

# Tradesignal DataConnect 6.1

## User Guide

---



**tradesignal**®

## Table of Contents

<b>1</b>	<b>Overview</b>	<b>7</b>
1.1	Architecture	7
1.2	Scope of the manual	8
1.3	Target audiences	8
1.4	Expressions and formats used in this manual	8
1.4.1	Note and Tip	8
1.4.2	Formatting	8
1.5	Glossary	8
<b>2</b>	<b>Deployment: Common Installation Configurations</b>	<b>9</b>
2.1	Scenario 1: Local installation alongside Tradesignal (single user)	9
2.2	Scenario 2: DataConnect on dedicated system	9
2.3	Scenario 3: Two systems	10
2.3.1	Scenario 3a: One system as failover	10
2.3.2	Scenario 3b: Two identical systems working in parallel	10
2.4	Scenario 4: Multiple systems	10
2.4.1	Scenario 4a: Distributing feeds and symbols across individual servers	11
2.4.2	Scenario 4b: Scaling number of clients beyond scenario 3	12
2.5	Scenario 5: Local installations using a centralized licensing DataConnect	12
<b>3</b>	<b>Installation</b>	<b>13</b>
3.1	Preparations for installing DataConnect	13
3.2	System requirements	13
3.2.1	Supported operating systems	13
3.2.2	Minimum system requirements	14
3.2.3	Recommended system requirements for the use of DataConnect	14
3.3	Software Requirements	14
3.3.1	Global Vision API (Trayport Feed Handler)	14
3.4	Setup	14
3.5	Updating	15
3.6	Uninstalling	15
3.7	Copying an existing data cache file to a new DataConnect	16



<b>4</b>	<b>DataConnect Console</b>	<b>17</b>
4.1	User interface overview	17
4.2	Application button	18
4.3	Symbol area	18
4.4	Status bar	19
4.5	Mouse-over information	19
4.6	Context menus	19
4.7	Field labels	19
<b>5</b>	<b>Starting / stopping the system</b>	<b>20</b>
<b>6</b>	<b>Managing datafeeds</b>	<b>21</b>
6.1	Setting up Thomson Reuters	22
6.1.1	Connecting to TREP-RT (RSSL/OMM, Triarch, RMDS, TIB)	22
6.1.2	Connecting to Eikon	23
6.2	Setting up Trayport GlobalVision	27
6.2.1	Backfilling data history from Trayport SQL server	27
6.2.2	Rolling	28
6.2.3	Mapping Trayport Symbol Identifiers	29
6.2.4	Product change detection for Trayport trade updates	31
6.2.5	Smart Match Support	32
6.3	Setting up Bloomberg	35
6.3.1	Available upstream connection options for Bloomberg	35
6.3.2	Enabling data collection for supplemental fields	37
6.4	Setting up GlobalView	40
6.5	Setting up a CSV Feed	41
6.5.1	Creating a CSV file format	42
6.5.2	CSV Feed Configuration settings	46
6.6	Setting up Thomson Reuters Tick History Feed	47
6.7	Setting up OpenConnect	48
<b>7</b>	<b>Managing Licenses and Users</b>	<b>49</b>
7.1	The license and user group manager window	49
7.2	Setting up a centralized license server	50
7.3	License properties and usage	50
7.4	Floating licenses	51
7.5	Adding a new license	51
7.6	Removing a license	51
7.7	Using user groups	52
7.8	Adding / editing / deleting users	53



7.9	Connecting to DataConnect only to access Top-Up data	53
7.10	Exporting / importing user groups	53
7.11	Viewing connected users	53
7.12	Generating an alert when a license is about to expire	54
7.13	Generating Usage Reports (for Trayport)	55
7.13.1	Clearing Collected Usage Report Data	56
<b>8</b>	<b>DataConnect Connections</b>	<b>57</b>
<b>9</b>	<b>Managing Symbols</b>	<b>58</b>
9.1	Editing reference data	60
9.1.1	Editing trading sessions	60
9.2	Editing price data	61
9.2.1	Editing price data fields	62
9.2.2	Deleting and disabling candles	62
9.2.3	Adding candles	62
9.2.4	Reverting not yet committed price data	63
9.2.5	Resetting modified or disabled data	63
9.2.6	Saving price data	63
9.3	Clearing and restoring data	64
9.3.1	Restoring default historic data	64
9.3.2	Restore default reference data	64
9.4	Editing data collection settings	65
9.5	Rolling Forward Symbols (User-Defined Continuations)	66
9.6	Pass-Through Mode vs. Persistent Mode	66
9.6.1	Pass-Through Mode for Bloomberg data feed	67
9.6.2	Pass-Through Mode for Eikon data feed	68
9.6.3	Pass-Through Mode for OpenConnect data feeds	68
9.7	Automatic copying of historic data for Thomson Reuters Expired Contracts symbols	69
9.8	Specifying Overrides for Currency Symbols	70
<b>10</b>	<b>Filtering Ticks</b>	<b>71</b>
10.1	Setting a bad tick filter for a single symbol	71
10.2	Setting up tick filtering for symbol groupings	72
10.3	Setting up the session filter	73
10.3.1	Configuring session extensions	73
10.4	Checking filter settings	74
10.5	Automatic removal of symbols based on usage	74
<b>11</b>	<b>Filtering Instruments (Whitelisting)</b>	<b>75</b>



<b>12 Backup / Restore / Synchronization</b>	<b>76</b>
12.1 Backing up a running (live) DataConnect	76
12.2 Backing up a stopped DataConnect	76
12.3 Restoring a DataConnect data cache	77
12.4 Synchronizing Systems	77
12.4.1 Enabling automated synchronization	78
12.4.2 Manual synchronization	79
<b>13 Export / Import</b>	<b>80</b>
13.1 Importing symbols	80
13.1.1 CSV file formats	83
13.2 Exporting symbols	83
<b>14 Offline Mode</b>	<b>84</b>
14.1 Offline Mode Options	85
14.2 Offline Mode Cache Files	86
<b>15 Simple Network Management Protocol (SNMP) Integration</b>	<b>87</b>
15.1 Configuring the SNMP Integration for DataConnect	87
15.1.1 Installing Windows SNMP Service on Windows XP or Windows Server 2003 R2	87
15.1.2 Installing Windows SNMP Service on Windows 7 or Vista	87
15.1.3 Installing Windows SNMP Service on Windows Server 2008 R2	87
15.1.4 Configuring Windows SNMP Service	88
15.1.5 Configuring DataConnect to Provide SNMP Integration	88
15.1.6 Disabling DataConnect SNMP Integration	89
15.2 Working with the DataConnect SNMP Integration	89
15.2.1 DataConnect Management Information Base (MIB)	89
15.2.2 Information retrievable from DataConnect via SNMP	90
15.2.3 SNMP Traps	90
<b>16 Logs, Alerts &amp; Troubleshooting Tools</b>	<b>91</b>
16.1 Logging console and log file	91
16.2 Log	92
16.3 Query	92
16.4 Command line	93
16.5 Query symbol	93
16.6 Sending automatic e-mail messages to administrators	93
16.7 Preparing information for sending by e-mail	94



<b>17 Scripting DataConnect</b>	<b>95</b>
17.1 Using batch/script files	95
17.1.1 Automatic execution of script files	95
17.1.2 Manually starting a script	95
17.2 Nightly Maintenance	96
17.3 Command Reference	97
17.3.1 Administrative commands	97
17.3.2 TS1 commands	98
17.3.3 Instrument and data commands	98
17.3.4 Informational commands	102
17.3.5 Recovery commands	103
<b>18 General Troubleshooting</b>	<b>104</b>
18.1 FAQ	104
18.2 Advanced configuration and troubleshooting of DataConnect for Thomson Reuters Data107	
18.2.1 Troubleshooting failed connections to Triarch/RWS (SSL4)	107
18.2.2 Manually configuring RMDS/RTIC (SASS3) infrastructures	107
18.2.3 Manually configuring TREP-RT RSSL/OMM connections	109
18.2.4 Entitlement Management with DACS	109
18.2.5 How to fix the “Thomson Reuters Excel Add-In not activated” error when using the Eikon data feed.	111
<b>Appendix A: Supported Trade Units</b>	<b>112</b>



# 1 Overview

Tradesignal DataConnect (formerly known as 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 Tradesignal DataConnect are:

- *Collection of real-time and historical data; maintaining a complete history of both if the upstream feed does not provide historical data capabilities.*
- *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, simultaneous connection to data from one or more of:*
  - **Bloomberg:** Bloomberg Professional Workstation (DAPI and SAPI)
  - **Bloomberg:** B-PIPE
  - **GlobalView:** Energy Market Data Feed
  - **Thomson Reuters:** TREP-RT (OMM/RSSL)
  - **Thomson Reuters:** Triarch/RWS (SSL4), Tibco TIC (SASS2), RMDS (SASS3)
  - **Thomson Reuters:** Eikon Desktop
  - **Thomson Reuters:** Tick History (version 6.0 and above)
  - **Trayport:** Trading Gateway & Direct broker connections
  - **Tradesignal OpenConnect:** Middleware for open interoperability
  - **Generic CSV Data** (version 6.0 and above)

## 1.1 Architecture

DataConnect runs in the background as a 64bit native 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, DataConnect will do its job. In addition to the DataConnect service, feed-handler services run for each configured data feed. Administration of DataConnect is done in the DataConnect Console. This central tool allows an administrator to control the system and gain reports on all aspects of DataConnect.



## 1.2 Scope of the manual

The scope of this manual is to deliver information regarding software setup, the 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, DataConnect 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 DataConnect
Tip	Information that makes working with DataConnect 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 DataConnect Console
Command Window	Windows console
Feed	Datafeed coming from Bloomberg, Reuters, Trayport, etc.
Symbol	An instrument that has been added successfully to DataConnect
DataConnect	Generic term for Tradesignal DataConnect
DataConnect Console	User interface of DataConnect
DataConnect 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)

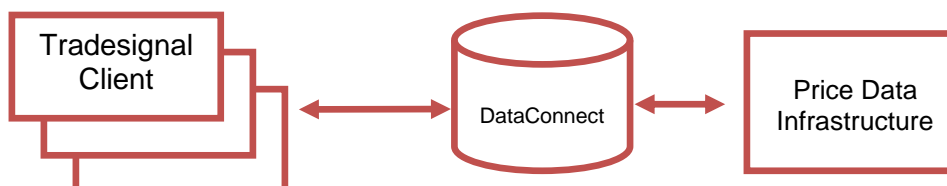
DataConnect is installed on each workstation and operates alongside the Tradesignal desktop products installed on the same computer. The advantages of this configuration include the fastest possible data retrieval rate and simplified administration for a single user at the cost of local system performance.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

*Note: When using Bloomberg or Thomson Reuters Eikon datafeeds, Scenario 1 must be deployed.*



### 2.2 Scenario 2: DataConnect on dedicated system

If a group of users need access to historical and real-time data, the installation of a dedicated system running DataConnect 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

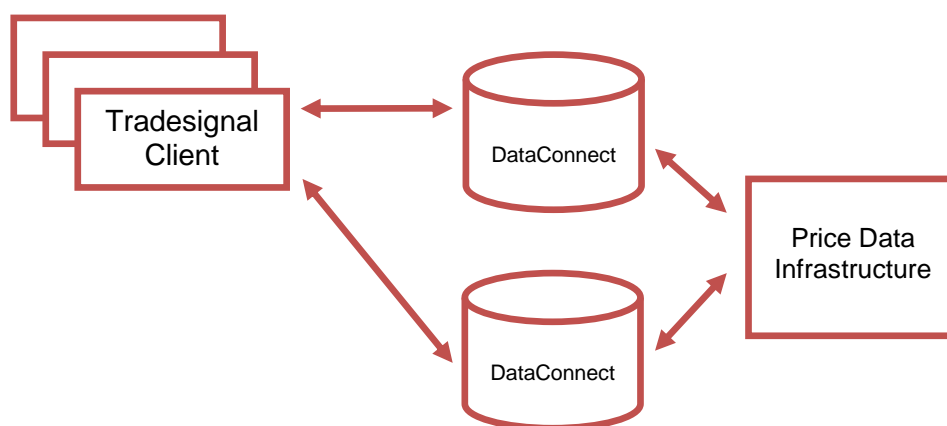
This option comes in three varieties. In each case both DataConnects must have access to the same data feeds and use the same prefixes.

