

# Tradesignal Market Data System

## User Guide

---



**tradesignal**<sup>®</sup>

## Table of Contents

<b>1</b>	<b>Overview</b>	<b>6</b>
1.1	Architecture	6
1.2	Scope of the manual	7
1.3	Target audiences	7
1.4	Expressions and formats used in this manual	7
1.4.1	Note and Tip	7
1.4.2	Formatting	7
1.5	Glossary	7
<b>2</b>	<b>Deployment: Common Installation Configurations</b>	<b>8</b>
2.1	Scenario 1: Local installation alongside Tradesignal (single user)	8
2.2	Scenario 2: TMDS on dedicated system	8
2.3	Scenario 3: Two systems	9
2.3.1	Scenario 3a: One system as failover	9
2.3.2	Scenario 3b: Two identical systems working in parallel	9
2.4	Scenario 4: Multiple systems	9
2.4.1	Scenario 4a: Distributing feeds and symbols across individual servers	10
2.4.2	Scenario 4b: Scaling number of clients beyond scenario 3	11
2.5	Scenario 5: Local installations using a centralized licensing TMDS	11
<b>3</b>	<b>Installation</b>	<b>12</b>
3.1	Preparations for installing TMDS	12
3.2	System requirements	12
3.2.1	Supported operating systems	12
3.2.2	Minimum system requirements	12
3.2.3	Recommended system requirements for the use of TMDS	13
3.3	Software Requirements	13
3.3.1	Global Vision API (Trayport Feed Handler)	13
3.4	Setup	13
3.5	Updating	14
3.6	Uninstalling	14
3.7	Copying an existing data cache file to a new TMDS	15



<b>4</b>	<b>TMDS Management Console</b>	<b>16</b>
4.1	User interface overview	16
4.2	Application button	17
4.3	Symbol area	17
4.4	Status bar	18
4.5	Mouse-over information	18
4.6	Context menus	18
4.7	Field labels	18
<b>5</b>	<b>Starting / stopping the system</b>	<b>19</b>
<b>6</b>	<b>Managing datafeeds</b>	<b>20</b>
6.1	Setting up Thomson Reuters	21
6.2	Setting up Trayport GlobalVision	22
6.2.1	Backfilling data history from Trayport SQL server	23
6.2.2	Rolling	24
6.3	Setting up Bloomberg	25
6.4	Setting up OpenConnect	25
<b>7</b>	<b>Managing Licenses and Users</b>	<b>26</b>
7.1	The license and user group manager window	26
7.2	Setting up a centralized license server	27
7.3	License properties and usage	27
7.4	Floating licenses	28
7.5	Adding a new license	28
7.6	Removing a license	28
7.7	Using user groups	29
7.8	Adding / editing / deleting users	30
7.9	Connecting to TMDS only to access Top-Up data	30
7.10	Exporting / importing user groups	30
7.11	Viewing connected users	30
<b>8</b>	<b>TMDS Connections</b>	<b>31</b>



<b>9</b>	<b>Managing Symbols</b>	<b>32</b>
9.1	Editing reference data	34
9.1.1	Editing server sessions	34
9.2	Editing price data	35
9.2.1	Editing price data fields	35
9.2.2	Deleting candles	36
9.2.3	Adding candles	36
9.2.4	Reloading / reverting price data	36
9.2.5	Saving price data	36
9.3	Clearing and restoring data	37
9.3.1	Clearing user edits	37
9.3.2	Restoring default historic data	37
9.3.3	Restore default reference data	37
9.4	Editing data collection settings	38
9.5	Rolling Forward Symbols (User-Defined Continuations)	39
<b>10</b>	<b>Filtering Ticks</b>	<b>40</b>
10.1	Setting a bad tick filter for a single symbol	40
10.2	Editing tick filter settings	41
10.3	Setting up the session filter	42
10.3.1	Configuring session extensions	42
10.4	Checking filter settings	43
<b>11</b>	<b>Filtering Instruments (Whitelisting)</b>	<b>44</b>
<b>12</b>	<b>Backup / Restore / Synchronization</b>	<b>45</b>
12.1	Backing up a running (live) TMDS	45
12.2	Backing up a stopped TMDS	45
12.3	Restoring a TMDS	46
12.4	Synchronizing Systems	46
12.4.1	Enabling automated synchronization	47
12.4.2	Manual synchronization	48
<b>13</b>	<b>Export / Import</b>	<b>49</b>
13.1	Importing symbols	49
13.2	Exporting symbols	49



<b>14</b>	<b>Logs, Alerts &amp; Troubleshooting Tools</b>	<b>50</b>
14.1	Logging console and log file	50
14.2	Log	51
14.3	Query	51
14.4	Command line	52
14.5	Query symbol	52
14.6	Sending automatic e-mail messages to administrators	52
14.7	Preparing information for sending by e-mail	53
<b>15</b>	<b>Scripting TMDS</b>	<b>54</b>
15.1	Using batch/script files	54
15.1.1	Implementing their automatic execution	54
15.1.2	Manually starting a script	54
15.2	Nightly Maintenance	55
15.3	Command Reference	56
15.3.1	Administrative commands	56
15.3.2	TS1 commands	57
15.3.3	Instrument and data commands	57
15.3.4	Informational commands	61
15.3.5	Recovery commands	62
<b>16</b>	<b>General Troubleshooting</b>	<b>63</b>
16.1	FAQ	63
16.2	Advanced configuration and troubleshooting of TMDS for Thomson Reuters Data	65
16.2.1	Troubleshooting failed connections to Triarch/RWS (SSL4) infrastructures	65
16.2.2	Manually configuring RMDS/RTIC (SASS3) infrastructures	66
16.2.3	Entitlement Management with DACS	67



# 1 Overview

The Tradesignal Market Data System (TMDS) is designed specifically to collect and distribute real-time market data to Tradesignal desktop applications. The main advantages of using the Tradesignal Market Data System are:

- *Collection of real-time and historical data; maintaining a complete history of both*
- *Servicing large numbers of users simultaneously and without loss of performance*
- *Use of advanced data accumulation techniques to ensure rapid data retrieval irrespective of requested time period*
- *Provision of instrument asset data including time zones and trade times*
- *Support for live backup so data can be protected without affecting users*
- *Scalable architecture with built-in failover and recovery support to ensure high availability with uninterrupted data collection and distribution*
- *Comprehensive license management including global floating, departmental floating and user-restricted configurations*
- *Import feature for previously collected data ensures even new installations can provide a comprehensive history of intraday data where that data is not already available from the source data feeds*
- *Automatic filtering of bad ticks and automated correction of historic deals (for feeds that provide such automated corrections)*
- *Direct connection to data from:*
  - **Bloomberg:** Bloomberg Professional Workstation
  - **Thomson Reuters:** Triarch/RWS (SSL4)
  - **Thomson Reuters:** Tibco TIC (SASS2)
  - **Thomson Reuters:** RMDS/RTIC (SASS3)
  - **Trayport GlobalVision:** trading gateway
  - **Trayport GlobalVision:** direct broker connections
  - **Tradesignal OpenConnect:** middleware for open interoperability

## 1.1 Architecture

TMDS runs in the background as a service on Windows-based systems. This has the advantage that no user needs to be logged in for the system to be collecting and distributing ticks; as long as the computer is turned on, the TMDS will do its job. In addition to the TMDS service, feed-handler services run for each configured data feed.

Administration of a TMDS is done in the TMDS Management Console. This central tool allows an administrator to control the system and gain reports on all aspects of the TMDS.



## 1.2 Scope of the manual

The scope of this manual is to deliver information regarding software setup, the Management Console (user interface and basic settings), administration processes, and troubleshooting. For many elements (fields, buttons, etc.) in the user interface, context-sensitive help is available as a tooltip when moving the mouse over the elements. Please refer to this help if this manual does not offer an explanation.

## 1.3 Target audiences

The target audiences are system administrators, TMDS administrators and data administrators.

## 1.4 Expressions and formats used in this manual

### 1.4.1 Note and Tip

Note	Important information for a functioning TMDS
Tip	Information that makes working with TMDS smoother

### 1.4.2 Formatting

Press <b>CTRL+C</b> .	buttons, fields, commands
<i>MDS.INI</i>	file names
<b>Tradesignal/MDS</b>	folder paths
<i>[General]</i>	sections in <i>MDS.INI</i>
Feed = TMDS_TRPT	entries in files
<b>Thomson Reuters</b>	emphasis

## 1.5 Glossary

Term	Explanation
command line	GUI element of the TMDS Management Console
command window	Windows console
feed	Datafeed coming from Bloomberg, Reuters, Trayport, etc.
symbol	An instrument that has been added successfully to the TMDS
TMDS	Tradesignal Market Data System
TMDS Management Console	User interface of the TMDS
TMDS service	The actual service that has to be running for the system to be online
TS1	Inter-day time series data from Reuters
query	Search and retrieve information about a symbol from the datafeed



## 2 Deployment: Common Installation Configurations

*Note: In the scenarios below, individual systems can concurrently collect data from any or all of the supported datafeeds.*

### 2.1 Scenario 1: Local installation alongside Tradesignal (single user)

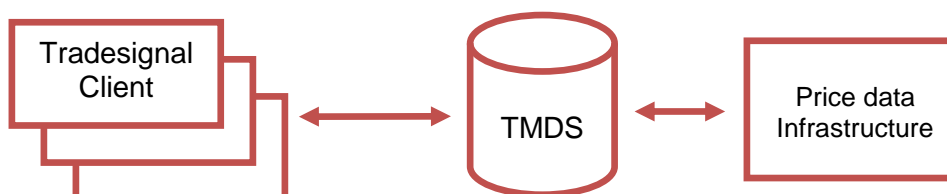
A TMDS is installed on each workstation and operates alongside the Tradesignal desktop products installed on the same computer. The advantages of this configuration include the fastest possible data retrieval rate and simplified administration for a single user at the cost of local system performance. If multiple users are configured in this way, sink distributor load would increase.

*Note: When using Bloomberg datafeeds, Scenario 1 must be deployed.*



### 2.2 Scenario 2: TMDS on dedicated system

If a group of users need access to historical and real-time data, the installation of a dedicated system running the TMDS is the preferred configuration. Such a configuration ensures reduced network traffic and significantly increased workstation performance over the single user configuration. Scenario 2 is recommended for 2 or more users.



## 2.3 Scenario 3: Two systems

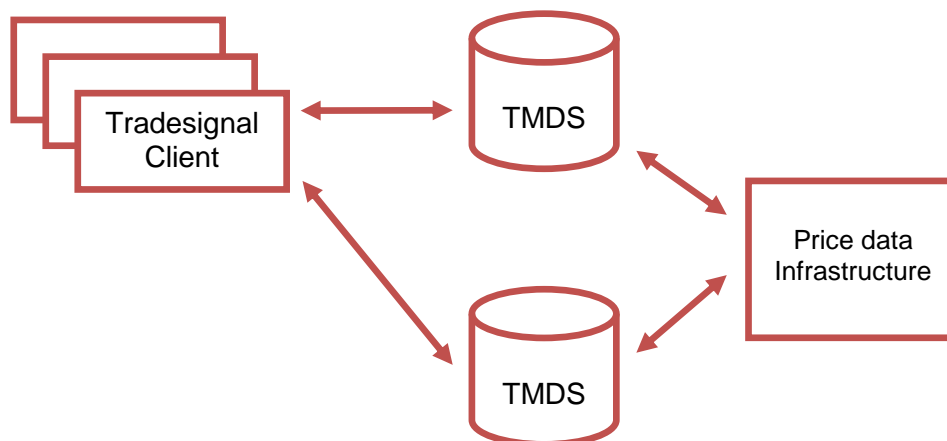
This option comes in three varieties. In each case both TMDSs must have access to the same data feeds and use the same prefixes.

*Where Thomson Reuters data feeds are being used, in order to guard against data holes when a server is restarted, each server should be entered in the Reuters datafeed settings as the **recovery host** for the other server (see “Setting up Thomson Reuters” on page 21).*

### 2.3.1 Scenario 3a: One system as failover

This allows desktop clients to swap automatically to the failover system in the event of a planned or unplanned outage in the primary system, significantly minimizing the detrimental effect of system outage.

This scenario is recommended for 2 or more users. For the settings, see “Synchronizing Systems” on page 46.



### 2.3.2 Scenario 3b: Two identical systems working in parallel

Both systems work as failover for each other; each is set up as a destination system using the other system as source. If one system fails, all users need to work on the second system, which might not be equipped for that many users.

This scenario can be seen as a substitute for scenario 4b, where provision of sufficient server hardware is limited.

## 2.4 Scenario 4: Multiple systems

In case of larger user groups and/or a larger number of instruments, multiple systems are necessary. The easiest configuration is to multiply the primary/failover system pairs.

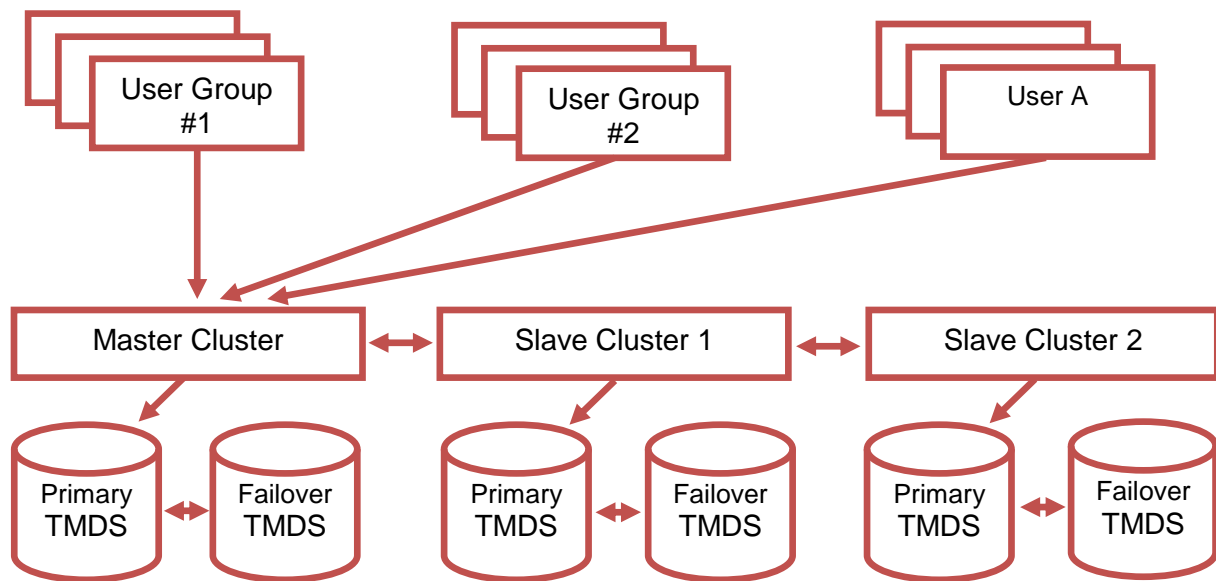
Depending on licenses, user groups, and users, users may be able to connect to any primary system or only to some of them. In principle, each TMDS can have all types of datafeeds, so it is possible that all primary/failover pairs are configured identically. This makes synchronization (automated or manual) very easy.



### 2.4.1 Scenario 4a: Distributing feeds and symbols across individual servers

*Note: This configuration is only available for use with Tradesignal Enterprise Edition 5.3 and above.*

The typical case for needing multiple servers is where the data collection requirements of a given feed are too great for the capabilities of a single server. In such a case, individual feeds can be collected on a single server (or server pair); or groups of Symbols from a single feed can be collected across multiple physical servers (see “Filtering Instruments (Whitelisting)” on page 44 on configuring whitelisting Symbols for a specific server).



*Note: All systems must be running on the same port and be using the same version of TMDS.*

Limitations:

- Currently 10000 RICs for each TMDS.
- After reaching the maximum number of RICs per TMDS, disable any further creation of RICs. This can be set for each TMDS via **Manage Settings** → **Service** → **Maximal allowed number of symbols**.

#### 2.4.1.1 Licensing of Master/Slave TMDS systems

The master system needs a normal license, while the slave systems need a license that will only register them as slave. If no slave license is present on a slave system, a desktop application will be unable to access data on that system.



### 2.4.1.2 Necessary settings in Tradesignal Enterprise Edition

To connect to a Master/Slave network from Tradesignal Enterprise Edition, enter the master system name in the **Host** field of the Tradesignal Enterprise Edition connection settings and enter the slave system names separated by a semi-colon (;) in the **Slave Hosts** field. Failover hosts should be entered in an identical manner (keeping the same order of host names).

