# This parameter defines the time interval (in seconds) the server
# will wait after mounting of the SSL Sink Channel and before activation
# (i.e. before parsing of the Feed Status file, mounting the SSL Source Channel,
# and calling sslSrcReady()). This interval must be long enough to receive all
# initial SSL service state events since the services which become UP after that
# will not be activated until USR1 signal is sent to the server
# (USR1 signal causes Feed Status file re-parsing).
# The value must be in the range of [5, 900].
BESTFEED_INIT_TIMEOUT=5
export BESTFEED_INIT_TIMEOUT

# Close Linger Timeout (default: 240)
# This parameter defines the time interval (in seconds) for which
# the data streams related to an instrument will be kept open after
# a CloseReq message receiving. The value must be in the range of [0, 1800].
BESTFEED_CLOSE_LINGER_TIMEOUT=60
export BESTFEED_CLOSE_LINGER_TIMEOUT

# Field Dictionary File (default: appendix_a)
# The server searches for the field dictionary in the following
# locations in that particular order of preference:
# 1. file pointed by the MASTER_FID_FILE environment variable;
# 2. appendix_a file in the executable's directory;
# 3. appendix_a file in the /var/triarch directory.
MASTER_FID_FILE=/var/triarch/appendix_a
export MASTER_FID_FILE

# Start the server
./bestfeed_serv_sol2.6.v1.0 &

```