*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 22).*

### 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 77.



### 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.

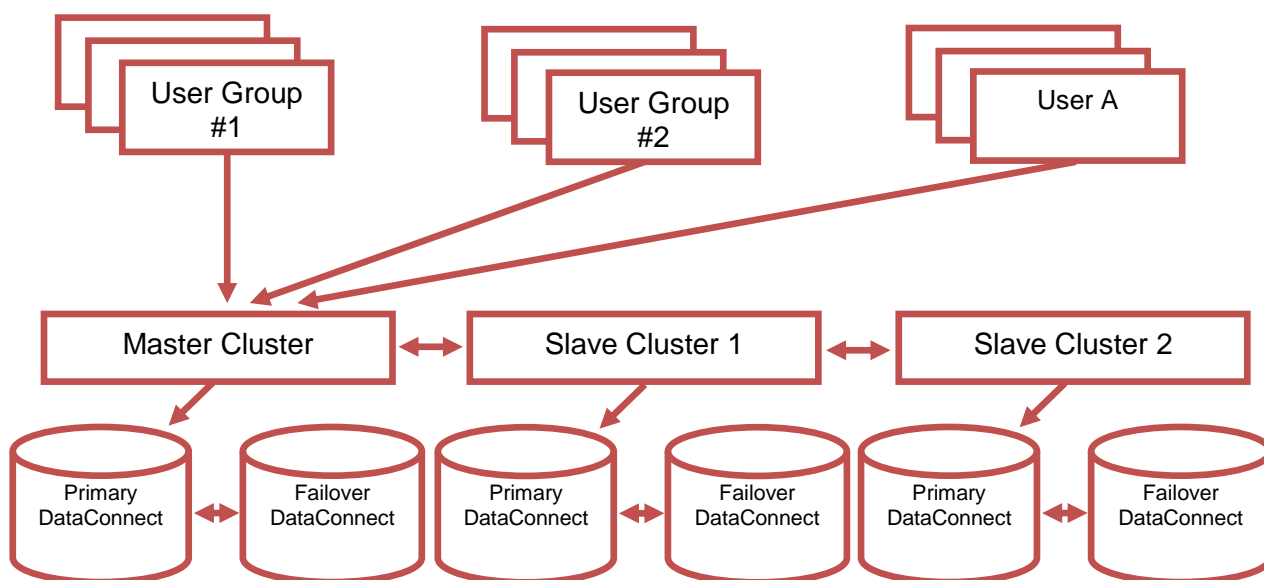
Depending on licenses, user groups, and users, users may be able to connect to any primary system or only to some of them. In principle, each DataConnect can have all types of datafeeds, so it is possible that all primary/failover pairs are configured identically. This makes synchronization (automated 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 75 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 DataConnect.*

Limitations:

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

### 2.4.1.1 Licensing of Master/Slave DataConnect 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).

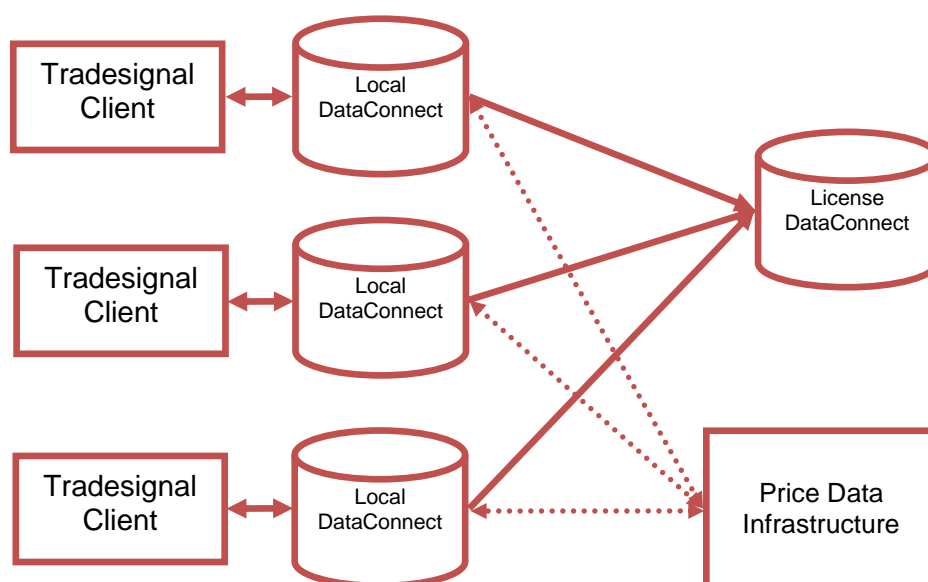
### 2.4.2 Scenario 4b: Scaling number of clients beyond scenario 3

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 DataConnect

As an administrator it may be difficult to handle the licensing of multiple local DataConnect installations throughout the company when licenses start expiring. This can be simplified by using a centralized licensing DataConnect. An example for this case would be to have several local Bloomberg DataConnect installations. In this case you can specify a centralized DataConnect as the license server. This license server will need additional remote licenses to grant feed access to all local DataConnect installations. All local DataConnect installations will be configured to use the license DataConnect instead of using their locally installed licenses (see section “Managing Licenses and Users” on page 49 for further information).



## 3 Installation

### 3.1 Preparations for installing DataConnect

- Have your installation CD-ROM ready.
- Define the DataConnect infrastructure and licensing model, see “Deployment: Common Installation Configurations” on page 9.
- 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 21.

We recommend installing DataConnect 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 DataConnect 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.*

*Note: DataConnect is designed to function in **virtualized environments** and it is strongly recommended to limit a single physical machine to running no more than two virtual machines (VMs) when one or both are hosting DataConnect. Depending on the data collection, concurrent users and instrument quantity, it may be necessary to have only a single virtual machine per physical machine. If running more than one VM on a single physical machine, it will be necessary to allocate a dedicated disk for each VM running a DataConnect to avoid performance degradation.*

#### 3.2.1 Supported operating systems

- Windows 2008 Server (64 bit) and R2
- Windows Vista (64 bit)
- Windows 7 (64 bit)
- Windows 8 (64 bit)
- Windows 8.1 (64 bit)
- Windows 2012 Server and R2

**Please note:** Starting with version 6.0, DataConnect is available as a 64bit application only.

The latest 5.x version of DataConnect can be used in cases where DataConnect must be run on a 32bit operating system (incl. Windows XP).



### 3.2.2 Minimum system requirements

- *Main processor: Pentium 4*
- *Main memory: 4 GB*
- *Free disk space: 50 GB; when installed on a non-system partition, additionally at least twice the amount of main memory free on the system (Windows) partition*

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

- *Main memory: 8 GB or more*
- *Multi-core (4+)*
- *Free disk space: 200 GB; when installed on a non-system partition, additionally at least twice the amount of main memory free on the system (Windows) partition*
- *Striped and mirrored disk array (RAID 10)*
- *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 DataConnect. For this purpose, DataConnect provides the dedicated Trayport Feed Handler, which relies on the Global Vision API (GV8API) to communicate with Trayport upstream servers. Please contact your Global Vision subscription support to obtain the latest GV8API installation package.

Please 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 DataConnect will have to be restarted after the GV8API installation).

## 3.4 Setup

1. Start the installation by running the installation file. Follow the instructions in the installer window.
2. After successful installation, Tradesignal DataConnect (as a service) and the DataConnect Console will both start up. Any problems with the configuration of the system will be shown immediately in the Logging console of the DataConnect 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 DataConnect Console, which is introduced on page 17.

- For adding feeds, see “Managing datafeeds” on page 21.
- For adding licenses and user (groups), see “Managing Licenses and Users” on page 49.
- For adding symbols, see “Managing Symbols” on page 58.

## 3.5 Updating

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

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

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

## 3.6 Uninstalling

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

1. To completely uninstall the Tradesignal DataConnect software, uninstall it and then remove the leftover files.
2. Open **Start → Control Panel → Add/Remove Programs**.
3. Choose **Tradesignal DataConnect** 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 DataConnect

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

1. Stop the old DataConnect if it is currently running.
2. Make a copy of the following files from the old DataConnect:
  - The data cache file *TMDS.DAT* (if the data cache file is called *MDSDB.FDB*, rename the copy to *TMDS.DAT*).
  - The data cache backup file *TMDS.DAT.BAK* (if the data cache backup file is called *MDSDB.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 DataConnect if it is currently running.
4. Copy the backed up files from step 2 into the installation folder of the new DataConnect, overwriting the current files if they exist.
5. If the data cache file was called *MDSDB.FDB* before renaming, open the *MDS.INI* file in a text editor and replace *MDSDB.FDB* with *TMDS.DAT* in the DataPath setting in the [General] section.
6. Restart the new DataConnect.

*Note:* If you need to copy a data cache file to a new DataConnect 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 DataConnect such as **Maximum Number of RICs** and **History Lengths** will ensure that the data is correctly described.

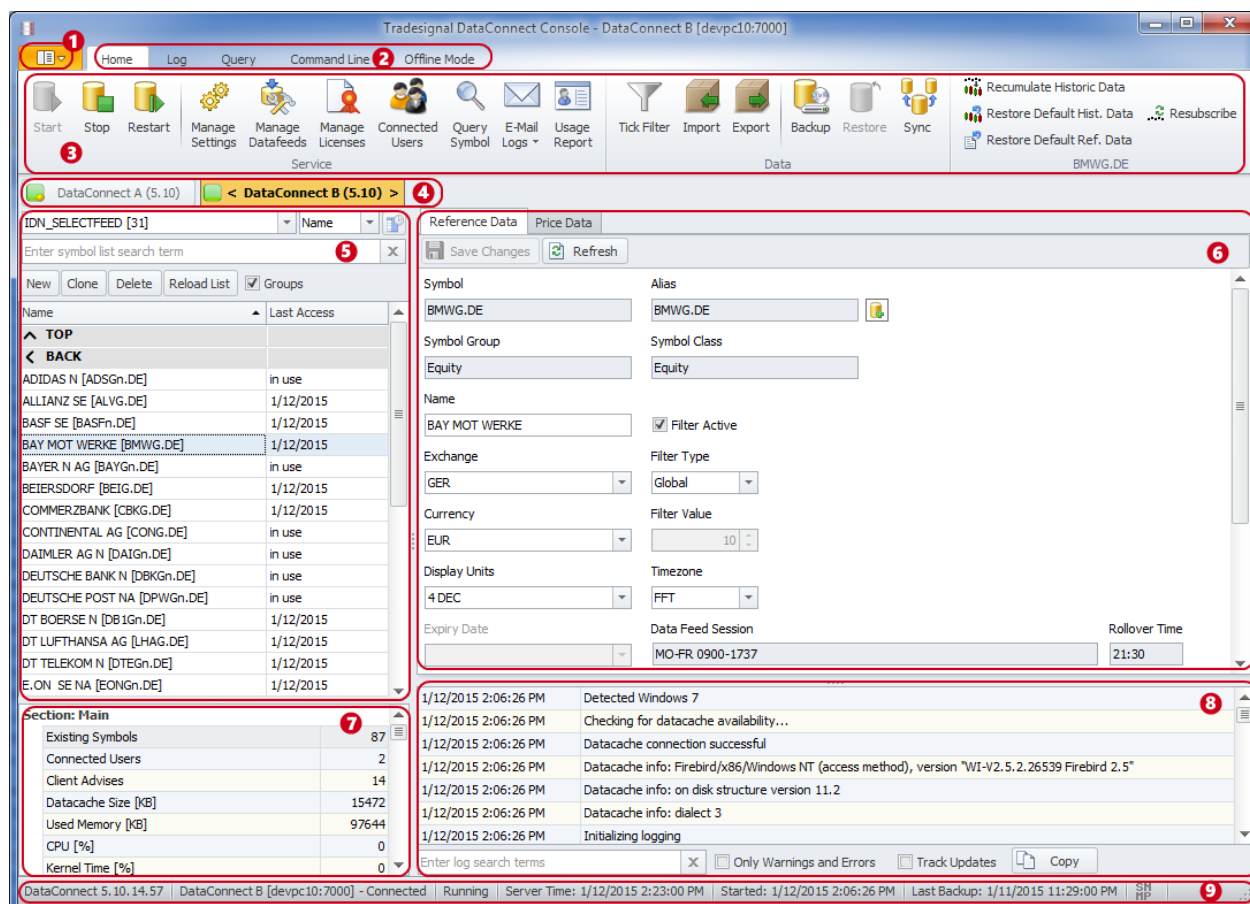
*Note:* If the data cache file is from a DataConnect version before 5, proceed as described above but be aware that the datafeed definition has changed. Start (or restart) the DataConnect Console after copying the files. It will attempt to convert the datafeed settings to the new format. DataConnect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



## 4 DataConnect Console

The Tradesignal DataConnect Console is the central user interface (UI) for the DataConnect System. Here, the connections, users, symbols, settings, etc. can be managed.

### 4.1 User interface overview

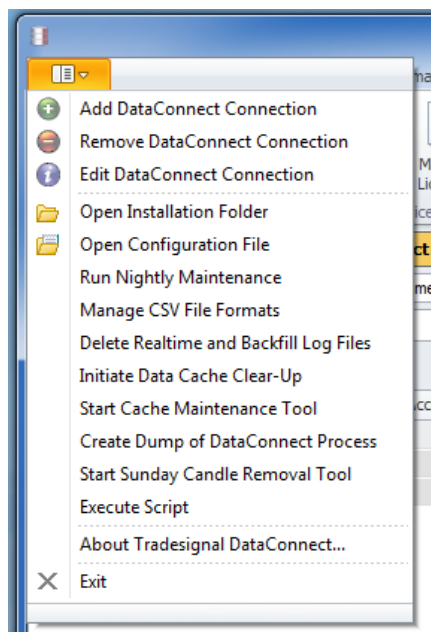


- 1: Application button, see “Application button” on page 18.
- 2: Tab bar with Home as default; for Log, Query and Command Line, see “Logs, Alerts & Troubleshooting Tools” on page 91.
- 3: Ribbon control with standard functions.
- 4: Connections to DataConnect services, see “DataConnect Connections” on page 57.
- 5: Symbol list with search filter, see “Managing Symbols” on page 58.
- 6: Per symbol: Reference data and price data, see “Managing Symbols” on page 58.
- 7: Information on various DataConnect service and computer aspects.
- 8: Logging console, can be searched/filtered for terms, see “Logging console” on page 91.
- 9: Status bar, see “Status bar” on page 19.



## 4.2 Application button

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



For connection functions, see “DataConnect Connections” on page 57.

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 96.