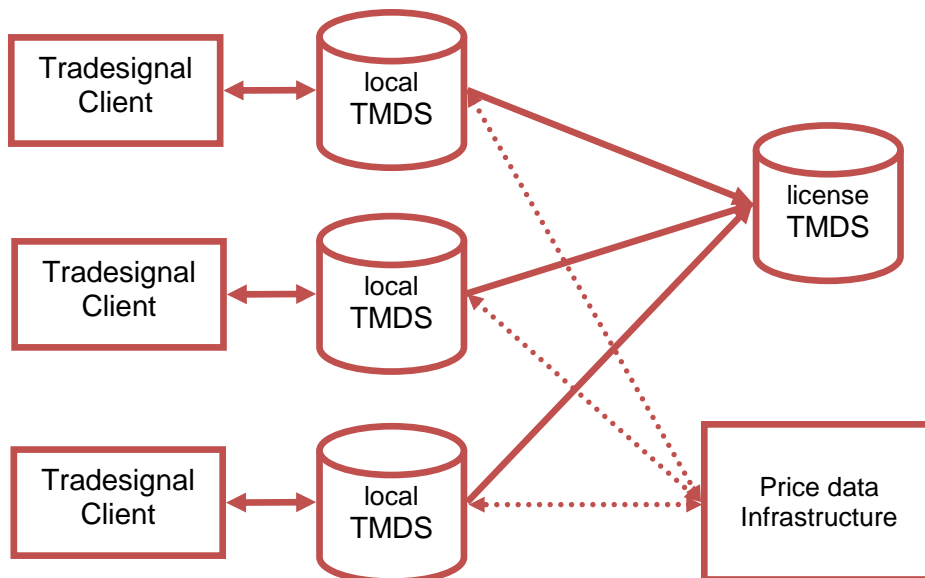
### 2.4.2 Scenario 4b: Scaling number of clients beyond scenario 3

A not so common variant of a multi system deployment is scaling scenario 3 to support larger numbers of users (or to improve access speed for different geographically located user groups).

Such deployments simply require identical clusters of servers as described in scenario 3, and restricting which users access each pair directly in the Tradesignal configuration settings.

## 2.5 Scenario 5: Local installations using a centralized licensing TMDS

As an administrator it may be difficult to handle the licensing of multiple local TMDS installations throughout the company when licenses start expiring. This can be simplified by using a centralized licensing TMDS. An example for this case would be to have several local Bloomberg TMDS installations. In this case you can specify a centralized TMDS as the license server. This license server will need additional remote licenses to grant feed access to all local TMDS installations. All local TMDS installations will be configured to use the license TMDS instead of using their locally installed licenses (see section “Managing Licenses and Users” on page 26 for further information).



## 3 Installation

### 3.1 Preparations for installing TMDS

- *Have your installation CD-ROM ready.*
- *Define the TMDS infrastructure and licensing model, see “Deployment: Common Installation Configurations” on page 8.*
- *The workstation must meet the hardware and software requirements for installation (see below).*
- *You must be logged in on the computer with an account that has administrative rights for the installation.*
- *You should have all necessary information ready for the data feed of your choice, see “Managing datafeeds” on page 20.*

We recommend installing TMDS on up-to-date hardware. A stand-alone system should have a high degree of fault tolerance components (RAID [1, 5 or 10], dual power supplies, Uninterruptable Power Supply, etc.); alternatively, use failover systems.

### 3.2 System requirements

*Note: The TMDS performance depends on many variables: the number of instruments, the frequency of feed updates, the number of connected clients, the size of the data cache files, and the number and type of connected data feeds.*

#### 3.2.1 Supported operating systems

- *Windows XP (SP3 or above)*
- *Windows 2003 Server (32 or 64 bit)*
- *Windows 2003 Server R2 (32 or 64 bit)*
- *Windows 2008 Server (32 or 64 bit)*
- *Windows 2008 Server R2*
- *Windows Vista (32 or 64 bit)*
- *Windows 7 (32 or 64 bit)*

#### 3.2.2 Minimum system requirements

- *Main processor: Pentium 4*
- *Main memory: 2 GB*
- *Free disk space: 200 GB*



### 3.2.3 Recommended system requirements for the use of TMDS

- Main memory: 4 GB
- Multi-core (4+)
- Free disk space: 500 GB – 1TB
- Striped and mirrored disk array
- Uninterruptable Power Supply (UPS)

## 3.3 Software Requirements

### 3.3.1 Global Vision API (Trayport Feed Handler)

This section is only relevant when you plan to feed Trayport prices into TMDS.

For this purpose, TMDS provides the dedicated Trayport Feed Handler, which relies on the Global Vision API (GV8API) to communicate with Trayport upstream servers.

Note that the Trayport Feed Handler can be installed without the GV8API present on that machine, however access to Trayport data will be unavailable until GV8API is installed (in that case TMDS will have to be restarted after the GV8API installation).

Please contact your Global Vision subscription support to obtain the latest GV8API installation package.

## 3.4 Setup

1. Start the installation by running the installation file. Follow the instructions in the installer window.
2. After successful installation, the Tradesignal Market Data System (as a service) and the TMDS Management Console will both start up. Any problems with the configuration of the system will be shown immediately in the Logging console of the TMDS Management Console, which serves as the user interface.
3. The datafeed wizard starts. You may proceed with entering datafeeds or skip that step for now; adding and editing datafeeds is always possible later.
4. If no license file is found, you are asked whether you want to add any. You may proceed with entering licenses or skip that step for now; adding and editing licenses is always possible later.

*Note: In case of questions and problems, please call the support hotline.*

All further configuration steps take place in the Tradesignal TMDS Management Console, which is introduced on page 16.

- For adding feeds, see “Managing datafeeds” on page 20.
- For adding licenses and user (groups), see “Managing Licenses and Users” on page 26.
- For adding symbols, see “Managing Symbols” on page 32.



### 3.5 Updating

*Note: To update (uninstall/install), you must have administrative rights on your computer.*

When installing a newer version of the TMDS software, uninstall the old version first.

1. Open **Start → Control Panel → Add/Remove Programs**.
2. Choose **Tradesignal Market Data System** and remove it. The configuration settings file *MDS.INI* and data cache file *mdsdb.dat* will be kept.
3. Install the new TMDS version as described above.

### 3.6 Uninstalling

*Note: To uninstall, you must have administrative rights on your computer.*

1. To completely uninstall the TMDS software, uninstall it and then remove the leftover files.
2. Open **Start → Control Panel → Add/Remove Programs**.
3. Choose **Tradesignal Market Data System** and remove it. The configuration settings file *MDS.INI* and data cache file *mdsdb.dat* will be kept.
4. For a complete removal, open Windows Explorer and delete the installation folder on your hard drive, usually located at **Program Files/Tradesignal/MDS**.



### 3.7 Copying an existing data cache file to a new TMDS

If you have a currently running old TMDS (before version 5) and want to copy the data cache file to a newly installed TMDS that uses the same Reuters infrastructure, do the following:

1. Stop the old TMDS if it is currently running.
2. Make a copy of the following files from the old TMDS:
  - The data cache file *tmds.dat* (if the data cache file is called *mtdsdb.fdb*, rename the copy to *tmds.dat*).
  - The data cache backup file *tmds.dat.bak* (if the data cache backup file is called *mtdsdb.fdb.bak*, rename the copy to *tmds.dat.bak*).
  - The settings file *MDS.INI*.

**It is strongly recommended that these backup copies are securely stored, in case of installation problems**

3. Stop the new TMDS if it is currently running.
4. Copy the backed up files from step 2 into the installation folder of the new TMDS, overwriting the current files if they exist.
5. If the data cache file was called *mtdsdb.fdb* before renaming, open the *MDS.INI* file in a text editor and replace *mtdsdb.fdb* with *tmds.dat* in the DataPath setting in the [General] section.
6. Restart the new TMDS.

*Note:* If you need to copy a data cache file to a new TMDS that uses a different Thomson Reuters infrastructure, settings from the old *MDS.INI* file will need to be copied by hand into the new *MDS.INI* file. Using certain configuration values from the old TMDS such as **Maximum Number of RICs** and **History Lengths** will ensure that the data is correctly described.

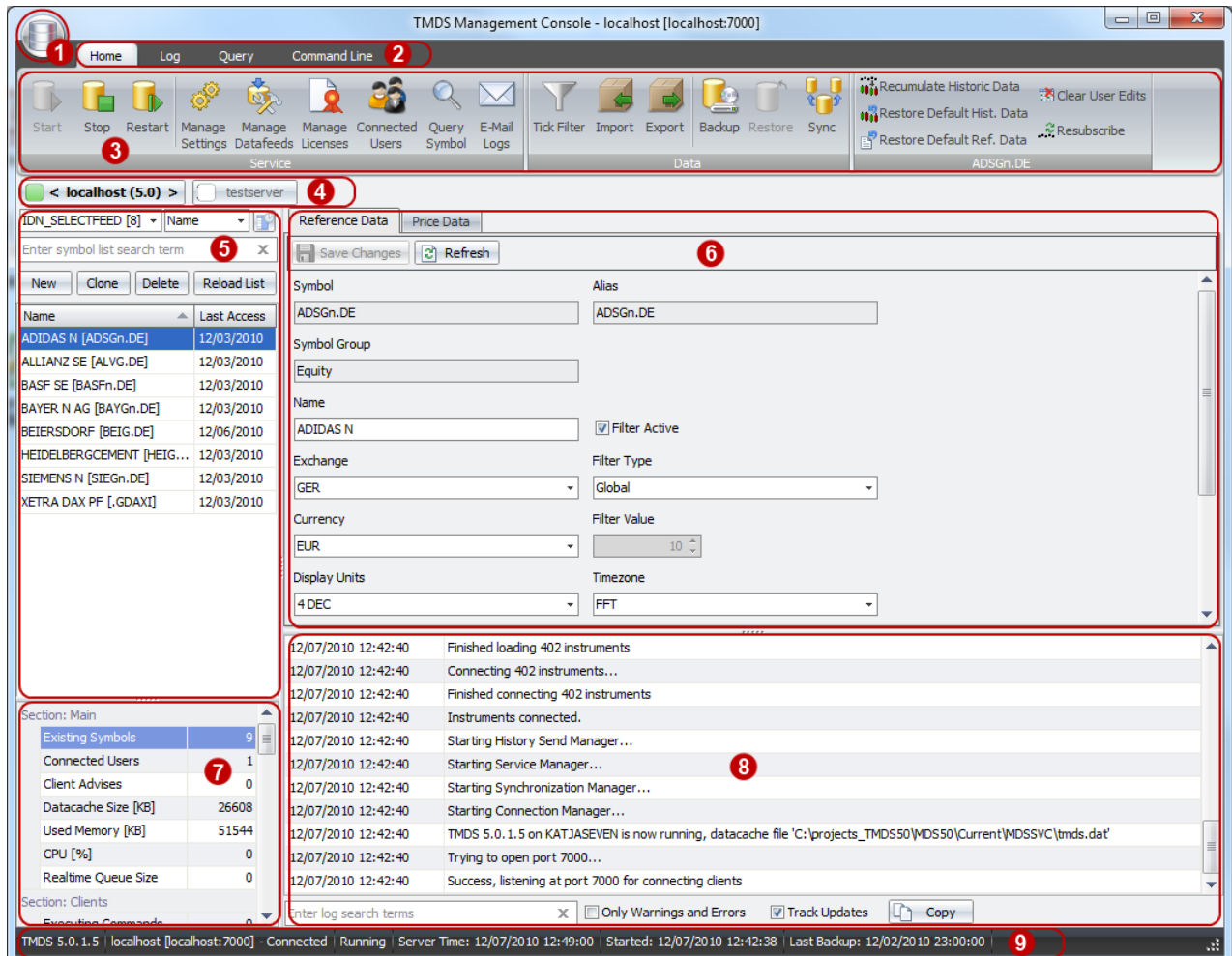
*Note:* If the data cache file is from a TMDS version before 5, proceed as described above but be aware that the datafeed definition has changed. Start (or restart) the TMDS Management Console after copying the files. It will attempt to convert the datafeed settings to the new format. TMDS will not start until a datafeed is configured correctly.



## 4 TMDs Management Console

The TMDs Management Console is the central user interface (UI) for the Market Data System. Here, the connections, users, symbols, settings, etc. can be managed.

### 4.1 User interface overview

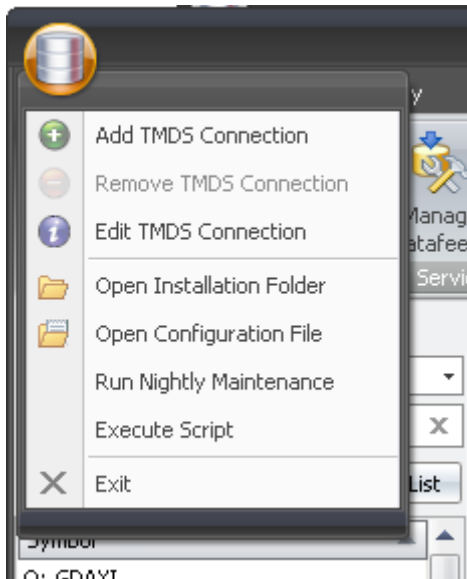


- 1: Application button, see “Application button” on page 17.
- 2: Tab bar with Home as default; for Log, Query and Command Line, see “Logs, Alerts & Troubleshooting” on page 50.
- 3: Ribbon control with standard functions.
- 4: Connections to TMDs services, see “TMDs Connections” on page 31.
- 5: Symbol list with search filter, see “9 Managing Symbols” on page 32.
- 6: Per symbol: reference data and price data, see “” on page 32.
- 7: Information on various TMDs service and computer aspects.
- 8: Logging console, can be searched/filtered for terms, see “Logging console” on page 50.
- 9: Status bar, see “Status bar” on page 18.



## 4.2 Application button

The Application button to the top left opens a menu with administrative options.



For connections functions, see “TMDs Connections” on page 31.

The installation folder is usually at Program Files/Tradesignal/TMDs.

The configuration file refers to the *MDS.INI* file in the installation folder.

For nightly maintenance, see “Nightly Maintenance” on page 55.

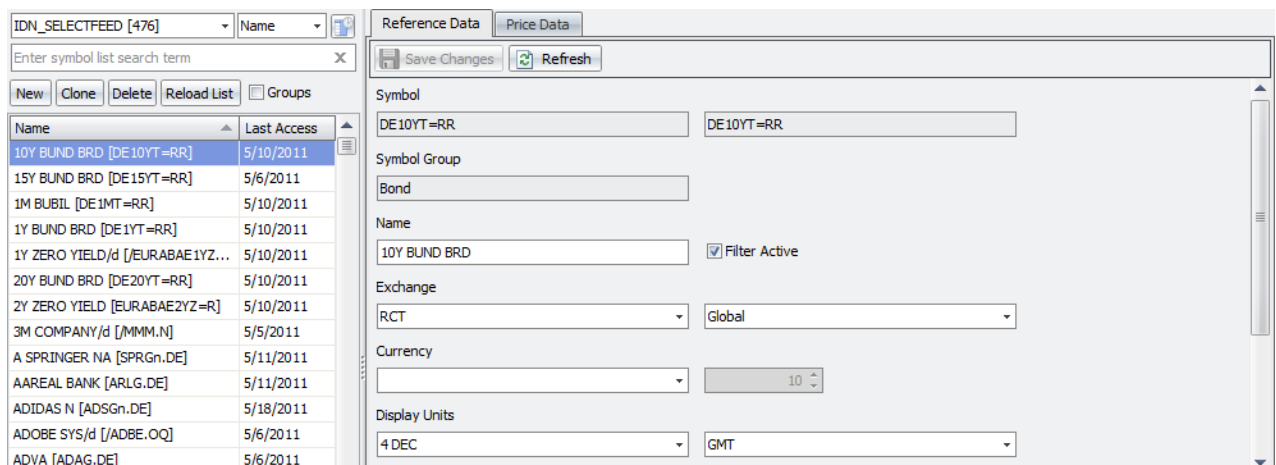
For scripting options, see “Scripting TMDs” on page 50.

## 4.3 Symbol area

The symbol area offers all datafeeds and their symbols. For each selected symbol, the Reference Data and Price Data tabs are available.

The search field offers automatic filtering while typing the search term.

For more on symbol handling, see “Managing Symbols” on page 32.