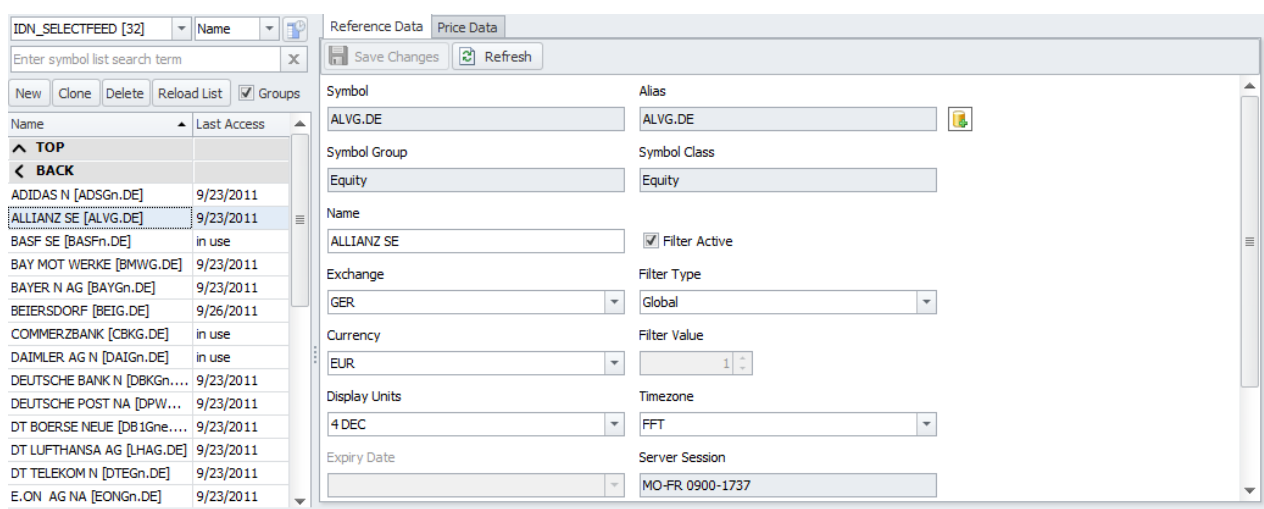
For scripting options, see “Scripting ” on page 95.

## 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 58.



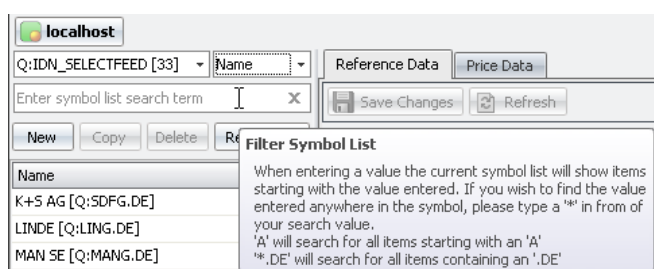
## 4.4 Status bar

DataConnect 5.4.0.15 | localhost [ingoseven2:7000] - Connected | Running | Server Time: 2/27/2012 11:14:00 AM | Started: 2/27/2012 11:11:54 AM | Last Backup: 2/23/2012 11:04:00 AM | **SNMP** | ..

A	B	C	D	E	F	G
A	Version of DataConnect software					
B	Host name, port, and state of the connection between the DataConnect Console and the DataConnect Service. Double-click to reconnect the DataConnect Console.					
C	State of the service (Starting, Running, Stopping, Stopped, Uninstalled)					
D	DataConnect time					
E	Service start time					
F	Time of the last backup					
G	Status of the SNMP Integration (Switched off, Online, Offline)					

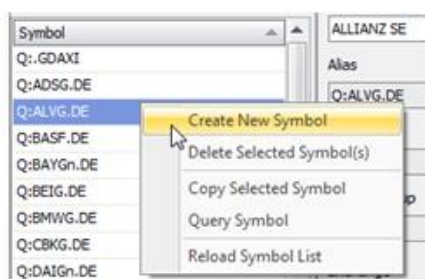
## 4.5 Mouse-over information

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.



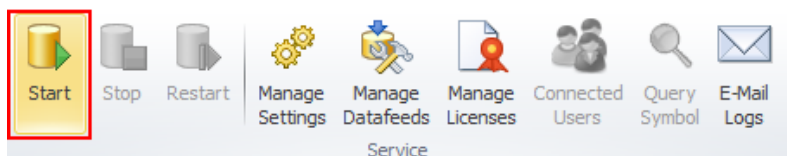
## 4.7 Field labels

-  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.

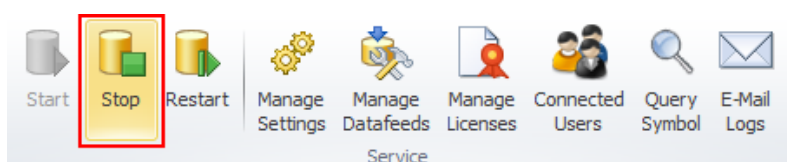


## 5 Starting / stopping the system

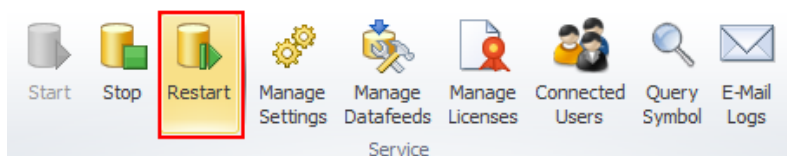
*Note: To start/stop the system, you must have administrative rights on your computer.*



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



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



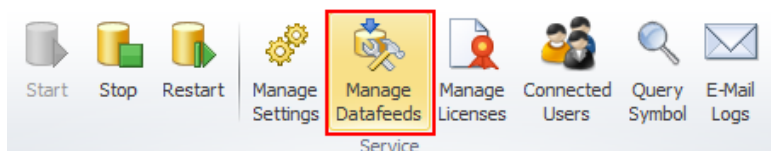
Click **Restart** to restart the DataConnect service and all feed-handler services.

This is usually done after changes in the settings and usually does not require administrative rights (may depend on operating system).

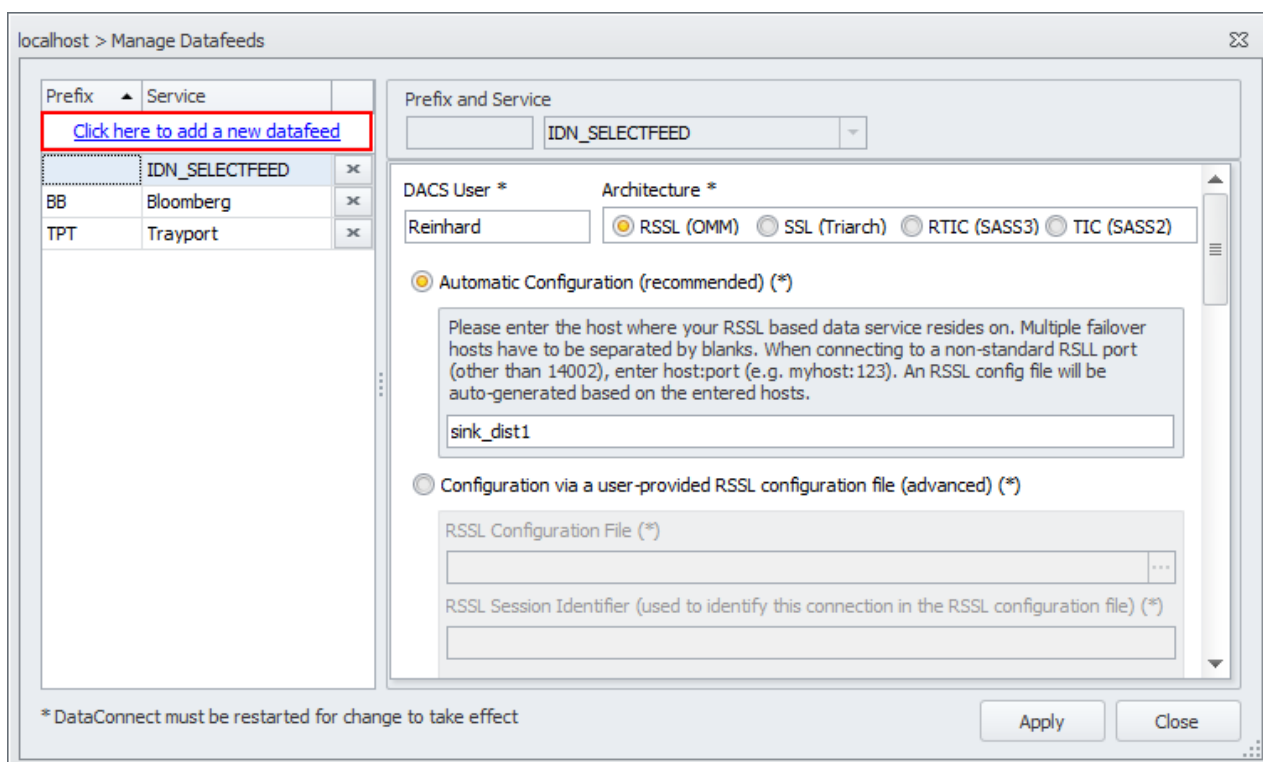
All process steps are logged and displayed in the Logging console.



## 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.

These types of datafeeds are supported:

- **Bloomberg** where the Bloomberg Professional Terminal is installed
- **Thomson Reuters** supporting RSSL/OMM, Triarch/RWS, RMDS/RTIC, Tibco TIC, Eikon and Thomson Reuters Tick History CSV data (from version 6.0 onwards)
- **Trayport** as trading gateway and for direct-broker connections
- **GlobalView**
- **CSV Data** generic CSV files in any data quality (from version 6.0 onwards)
- **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 DataConnect for Thomson Reuters Data” on page 106.

### 6.1.1 Connecting to TREP-RT (RSSL/OMM, Triarch, RMDs, TIB)

In the Manage Datafeeds Wizard, pick Thomson Reuters Enterprise Platform and enter:

- *Reuters Architecture*
- *Prefix and service*
- *DACS User which the DataConnect will use for identification at the Reuters infrastructure*
- *SSL host and port (default port is “8101”)*
- *DACS entitlement 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 RSSL (OMM) and Triarch (SSL4) installations, select **IDN\_SELECTFEED**.*
- *For RWS (SSL4) installations, select **IDN\_RWS**.*
- *For TIC (SASS2) and RTIC (SASS3) installations, select **IDN\_RDF**.*

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

localhost > Configure New Datafeed

### DACS Permissioning for Connected Users

DataConnect will perform DACS entitlement checks on behalf of connected users. In accordance to exchange rules, DACS permissioning has to be enabled for all feeds that support DACS entitlement control.

Please enter the host where your DACS daemon / OpenDACS service resides on. Multiple servers have to be separated by blanks. The default DACS connection port is already entered, you only have to adjust this value, if your desired data service uses a different setting.

DACS Daemon / Open DACS Server(s)

DACS Daemon Port

Next > Cancel

As of version 5.2, DataConnect 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 DataConnect. Enter the DACS Daemon / OpenDACS host that should be connected in order to perform entitlement checks for symbols served by the data feed. When more than one DACS Daemon / OpenDACS host is available for this feed, please enter all hosts (separated by a space character).

In the datafeed configuration, additional information can be added.

localhost > Manage Datafeeds

Prefix Service

[Click here to add a new datafeed](#)

IDN\_SELECTFEED

CSV CSVFeed

Prefix and Service

IDN\_SELECTFEED

entitlement control

DACS Daemon / OpenDACS Server(s) \*

sink\_dist1

DACS Daemon Port \*

8211

Pass Through/Persistent Mode \*

☐ Pass-through price data directly to the client without storage

☒ Store historic prices persistently in DataConnects data cache

The following settings are shared by all Reuters datafeed connections:

☒ Backfill securities with data from CSV feed (Securities in both feeds must have same symbol and FID names)

Recovery host

\* DataConnect must be restarted for change to take effect

Apply Close

To use a recovery host, see “Scenario 3: Two systems” on page 10.

#### 6.1.1.1 Obtaining historic inter-day and intraday data via CSV Feed

Thomson Reuters data can be backfilled using data from any of the sources configured as a CSV feed. To enable this feature, check the “Backfill securities with data from CSV Feed” option. This also allow backfilling data from Thomson Reuters Tick History files (See section 6.6 for more information on how to set up TRTH source files).

Important: The Symbol and FID in the CSV feed must exactly match the Thomson Reuters RIC and FID. This is more-or-less guaranteed when using the Thomson Reuters Tick History format.

#### 6.1.2 Connecting to Eikon

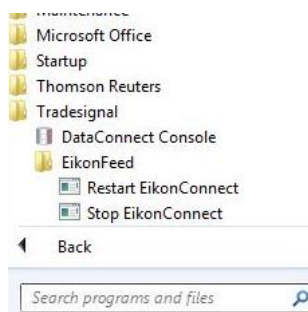
The following applications are required to be installed on the same local machine as DataConnect to be able to access Thomson Reuters Eikon data:

- Thomson Reuters Eikon
- Optional (for Eikon 3.x and earlier): Microsoft Excel 2003 or above with activated Thomson Reuters Eikon Excel Add-In

For Eikon version 3.x and earlier, the Excel add-in is necessary for historic data retrieval as Eikon does not offer a dedicated API for direct history downloading.



To access the Eikon realtime data API, Thomson Reuters requires that an application has to run under a normal user account (usually the same account that runs the Eikon application). Unlike other DataConnect extension services (e.g. Trayport, Bloomberg, GlobalView) that run as Windows services under the LocalSystem account, the Eikon feed handler process must therefore be run as a non-service application. In order to provide a similar level of reliability in terms of exceptional situations, the Eikon feed handler comes with a separate safeguard program that restarts the Eikon feed handler when it is unexpectedly terminated. The DataConnect setup will also install shortcuts to the Windows Program menu that allow fast access to restart or shutdown the Eikon feed handler process manually.