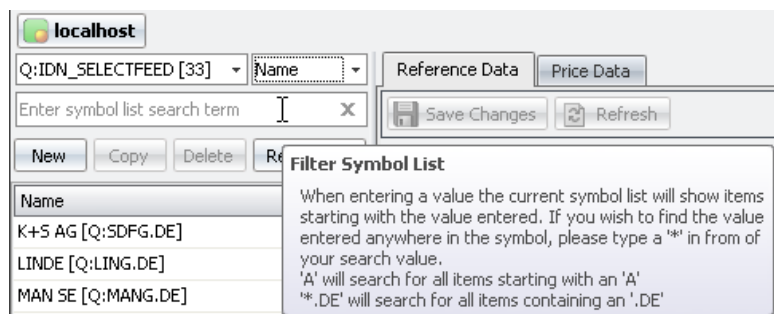
## 4.4 Status bar

TMD5 5.0.0.60 | localhost [localhost:7000] - Connected | Running | Server Time: 06/18/2010 21:26:00 | Started: 06/16/2010 21:33:13 | Last Backup: 06/17/2010 23:31:00

A	B	C	D	E	F
A	Version of TMD5 software				
B	TMD5 name, port, and state of the connection between the Management Console and TMD5. Double-click to reconnect the Management Console.				
C	State of the service (Starting, Running, Stopping, Stopped, Uninstalled)				
D	TMD5 time				
E	Service start time				
F	Time of the last backup				

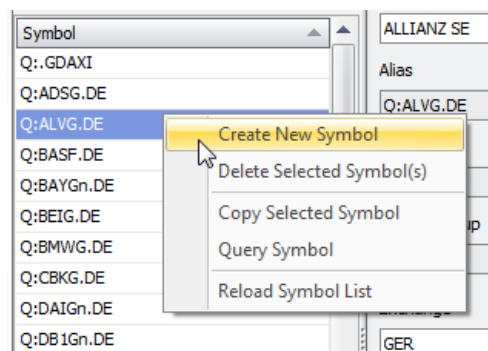
## 4.5 Mouse-over information

Move the mouse over UI elements to find tooltips with explanations for the element/field/feature:



## 4.6 Context menus

In some areas, context menus are available. Open them by right-clicking the area/element.



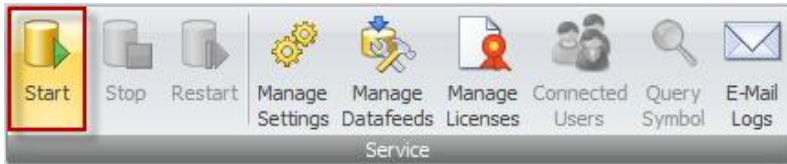
## 4.7 Field labels

- ✘ Field cannot be left empty or the entered value is not valid. Move the mouse over the label to read the error message in a tooltip.
- ⚠ Field has a warning. Move the mouse over the label to read the warning in a tooltip.

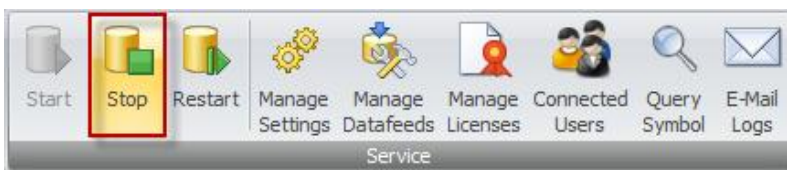


## 5 Starting / stopping the system

Note: To start/stop the system, you must have administrative rights on your computer.



Requires administrative rights on the system!  
Click **Start** to start the TMDS service and all feed-handler services.



Requires administrative rights on the system!  
Click **Stop** to stop the TMDS service and all feed-handler services.

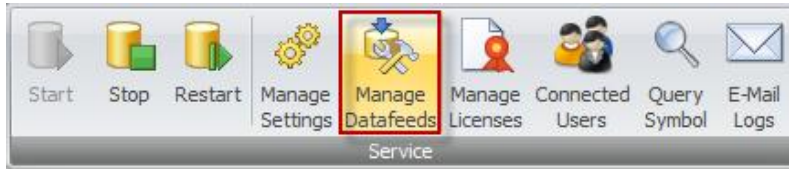


Click **Restart** to restart the TMDS service and all feed-handler services. This is usually done after changes in the settings and usually does not require administrative rights (may depend on operating system).

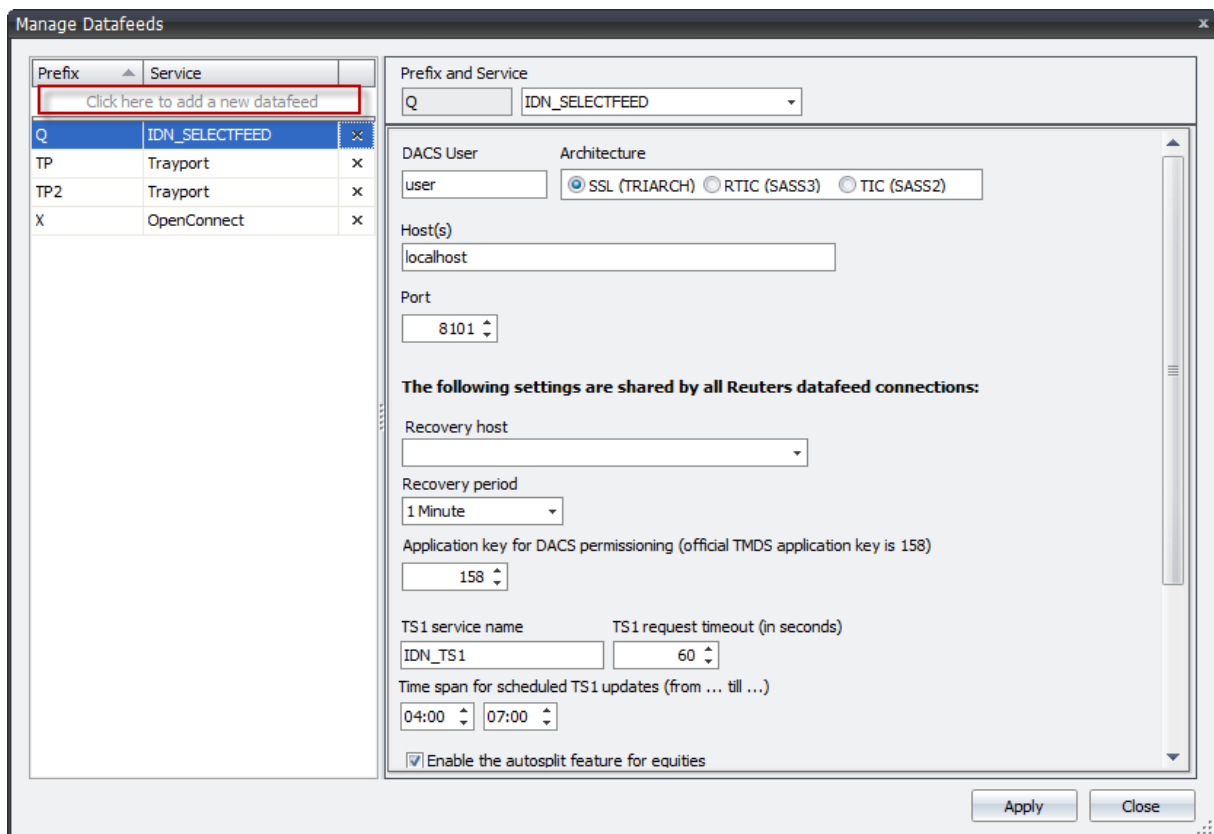
All process steps are logged and displayed in the Logging console.



## 6 Managing datafeeds



Click Manage Datafeeds to add, edit, and delete datafeeds.



All types of datafeeds are created by clicking into the field “**Click here to add a new datafeed.**” This will open the datafeed wizard, in which you can select the datafeed type and then make some basic settings.

Four main types of datafeeds are supported:

- **Bloomberg** where the Bloomberg Professional Terminal is installed
- **Thomson Reuters** in the versions Triarch/RWS), Tibco TIC), RMDS/RTIC
- **Trayport** as trading gateway and for direct-broker connections
- **OpenConnect** as middleware for open interoperability

After the datafeed creation, check your datafeed settings and edit them, if necessary.



## 6.1 Setting up Thomson Reuters

For troubleshooting, see “Advanced configuration and troubleshooting of TMDS for Thomson Reuters Data” on page 65.

In the Wizard, enter:

- Reuters Architecture
- Prefix and service
- DACS User which the TMDS will use for identification at the Reuters infrastructure
- SSL host and port (default port is “8101”)
- DACS permissioning for connected users (see below)

The datafeed prefix is used to distinguish internally between the different data providers. It will be part of each symbol name created for that datafeed.

For the service, the following entries are available as default:

- For Triarch (SSL4) installations, select **IDN\_SELECTFEED**.
- For RWS (SSL4) installations, select **IDN\_RWS**.
- For TIC (SASS2) installations, select **IDN\_RDF**.
- For RTIC (SASS3) installations, select **IDN\_RDF**.

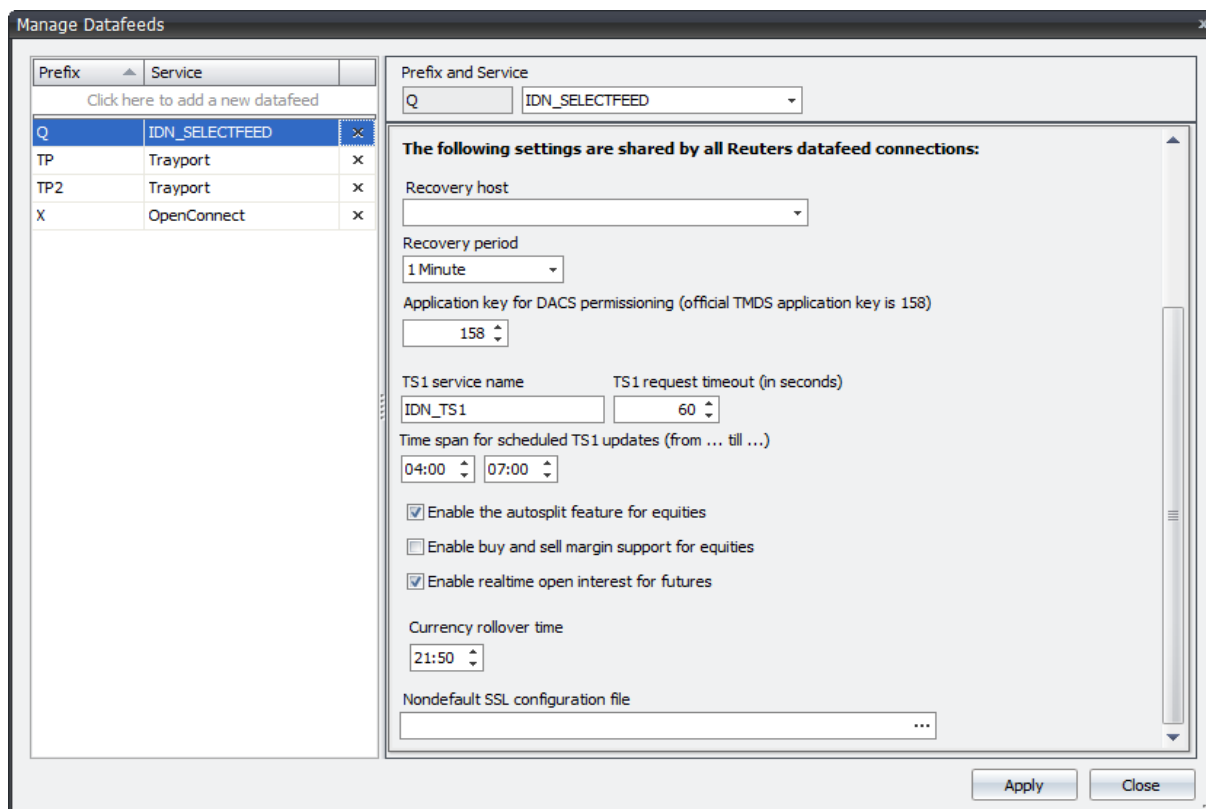
Other feed names can be entered manually by editing the field.

As of version 5.2, TMDS is able to perform entitlement checks on behalf of connected users utilizing the Thomson Reuters DACS infrastructure (Data Access Control System). In



accordance to exchange rules, DACS permissioning has to be performed for all feeds that support DACS entitlement control in order to control access rights to data for every user connected to TMDS. Enter the DACS Daemon host that should be connected in order to perform entitlement checks for symbols served by the data feed. When more than one DACS Daemon host is available for this feed, please enter all hosts (separated by a space character). DACS permissioning must only then be disabled if the data feed does not provide access controllable symbols (e.g. custom publishers).

In the datafeed configuration, additional information can be added.



To use a recovery host, see “Scenario 3: Two systems” on page 9.

## 6.2 Setting up Trayport GlobalVision

Please cf. section 3.3.1 for specific software requirements in order to access Trayport prices.

In the Wizard, enter:

- *prefix*
- **host name and port** (default port is “27361”) of the feed server
- **username and password** of the Trayport user account

In the UI, additional information can be added, like the SQL database settings.



Click the Log tab to read the logs.

### 6.2.1 Backfilling data history from Trayport SQL server

*Note: For the SQL Database server settings, please seek the help of your SQL Database Administrator. Leave these fields empty if in doubt. These settings are only necessary when you need to access longer histories. Entering invalid information may result in no histories at all.*

<b>Trayport SQL Database Server:</b>	The address of the (Trayport) Microsoft SQL Server data file
<b>Trayport SQL Database Name:</b>	The name of the database on the given server (typically GVLOCAL80)
<b>Database Username:</b>	The Microsoft SQL Server data file username to access the Trayport tables
<b>Database Password:</b>	The Microsoft SQL Server data file password

*Note: The SQL Data file username and password will normally need to be specially created in the SQL Server Database by an administrator and configured to access the tables listed below in point 3.*

*Note: The TMDS system must have the Microsoft SQL Server ODBC driver installed on it. This is normally installed with the SQL Server Client Components setup.*

TMDS will obtain historic deals via the Trayport GlobalVision API, for which there is likely to be a rather small quantity of data (the amount of data is often capped by the API at two weeks of data). TMDS provides a way to obtain historic deals directly from an ODBC data file (such as the server used by GlobalVision to store the complete deal history). If the TMDS is able to connect to the SQL Server directly, follow these steps to obtain historic deals.

1. Ensure the server is reachable from TMDS by typing ping <servername> into the command prompt.
2. Ensure that the Microsoft SQL Server ODBC driver is installed on the TMDS system (this normally will be installed when the SQL Server client components are installed on the machine using the SQL Server setup)



**3. Obtain a username and password for the data file that has read access to the following data files and tables:**

- a. GVGLOBAL80 - SequenceItems
- b. GVGLOBAL80 - SequenceTypes
- c. GVGLOBAL80 - Sequences
- d. GVGLOBAL80 - InstTreeGroups
- e. GVGLOBAL80 - InstTreeInstruments
- f. GVGLOBAL80 - Version
- g. GVLOCAL80 - DealsDone
- h. GVLOCAL80 - DealTreeNodes
- i. GVLOCAL80 - Orders
- j. GVLOCAL80 - Version

**4. Edit the Trayport feed via **Manage Datafeeds**.**

**5. Check the option **Get deal history directly from Trayport SQL Database Server**.** This will automatically set ODBC as backfill method. The settings for the server, database, database user and password are then used to create the ODBC connection string.

*Note: If you are unable to connect and the log file reports that the ODBC driver cannot be found, you will need to install the SQL Server client components on this machine (check with the administrator before doing this).*

**6. Restart the TMDS and query a symbol.**

### 6.2.2 Rolling

The TMDS provides a number of special symbols for each Trayport sequence in a symbol list marked as *Rolling*. These symbols represent instruments that will automatically roll from one sequence item to the next based on the defined *next contract* (NEAR, +1, +2, etc.). The rollover is strictly based on the defined trading end date of the current contract. The TMDS will also back-adjust the historic process after the first tick in the new contract to remove price gaps introduced by the rollover.

Absolute adjustment means that historic prices will be reduced or increased by the difference between the close of the previous contract and the first tick of the new contract after the rollover. Relative adjustment means that historic process will be multiplied by the ratio between the old and new prices.



## 6.3 Setting up Bloomberg

The TMDS can be installed on any computer that is running a Bloomberg Professional Terminal. The TMDS will use the BBCOM Version 3.0 Desktop API to obtain data.