In order to configure an Eikon data feed for DataConnect, pick *Thomson Reuters Eikon* in the Manage Datafeeds Wizard and enter:

- *Prefix (only configurable when a default feed with empty prefix is already set up)*
- *Default data source (e.g. "IDN", "IDN\_RDF" or "IDN\_RDF;IDN")*

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. The suggested setup when using the Eikon feed is to configure Eikon as the main feed (with no prefix). Additional Eikon sub-feeds can then be configured with dedicated prefixes in the Manage Datafeeds dialog (see below).

**Breaking Change:** When using versions of Eikon after 4.0.26424, it is very likely the data source feed name will need to be changed to: *IDN\_RDF;IDN*

### 6.1.2.1 Advanced Eikon data feed configuration

When necessary, additional configuration options can be set once the Eikon data feed has been added to the feed configuration. Select the *EIKON* entry in the Manage Datafeeds Dialog to display the specialized configuration pane for the Eikon data feed. The remainder of this chapter describes the available configuration options.



### Section: General

Push/Backfill port	The TCP/IP ports used by DataConnect to talk to the Eikon feed handler service. Can be changed when there are conflicts with existing applications.
Daily session reload time	The daily local system time when trading session information should be updated.
Expired contracts chain years	Security lists containing expired futures contracts will be filled with contracts reaching back this amount of years. Note that Thomson Reuters stops providing historic data for older contracts so this can be used to reduce the number of "empty" contracts in expired symbol lists.

### Section: Pass Through / Persistent Mode

Pass-through price data directly to the client without storage	DataConnect does not store historic prices in its data cache in this mode.
Store historic prices persistently in DataConnect's data cache	DataConnect stores historic prices in its data cache in this mode (which is the default mode).

Note: See section 9.6 (Pass-Through Mode vs. Persistent Mode) for more information on this topic.

### Section: Thomson Reuters Eikon Feeds

In order to access multiple Eikon sub-feeds, every such feed has to be assigned a unique prefix. This can be achieved by adding data sources in the Eikon Feeds section. In the above screenshot, the default feed was configured as *IDN\_RDF* meaning symbols without a prefix will be assigned to this feed. Additionally, two sub-feeds named *MARKETLINK* (with alias *M*) and *USER\_FEED* (with alias *U*) have been set up. For example, the symbol *ABC* would then be mapped to *IDN\_RDF*, the symbol *M:ABC* to *MARKETLINK* and *U:ABC* to *USER\_FEED*. Note that in order to access sub-feeds it is required to configure the Eikon data feed as the default feed in DataConnect (having no feed prefix assigned).



### **Section: Advanced Logging**

Record Creation	Logs creation/deletion information on symbol subscriptions.
Record Clients	Logs client information on symbol subscriptions
Record Updates	Logs every realtime field update for a symbol subscription
Backfill	Logs information on historic data forwarding
Communication	Logs requests sent by DataConnect to the OpenConnect service hosting the Eikon feed adapter module

Note: All Logging will be written to the OpenConnect.log file for the Eikon feed handler.

### **Section: Excel**

Pool Size	The number of independent Excel processes to utilize for historic data retrieval.
Download Retries	Specifies how many times a failed Excel history download should be retried before giving up.
Download Timeout	Specifies the time in seconds before an outstanding Excel history download gets cancelled.
Maximum Total History Length	Excel restricts the number of data points that can be retrieved per request. Therefore downloading very long histories may take more time as multiple requests have to be issued. This setting allows restricting the number of bars to be retrieved in favor of quicker execution.
Visible Excel Instances	Normally, all Excel instance windows are hidden. This setting allows making them visible.

### **Section: Periodic History Refresh**

DataConnect can be instructed to reload recent historic data at a specified local time. This helps correcting local histories that may have changed on the upstream server side (e.g. through price corrections). The reload should ideally be scheduled during times of low activity to avoid interfering with other tasks. The number of bars to be updated can be configured on a per-period basis. Note that reloading ticks can take a comparatively long time to complete, depending on the number of symbols with active tick collection, the type of symbols, and the setting how many hours of tick data have to be refreshed.

### **Section: Expired Contract History Transfer**

Please refer to section 9.7 (Automatic copying of historic data for Thomson Reuters Expired Contracts symbols) for more information on this topic.



## 6.2 Setting up Trayport GlobalVision

Please see. section 3.3.1 for specific software requirements in order to access Trayport prices. The pre-configured prefix for symbols connected to the Trayport data feed is `TPT:`.

In the Wizard, enter:

- **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 DataConnect system must have the Microsoft SQL Server ODBC driver installed on it. This is normally installed with the SQL Server Client Components setup.*

DataConnect 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). DataConnect 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 DataConnect is able to connect to the SQL Server directly, follow these steps to obtain historic deals.

1. Ensure the server is reachable from DataConnect by typing ping <servername> into the command prompt.
2. Ensure that the Microsoft SQL Server ODBC driver is installed on the DataConnect 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 - DealTreeNodees
- 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 DataConnect and query a symbol.**

**Please note:** Even with this mode active, DataConnect will still attempt to get the last 24 hours' worth of trades from the API instead of the database, since in most installations the database is only updated once per day for performance reasons. This time period can be modified or disabled in the Trayport settings in the DataConnect Console.

## 6.2.2 Rolling

DataConnect 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. DataConnect 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.



### 6.2.3 Mapping Trayport Symbol Identifiers

Using DataConnect version 5.5 and below, the only way to assign user-defined identifiers to Trayport symbols is setting up explicit one-to-one mapping entries in the [SymbolMap] section of the *TMDSTRPT.INI* file, e.g. GERBSLDMONJAN13=TP10001126.10000104.109  
While this approach may be viable for many use cases, it has a number of drawbacks:

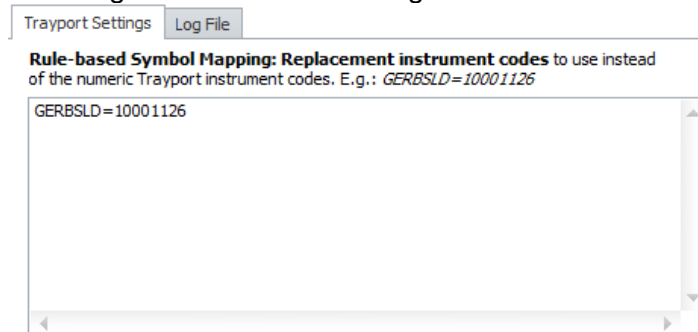
- One alias entry has to be created for each instrument/sequence/sequence item triple (where one sequence can contain possibly hundreds of sequence items)
- When new sequence items are added by the upstream Trayport service, the [SymbolMap] section has to be manually updated with new aliases
- Symbols for server-side continuous contracts (UDCs) cannot be mapped using this approach

Additionally to the above, DataConnect is therefore offering rule-based symbol mapping to remedy these deficiencies starting with version 5.6.

The basic idea is to split the alias in three parts, one for the instrument (e.g. Euro Power Germany Baseload), one for the sequence (e.g. Months), and one for the sequence item (e.g. January 2013). These three parts concatenated then form the specific alias. The instrument and sequence alias parts are defined using a one-to-one mapping (e.g. GERBSLD maps to Trayport instrument identifier 10001126 and MON maps to the Trayport sequence identifier 10000104). The sequence item alias part is specially mapped using a rule that is applied to the sequence item information provided by Trayport. This approach involves picking certain parts from a sequence item's name and/or its expiry date to form the sequence item alias.

Example: Symbols for the Euro Power Germany Baseload (instrument id 10001126) should have a common symbol alias starting with GERBSLD, monthly contracts should be denoted with MON and items from the German Baseload monthly sequence should follow the pattern MMMYY (e.g. JAN13). The concatenated alias for the January 2013 monthly Germany Baseload contract would then be named GERBSLDMONJAN13. Seasonal contracts should be denoted with the alias SEASONS, and contained sequence items should be accessible using the season abbreviation from Trayport's provided sequence item information (e.g. Win, Sum) followed by a dash and the four digit year indicator. The complete alias for Germany Baseload Winter season 2013 would then be GERBSLDSEASONSWIN-2013. If necessary, DataConnect allows delimiting the three sub-parts with a separator character (e.g. the dot, in which case the aliases would be named GERBSLD.MON.JAN13 and GERBSLD.SEASONS.WIN-2013).

The following settings would generate such a naming scheme:



The screenshot shows a window titled "Trayport Settings" with a "Log File" button. Below the title bar, there is a section titled "Rule-based Symbol Mapping: Replacement instrument codes to use instead of the numeric Trayport instrument codes. E.g.: GERBSLD=10001126". A text area below this title contains the entry "GERBSLD=10001126".



**Rule-based Symbol Mapping: Replacement sequence / sequence item rules to**  
use instead of the numeric Trayport sequence and sequence item codes.  
Please see the documentation for detailed information.

M=10000104	.	*	[MMM][YY]
SEASONS=10000107	(?<season>[a-z]{3})\s[0-9]{2}		<season>-[YYYY]

**Separator character to be applied for rule-based symbol mapping.**

The separator delimits the instrument, sequence, and sequence item parts, e.g.  
*COAL-MONTHS-JAN13* (minus character).

The separator can also be left empty, e.g. *COALMONTHSJAN13*.

The sequence item rule consists of a regular expression (regex) and an output pattern. The output pattern defines which pieces of information are used to generate the sequence item alias. The regular expression can be used to extract arbitrary parts from the sequence item name as published by the Trayport upstream source. The DataConnect Trayport Feed Handler will store the provided instrument and sequence information in Xml format in C:\Documents and Settings\LocalService\AppData\Tradesignal\TMDSTRPT, the *sequenceitems.xml* file will contain sequence item information (ItemName and TradingEnd Xml attributes specifically).

In order to use parts from the sequence item name, so-called named sub-expressions have to be used in the regex. Named sub-expressions have the form `(?<name>exp)`. The extracted text can then be referenced by `<name>` in the output pattern. In case the item name shall not be used to form the item alias, `. *` should be specified as the regex pattern.

The output pattern can contain other building blocks like date parts picked from a sequence item's expiry date. These blocks share the common form `[name]`. The following table shows all predefined output pattern blocks:

Pattern	Description	Resulting Replacement Text
[DD]	Numeric, two digit zero prefixed calendar day	01, 02, 03, ... , 31
[WWW]	Abbreviated English three letter weekday	MON, TUE, WED, THU, FRI, SAT, SUN
[MM]	Numeric, two digit zero prefixed calendar month	01, 02, 03, 04, 05, 06 07, 08, 09, 10, 11, 12
[MMM]	Abbreviated English three letter month	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
[YY]	Numeric, two digit calendar year	99, 13, 20
[YYYY]	Numeric, four digit calendar year	1999, 2013, 2020
[I]	Numeric Trayport sequence item ID	109
[N]	Trayport sequence item name	Jan-13

All expiry date based patterns [DD], [WWW], [MM], [MMM], [YY], [YYYY] additionally allow an offset of a number of days in the form `+/-nD` to be applied to the sequence item expiry information. For example, the expiry date of a yearly contracted may be defined as January 1<sup>st</sup>



the following year. The rule [YYYY] would then yield e.g. 2014 instead of 2013 for a 2013 yearly contract. Using [YYYY-1D] would first deduct one day from the listed expiry date before generating the alias.

An output string can contain additional verbatim text that will be copied to the alias. All text that is not of the form <...> or [...] will be taken as verbatim text, e.g. the pattern "[MMM]\_OF\_[YYYY]" might generate the sequence item alias "FEB\_OF\_2013". Note that spaces are not allowed in alias names.

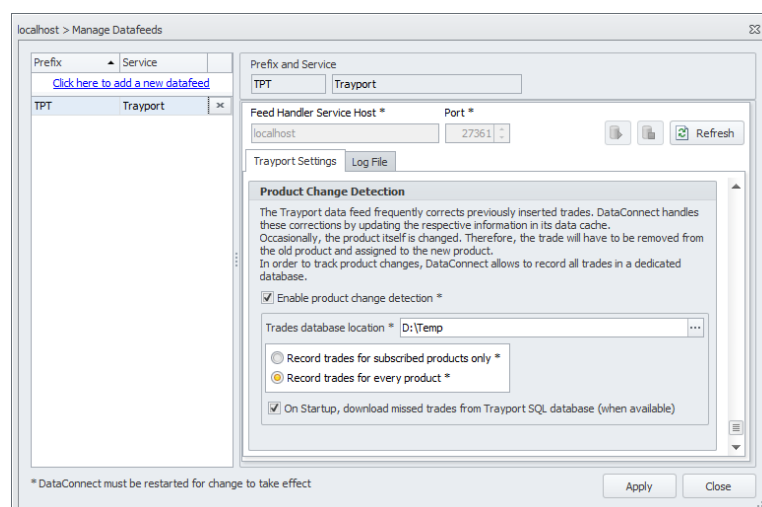
Note that [N] will map to the sequence item name with all spaces removed (e.g. "Wk 1-10 (WD)" becomes "WK1-10(WD)"). Alternatively, spaces can be replaced with any character using [Nc] where c denotes the space replacement character (e.g. using [N\_] would yield "WK\_1-10\_(WD)").

## 6.2.4 Product change detection for Trayport trade updates

The Trayport data feed has a means to publish updates to already committed orders and trades as corrections that are then picked up by DataConnect which in turn corrects affected stored prices. A trade correction can be either of type "update" or "delete". A deal update normally involves modifying the price or size of a deal, but Trayport also allows a modification of the underlying product that was traded.

In this regard, the Trayport system behaves somewhat idiosyncratic in that it does not send out one update message involving the new product and one deletion message for the trade in the incorrect product. Only the update message is being generated. Consequently, product change modifications become notoriously hard to handle by feed handlers like DataConnect as this involves locating the product that was previously associated with a trade. Not removing such mistrades may result in unwanted spikes in the history of a traded product.

DataConnect version 5.7 and above implement a strategy to overcome this problem in an efficient way. This involves recording all deals for every product supported by the connected trading gateway in a dedicated database which is then being consulted when a trade update message is received in order to find out if a product change has occurred. In case the old product is collected by DataConnect, the affected trade is removed from the history of the mistraded product, thereby guaranteeing consistent trade data for both products.



Recording every trade dealt in a Trayport trading gateway is not expected to affect performance noticeably due to the comparatively low number of updates in the provided energy markets. Still the DataConnect Console allows to reduce the load in different ways should this become necessary. The product change detection settings are available in the Trayport section of the Manage Datafeeds dialog.

To disable the deal database for product change detection altogether, uncheck the “enable product change detection” tick box.

The number of trades recorded can be minimized by only subscribing to the products collected by DataConnect. Note that this great reduction of the number of feed messages to handle is paid for by the possibility of undetectable trade product changes (those where the new product is none of the collected items by DataConnect). Therefore it is recommended to use this feature only if the sheer number of update messages is too high to be handled without adversely affecting performance.

For best performance, the trades database ought to be located on a physically different hard drive than the DataConnect data cache file. The storage location can be changed in the Trades database location folder picker. Note that folders on a network share are not suitable for the trades database.

The availability of a Trayport SQL server allows getting hold of all recent trade/product combinations. This deal download is enabled by default, if such an SQL database is configured (see section 6.2.1, Backfilling data history from Trayport SQL server). On every startup, the Trayport feed handler will try to retrieve all deals missed since it was operative last. If necessary, accessing the Trayport SQL database for this reason can be inhibited by unchecking the corresponding tick box.

### 6.2.5 Smart Match Support

Trayport offers SmartMatch as a tool allowing the creation of single tradable instruments that can support multiple execution venues (i.e. Bilateral, Cleared). Traditionally, support for instruments that have multiple execution venues has been achieved by creating separate instruments in the Trayport broker platforms.

By using SmartMatch, brokers and traders can make use of a single instrument and point of liquidity that provides the ability to trade agnostic to execution type (‘Smart’ orders), or by specifying an explicit execution order type.

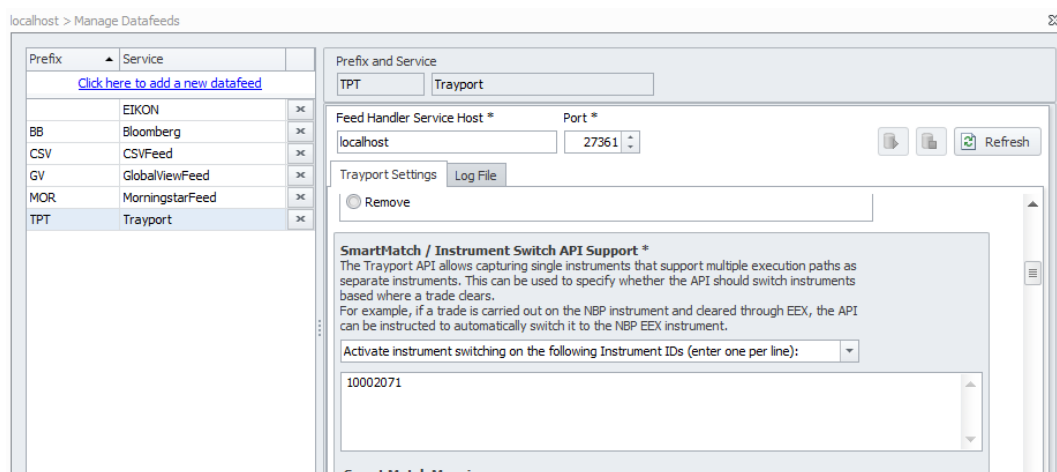
As SmartMatch will combine multiple instruments to a single Bilateral/Cleared instrument, trades that used to flow down the API on separate InstrumentID’s will now come down using a single InstrumentID.

The DataConnect Trayport feed handler integration can take advantage of the new Trayport Instrument Switch feature of the API which will correctly send trades to the API on a separate instrument. Trayport offers this feature starting with the GlobalVision API 8.9.2. Please contact Trayport support on more information regarding SmartMatch.

The DataConnect console allows enabling instrument switching on a per instrument ID basis directly using the GlobalVision API function SetAPIOptions(). Trayport also allows instrument



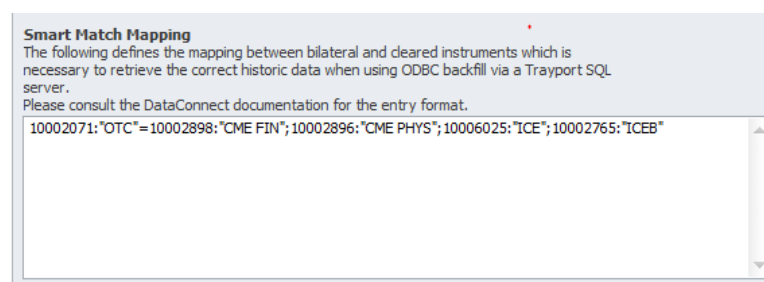
switching configuration via certain registry settings, however in order to rule out misconfiguration or unexpected side effects, it is recommended to use the DataConnect Console as the only place to configure Trayport instrument switching.



The Smart Match configuration is found in the Trayport section of the Manage Datafeeds dialog. Three options are available:

- *Do not switch any instrument automatically.* This option will not enable instrument switching at all, so trades will always be sent to the original instrument ID.
- *Activate instrument switching on the following Instrument IDs.* This will activate instrument switching via the GlobalVision API for all instruments having the IDs listed in the text box below the drop down field. Please enter one ID per line. In the screenshot above, the instrument ID of NBP (10002071) was configured, meaning that Trades dealt using SmartMatch Instrument Execution on NBP may be switched to the respective NBP instrument based on execution venue (e.g. NBP ICE, 10006035 or NBP EEX, 10002317).
- *Activate instrument switching on all instruments.* This will activate instrument switching via the GlobalVision API for all instruments provided by Trayport that support SmartMatch.

**Important:** When using SmartMatch in conjunction with the ODBC backfill option (retrieving historic data from an SQL database server, see section 6.2.1), it is important to perform additional configuration in order to guarantee that downloaded deals will be sorted into the correct instruments. The Console offers the Smart Match Mapping table for this purpose, found below the instrument switch configuration.



There is already one mapping present by default to define the correct instrument and venue mapping for the NBP market. Note that all configured entries will only then be used when the respective bilateral instrument was set up in the instrument switch configuration (see above).

Every line in the Smart Match Mapping textbox defines the relationship between a bilateral instrument and all respective cleared ones based on execution venue. The entered information will be used to correctly assemble the database request so that deals are sorted into the correct instruments.

A mapping entry starts with the bilateral instrument id, followed by a colon and a quoted comma-separated list of execution venue names. The other part of the mapping is a semi-colon separated list of all cleared instrument definitions (using the same syntax, i.e. ID:"VENUE1","VENUE2",...). Both parts of the definition are separated by the equals sign ('=').



## 6.3 Setting up Bloomberg

The pre-configured prefix for symbols connected to the Bloomberg data feed is **BB** :.

Once a Bloomberg feed has been configured via the New Datafeed Wizard, DataConnect will attempt to connect to the upstream service using on the configured access modes after a restart (see below).

### 6.3.1 Available upstream connection options for Bloomberg

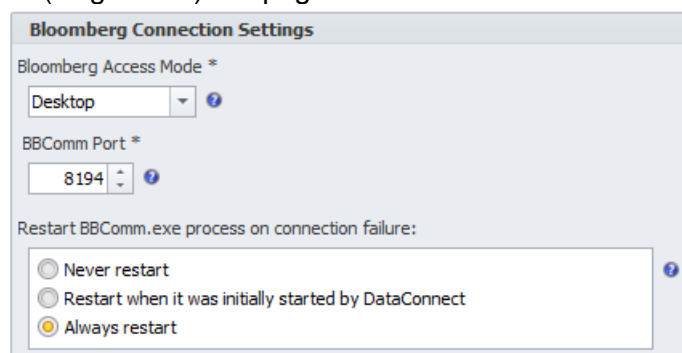
DataConnect supports connecting to the Bloomberg data feed via the Desktop API, B-Pipe, or the Server API.

#### 6.3.1.1 Desktop-API (via BBComm)

When using the Bloomberg Desktop API, Tradesignal and DataConnect must run on the same physical machine. DataConnect can be installed on any computer that is running a Bloomberg Professional Terminal. DataConnect will use the BBComm Version 3.0 Desktop API to obtain data.

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

*Note:* If using the Bloomberg Desktop API with DataConnect, the Tradesignal desktop application has to be installed on the same computer, see “Scenario 1: Local installation alongside Tradesignal (single user)” on page 9.



Chose “Desktop” in the “Bloomberg Access Mode” dropdown to activate this mode. If necessary, adapt the port used to connect to the BBComm host process. If wanted, BBComm can also be restarted when DataConnect is being restarted.

#### 6.3.1.2 B-Pipe

Bloomberg B-Pipe allows multiple Tradesignal clients to connect to a central DataConnect which communicates with the so-called B-Pipe appliance residing on a dedicated host.

User entitlements will be managed in the Bloomberg Terminal's EMRS function by the firm's EMRS Administrator. Every user is assigned an EMRS Name that is associated with all exchanges that the user is allowed to access. Depending on the EMRS configuration, a Bloomberg Terminal may need to be installed on the Tradesignal user's machine.

DataConnect has to be set up in EMRS as type "Application" in order to be able to authorize all Tradesignal user data requests via a user's EMRS Id and IP address.

Tradesignal users can enter their EMRS Id in Tradesignal in the Datafeeds options.



Click “Show more options” to display extended configuration options and enter the EMRS Id in the text field labeled “EMRSId (B-Pipe) / UUID (Server API)”.

**Datafeed** Tradesignal DataConnect

Username bob

Password Optional

Master Server localhost 7000

Failover Server Optional 7000

Slave Servers Semi-colon separated list

Slave Failover Servers Semi-colon separated list

[Show less options](#)

General Thomson Reuters **Bloomberg** Trayport

☐ Use Bloomberg function keys

EMRS Id (B-Pipe) / UUID (Server API): MYDOMAIN\bob

Chose “B-Pipe” in the “Bloomberg Access Mode” dropdown to activate this mode. Enter the EMRS Application Name that was configured by your EMRS administrator and the host name or IP address of your network’s B-Pipe appliance in the Bloomberg section of the Manage Datafeeds dialog.

**Bloomberg Connection Settings**

Bloomberg Access Mode \*  
B-Pipe

EMRS Application Name for Tradesignal DataConnect \*  
Tradesignal:DataConnect

B-Pipe Host Name(s) \*  
192.168.1.91

### 6.3.1.3 Server API

Server API allows Tradesignal clients to connect to a DataConnect which communicates with the so-called Server Process residing on a dedicated host. DataConnect authorizes all client data requests using the client’s UUID provided by Bloomberg in conjunction with the IP address. Tradesignal users can enter their UUID in Tradesignal in the Datafeeds options. Click “Show more options” to display extended configuration options and enter the UUID in the text field labeled “EMRSId (B-Pipe) / UUID (Server API)”.

**Bloomberg Connection Settings**

Bloomberg Access Mode \*  
Server

Server Process Host Name(s) \*  
192.168.1.91

Choose “Server” in the “Bloomberg Access Mode” dropdown to activate this mode.



Enter the host name or IP address of your network's Server API Process host in the Bloomberg section of the Manage Datafeeds dialog.

### 6.3.2 Enabling data collection for supplemental fields

Depending on the security type, Bloomberg symbols usually provide a subset of the following fields for historic data collection and realtime data provision: Last price with Volume (LAST/VOL), Best Bid prices with size (BID/BVOL), Best Ask prices with size (ASK/AVOL), and Open Interest (OI). The names in brackets denote the so-called *FIDs* (field identifiers) that are used by Tradesignal to make the data available for charting etc.

Beginning with v5.7, DataConnect allows configuring virtually any field supplied by Bloomberg (as long as it describes numeric data) to be exposed as additional FIDs. Prerequisite for enabling any additional FID for data collection is the knowledge of the underlying data fields supplied by the upstream feed. The Bloomberg API draws a strict distinction between fields available for realtime push updates (called *Realtime Fields*), fields that have to be snapped in regular intervals to update their latest values (called *Static Fields*) and fields available for retrieving a series of past (daily) values (*Static History Fields*). One way to find out the Bloomberg field names needed to define additional FIDs is the *FLDS* function provided by the Bloomberg Terminal, which allows browsing all available fields and current values for a security, filtered by field type (realtime, static, historical).

Supplemental fields can be configured in two dedicated sections in the Bloomberg feed handler configuration file, *TMDSBB.INI*, residing in the DataConnect executable directory (note that administrator rights may be necessary to modify this file). When the goal is to display latest field values in Tradesignal watch lists only, an entry in the [QuoteFields] section is sufficient. In order to expose a field as a chartable FID, additionally an entry has to be set up in the [UserDefinedFIDs] section.