In the Wizard, enter:

- **prefix** of the *Bloomberg* service

TMDS will attempt to connect to your service after restart.

*Note:* A user must log-in to the Bloomberg Professional Terminal at least once per week; otherwise, the TMDS will report *no permission* errors when establishing a connection.

*Note:* If using Bloomberg with TMDS, the Tradesignal desktop application has to be installed on the same computer, see “Scenario 1: Local installation alongside Tradesignal (single user)” on page 8.

## 6.4 Setting up OpenConnect

The Tradesignal OpenConnect Software Development Kit (SDK) is a package that allows a licensed developer to create and distribute feed-handling modules that interface TMDS to third-party datafeeds and data files. Once an OpenConnect derived service is packaged and deployed to a suitable server, the TMDS will need to be informed of the details of this service either during installation or manually by modifying settings files.

In the Wizard, enter:

- **prefix** (optional) and **service name** of the *OpenConnect* service
- **user name** (optional)
- **host name** and **port** (default port is “27367”) of the feed server

TMDS will attempt to connect to your service after restart.

*Note:* If the *OpenConnect* service has been set up to use non-standard ports for *Push* or *Backfill*, the corresponding ports have to be set.



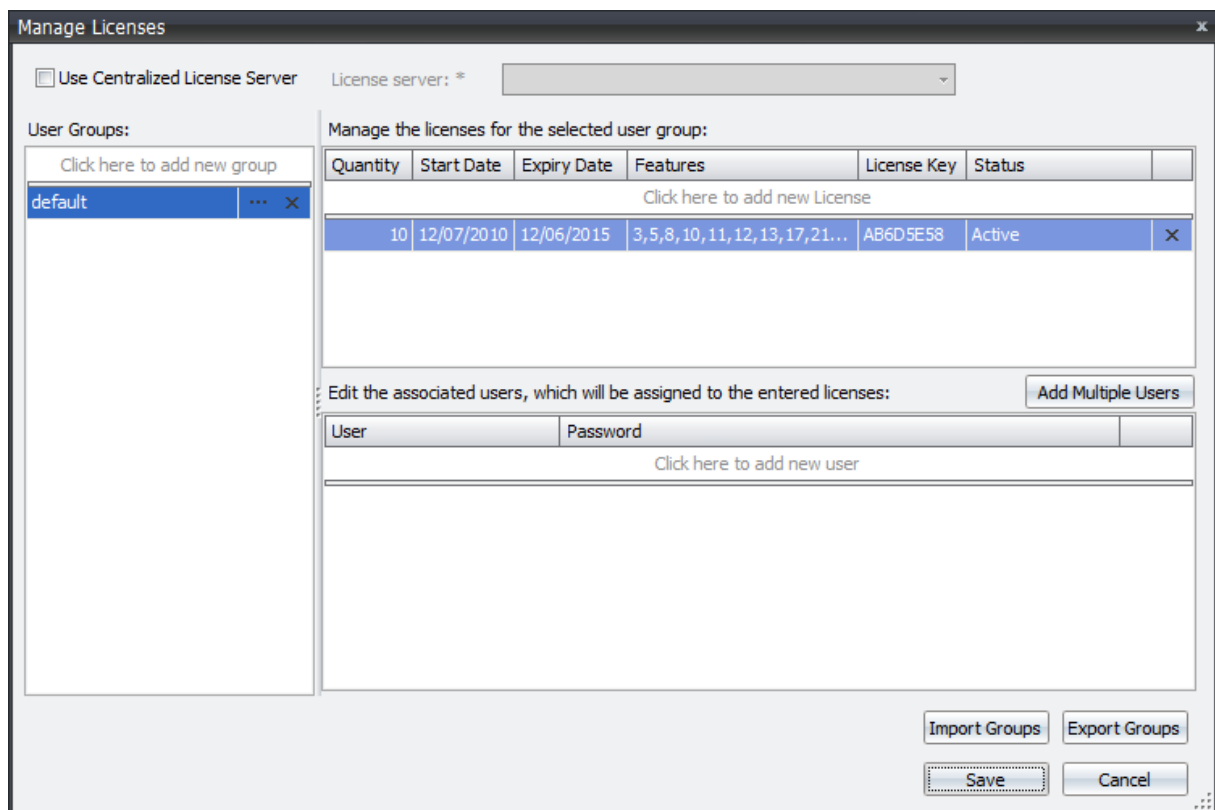
## 7 Managing Licenses and Users

Before client applications can use the TMDS to obtain market data, the system must first have client-access licenses installed.



**Click Manage Licenses to open license management.**

### 7.1 The license and user group manager window



## 7.2 Setting up a centralized license server

Setting up central licensing is a two-step process:

1. Install a TMDS on a central server and obtain the central license code from Tradesignal Support.
2. Direct all other TMDS installations to this central license server in the following way:
  - a. Ensure the license server is available from the Management Console.
  - b. Select the TMDS server that needs to connect to the license server.
  - c. Display the *Manage Licenses Dialog*.
  - d. Check the *Use Central License Server* box and pick the License Server from the list.
  - e. Click **Save** and Restart the TMDS.

## 7.3 License properties and usage

Each TMDS license has the following properties:

Quantity	The number of licenses of this type that are available.
Start Date	The date on which the license becomes valid.
Expiry Date	The date after which the license ceases to be valid.
Features	The features this license allows: see below.
License Key	The License key itself.
Status	Indicates whether this license is valid, invalid, superseded, duplicated, expired or pending.

Important points about licenses:

- Licenses are connected to the installation by the Machine ID of the computer.
- Licenses are cumulative, so if a system has two valid licenses, each allowing 5 connections to the system, this system will allow a total of 10 connections.
- Multiple identical licenses (with the same license code) are not allowed, and will result in all but one of them having an invalid status.
- Each license contains a numeric list of features. These features define what the TMDS (and in some cases a Tradesignal desktop application) are allowed to do and what features they themselves can offer. In general, the features provided will be detailed in your license agreement.
- When requesting a license for a given TMDS, it is important to state the datafeed that the TMDS will be connected to, because each datafeed vendor requires a different license code. For example, if you change a TMDS connection from one datafeed (e.g., Thomson Reuters) to another from a different vendor (e.g., Trayport or OpenConnect), you will require a new license code.



- *If the use of a centralized license server is intended state this when requesting for license codes because client TMDS installations will use special features that need to be present on the license server.*
- *For the licensing of multiple Tradesignal Market Data Systems in a Master/Slave configuration, see “Scenario 4: Multiple systems” on page 9.*

Normally, it is not necessary to enter any users. If different users will be able to use different licenses, user groups can be created, see “Using user groups” on page 29.

## 7.4 Floating licenses

Floating licenses, users, and user groups work as follows:

- *If ten floating licenses are available, up to ten users can connect to the TMDS. The users don't have to be entered by name.*
- *If, in a user group, a floating license and users are entered, the number of necessary licenses is reserved for these users. If unreserved licenses are left, any user may use them to connect to the TMDS.*
- *If all licenses are reserved for user groups, only those can be used. Users that are not members of a group will not be able to connect to the TMDS.*

## 7.5 Adding a new license

1. Select a user group, if necessary.
2. Click into the field "Click here to add new license".  
A dialog opens in which a *Machine ID* is displayed, and a **License Code** can be entered. The *Machine ID* is unique to a specific system, and ensures that a license code will only be valid for that specific machine.
3. Call the licensing hotline number contained in your support agreement and provide the *Machine ID* displayed in the dialog.
4. The agent on the licensing hotline will issue a 26 character code which must be entered into the **License Code** section in the dialog above. The license code contains all license details.
5. Click **OK** to close the dialog. The new license will now be visible in the license list.

*Tip: To reserve licenses for specific users, create a group for them.*

## 7.6 Removing a license

To remove an invalid or expired license, click **X** and confirm the deletion.

Once a license has been removed, it can be added again as long as the original license code is still available. Simply re-enter it.



## 7.7 Using user groups

If different users will be able to use different licenses, user groups can be created. Adding users to licenses guarantees that these licenses are available to these users.

The workflow is as follows:

1. Enter the user group by clicking into the field “**Click here to add new group**”.
2. Add licenses and users to the selected group.
3. Click **Save** to save any changes upon leaving.

The screenshot shows the 'Manage User Groups' window with the following components:

- User Groups:** A list containing 'default' (selected) and 'Superuser'.
- Manage the licenses for the selected user group:** A table with columns: Quantity, Start Date, Expiry Date, Features, License Key, Status, and a delete icon.
 

Quantity	Start Date	Expiry Date	Features	License Key	Status	
1	07/06/2010	07/05/2015	34	717C9684	Active	×
1	06/01/2010	07/06/2010	34	D200955E	Active	×
- Edit the associated users, which will be assigned to the entered licenses:** A table with columns: User, Password, and a delete icon.
 

User	Password	
user1	No	×
user2	Yes	×

Buttons at the bottom: Import Groups, Export Groups, Save, Cancel.



## 7.8 Adding / editing / deleting users

*Note: Users only need to be entered explicitly if you want to restrict licenses to certain user groups.*

Users are handled in the area on the lower right.

1. Select the user group.
2. To add a single user to the selected group, click into the field "**Click here to add new user**".
3. As a user name, enter the DACS user name (ID).
4. In addition, a password can be entered for each user.

Alternatively, multiple users can be added by clicking **Add multiple users**. Entering a comma-separated list of names will add all these users at once. To give them passwords, you will have to edit each one.

To edit a user, double-click the user name.

To delete a user, click **X** and confirm the deletion.

## 7.9 Connecting to TMDS only to access Top-Up data

*This relates to a Tradesignal Enterprise feature.*

If a user only needs to access top up data (i.e., has no DACS ID to access Reuters data), this can be achieved by giving them a username prefixed by a minus character (-).

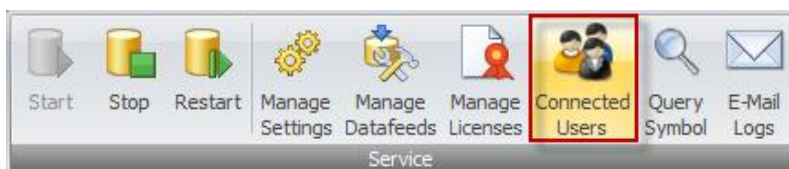
## 7.10 Exporting / importing user groups

User groups can be exported to a file by clicking **Export Groups**. All groups and their users are exported. The file has XML format and is stored in the folder **Console/export** in the installation folder.

Groups are imported by clicking **Import Groups** and selecting a previously exported groups xml file.

*Note: This will overwrite all currently existing groups and all entered licenses.*

## 7.11 Viewing connected users



Click **Connected Users** to get a list of all users currently connected to the TMDS.

To kick (disconnect) a user, click **X** in the last column of the user list.



## 8 TMDS Connections

TMDS connections can be of various types, depending on the available systems.

- *Client – local TMDS installation: for local installation alongside Tradesignal desktop application (necessary for Bloomberg)*
- *Client – remote TMDS installation: for installations with many clients per system*
- *TMDS to TMDS: for recovery and failsafe strategies, for example synchronization*

For more on deployment strategies and setups, see “Deployment: Common Installation Configurations” on page 8.



**Click Application button → Add TMDS Connection.**

**The Connection Properties dialog opens. Enter the system name/IP and port of another TMDS.**







The TMDS Management Console will try to connect to the TMDS. The result of the connection attempt will be displayed in the status line. The new connection will appear in the connection tabs.



The currently selected connection is marked with < >.

To delete a connection, select it and then remove it via **Application button → Remove TMDS Connection**. The connection is immediately removed.

Possible states:

-  TMDS and all feeds are running.
-  TMDS is not running.
-  TMDS is running but one or more feeds are not active.
-  Console is not connected to the TMDS.
-  A warning alert is active. The warning is displayed in the Log.
-  A severe error alert is active. The error is displayed in the Log.



## 9 Managing Symbols

Symbols are listed per datafeed in the symbol area. They are displayed by their names or symbols. You select this from a dropdown on top of the symbol area. A column showing the date of the last access to a symbol can be shown or hidden by pressing the button next to this dropdown.

Name	Last Access
10Y BUND BRD [DE10YT=RR]	5/10/2011
15Y BUND BRD [DE15YT=RR]	5/6/2011
1M BUBIL [DE1MT=RR]	5/10/2011
1Y BUND BRD [DE1YT=RR]	5/10/2011
1Y ZERO YIELD/d [EURABAE1YZ=RR]	5/10/2011
20Y BUND BRD [DE20YT=RR]	5/10/2011
2Y ZERO YIELD [EURABAE2YZ=R]	5/10/2011
3M COMPANY/d [MMM.N]	5/5/2011
A SPRINGER NA [SPRGn.DE]	5/11/2011
AAREAL BANK [ARLG.DE]	5/11/2011
ADIDAS N [ADSGn.DE]	5/18/2011
ADOBE SYS/d [ADBE.OQ]	5/6/2011
ADVA [ADAG.DE]	5/6/2011

Starting with TMDS 5.2, symbols can be logically grouped by their respective market sectors. Tick the **Groups** checkbox to enable grouping:

Name	Last Access
<b>ALL (476)</b>	>
<b>Bond (7)</b>	>
<b>Currency (16)</b>	>
<b>Equity (165)</b>	>
<b>Future / Commodity (274)</b>	>
<b>Index (12)</b>	>
<b>Others (2)</b>	>

You can drill down into groups by clicking one of the entries. For example, selecting **Bond (7) >** will filter the list of symbols to contain bond-type instruments only:



Name ▲	Last Access
^ <b>TOP</b>	
< <b>BACK</b>	
10Y BUND BRD [DE10YT=RR]	5/10/2011
15Y BUND BRD [DE15YT=RR]	5/6/2011
1M BUBIL [DE1MT=RR]	5/10/2011
1Y BUND BRD [DE1YT=RR]	5/10/2011
1Y ZERO YIELD/d [/EURABAE1YZ=R]	5/10/2011
20Y BUND BRD [DE20YT=RR]	5/10/2011
2Y ZERO YIELD [EURABAE2YZ=R]	5/10/2011

Depending on the connected upstream data feed, groups can be organized into multi-level hierarchies. The entry **^ Top** will always transport you back to the top-most level in the hierarchy, while **< Back** will get you to the previous level in the hierarchy.

Symbols can be added, cloned, deleted, and the list can be reloaded.

The context menu offers the additional option **Query Symbol**. This can be used to troubleshoot the datafeed.

Search terms can be entered in the search field. Add an asterisk for a wildcard search (for example, **"\*as"** finds Adidas and BASF)

For each selected symbol, the reference data and the price data can be displayed to the right.

- For exporting/importing symbols data, see *“Export / Import” on page 49.*
- For setting up datafeeds, see *“Managing datafeeds” on page 20.*
- For synchronizing symbols and historical data, see *“Synchronizing Systems” on page 46.*



## 9.1 Editing reference data

The reference data for each symbol can be edited.

For the Filter settings, see “Filtering Instruments (Whitelisting)” on page 44.

### 9.1.1 Editing server sessions

When the TMDS adds an instrument to its data cache file, the session times for that instrument are automatically requested from the Reuters network. If the default session times need to be globally modified for this instrument, edit the **Active Session** settings.

Click  to add further sessions. For each, a start and end time and the valid day(s) can be entered.

Click  to remove a session.

Click **Load from Symbol** to copy session information from another symbol.

Click **Server Defaults** to delete the active session and revert to the value in Server Session.

Start Time	End Time	Mo	Tu	We	Th	Fr	Sa	Su	
09:00	17:37	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="x"/>
10:00	14:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="x"/>

*Note: Modifying an instruments session does not modify what data is collected, unless a session based filtering has been set up. All data is collected and sent to the Tradesignal desktop application, where data outside the active session will be hidden. This can be switched off in the Tradesignal desktop application, in which case all data would be displayed.*



## 9.2 Editing price data

For last, bid, ask, and the available time periods, the price data of the selected symbol can be displayed and edited.

Select the **Field** and **Period** first. The data is then loaded.