### 6.3.2.1 Section [QuoteFields]

This section defines the mapping between a quote field (as available in a Tradesignal watchlist column) and the underlying realtime or static field provided by the Bloomberg API. Each line has to follow this syntax:

```
QuoteField = Id | DataType | Name | Description | BloombergField | UpdateMethod
```

Item	Description	Example(s)
QuoteField	The quote field short name as to be published by DataConnect. Quote field names are required to be unique.	BB_ASSET_SWAP_SPD_ASK
Id	A unique numerical identifier of the field to add (in the range 210000 - 210100).	210000
DataType	The data type of the field; one of <i>double</i> , <i>int</i> , or <i>string</i> . Note that string fields cannot be used for data collection in [UserDefinedFids] (see below).	double
Name	The (short) name of the field (this will appear in Tradesignal's watchlist column picker).	Asset Swap SPD ASK
Description	An optional description of the field.	Ask Asset Swap Spread
BloombergField	The native Bloomberg realtime or static field name that will be used to generate realtime updates.	ASSET_SWAP_SPD_ASK
UpdateMethod	Defines how realtime updates will be triggered for this field. When the field is fed from a Bloomberg realtime (push) field, specify <i>R</i> . For a static (pull) Bloomberg field, specify <i>P</i> . The feed handler will have to periodically snap the field to simulate realtime behaviour. The update interval has to be given in minutes. Alternatively, one or two daily snap times can be specified in the format "hh:mm" (UTC).	R P 5 P 15:00 P 10:00 20:00

Examples (note that the line breaks printed here for clarity are not supported in the actual INI file):

```
BB_ASSET_SWAP_SPD_ASK=210000|double|Asset Swap SPD ASK|
Ask Asset Swap Spread|ASSET_SWAP_SPD_ASK|P 5
```

This would expose a new floating point double precision quote field called "BB\_ASSET\_SWAP\_SPD\_ASK" which is automatically updated every 5 minutes by retrieving the underlying static Bloomberg field named "ASSET\_SWAP\_SPD\_ASK".

```
BB_VWAP=210001|double|VWAP|Volume Weighted Average Price|VWAP|R
BB_NUM_TRADES=210002|double|Num Trades|Number of Trades|NUM_TRADES_RT|R
```

This would expose two new floating point double precision quote fields called "BB\_VWAP" and "BB\_NUM\_TRADES" which update in realtime, fed by the Bloomberg realtime fields "VWAP" and "NUM\_TRADES\_RT", respectively.

```
BB_DAYS_TO_EXPIRY=210003|int|Days Till Expiry|Days Until Contract Expiration Date|
FUT_ACT_DAYS_EXP|P 06:00
```



This would expose a new integer quote field called “BB\_DAYS\_TO\_EXPIRY” which is updated daily at 6 am (UTC) by snapping the current value of the static Bloomberg field named “FUT\_ACT\_DAYS\_EXP”.

```
BB_EXCH_STATUS=210004|string|Exchange Market Status|
Exchange specific representation of the trading status of the security|
EXCH_MARKET_STATUS|P 08:00 16:00
```

### 6.3.2.2 Section [UserDefinedFIDs]

Precondition for defining a new user-defined FID is an entry in the [QuoteFields] section described in paragraph 0.

The syntax for defining a new FID is as follows:

```
FID=RegisteredRealtimeField BloombergHistoryField InstrumentPattern
```

Item	Description	Example
FID	The name of the user-defined FID as it shall appear in DataConnect and Tradesignal	ASWPASK
RegisteredRealtimeField	The field which feeds realtime updates for this FID (must be one of the Fields registered in the [QuoteFields] section)	BB_ASSET_SWAP_SPD_ASK
BloombergHistoryField	The name of the Bloomberg static history field that feeds historic (daily) data; or '-' if historic backfill is unavailable. Note that Bloomberg does not provide intraday data for supplemental fields.	ASSET_SWAP_SPD_ASK
InstrumentPattern	A Regular Expression (Regex) which defines the symbols that the defined FID will be added to.	.* Corp

Examples:

```
ASWPASK=BB_ASSET_SWAP_SPD_ASK ASSET_SWAP_SPD_ASK .* Corp
ASWPBID=BB_ASSET_SWAP_SPD_BID ASSET_SWAP_SPD_BID .* Corp
ASWPMID=BB_ASSET_SWAP_SPD_MID ASSET_SWAP_SPD_MID .* Corp
```

Defines three user-defined FIDs ASWPASK, ASWPBID, ASWPMID for all symbols of the Corporate Bond Bloomberg security category. Notice that the regular expression “.\* Corp” matches all symbols with the “Corp” appendix.

```
VWAP=BB_VWAP EQY_WEIGHTED_AVG_PX .* Equity
```

This entry defines the additional FID VWAP available for all Equity symbols.

```
TRDNUM=BB_NUM_TRADES - .*
```

This defines the additional FID TRDNUM to be available for all symbols. Note the minus character which states that this FID does not support the retrieval of historic (daily) data for this FID.



## 6.4 Setting up GlobalView

DataConnect supports the GlobalView energy data market feed.

The pre-configured prefix for symbols connected to the GlobalView data feed is `GV:`.

In the Wizard, enter:

- ***User name and password of the GlobalView user account***

Once the feed is added, additional options, e.g. regarding logging, can be configured in the GlobalView pane of the Manage Datafeeds dialog.

Click the Log tab to access the log produced by the GlobalVision feed handler component.



## 6.5 Setting up a CSV Feed

CSV Feed is part of the DataConnect suite (*from version 6.0 onwards*) and can be installed by selecting the respective option in the DataConnect setup. This data feed scans one or more specified file locations containing price data definition files in various pre-defined or configurable formats, making the data available to Tradesignal clients. Files are periodically being rescanned to propagate possible updates.

### Steps to set up a CSV Feed

1. To set up CSV Feed after its installation, start by selecting *Manage Datafeeds* on the DataConnect console's toolbar. Next click on the **Click here to add a new datafeed** link to start the wizard. Now select the **CSV Feed** option in the wizard. In the next page of the wizard, a three-letter Prefix can be given to the CSV Feed, after completing the wizard the newly added feed would be visible in the table on the left hand side of the *Manage Datafeed* window.
2. Select this entry to start configuring the data feed (The CSV Feed settings panel will be visible on the right hand side of the window). To add source files to the CSV Feed, the folder(s) containing source files should be added with appropriate file formats assigned to those folders.
3. A new folder can be added by clicking the **Add new Source Data Folder** button.
4. After selecting the Folder, the correct csv file format has to be assigned to it; this can be done by selecting the appropriate entry from the *CSV File Format* dropdown. If the file format for the folder is not already present in the dropdown, a new CSV File format can be created using the *Create New CSV File Format* option. 6.5.1 Section explains in detail the creation of CSV file formats. *Include sub folders* should be selected if there are source data files residing in subfolders of the main selected folder location.

5. Set the “*Check for updates in this folder every ...*” option according to how often the source files should be checked for updates.
6. After adding the folder to the feed, press the *Apply* button in the CSV Feed settings panel. The changes made in the settings panel won't be saved until the *Apply* button is pressed.

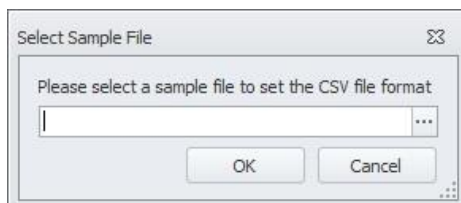
Wait for the initial internal index generation to finish, the time taken will depend on the size of the files inside the added folders. The list of symbols present in the source files will be



displayed in symbol lists in Tradesignal under *CSVFeed*. When a frequently updating source file is present in the added folders the *File Scanning Interval* should be defined correctly to propagate the updates to Tradesignal soon after they happen (Please refer section **Error! Reference source not found.**)

### 6.5.1 Creating a CSV file format

When this option is selected from either *Add new Source Data Folder* or *Manage CSV File Formats* options a dialog for selecting a sample file for the CSV file format will appear (Since it is necessary to have a sample file for preview purpose to create a new file format).



After picking a sample file, the *New CSV File Format* dialog will be displayed. Once a sample file is selected for a file format it will be remembered as long as the file format is assigned to a source folder. Most of the options displayed in the *New CSV file format* are similar to the *Create CSV File Format* option in File Import, there are a few functionalities which are different since CSV Feed supports user defined FIDs.

New CSV file format

CSV File format name:

Column Delimiter: ☐ TAB ☒ Semicolon ☐ Space ☐ Comma ☐ Other:

Timezone:  Date format:

Skip first ... rows:  Decimal delimiter:

☒ Parse file name Display name delimiter:

File name:

Pattern:

☐ Ignore zero values ☐ Ignore negative values

Symbol List Sort Pattern:

Symbol Sort Replace Pattern:

Zero Padding for Symbol List Sorting:

DateTime	Open	High	Low	Close	Volume
[TIMESTAMP]	[OPEN]	[HIGH]	[LOW]	[CLOSE]	[VOLUME]
08.11.2011 00:00	52.37	51.74	50.3	50.87	1138398
09.11.2011 00:00	54.6	51.68	49.54	50.19	1109588
10.11.2011 00:00	42.6	50.76	49.295	49.14	880553
11.11.2011 00:00	55.02	52.1	49.885	51.15	993751
14.11.2011 00:00	56.47	54	52.3	52.25	1955740
15.11.2011 00:00	58.3	53.3	51.7	52.02	1189313



The CSV file format name is case insensitive and without spaces. The correct column delimiter should be selected till the data preview shows the columns correctly. The date format determines in which order Day, Month and Year are specified in the date column in the data. If the date format specified is wrong, the data in the file will not be processed.

### 6.5.1.1 Specifying CSV File Format Columns

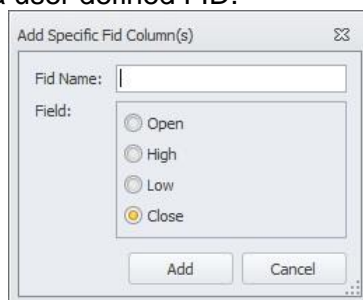
Once the correct column delimiter is selected the preview shows first few lines of data from the selected sample file. Correct columns can be set by clicking on the column headers of the preview. If there the data in a particular column should not be processed then *<skip>* should be selected as the column header, else the file might not be processed correctly.

The CSV file format will be inconsistent without a specified close column; if one or more user defined FIDs are being specified then each FID should have a close column. (Refer section 6.5.1.1.1 for more details on user defined FIDs)

CSVFeed allows symbol names or display names to be derived from more than one column; the order in which they should be combined can also be specified (Refer section 6.5.1.1.2 for details)

#### 6.5.1.1.1 Specifying User Defined Fields (FIDs)

By selecting the *Add User Defined FID* option from the column header, any name other than that of standard FIDs can be used as a user defined FID.



When a user defined FID column is specified, the columns which do not have any user defined FID associated with it is considered to be part of the Default Fid specified for the Folder.

	DateTime	Close	YIELD:Open	YIELD:High	YIELD:Low	YIELD:Close
►	Date	CLOSE	YIELD:OPEN	YIELD:HIGH	YIELD:LOW	YIELD:CLOSE
	2014-04-07T15:20:20:420	75	76	72	73	32
	2014-04-06T15:20:20:420	90	91	87	88	20

Consider the above example, here the user defined FID is "YIELD" but the second column does not have any user defined FID defined; hence it will be associated with the Default FID specified for the source folder or the FID derived from the file name using *Parse file name* option (Refer section 0 Source File Naming Convention for details).

#### 6.5.1.1.2 Multi-column Symbol/Display Names

One or more columns can be combined in any order to derive symbol or display name from data.



This can be done by selecting the *Add Multi-Column Symbol Name/Add Multi-Column Display Name* options from the column header Dropdown.

If order is not specified or if more than one symbol/display column have same order number the column on the left will be considered to have the lower order number. A column can have both symbol and display name headers. This can only be done by using the *Add Multi-Column Display Name* option, and by selecting ☒ *Set this column as display name:* (*Set this column as display name*) option. To remove the secondary display name header, deselect this option.



Symbol column values are combined with “\_” as the default delimiter, for display names the delimiter can be specified in *Display name delimiter* option in the CSV file format.

DateTime	Symbol   DisplayName(3)	<skip>	<skip>	<skip>	<skip>	Symbol   DisplayName(2)	Symbol   DisplayName(1)	Close	Volume
05.11.2014 10:16:00	05.11.2014	6	9	4	6	5	1	5.0	19.4

Consider the above example the resultant symbol name would be “05.11.2014\_5\_1” and the Display name would be “1505.11.2014” assuming that the *Display name delimiter* option was left empty.

### 6.5.1.2 Symbol List Sort Options

These settings can be used to determine how the symbol list provided by the CSV Feed will be sorted in Tradesignal. *Symbol List Sort Pattern* will be matched to the display name of the symbol and will be replaced by the pattern given in the *Symbol Sort Replace Pattern* option so the symbols will be sorted according to the order specified.

Symbol List Sort Pattern:  ⓘ

Symbol Sort Replace Pattern:

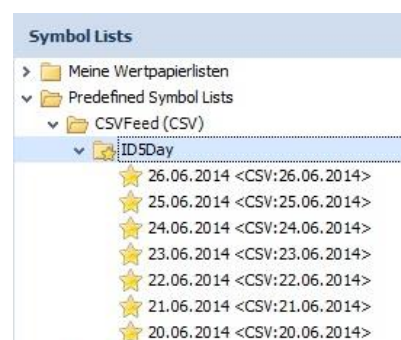
Zero Padding for Symbol List Sorting

☐ Ascending  
☒ Descending

This option is only significant when there is more than one symbol present in a CSV file.

The example above shows how symbol names derived from a column with date data can be sorted in the order of latest date first.

For the file shown in the CSV File Format creation example, with the above symbol sort settings, the Symbol list in Tradesignal will be listed as below.



### 6.5.1.3 Source File Naming Convention

In order to reference available data files containing price data, a mapping has to be defined between a requested symbol (alongside FID and period) and its dedicated data file.

The *Parse file name* option allows to setup such a mapping in a generic way by providing a regular expression that defines where the requested symbol, FID, and period parts are to be found in a CSV file name.