Date	Open	High	Low	Close	Volume	Source
06/01/2010 14:30	80.39	80.74	80.39	80.7	30749	I
06/01/2010 14:00	79.97	80.56	79.89	80.36	151580	I
06/01/2010 13:30	79.989	80.07	79.9	80.01	79958	I
06/01/2010 13:00	79.74	80.05	79.7	80	39780	I
06/01/2010 12:30	80.03	80.08	79.62	79.78	50212	I
06/01/2010 12:00	79.75	80.06	79.68	80	62881	I
06/01/2010 11:30	79.54	79.92	79.54	79.73	74079	I
06/01/2010 11:00	79.62	79.65	79.5	79.5	59321	I
06/01/2010 10:30	79.65	79.85	79.6	79.6	63589	I
06/01/2010 10:00	79.7	79.97	79.65	79.65	96218	I
06/01/2010 09:30	79.72	79.81	79.55	79.68	90409	I
06/01/2010 09:00	79.92	79.96	79.62	79.77	107524	I
06/01/2010 08:30	79.601	80.28	79.6	79.9	154218	I
06/01/2010 08:00	79.78	80.07	79.52	79.64	133428	I
06/01/2010 07:30	80.84	80.99	79.55	79.78	303668	I
05/31/2010 16:00	81.38	81.38	81.38	81.38	387364	I
05/31/2010 15:30	81.48	81.51	81.21	81.33	68234	I

Sources can be

- **I**: internal, the source is realtime updates from the datafeed
- **U**: user, the source has been edited
- **T**: TS1 (Reuters only), the source is historical data from the datafeed

Beneath the Price Data tab, the number of candles is displayed (in the screenshot above, “212”).

### 9.2.1 Editing price data fields

Click into the fields to edit the data. Changed fields are highlighted by a light-blue background.

To undo edits, click or open the context menu and choose **Restore Selected Candle(s)**. This works only for unsaved changes.



### 9.2.2 Deleting candles

To delete a complete candle (time point), click **X**. Alternatively, open the context menu and choose **Delete Selected Candle(s)**. Deleted candles/time points are highlighted by an orange background and strike-through.

To undo a deletion, open the context menu and choose **Restore Selected Candle(s)**. This works only for unsaved deletions.

### 9.2.3 Adding candles

To add a new candle (time point), click in the field "**Click here to add new candle**" or open the context menu and choose **Create New Candle**.

### 9.2.4 Reloading / reverting price data

Click **Refresh** to reload all data. This reverts all changes made to this point.

### 9.2.5 Saving price data

Click **Save Changes** to save the data.

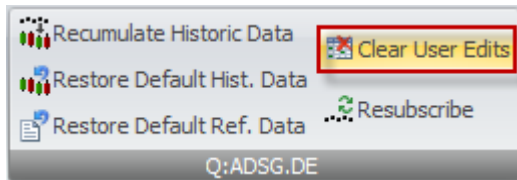
Changed data is checked for plausibility; if this check fails, the data cannot be saved. Once changes are saved, they cannot be reverted/undone. The **Source** field is changed to "U".



## 9.3 Clearing and restoring data

### 9.3.1 Clearing user edits

User edits are saved in a way that prevents them from being overwritten when restoring default historic data (as obtained from the datafeed). To prepare a restore, clear the user edits first.



**Click Clear User Edits to clear the edits for the current or all symbol.**

### 9.3.2 Restoring default historic data

This function will restore the historic data as received from the datafeed.



**Click Restore Default Historic Data to restore the historic data for the selected symbol from the datafeed, if available.**

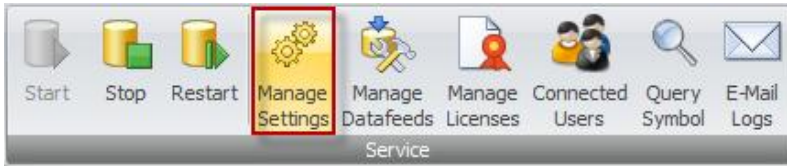
***Unless user edits are explicitly cleared (see “Clearing user edits” above), they will not be overwritten.***

### 9.3.3 Restore default reference data

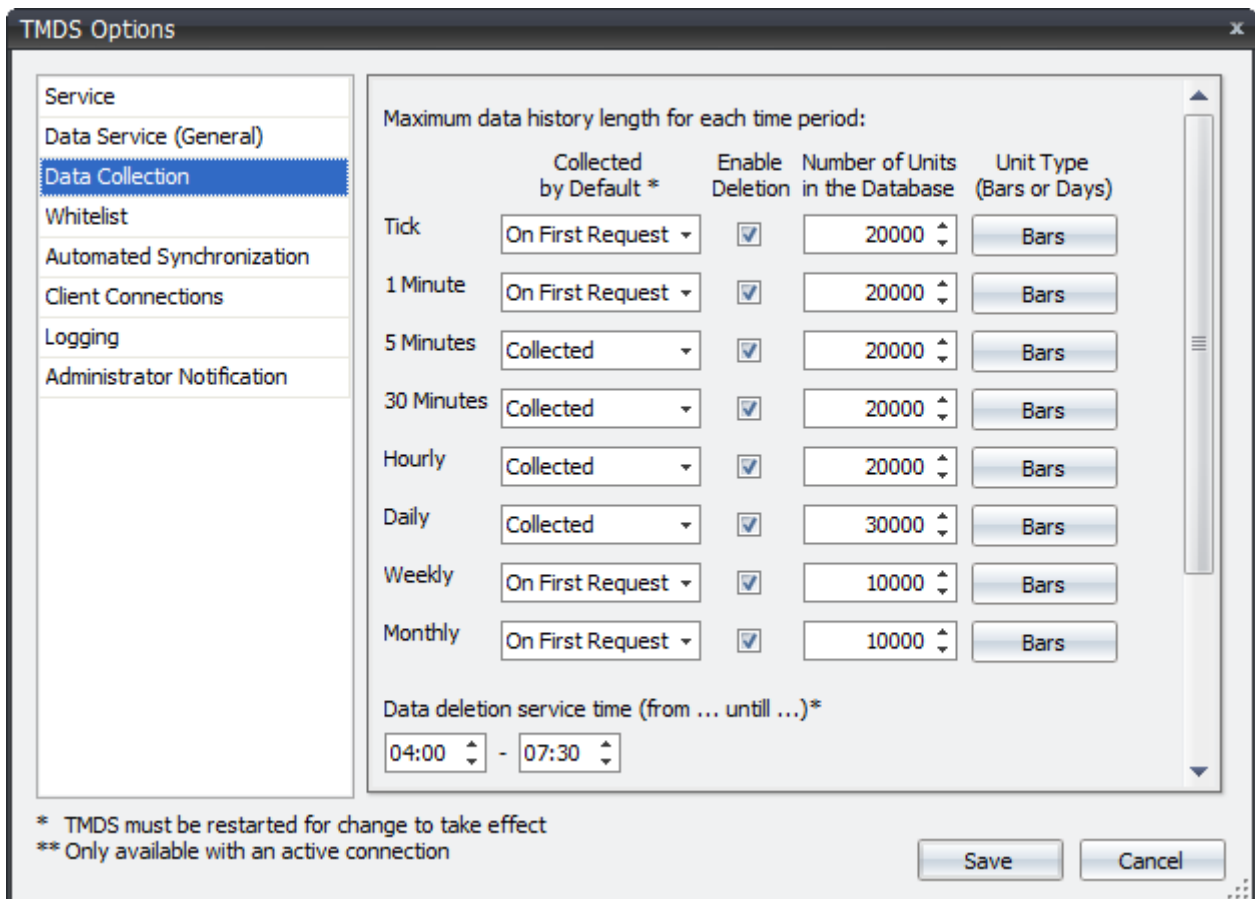
This function will restore the reference data of the symbol as received from the datafeed.



## 9.4 Editing data collection settings



Click Manage Settings  
→ Data Collection.



For each period, the data collection can be disabled, enabled by default or enabled on first request.

The unit can be set to **bars** or **days**. For example, for a period of “1 Minute” for “90 days”, this means that the data of up to the last ninety days is kept in the data cache. This way, year-old data of expired contracts might be kept.

A **data deletion service time** can be entered in which the data for time periods with **Enable Deletion** checked will be deleted. If the time frame is too short to delete the old data of all symbols, the symbols for which the data could not be processed are prioritized for the next service run.



## 9.5 Rolling Forward Symbols (User-Defined Continuations)

Starting with version 5.2, TMDS is able to publish server-calculated Rolling Forward symbols with real-time rollover, also known as User-Defined Continuations. Usually, one would create such a symbol by picking it directly from a predefined symbol list or by building one freely using Tradesignal's **New Symbol Wizard**.

TMDS Rolling Forward symbol names are constructed following a specific syntax:

*<root code><shift>.<back adjustment>.<candle calculation>.<period offset>.<months>*

<i>&lt;root code&gt;</i>	The future symbol root code
<i>&lt;shift&gt;</i>	Contract shift: <i>.NEAR</i> or <i>+1, +2, +3, ...</i>
<i>&lt;back adjustment&gt;</i>	<i>A</i> (absolute), <i>P</i> (proportional), or <i>U</i> (unadjusted); default: <i>A</i>
<i>&lt;candle calculation&gt;</i>	<i>CC</i> (close-close), <i>OO</i> (open-open), <i>CO</i> (close-open), <i>OOO</i> (open-open continuous); default: <i>CO</i>
<i>&lt;period offset&gt;</i>	<i>D</i> (days) or <i>M</i> (months) followed by a day offset, e.g. <i>-1</i> ; default: <i>D0</i>
<i>&lt;months&gt;</i>	The month future codes to include ( <i>M+</i> followed by a combination of the letters <i>FGHJKMNQUVXZ</i> in this order); default: <i>FGHJKMNQUVXZ</i>

Example: *FDX.NEAR.CO.D-1.M+HMUZ*

This specifies the rolling forward based on the German DAX future (Thomson Reuters root code *FDX*) with the current contract as the front month (*NEAR*), stitching the close of the expiring contract to the open of the continuing contract (*CO*) one day before the expiry date (*D-1*) using the four quarterly contracts Mar/Jun/Sep/Dec (*M+HMUZ*).

Note that there is no *<back adjustment>* specified in this example. All Rolling Forward symbol part names except *<root code>* and *<shift>* are optional, default settings will apply when omitted.



## 10 Filtering Ticks

There are a number of configuration options to instruct TMDS

- to ignore prices that deviate too much from their predecessors (value filter) and
- to ignore updates that are received outside an instrument's session (session filter).

Price filtering is configurable

- on a per symbol basis, see “Setting a bad tick filter for a single symbol” on page 40.
- on a per instrument class basis, see “Editing tick filter settings” on page 41.

*Note: Once a price is terminally filtered (i.e. not confirmed by a successive tick in the same price range) it cannot be restored, as filtered prices are not stored in the TMDS data cache file. Therefore, use the filter mechanism very cautiously to avoid filtering legitimate ticks.*

### 10.1 Setting a bad tick filter for a single symbol

To set up a bad tick filter for a single symbol:

1. Click the symbol for which a filter is required.
2. On the Reference Data tab, check the **Filter Active** option.
3. Select the desired **Filter Type** and **Filter Value** according to the following table:

Filter Type	Filter Value	
Global	n/a	Use the global relative filter specified in the tick filter settings (which can be configured on a per instrument class basis).
Relative	Percent	Filters a tick if it is greater or less than the preceding tick by a specified percentile amount
Absolute	Fixed	Filters a tick if it is greater or less than the preceding tick by a fixed amount



## 10.2 Editing tick filter settings

*Note: The TMDS will have to be restarted if you change the filter settings described in this section.*

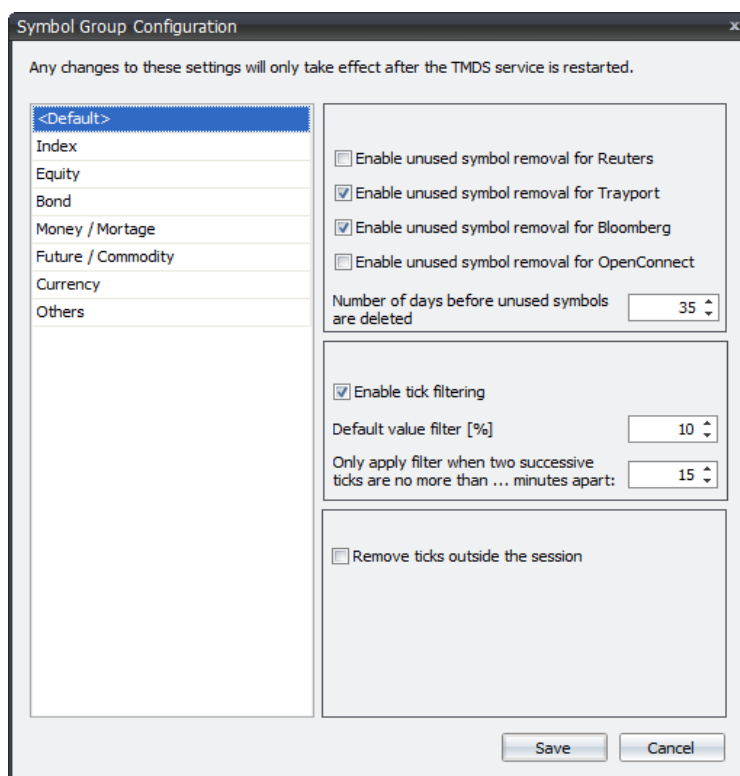
If no explicit price filter configuration had been performed for an instrument, a global filter will apply. The global filter is a relative value filter that will ignore a tick if it is greater or less than the preceding tick by a specified percentile amount. It is set in Tick Filter Settings.



**Click Tick Filter to edit Tick Filter Settings.**

**The Symbol Group Configuration dialog opens.**

The **<Default>** values apply to all instrument classes unless configured otherwise.



For the **<Default>**, as well as for all instrument classes, **Enable unused symbol removal** can be set.

Tick filtering works with two values: the value filter applied to the price data, and the time elapsed.

The value (price) filter will only be applied if not too much time has elapsed between two successive tick updates. This amount of time (in minutes) can also be set per instrument class.



### 10.3 Setting up the session filter

*Note: The default session filter only offers tick removal. The following advanced settings must be configured per instrument class.*

Instruments have a trading session applied to them. The trading session is usually defined by the datafeed. If no trading session is defined, 24/7 trading is assumed.

By default, TMDS will not filter ticks outside an instrument's session, but will store all updates in its data cache file and forward updates to connected clients. The client (e.g. Tradesignal) may then decide to omit the update from a chart when it is received out-of-session.

In order to enable TMDS-side filtering based on trading sessions, check **Remove ticks outside the session**.

A trading session defined by the datafeed can be overridden on a per instrument basis.

For a Trayport datafeed, the **standard session** and **standard timezone** can be edited for the datafeed via **Manage Datafeed**.

#### 10.3.1 Configuring session extensions

Checking **Remove ticks outside the session** will enable the configuration of session extensions (in minutes). Session extensions can be used to expand an instrument's session slightly, e.g. in order to include market close tick information which may be published shortly after a session ends. The extension values for start/end of session will only be applicable when you have more than one session per day.



Above is a schematic visualization of two daily sessions where session (1) runs from 08:00 till 12:00 and another session (2) runs from 15:00 till 18:00. The extension marked as (3) would refer to the start of day, the one marked (4) to the end of session, the one marked (5) to the start of session, and the one marked (6) to the end of day.



## 10.4 Checking filter settings

To inspect the filter settings for the selected symbol (price and session filter), open the Query tab and click **Filter**. The current settings for the selected symbol will then be printed to the TMDS log.

To get a list of the actually filtered ticks printed to the TMDS log, open the Log tab. In the section Filter, select

- **Price** to log ticks being filtered due to a big leap in price.
- **Session** to log ticks being filtered due to the update being received outside the configured session.

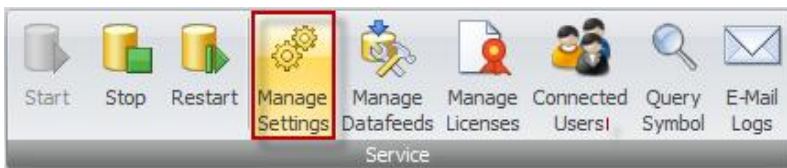


## 11 Filtering Instruments (Whitelisting)

In order to handle a large number of instruments, multiple TMDS can be set up so that certain instruments are only collected/distributed on specific servers.