Example:

If the source file contains “LAST” FID information of symbol “ADAG.DE” for a periodicity of “DAILY”, the file name can be “ADAG.DE DAILY LAST.CSV” which should be stored in the folder with a CSV file format using the regular expression given in the image as a *Pattern*.

File name	ADAG.DE DAILY LAST.CSV
Pattern	(?<symbol>[^\s]+)\s(?<period>[^\s]+)\s(?<fid>[^\s]+).csv

Note that in the example a white space character is used to separate Symbol, Fid, and Period, therefore \s is used to separate the different groups mentioned in the regular expression. All predefined group names are case insensitive.

In addition to symbol, fid, and period, the following metafields can likewise be specified in the file name using the (?<metafield>...) syntax:

- *Exchange*
- *Currency (given as three-letter ISO 4217 short code)*
- *DisplayUnit (number of decimal places given as an integer)*
- *TradeUnit (trade unit key, see Appendix A: for details)*
- *ContractSize (given as a floating point number)*
- *TickSize (given as a floating point number)*

Note that hovering the mouse above the blue question mark symbol to the right of the pattern edit box will display metafields as extracted from the given file name using the specified pattern. This is a simple way to check the entered expression for validity.

The *File name* option will show the path of the selected file without the root folder (Folder specified in the *Add new source folder* dialog). This will be useful in case symbol needed to be derived from folder names of the source files.

NOTE: When opening a CSV format for editing through *Manage CSV File Format* option, if the CSV format is not assigned to any folder, the *File name* field will show the full file path, care should be taken while setting a *Parse file name pattern*, since root folder will be removed from the path when it is assigned to a data source folder later.



## 6.5.2 CSV Feed Configuration settings

The default Push port and Backfill port for CSV data feed is 27372 and 27373 respectively. If necessary, port numbers can be adapted in the CSV data feed configuration available in the Console.

### 6.5.2.1 Source Data Folders

Contains the list of added folders containing source data files with the assigned csv file formats.

#### Default FID

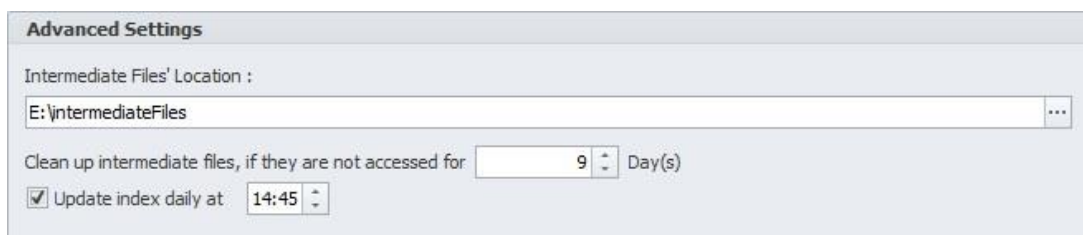
If there is no FID column specified in the CSV format then this value is taken as the FID for that symbol.

#### Default Period

If there is no Period column defined in the CSV file format, and if it cannot be derived from the file name, Default period is used for that symbol.

### 6.5.2.2 Intermediate Files' Location

CSV Feed creates temporary files for each subscribed symbol, fid, and period combinations in a way that supports efficient data retrieval. This field defines where these temporary files should be stored.



The screenshot shows the 'Advanced Settings' dialog box. It has a title bar 'Advanced Settings'. Inside, there is a section 'Intermediate Files' Location :'. Below this is a text input field containing 'E:\IntermediateFiles' and a browse button '...'. Below the text field is a label 'Clean up intermediate files, if they are not accessed for' followed by a spin box set to '9' and the text 'Day(s)'. At the bottom, there is a checked checkbox 'Update index daily at' followed by a time selection box set to '14:45'.

### 6.5.2.3 Index Update Daily

This can be activated and set so that the internal index is updated at a specific time. Updating the index helps the CSV Feed to detect newly added source data.

### 6.5.2.4 Clean up intermediate files

This is the amount of time an intermediate file has to be inactive at least before it is automatically removed.



## 6.6 Setting up Thomson Reuters Tick History Feed

Thomson Reuters Tick History Feed can be set up using CSV Feed and can be installed by selecting the *CSV Feed* option in the DataConnect setup. This data feed scans one or more specified file locations containing price data definition files in Thomson Reuters Tick History format, making the data available to Tradesignal clients. Files are periodically being rescanned to propagate possible updates.

### Steps to set up Thomson Reuters Tick History Feed

1. To set up Thomson Reuters Tick History Feed after its installation, start by selecting the **Manage Datafeeds** button on the DataConnect Console's toolbar. Then click on the **Click here to add a new datafeed** link to start the wizard. Select the option **CSV Feed** in the wizard. In the next page of the wizard, a three-letter prefix can be given to the feed, after completing the wizard the newly added feed would be visible in the table on the left hand side of the *Manage Datafeed* window. Select this entry to start configuring the data feed (The CSV Feed settings panel will be visible on the right hand side of the window).
2. To add source files to the CSV Feed, the folder(s) containing source files should be added with appropriate file formats assigned to those folders. A new folder can be added by clicking the **Add new Source Data Folder** button.
3. Select the corresponding folder, and then select *Thomson Reuters Tick History* option from the *CSV File Format* dropdown.



4. After adding the folder to the feed, press the *Apply* button in the CSV Feed settings panel. The changes made in the settings panel won't be saved until the *Apply* button is pressed.

Wait for the initial internal index generation to finish, the time taken will depend on the size of the files inside the added folders. The list of symbols present in the source files will be displayed in symbol lists in Tradesignal under *CSVFeed*.



## 6.7 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 DataConnect to third-party datafeeds and data files. Once an OpenConnect derived service is packaged and deployed to a suitable server, DataConnect 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

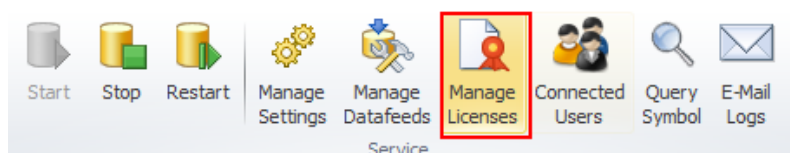
DataConnect will attempt to connect to your service after restart.

*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.*



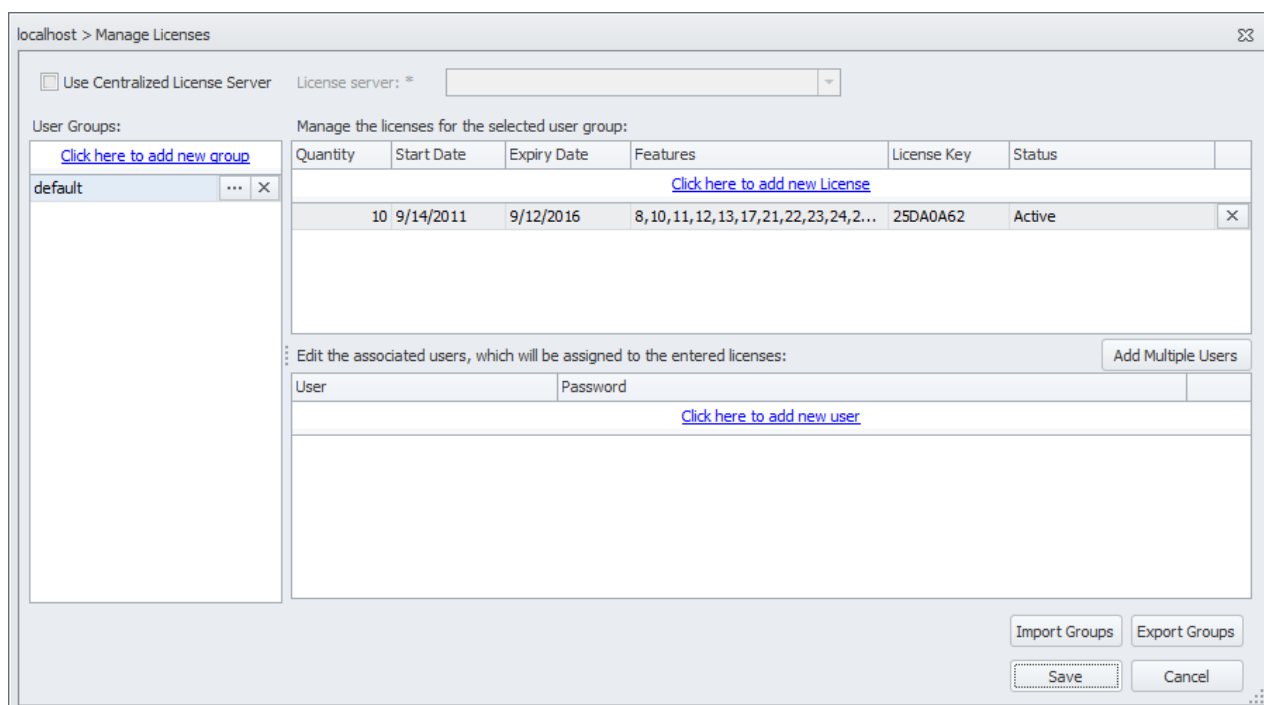
## 7 Managing Licenses and Users

Before client applications can use DataConnect 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 DataConnect on a central server and obtain the central license code from Tradesignal Support.
2. Direct all other DataConnect installations to this central license server in the following way:
  - a. Ensure the license server is available from the DataConnect Console.
  - b. Select the DataConnect 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 DataConnect.

The screenshot shows a user interface for configuring licenses. On the left, there is a checkbox labeled 'Use Centralized License Server' which is checked. To its right is a label 'License server: \*'. Further right is a dropdown menu currently displaying 'License Server [licServer:7000]'.

## 7.3 License properties and usage

Each DataConnect 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 DataConnect (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 DataConnect, it is important to state the datafeed that DataConnect will be connected to, because each datafeed vendor requires a different license code. For example, if you change a DataConnect 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 DataConnect installations will use special features that need to be present on the license server.*
- *For the licensing of multiple Tradesignal DataConnect instances in a Master/Slave configuration, see "Scenario 4: Multiple systems" on page 10.*

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 52.

## 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 DataConnect. 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 DataConnect.*
- *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 DataConnect.*

## 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.

localhost > Manage Licenses

☐ Use Centralized License Server License server: \*

User Groups:

[Click here to add new group](#)

team1	...	X
team2	...	X

Manage the licenses for the selected user group:

Quantity	Start Date	Expiry Date	Features	License Key	Status
<a href="#">Click here to add new License</a>					
10	9/14/2011	9/12/2016	8,10,11,12,13,17,21,22,23,24,2...	25DA0A62	Active

Edit the associated users, which will be assigned to the entered licenses: [Add Multiple Users](#)

User	Password
<a href="#">Click here to add new user</a>	
guest	No
horst	No

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 DataConnect only to access Top-Up data

*This relates to a Tradesignal Enterprise feature.*

If a user only needs to access local file or web based top up data, this can be achieved by assigning a username in Tradesignal prefixed by a minus character (-). Note that access is then restricted to top-up symbols solely (DataConnect will not provide any symbols).

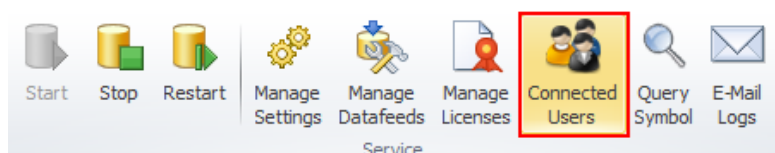
## 7.10 Exporting / importing user groups

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 DataConnect.

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



## 7.12 Generating an alert when a license is about to expire

Starting with version 5.3, DataConnect is able to generate a warning into the DataConnect log and the Windows Application Event Log and optionally an Email to the Administrator or an SNMP Trap when a license is going to expire soon.

The behaviour can be controlled in the options dialog shown using Manage Settings→Administrator Notification:

localhost > DataConnect Options

Service

Data Service (General)

Data Collection

Whitelist

Automated Synchronization

Client Connections

Logging

Administrator Notification

Offline Mode

SNMP

TMD5@tradesignal.com

SMTP User (optional):

SMTP Password (optional):

Interval before identical messages are resent (in seconds):

3600

Test Current E-Mail Settings

**DataConnect can generate an alert to the DataConnect Log and Windows Application Event Log when a license is about to expire**

Days before expiry (set to 0 to deactivate alert):

7

☐ Send an Email to the administrator entered above

☐ Generate SNMP Trap (DataConnect SNMP extension must be activated)

\* DataConnect must be restarted for change to take effect  
\*\* Only available with an active connection

Save Cancel

Note that the pending license expiry alert can be switched off by setting the option *days before expiry* to 0.

Please refer to section 15 on how to setup SNMP integration when you plan to generate SNMP traps on license expiry.



## 7.13 Generating Usage Reports (for Trayport)

Starting with DataConnect 5.10.16, usage reports can be generated for the Trayport data feed. DataConnect will record data feed usage on a per-user basis. Reports can be generated with monthly granularity either in textual form or in a more structured way as XML. Note that Tradesignal users need to use client version 7.6.3 or later in order to facilitate usage reporting.

The Usage Report wizard will be started by clicking on the Usage Report icon in the Home ribbon:



Usage reports are generated taking data from every DataConnect into account that is currently attached to the Console. Results will be combined into a single report.