*Tip: This is especially useful in a Master/Slave deployment scenario, see “Scenario 4a: Distributing feeds and symbols across individual servers” on page 10*

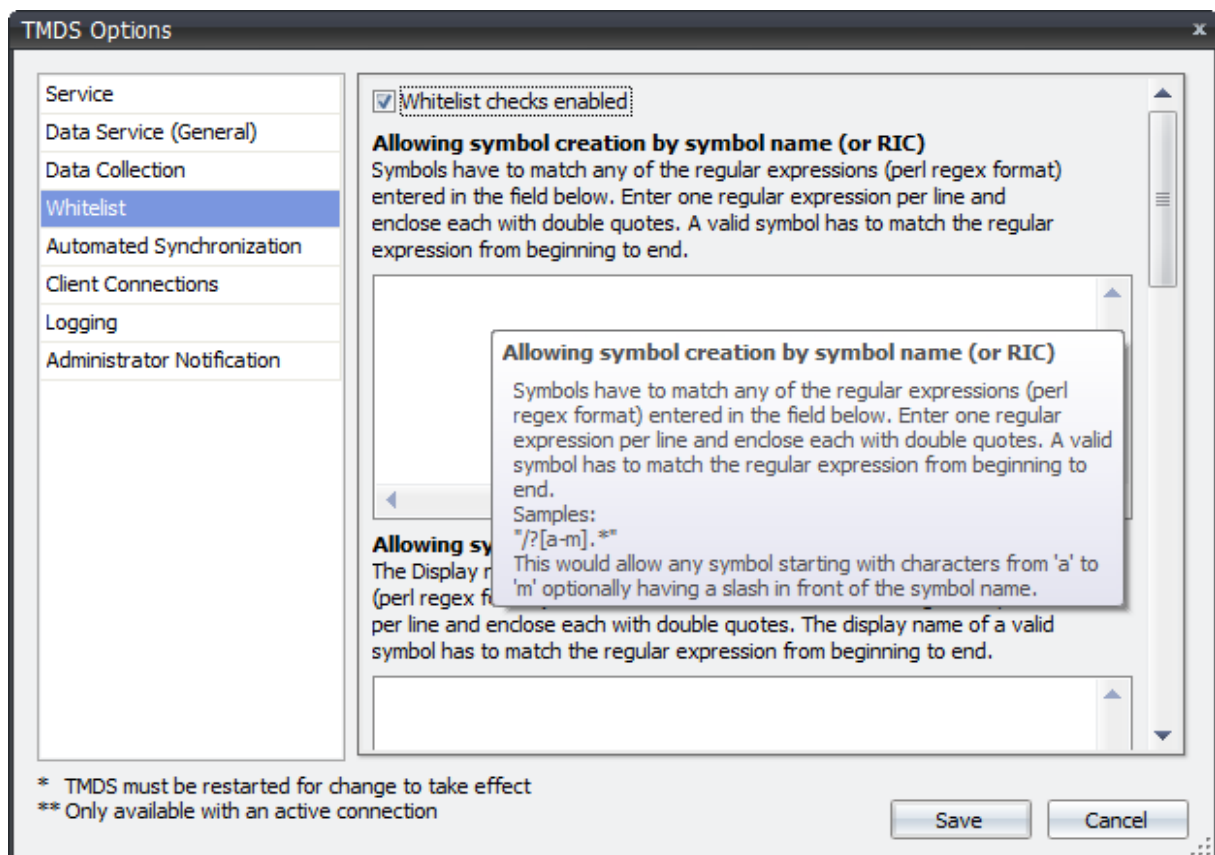
TMDS will match each new symbol against every whitelist section. The symbol will be created by TMDS as soon as one matching whitelist entry is found. If no whitelist filtering should take place, the section(s) need to be left blank or any existing entries have to be commented out.



**Click Manage Settings  
→ Whitelist to edit the  
whitelist settings.**

Select **Whitelist checks enabled** to switch on/off the whitelist filtering for new instruments.

For further information, please refer to the help texts and the information that appears when moving the mouse over the entry fields.



## 12 Backup / Restore / Synchronization

This section offers information on manual backup and restore of the data cache file. For automatic backups, see “Nightly Maintenance” on page 55.

A typical reason for a manual backup is to make a TMDS data cache file transportable, for example if you want to duplicate a data cache file on another TMDS machine. "Raw" TMDS data cache files (*tmds.dat* files) cannot be copied while the TMDS is running.

Backing up a TMDS data cache file will create a *tmds.dat.bak* file, which can be copied to another TMDS machine and be restored later.

*Note: Backing up and restoring a data cache file may take up to several hours for very large data cache files.*

### 12.1 Backing up a running (live) TMDS



Click Backup and confirm.

The status of the backup process is displayed on the right side of the status bar.

Resulting files in the installation folder:

- backup file *tmds.dat.bak*
- log file *tmds.dat.backup.log*

### 12.2 Backing up a stopped TMDS



Click Backup and confirm.

A command window opens with status information. After the backup, press any key to close the command window.

Without a running TMDS, the backup is performed by the *gbak* utility. TMDS Console will start this utility via the *backup.cmd* script in the installation folder. You can also start this script directly.

Resulting files in the installation folder:

- backup file *tmds.dat.bak*
- log file *tmds.dat.backup.log*



## 12.3 Restoring a TMDS

Note: A backup file *tmds.dat.bak* must be available in the installation folder.



Click **Stop** to stop the TMDS service.

*This is necessary for a restore.*



Click **Restore** and confirm. A command window opens.

To proceed with the restore, press any other key in the command window. To cancel the restore, press **CTRL+C**.

The status information is displayed in the command window. After the restore, press any key to close the command window.

The restore is performed by the *gbak* utility. TMDS Console will start this utility via the *restore.cmd* script in the installation folder. You can also start this script directly.

Resulting file in the installation folder:

- *tmds.dat.restore.log*

## 12.4 Synchronizing Systems

TMDS installations can be synchronized in their symbols and historical data. The destination system synchronizes with a source system by pulling data from the source system.

The synchronization can be done in two ways:

- *automatically*
- *manually*

For both synchronization methods, a working connection to the source system must be available, see “TMDS Connections” on page 31.

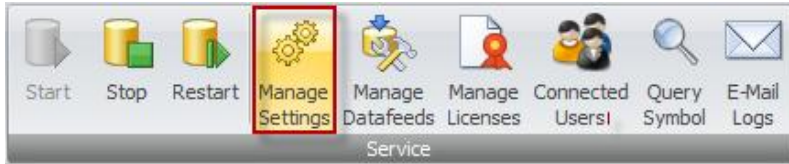
*Automated synchronization is an important feature for deployment configurations, see “Deployment: Common Installation Configurations” on page 8.*



### 12.4.1 Enabling automated synchronization

Note: The TMDS must be restarted for changes to take effect.

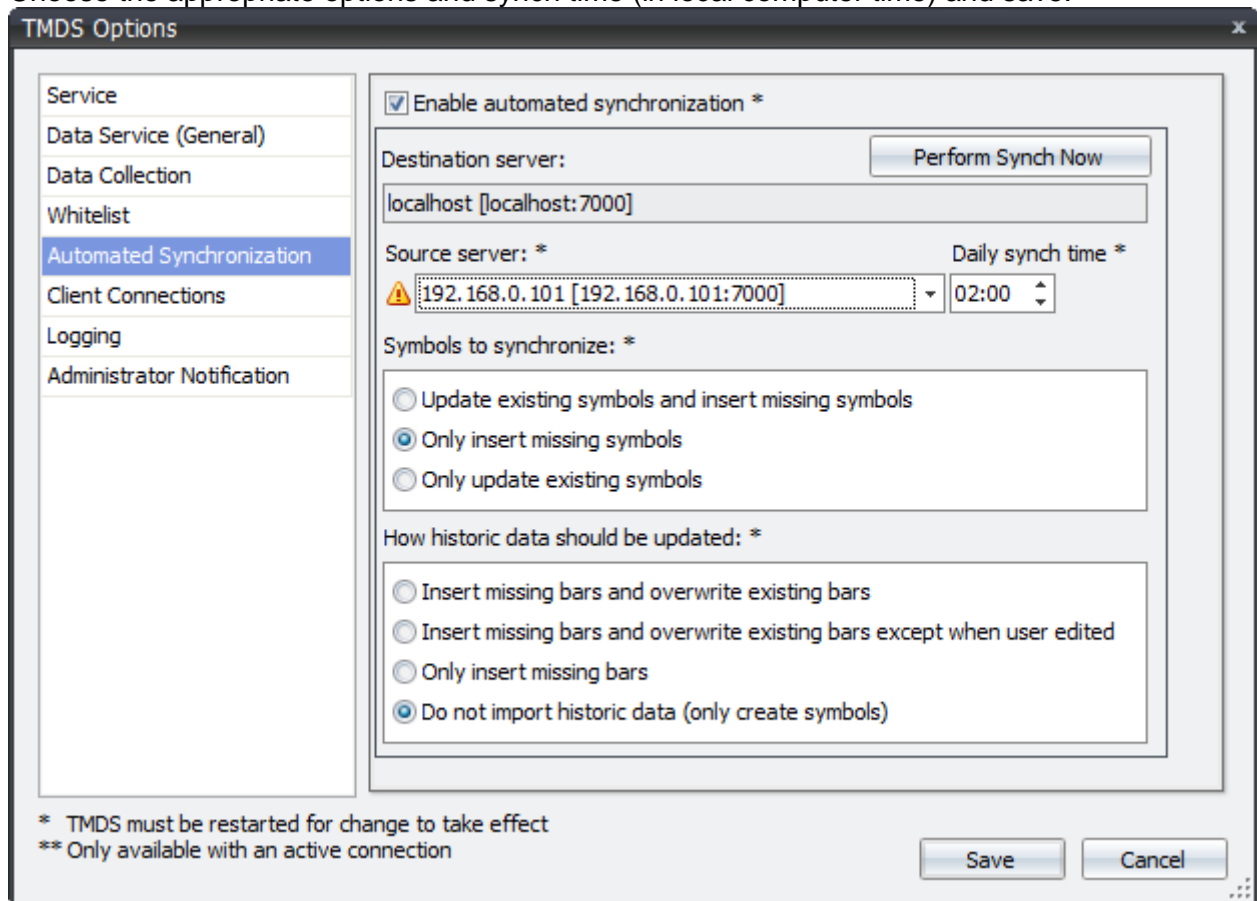
Automated synchronization requires identical datafeed setups (including the prefixes). This results in two TMDS with identical setups that can act as one.



Click **Manage Settings**  
→ **Automated Synchronization**.

The synchronization settings open. The destination system is always the system you currently work with.

Choose the appropriate options and synch time (in local computer time) and save.

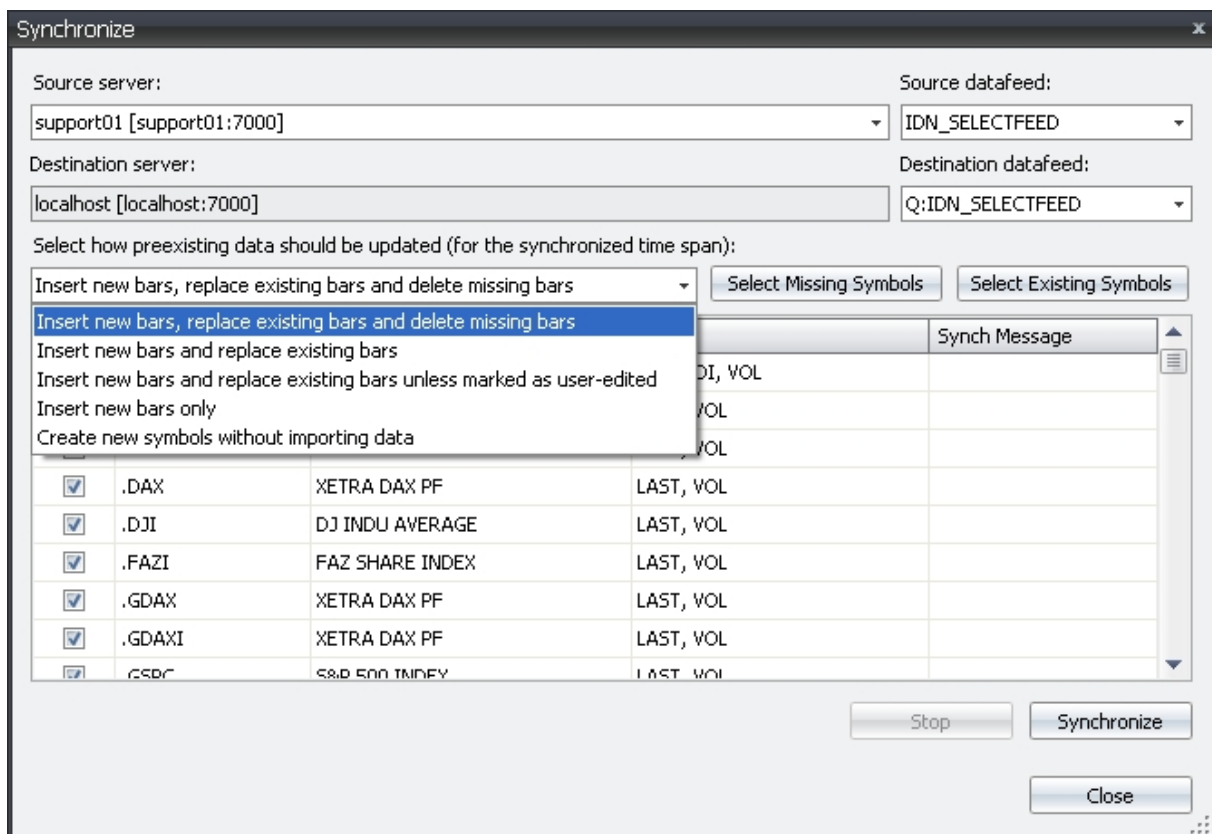


### 12.4.2 Manual synchronization



Click **Synchronize** to start the synchronization manually.

The dialog offers similar options to the automated synchronization. The destination system is always the system you currently work with. Enter the **source server**, the **source datafeed**, and the **destination datafeed**. The symbols available on the source are then displayed.



*Note: The chosen option only has an effect within the range of available data on the source server. For example, if the source system has historic data of one year and the destination system has historic data of five years; all options will only be applied to the one year range.*

For easier selection of symbols on the source system, the two buttons **Select Missing Symbols** and **Select Existing Symbols** are available.

Upon synchronization, TMDS will check whether the prefixes fit and are compatible.

- If the prefixes are different, a warning will be issued that you can override.
- If the synchronization fails, a message with further information will be issued.



## 13 Export / Import

This section refers to the export/import of symbol data.

- For the export of user groups, see “Exporting / importing user groups” on page 30.
- For the export of the data cache files, see “Backup / Restore” on page 45.
- For the export of TMDS logs and information, see “Preparing information for sending by e-mail” on page 53.

### 13.1 Importing symbols

For importing symbols, files in a supported format have to be available.

Two import formats are supported:

Binary: \*.zdbu, TMDS format

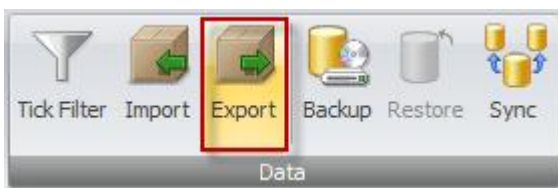
CSV: \*.csv, comma separated values (single or multiple symbols), compatible with Thomson Reuters DBU format



Click **Import** to start the import wizard.

It leads you through several selection steps. Add one or more files for import, symbols from the selected files, the datafeed the symbol(s) belong to, and the import mode.

### 13.2 Exporting symbols



Click **Export** to start the export wizard.

It leads you through several selection steps for the datafeed, one or more symbols, periods, time ranges, and the export format.

Three export formats are supported:

Binary: \*.zdbu, only useful for import into TMDS

CSV: \*.csv, comma separated values, compatible with Reuters DBU format

Multi CSV: \*.csv, one csv file per symbol, compatible with Reuters DBU format

By default, the export file(s) are written to the folder **Console/export** in the installation folder.



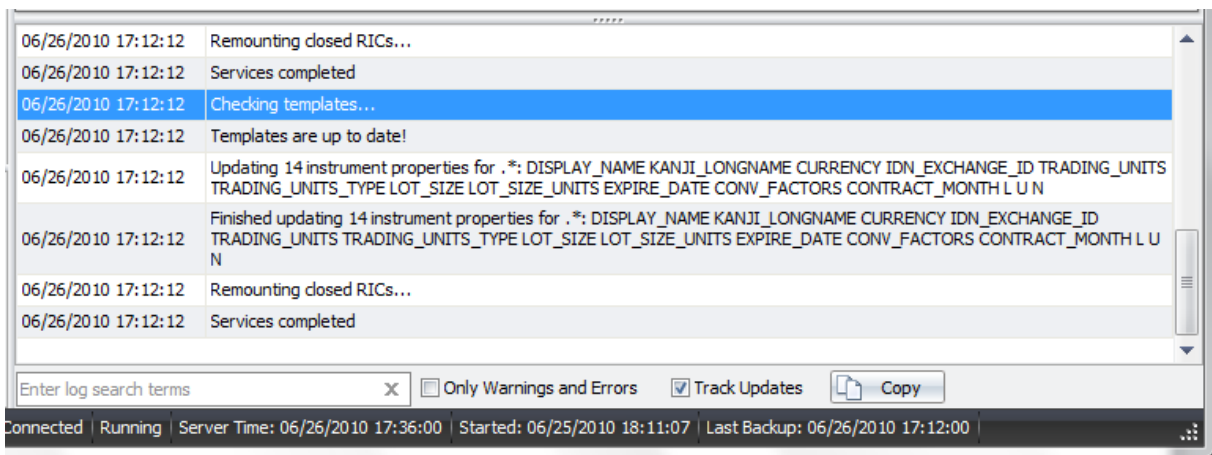
## 14 Logs, Alerts & Troubleshooting Tools

Helpful information about the system and the current or latest activities is given in the TMDS Management Console.

### 14.1 Logging console and log file

The logging console displays the contents of the TMDS log file (*mds.log*), which is written in the folder **WD/** in the installation folder.

By default, only significant events on a TMDS are logged. More exotic events, which can result in large quantities of data, can be enabled and disabled on the Log tab, see below.



Logged items are color-coded in the logging console.

Color	Severity
Black	Normal log message
Light Grey	Minor informative message, normally ancillary information
Orange	Important notification
Red	Critical error

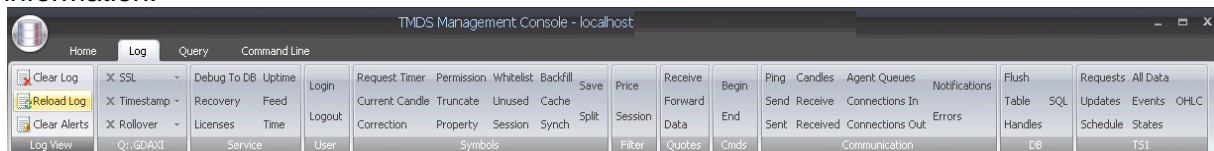
Options can be found in the context menu, for example to clear the log or the alerts (errors and warnings).



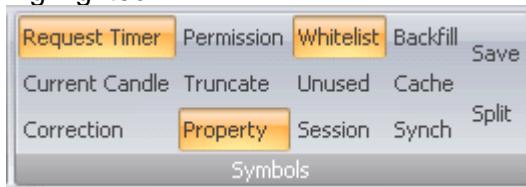
## 14.2 Log

*Note: This tab is normally only used in association with support staff to diagnose reported issues. Activating some or all of the commands can lead to significant performance problems and should therefore only be used by experts.*

The Log tab offers very granular logging of activities within TMDS. The available data is grouped by common theme. Move the mouse over the buttons to see a tooltip with information.



Click the buttons to switch the logging on or off for each data set. Data being logged is highlighted.



Selected logging options are persistent and will be re-activated when the TMDS is restarted.

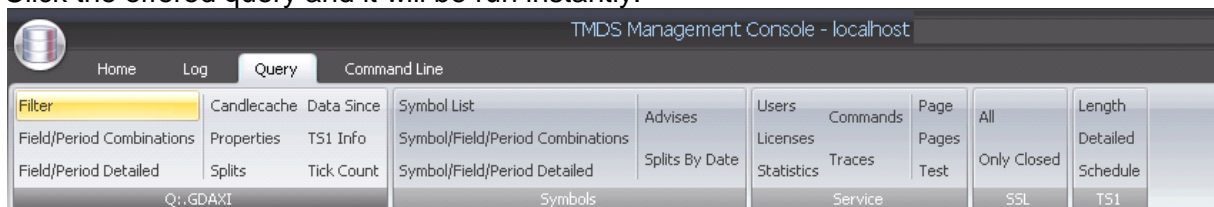
*Note: Logging can also be enabled manually by entering **trace** commands in the command line. Manually entered trace commands are not persistent, i.e., will not be re-activated when TMDS restarts.*

## 14.3 Query

*Note: This tab is normally only used in association with support staff to diagnose reported issues. Activating some or all of the commands can lead to significant performance problems and should therefore only be used by experts.*

The Query tab offers the ability to run queries that will be written to the TMDS log file (*mds.log*) and displayed in the Logging console.

Click the offered query and it will be run instantly.



*Tip: Queries can be run manually by entering **query** commands in the command line.*

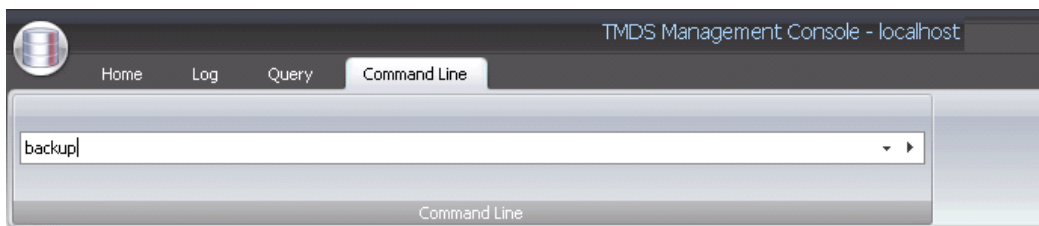


## 14.4 Command line

*Note: This tab is normally only used in association with support staff to diagnose reported issues. Activating some or all of the commands can lead to significant performance problems and should therefore only be used by experts.*

With the command line, script commands can be issued directly to any connected TMDS. For a complete list of suitable commands, see “Command Reference” on page 56.

- For valid commands, the response will be added to the log.
- For invalid commands, an error dialog will be displayed.



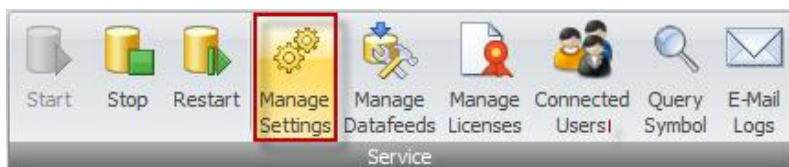
*Note: Commands can only be executed if the TMDS Service is running and the TMDS Management Console has a valid connection to it. Both of these states can be validated in the status bar, see “Status bar” on page 18.*

## 14.5 Query symbol



Click **Query Symbol** to check whether an instrument is available in the datafeed. A report is delivered on the instrument. This is especially useful for troubleshooting.

## 14.6 Sending automatic e-mail messages to administrators



Click **Manage Settings** → **Administrator Notification**.

Enter a comma-separated list of email addresses and an SMTP server.

*Notifications will be sent for critical events only.*



## 14.7 Preparing information for sending by e-mail



Click **E-Mail Logs** to create a *.zip* file with information about your system.

The *.zip* file is written to the folder **Console/export** in the installation folder. The naming convention is

*[system name]\_MDSLogs\_YYYY\_MM\_DD\_HH\_MM\_SS.zip*.

After creation, Windows Explorer opens. The *.zip* file could now be sent to Tradesignal support by selecting **Send to** from the context menu, for example.



## 15 Scripting TMDS

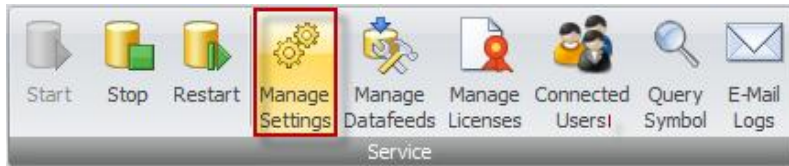
The command line in TMDS offers the ability to enter single commands, see “Command line” on page 52. These commands can also be used for more thorough scripting of the TMDS software with batch files.

Every aspect of TMDS can be scripted. For a full list of all possible scripting commands, see “Command Reference” on page 56.

### 15.1 Using batch/script files

Batch files must be in text format.

#### 15.1.1 Implementing their automatic execution



Click **Manage Settings** → **Service**.

The service settings open.

Two scripts can be added:

- **Service script file:** A file containing TMDS commands that will be consecutively executed at the given service time.
- **Startup script file:** A text file containing TMDS commands that will be consecutively executed at start up.

#### 15.1.2 Manually starting a script

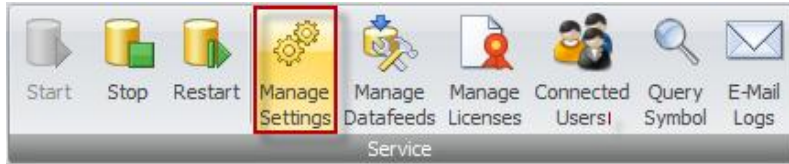
A script can be started directly via **Application button** → **Execute Script**.

Alternatively, a script can be called up in the command line with **script filename**.



## 15.2 Nightly Maintenance

TMDS will automatically back up its data cache file on a regular basis. This backup is part of the nightly service, which will usually commence at 23:00 local (computer) time. If needed, this time can be changed.



Click **Manage Settings**  
→ **Service**.

Edit **Daily Service Time**  
to change the time of the  
nightly maintenance.

Nightly maintenance can be started manually via **Application button** → **Nightly Maintenance**. It offers four options:

**Complete nightly service:** Includes the single options plus a backup of the data cache file as described in “Backup / Restore” on page 45.

**Only force immediate TS1 requests** (Reuters only): Informs all instruments to request an immediate TS1 update (command **requestts1**).

- **Only force template reload:** Forces a reload of session and time zone data from the data stream (command **reloadtemplates**).
- **Force template reload and export them:** Forces a reload of the session and time zone data from the data stream and the template pages are written as text files to the ‘Templates’ directory (command **reloadtemplates [export]**).



## 15.3 Command Reference

In general all *time* arguments will be in the following format: yyyy.mm.dd#hh:mm:ss:mmm, and *period* will be one of: TICK, 1MIN, 5MIN, 30MIN, 1HOUR, DAILY, WEEKLY, or MONTHLY. Parameters called *symbol* will most likely be a standard RIC, although they could be an alias for a RIC of an artificial instrument.

All commands below take the following rough form: **command** *variable* [optional]

**alternate1|2**

### 15.3.1 Administrative commands

#### **email** *text*

This command can be used to test the email notification feature or to send arbitrary notifications to the list of administrators. The sent email will contain whatever *text* is. For example, "email going to shut down now!" will send an email containing the message "going to shut down now!" to all administrators.

#### **backup**

Starts creating a live backup of the TMDS data cache file. For more information, see "Backup / Restore" on page 45.

#### **updatelic**

Re-reads the license XML file and notifies connected clients to reacquire all licenses.

#### **dump lic**

Displays all licenses, and whether they are in use or not.

#### **script** *filename*

Executes all TMDS commands contained in the text file pointed to by the filename argument.

#### **skipdebug** *mode*

Defines what will happen with debug output: mode 0 - skips nothing; 1 - skips data file writes; 2 - skips sending to clients; 3 - (= 1 + 2) skips data file writes and send to clients.

#### **trace permission on|off**

Permission cache additions and removals can be traced with "trace permission".

#### **trace requesttimer on|off**

This traces information on pending instrument requests.



### 15.3.2 TS1 commands

#### **clearts1**

Removes all current TS1 requests (purges normal, VIP and Subito queues).

#### **dump ts1**

Displays the length of the TS1 queues.

#### **dump ts1 all**

Displays the contents of the TS1 queues.

#### **dump ts1 cache**

Displays general information about the TS1 page cache.

#### **dump ts1 cache all**

Displays the complete TS1 page cache.

#### **getts1info** *symbol fid period*

Returns the time of the last TS1 update, the time of the last successful TS1 update, and the time of the next scheduled TS1 update.

#### **requestts1**

Informs all instruments to request an immediate TS1 update. The command *clearts1* can be used to subsequently clear this request.

#### **trace ts1 on|off|level**

Switches TS1 dumping on or off. *level* is a number specifying what has to be traced. Add the following values to find the desired level: *ts1trace\_off* = 0 (trace nothing, same as OFF); *ts1trace\_requests* = 1 (trace start/end of TS1 requests); *ts1trace\_cache* = 2 (trace page and cache info); *ts1trace\_schedule* = 4 (trace schedule time); *ts1trace\_block\_from\_clients* = 8 (trace ts1 blocking originating from connected clients); *ts1trace\_block\_from\_server* = 16 (trace ts1 blocking originating from TMDS itself); *ts1trace\_dump* = 32 (traces the candle data received from TS1); *ts1trace\_dump\_facts* = 64 (traces TS1 facts); *ts1trace\_dump\_events* = 128 (traces aperiodic events); *ts1trace\_states* = 256 (traces SFC states like complete, error, initialized); \* ON switches all tracing on except *ts1trace\_block\_from\_server* and *ts1trace\_dump*.

#### **updatehistorical** *symbol*

Requests historical inter-day data for the instrument, for example: *updatehistorical EGRP.O*

### 15.3.3 Instrument and data commands

#### **datasince** *symbol period fid*

Retrieves the date of the oldest tick in data file for the instrument identified by the symbol, period, and FID tuple.

#### **createstats**

Creates all TMDS specific statistic RICs in the data file.

#### **deletedatarange**://*symbol fid period time*

Deletes a candle, for example: *REUTER://EUR= BID 1HOUR 2010.05.21#15:00:00:000*.



**deletedatarange** *feed://symbol fid period from\_time to\_time*  
Deletes a range of candles.

**deleteinstrument** *feed://symbol*  
Deletes the instrument with the given RIC, for example: *REUTER://EUR=*.

**dump advises**  
Displays all instruments currently advised by client applications.

**dump insts**  
Displays all instruments.

**dump insts** *symbol*  
Displays all instruments that share the given RIC.

**dump insts** *symbol db*  
Displays all instruments that share the given *symbol* or RIC and includes the last tick time and the number of ticks in the data file.

**dump insts all**  
Displays all instruments and sub-instruments.

**dump insts all db**  
Displays all instruments and sub-instruments and shows the last tick time and number of ticks contained in each.

**dump page** *page\_id*  
Displays the QQ page identified by *page\_id*, for example: *dump page QQCZ*.

**dump pages** *page\_id*  
Displays a series of QQ pages starting with the page called *page\_id*, for example: *dump pages QQDA*.

**dump props** *symbol*  
Displays the properties of the given *symbol* or RIC.

**dump splits** *symbol*  
Displays information about any splits that have been processed for the given symbol or RIC.

**dump splits** *date\_from date\_to*  
Displays information about any splits that have occurred during the specified period.

**dump ssl [closed]**  
Displays all symbols currently advised at the infrastructure in the log. If the *closed* argument is given, only symbols that are closed (e.g. dropped) will be printed.

**getlastaccess** *feed://symbol fid period*  
Returns the last access time for the specified instrument, for example: *REUTER://GBP= BID 1MIN*.



**getticks** *symbol fid period*

Retrieves the number of ticks stored in the data file for the given instrument, for example: EUR= BID DAILY.

**isEnabled** *symbol fid period*

Checks if the given instrument is enabled (1) or disabled (0).

**newinstrument** *feed://ric symbol*

Creates a new instrument and assigns a symbolic name to it (normally this will just be the name of the RIC), for example: *newinstrument REUTER://EUR= EUR=*.

**recalccandlerange** *feed://symbol fid period from\_time to\_time*

Recalculates all related instruments to the given instrument having longer period based on the data found in this instrument, for example: *recalccandlerange REUTER://GBP= BID 1MIN 36225.1234 36225.678*.

**reconnect** *ric*

Reconnects the given RIC that may have been dropped.

**reloadallhistoricdata** *ric [startdate]*

For feed handler instruments (like Trayport), this will retrieve historic data via the backfill mechanism and refresh existing data (making sure that user-changed prices will not be overridden). For Reuters instruments, this is just an alias for the updatehistorical command (refreshing TS1 data for daily, weekly, monthly price series). When SHORTNAME is given as \*, all instruments will be reloaded. The start date argument is optional and defines the start date for data reloading in the format YYYY.MM.DD#HH:MM:SS:mmm where the :mmm, :SS:mmm, #HH:MM:SS:mmm parts are optional. Valid start dates would be, for example: 2009.10.24, 2009.10.24#14:00, or 2009.10.24#14:00:06.

**reloadtemplates [export]**

Forces a reload of the session and time zone data from the data stream. The *export* parameter advises the TMDS to write the template pages as text files to the 'Templates' directory. Note that the templates directory must exist; otherwise, you will get an error message. Also note that the next template reload will use the stored templates located in the TMDS installation directory: these template must be deleted if it is not the desired behavior to reload them.

**remount**

Reconnects all instruments at the infrastructure.

**remount closed**

Reconnects all currently disconnected (closed) instruments.

**trace del on|off**

Switches deletion of obsolete data dumping on or off.

**trace expire on|off**

Switches dumping of date/time expiry information on or off (mainly for odd ticks for Futures instruments).



**trace feedinfo on|off**

Switches tracing of informational datafeed infrastructure messages concerning instruments on or off.

**trace flush on|off**

Switches instrument cache flushing dumping on or off.

**trace propchange on|off**

Switches logging of instrument property change notifications (like a changed display name) on or off.

**trace rollover on|off ric**

This will enable/disable writing of rollover logs for the given RIC into the TMDS directory specified in MDS.INI.

**trace sessions on|off**

When this is switched on, QQ session download will be logged in the files *sessions\_raw.log* and *sessions.log*. To force a session download, use the *reloadtemplates* command.

**trace split on|off**

Switches tracing information on performed stock splits on or off.

**trace ssl file|log|on|off ric1 ric2 ...**

Switches tracing of realtime data as it gets received on or off. If RICs are specified, only messages applying to these will be displayed. You can choose either the TMDS log or a file as the target for the realtime data logging. In the latter case, the file name will be the RIC name and .txt appended. The parameters LOG and ON are synonymous. If no RICs are specified, realtime data for all connected instruments will be displayed.

Examples:

<i>trace ssl log EUR=</i>	-- will continuously display realtime data for the EUR=
<i>trace ssl file .DJI .GDAXI .SPX</i>	-- will log realtime data for .DJI, .GDAXI, and .SPX into three files
<i>trace ssl off .DJI</i>	-- will switch off tracing realtime data for .DJI only
<i>trace ssl off</i>	-- will switch off all realtime data traces

**trace timestamp on|off symbol**

When switched on, dumps information on the calculation of date/time stamps from the original data stream fields for every incoming update. If *symbol* is omitted, this will be dumped for every instrument.



### 15.3.4 Informational commands

#### **dump commands**

Displays all currently active commands.

#### **dump permission**

Displays the contents of the permission cache.

#### **dump stats**

Displays the current values of all TMDS statistic counters.

#### **dump traces**

Displays all traces and their current values.

#### **dump users**

Displays information about any connected users.

#### **getclassname** *symbol*

Returns the TMDS class name that the given RIC is contained in.

#### **getstarttime**

Displays the time that the TMDS started.

#### **trace commands** *on|off|level*

Switches command logging on or off. All commands coming from any client will be logged. Add the following values to find the desired level: 0 - no trace (same as off); 1 - trace starting of commands; 2 - trace finishing of commands.

#### **trace send** *on|off*

Switches tracing of candle block sending on or off.

#### **trace stat** *on|off* *statistic*

Switches dumping of statistical data concerning the TMDS on or off. If no RIC is specified, this applies to all TMDS statistics.

#### **trace table** *on|off*

Switches logging of table creations on or off.

#### **trace users** *on|off|level*

Switches user logging on or off. Add the following values to find the desired level: 0 - no trace (same as off); 1 - trace user login; 2 - trace user logout; *trace users on* switches all these options on.



### 15.3.5 Recovery commands

#### **clearrecovery**

If issued of a *master* TMDS, this command removes all outstanding items received from the recovery system pending insertion into the data file. If executed of a recovery TMDS, the command deletes all requested but not yet sent items, including items in the retry loop.

#### **recoverdatafeed** *symbol fid period*

Requests recovery data from the recovery system specified in *MDS.INI*, and recalculates the historical data. The operation works with wild cards, for example: *recoverdata \* EU\* ASK \** will request data for all ASK instruments starting with EU for any period.

#### **recoverydelinsts**

If this is a recovery system, this instruction will check for and delete any instruments that have not been used for a configurable time period.

#### **reportinsts**

Forces a *master* TMDS to send its instrument list to the recovery system defined in the *MDS.INI* file.

#### **trace recovery on|off**

Switches logging of data recovery related processes on or off.



## 16 General Troubleshooting

### 16.1 FAQ

***I have started the Market Data System but I do not appear to be getting any data***

This is normally because of a problem configuring the TMDS to your specific Reuters infrastructure, please refer to “Advanced configuration and troubleshooting of TMDS for Thomson Reuters Data” on page 65 for a list of common installation problems and their solutions for each of the main Reuters infrastructures.

***The TMDS appears to have difficulties failing over to a secondary P2PS when the main P2PS server goes down and there are messages like this in the TMDS log: “Loss of Channel Detected. Channel Timed-Out. Recovery Underway” (Thomson-Reuters only)***

This may apply to RMDS infrastructures running in TRIARCH mode with multiple P2PS configured in the *sslapi.cnf* file.

When TMDS detects a connection loss to the primary P2PS, it will try to re-request all previously subscribed RICs from a secondary P2PS. If the P2PS is not prepared to handle the amount of requests in a timely manner, connection channels may be overflowed and cut by the P2PS, resulting that TMDS will try to re-subscribe all symbols again, which may fail again and so on. You may be able to remedy this by tweaking the size of the communication buffers on the P2PS side, thereby giving the P2PS more time to process the requests. You may want to add or adapt the following settings in the P2PS configuration file (*triarch.cnf* for P2PS v5 or *rmds.cnf* for P2PS v6):

```
*p2ps*guaranteedOutputBuffers : 2000
*p2ps*maxOutputBuffers : 4000
*p2ps*outputThresholdBreach : 3000
*p2ps*outputThresholdOK : 1000
*p2ps*poolSize : 16000
```

You will have to restart the P2PS service in order to make your changes effective. If this is still not sufficient you may want to throttle the number of pending requests, which can be done on the client side. Starting from version 3.0.2.4, TMDS supports an additional feed configuration file. Enter the file via **Manage Datafeed**, field **Nondefault SSL configuration file**. This changes the `ConfigFile` setting in the `[SSL]` section of *MDS.INI*.

Try adding this line to the feed config file (which is called *tmds.cfg* in the above example) and restart the TMDS:

```
*max_pending_limit : 20
```

More information on channel cuts and P2PS configuration issues can be found in Reuters documentation (e.g. RMDS Tuning Guide).



***I have TMDS running in an RWSHi environment where the VPN tunnel gets cut every 16 hours. For some reasons, the TMDS sometimes does not automatically re-establish the connection so that all my instruments are not updating anymore meaning I have to restart the TMDS. The feed log file log.out contains an entry: “Logon failed. User <username> has already logon to resilience cou” (Thomson-Reuters only).***

The probable reason for this condition is that there is a failover sink distributor specified in the `sslapi.cnf` file and there is another sink application using the same DACSID running on the same machine (for example Reuters Kobra). After the VPN connection is cut and re-established, both applications will automatically try to re-establish the datafeed connection. When one application successfully logs-in to the primary sink distributor, and the other tries to log-in to the secondary (failover) sink distributor, the Thomson Reuters infrastructure will reject the second log-in as one user is already connected to an alternate sink distributor in the infrastructure. The behavior of the Reuters SFC API is to never retry other sink distributors if the reported error is received.

There are some possible workarounds for this problem suggested by Thomson Reuters development staff:

- *Configure the infrastructure to remove the one target sink distributor restriction for a DACS ID*
- *Set up the DACSID to support two positions*
- *Use an alternate DACS ID for each application*
- *Ensure that all applications only use a single sink distributor. Although this will mean that if the sink distributor fails then no failover would be available.*

In all cases, it is strongly recommended to consult a Thomson Reuters engineer before applying any of these or other changes to a live infrastructure.

***My TMDS only appears to be getting real-time tick data, but no historical (daily) TS1 data (Thomson-Reuters only)***

If you have a Tibco TIC (SASS2) or RMDS (SASS3) infrastructure, this situation is occurring because you are currently running the older TIB 9 software. The Market Data System supports TIB 10 and above. To resolve this problem, contact your Reuters technical representative and request an upgrade to the more recent software.

***I have recently upgraded my TMDS from version 1.4 on my Windows XP or 2003 server and I am experiencing significant slowdown while accessing data***

This can be caused by Windows XP mistaking the TMDS data cache file for another type of file. If you have a file in your TMDS installation directory called `mdsdb.gdb`, simply stop the TMDS, rename the file to `mdsdb.dat`, open the `MDS.INI` file in a text editor and replace the entry `Data filePath` in the `[General]` section with the new filename, then restart the TMDS.

***I have recently lost power to my Market Data System and now the data cache file appears to be locked***

Simply start the Management Console using an account with administrative rights, and stop and restart the service. If that approach does not correct the data cache file lock, shutdown the computer and reboot.



***I have TMDS running in an RWSHi environment but the TMDS service will not connect after I have logged in (Thomson-Reuters only).***

When the machine starts up, TMDS will start immediately (as a Windows service) and try for 30 seconds to establish a connection. The problem is the Reuters VPN is only connected when a user logs in which normally takes longer than 30 seconds and the VPN is needed to connect to the Reuters data servers. The solution requires that the 30 second connect timeout is increased to a very high value to give the user an opportunity to sign-in. Set the following *MDS.ini* setting to a very high number (e.g. seconds in a day: 86400) and restart the TMDS.

```
[SSL]
ServiceStartTimeout=86400
```

## 16.2 Advanced configuration and troubleshooting of TMDS for Thomson Reuters Data

### 16.2.1 Troubleshooting failed connections to Triarch/RWS (SSL4) infrastructures

Assuming all settings are correctly set when the Tradesignal Market Data System is started from the Tradesignal MDS Control Center, the message “TMDS is now running” will be displayed as the last line in the log.

Should connecting to the feed infrastructure fail for some reason, the TMDS will display as much information on the error as possible to help with the diagnosis; the reason for connect failure is usually one of the following:

- *The wrong feed (record service) was configured; for example, setting the record service to IDN\_RWS when you have IDN\_SELECTFEED. To solve this, set the **Service** field in the Manage Datafeeds pane.*
- *A configuration parameter has been misspelled: IDN\_SELECTFED instead of IDN\_SELECTFEED.*
- *An incorrect DACS user name was set. Please ask your DACS administrator for verification.*
- *A valid DACS user name was used, but connecting fails because another user is using that same name on a different workstation. Use a different DACS user name.*
- *The sink distributor name is not correctly entered in the **Host(s)** field of the Manage Datafeeds pane. If the name is correct, try pinging the sink distributor to ensure that it is up. This could also be a name resolution problem so ask your DNS administrator for assistance if necessary.*

If there is the need for additional feed configuration parameters, for example to aid troubleshooting or for tuning, use a **Nondefault SSL configuration file** which can be defined in the Manage Datafeeds pane. For Triarch infrastructures, this configuration file is usually called *sslapi.cnf*.



## 16.2.2 Manually configuring RMDS/RTIC (SASS3) infrastructures

The TMDS Management Console allows configuring datafeeds in a simple and straightforward manner. A manual configuration is therefore seldom necessary. TMDS nevertheless supports manual configuration of the data feed using a so-called **Nondefault SSL configuration file**.

This section describes the steps necessary to manually set up a connection of the TMDS to a RMDS (SASS3) infrastructure using a configuration file (usually called *sslrvcnf* for RMDS-based infrastructures). As with all feed infrastructures, you must have correctly configured your DACS settings in the UI.

*Note:* The TMDS only supports TIB 10 and above. Please ensure that your infrastructure meets this minimum requirement.

In the Manage Datafeeds pane, pick your *sslrvcnf* file in the **Nondefault SSL configuration file** field.

Assuming all settings are correctly set when the Tradesignal Market Data System is started from the Control Center, the message “TMDS is now running” will be displayed.

Should start up fail for some reason, the TMDS will display as much information on the error as possible to help with the diagnosis; the reason for start-up failure is usually one of the following:

- *The wrong record service was configured; e.g. setting the record service to IDN\_SELECTFEED when you have IDN\_RDF. To solve this, set the **Service** field in the Manage Datafeeds pane.*
- *A configuration parameter has been misspelled: IDN\_RDE instead of IDN\_RDF.*
- *An incorrect DACS user name was set. Please ask your DACS administrator for verification.*
- *A valid DACS user name was used, but connection fails because another user is using that same name on a different workstation. Use a different DACS user name.*
- *If the parameter `*useSeparateUpdateSession` is set to true, then all of the following options must additionally be present in the *sslrvcnf* feed configuration file, the actual values will vary from system to system, the important thing is that they are duplicated:*

```
*update_service : 9515
*updateService : 9515
*update_daemon : tcp:9500
*updateDaemon : tcp:9500
*update_network : 192.44.127.255;250.99.15.1
*updateNetwork : 192.44.127.255;250.99.15.1
```



- *The `sslrvcnf` feed configuration file does not contain all of the necessary configuration settings, please ensure that the following items are present:*

```
*serviceList : IDN_RDF
*recordServices : IDN_RDF
```

- *If the parameter `*useSeparateEntSession` is set to `true`, then all of the following options must additionally be present in the `sslrvcnf` feed configuration file, the actual values will vary from system to system, the important thing is that they are duplicated:*

```
*ent_service : 7500
*entitlementService : 7500
*ent_daemon : tcp:9500
*entitlementDaemon : tcp:9500
*ent_network : 192.44.127.255;250.99.15.1
*entitlementNetwork : 192.44.127.255;250.99.15.1
```

## 16.2.3 Entitlement Management with DACS

### 16.2.3.1 Advanced configuration of server-side DACS permissioning

TMDS supports performing entitlement checks on behalf of connected users utilizing the Thomson Reuters DACS infrastructure (Data Access Control System). In accordance to exchange rules, DACS permissioning has to be performed for all feeds that support DACS entitlement in order to control data access rights for every user connected to TMDS. This section explains how server-side DACS permissioning is performed and which advanced configuration options are available.

Server-side DACS permissioning is implemented using a dedicated Windows service residing on the same host as the TMDS. This service is called *TMDS Permission Server (TMDSPPS)* and its lifetime is managed transparently by TMDS without the need for user intervention. TMDS talks to the Permission Server via TCP/IP using the local default port number 27359. Should this port already be in use by another application, it can be changed in the TMDS Console's *Manage Datafeeds* dialog in the Reuters shared settings section (local TMDS Permissioning Server Port).

TMDSPPS will establish permanent connections to all configured DACS daemons. Should a connection fail, TMDS will have to disallow access to all symbols provided by the underlying data feed. This behavior is dictated by exchange regulations. In such a case, adequate warnings will be displayed in the TMDS log and alerts are sent out to connected clients using these feeds.

When a user connected to TMDS from a remote host requests accessing a symbol's data (for example a Tradesignal user opening a chart), TMDS will forward the symbol request to the TMDSPPS which in turn will interact with the DACS infrastructure to request a DACS lock on this symbol. When the DACS rejects accessing this symbol, TMDS will return an adequate error message back to the client. Usually, this process is performed within fractions of a second. If for any reason the request is outstanding for too long, the request is aborted and the client will not be allowed to access the symbol. This permission check request timeout can be configured in the TMDS console (the default is set to 30 seconds).



### **16.2.3.2 Connecting a Tradesignal 6.1 or earlier client to TMDS 5.2 or above**

Older Tradesignal versions (6.1 or earlier) were performing entitlement checks directly on the client machine, so there is no need to repeat permissioning on the server side. TMDS 5.2 and above will detect that an older Tradesignal client is connected and will therefore skip the entitlement checks for those clients.

### **16.2.3.3 Connecting a Tradesignal 6.2 or later client to TMDS 5.1 or earlier**

TMDS versions before 5.2 did not support server-side permissioning, so in order to ensure exchange-compliant access control, Tradesignal needs to perform client-side permissioning. To enable Tradesignal 6.2 or above to perform client-side permissioning, the Thomson Reuters SFC 4.6.0 components will have to be installed in Tradesignal's TSx components directory (under Programs\Common Files\Tradesignal\TSx).

The following SFC components are needed:

- dacs4w32.dll
- sipc3290.dll
- ssl48w3290.dll

When Tradesignal is unable to find these SFC components, access to Thomson Reuters provided symbols on remotely connected TMDS will be disallowed.

Note that Tradesignal does not need to perform client-side permissioning when connected to a TMDS that is running on the same machine as Tradesignal. In that case, Thomson Reuters symbols can be accessed by Tradesignal as they will be properly access controlled using the DACS user configured in TMDS.