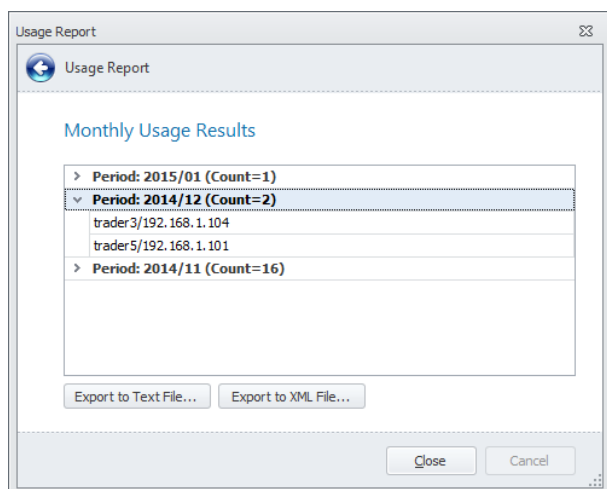
**Please note:** The Trayport data feed is currently the only data feed that supports Usage Report generation. Trying to generate a Usage Report when none of the connected DataConnect servers have a Trayport feed configured will result in an error message.

The wizard will allow you to select a data range to restrict the number of months for which usage data should be reported. The available choices include the recent three months, current month, previous month, current year, previous year, all available data, and a custom range.

The following wizard page will let you select which information should be used to construct the unique identifier (UUID) listed in the usage report (by username, by IP address or a mix of both).

The final wizard page displays a preview of the report in a grid-based fashion, grouped by monthly period.

The following example lists a three-month report with username and IP address as UUID format:



Exporting the report to text file will allow you to write a textual form of the report to a selected target file. The format is as follows:

*First line:* Header listing the feed name and the generation time stamp in ISO 8601 format.  
*Second line:* Column header

*Remaining lines:* space separated list of monthly period (YYYY/MM), total number of users, and semicolon delimited list of UUIDs.

**Example:**

```
DataConnect Usage Report for 'TMDS_TRPT' generated 2015-01-12T13:09:26.5871807+01:00
Year/Month Total_Number_Users UUID_List
2015/01 2 trader3/192.168.1.103;trader4/192.168.1.104
2014/12 2 trader2/192.168.1.102;trader2/192.168.1.104
2014/11 4 trader1/192.168.1.101;trader2/192.168.1.102;trader3/192.168.1.103;trader4/192.168.1.104
```

The following section shows the report for the same data generated in XML format:

```
<?xml version="1.0" encoding="utf-8"?>
<DataConnect-Usage-Report Service="TMDS_TRPT" Date="2015-01-12T13:19:08.3793460+01:00">
  <Period Year="2015" Month="1" Count="2">
    <User UUID="trader3/192.168.1.103" />
    <User UUID="trader4/192.168.1.104" />
  </Period>
  <Period Year="2014" Month="12" Count="2">
    <User UUID="trader2/192.168.1.102" />
    <User UUID="trader2/192.168.1.104" />
  </Period>
  <Period Year="2014" Month="11" Count="4">
    <User UUID="trader1/192.168.1.101" />
    <User UUID="trader2/192.168.1.102" />
    <User UUID="trader3/192.168.1.103" />
    <User UUID="trader4/192.168.1.104" />
  </Period>
</DataConnect-Usage-Report>
```

### 7.13.1 Clearing Collected Usage Report Data

There is no automatic process in place that discards old recorded usage data. If there is the need to clear out existing usage data, this can be achieved manually using the Command Line. The command “usagereport delete [num]” will remove existing usage data, keeping the most recent “num” months of records intact. Specifying “1” would keep only the current month worth of records. Omitting the “num” argument would remove all collected data.

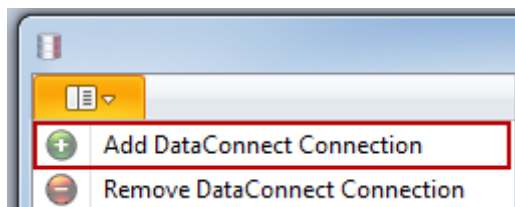


## 8 DataConnect Connections

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

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

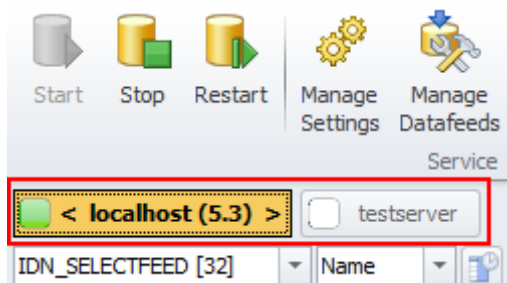
For more on deployment strategies and setups, see “Deployment: Common Installation Configurations” on page 9.



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

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








The DataConnect Console will try to connect to DataConnect. 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 drawn with a yellow-orange background.

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

Possible states:

-  DataConnect and all feeds are running.
-  DataConnect is not running.
-  DataConnect is running but one or more feeds are not active.
-  DataConnect is running in offline mode.
-  Console is not connected to DataConnect.
-  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 and can be displayed by name or ticker symbol. A column showing the date of the last access to a symbol can be shown or hidden by pressing the button next to the name/symbol selection drop down.

The screenshot shows the Symbol Management interface. On the left, there is a table with columns 'Name' and 'Last Access'. The table lists various symbols like ADIDAS N, ALLIANZ SE, BASF SE, etc. On the right, there are tabs for 'Reference Data' and 'Price Data'. The 'Reference Data' tab is active, showing fields for Symbol, Alias, Symbol Group, Symbol Class, Name, Exchange, Currency, Display Units, Filter Active, Filter Type, Filter Value, and Timezone.

Name	Last Access
ADIDAS N [ADSGn.DE]	9/23/2011
ALLIANZ SE [ALVG.DE]	9/23/2011
BASF SE [BASFn.DE]	in use
BAY MOT WERKE [BMWVG.DE]	9/23/2011
BAYER N AG [BAYGn.DE]	9/23/2011
BEIERSDORF [BEIG.DE]	9/26/2011
COMMERZBANK [CBKG.DE]	in use
DAIMLER AG N [DAIGn.DE]	in use
DEUTSCHE BANK N [DBKGn....]	9/23/2011
DEUTSCHE POST NA [DPW....]	9/23/2011
DJ INDU AVERAGE [DJII]	9/26/2011
DT BOERSE NEUE [DB1Gne....]	9/23/2011
DT LUFTHANSA AG [LHAG.DE]	9/23/2011

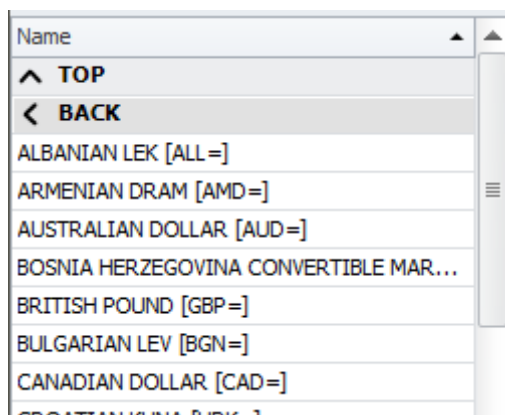
Starting with DataConnect 5.2, symbols can be logically grouped by their respective market sectors. Tick the **Groups** checkbox to enable grouping:

The screenshot shows the Symbol Management interface with the 'Groups' checkbox checked. The list of symbols is now grouped into categories like ALL (352), Bond (1), Currency (31), Equity (34), Future / Commodity (275), Index (10), and Others (1).

Name	
ALL (352)	>
Bond (1)	>
Currency (31)	>
Equity (34)	>
Future / Commodity (275)	>
Index (10)	>
Others (1)	>



You can drill down into groups by clicking one of the entries. For example, selecting **Currency (31) >** will filter the list of symbols to contain currency-type instruments only:



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 80.*
- *For setting up datafeeds, see "Managing datafeeds" on page 21.*
- *For synchronizing symbols and historical data, see "Synchronizing Systems" on page 77.*



## 9.1 Editing reference data

The reference data for each symbol can be edited.

For the Filter settings, see “Filtering Instruments (Whitelisting)” on page 75.

### 9.1.1 Editing trading sessions

When DataConnect 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.

Starting with DataConnect version 5.6.1, the session management has been updated to allow a finer control of sessions spanning more than one week day (e.g. overnight sessions or afternoon sessions that have to be rolled over into the next daily candle).

Timezone  
TOK

Data Feed Session  
MO-FR 0844-1103,1230-1505,1530-2330

Rollover Time  
15:15

Active Session

Start	Day	End	Day	Mo	Tu	We	Th	Fr	Sa	Su	
15:30	3 days prior	23:30	3 days prior	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X
08:44	Trade day	11:03	Trade day	<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"/>	X
12:30	Trade day	15:05	Trade day	<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"/>	X
15:30	Prev day	23:30	Prev day	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X

This example shows how the session for the 10 year Japanese Government Bond (JGB) will be visualized in the Console.

Click [Add new session](#) to add further sessions. For each session, a start and end time and the respective trading day(s) can be entered. Note that the session editor control will reject invalid sessions (e.g. overlapping session spans) and will visualize this with a thick red border.

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 Data Feed Session.

*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

Price data stored in the data cache can be displayed for the selected symbol. The constituent values of a bar, such as open/high/low/close/volume, can be modified (see section 9.2.1).

When necessary, bars can either be deleted from the data cache or can be tagged as disabled (see section 9.2.2).

Select the **Field** and **Period** first. The price data will then be loaded from the data cache.

Reference Data		Price Data							
LAST (VOL)		Daily		Save Changes		Refresh		Filter: All Candles	
Date	Open	High	Low	Close	Volume	Source			
<a href="#">Click here to add new candle.</a>									
9/4/2012	62.57	62.59	61.87	62.29	436142	I	X	↺	↻
9/3/2012	61.66	62.64	61.51	62.44	2359994	T	X	↺	↻
8/31/2012	61.08	62.34	61.03	61.81	3084550	T	X	↺	↻
8/30/2012	62.2	62.2	60.91	61.22	3151278	T	X	↺	↻
8/29/2012	62.52	62.6	61.77	62.52	1677339	D	X	↺	↻
8/28/2012	62.41	62.69	62.09	62.45	1687322	T	X	↺	↻
8/27/2012	61.96	62.84	61.67	62.82	1348875	U	X	↺	↻
8/24/2012	61.6	62.4	61.24	62.13	2227577	T	X	↺	↻
8/23/2012	62.78	63.19	61.46	61.68	3214006	T	X	↺	↻
8/22/2012	62.52	64	62.17	62.32	2578378	T	X	↺	↻
8/21/2012	62.57	63.28	62.4	63.04	2271832	T	X	↺	↻
8/20/2012	62.05	62.52	61.98	62.3	2029222	T	X	↺	↻
8/17/2012	61.8	62.17	61.37	62.05	3853766	T	X	↺	↻
567									




The source column indicates the origin of the displayed values:

- **I: realtime**, the source is realtime updates from the datafeed.
- **T: historic**, the source is historical data from the datafeed.
- **U: user**, the bar has been modified by the user. The respective row is displayed using **bold** font.
- **D: disabled**, the bar has been disabled by the user. The respective row is displayed using **grey** font (available as of DataConnect version 5.5).

A dropdown filter menu is available that allows to show all user-edited data at a glance or to hide disabled candles.



The rightmost column contains the following buttons:


-  Deletes or disables the selected bar (see section 9.2.2).
-  Only enabled when the selected bar is marked as previously edited or disabled. Resets the overwrite protection so that the bar can be overwritten again by historic data from the data feed (see section 9.2.5).
-  Only enabled when the selected bar was modified, but not yet committed using **Save Changes**. Undoes the modification and restores the original state (see section 9.2.4).

Beneath the Price Data tab, the number of candles is displayed (in the screenshot above, “567”).


Note that prices for symbols published by pass-through data feeds (see section 9.6) can neither be displayed, nor be edited.

### 9.2.1 Editing price data fields

Click into the fields to edit the data. Changed fields are highlighted by a light-blue background as long as the changes have not yet been committed. Edited bars will be marked with the *U* source flag in the datacache, effectively protecting them from being overwritten by history provided by the data feed. You may want to use the reset feature to remove this kind of overwrite protection when no longer necessary, e.g. when the upstream data have been corrected (see section 9.2.5).

To undo edits, click  or open the context menu and choose **Undo Pending Modifications for the Selected Candles**. This works only for unsaved changes.

### 9.2.2 Deleting and disabling candles

To delete or disable a complete bar (time point), click . Alternatively, open the context menu and choose **Delete Selected Candle(s)**. Deleted but not yet committed bars are highlighted by an orange background and strike-through font.

To undo a deletion that was not yet committed using **Save Changes**, click  or open the context menu and choose **Undo Pending Modifications for the Selected Candles**.


Deletions will be committed after using **Save Changes**, which will open a dialog that asks if to delete or disable the data. The difference is that deleted bars will come back when the default history is restored, whereby disabled candles will stay invisible to the Tradesignal user, even after the history has been reloaded. This helps to suppress bad data, such as spikes, as long as the faulty data has not been corrected by the upstream data feed. You can use the reset feature to remove this kind of overwrite protection (see section 9.2.5). Note that disabled candles are available as of DataConnect version 5.5.

### 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 Reverting not yet committed price data

Click  next to a row containing uncommitted changes to restore the bar to the previous values. To undo changes in multiple bars, you can select all respective rows and then pick **Undo Pending Modifications for the Selected Candles** from the context menu. To undo all uncommitted changes for all bars, click **Refresh** to reload all data.

### 9.2.5 Resetting modified or disabled data

User-modified and disabled bars are protected against overwrite from historic bars provided by the data feed. If you want to re-enable overwriting with default historic data, you can do so by using the reset button  on specific bars. Alternatively, use **Reset Selected Candles** from the context menu (which also works on multiple rows at once). Note that the reset feature is available starting from DataConnect version 5.5.

In case you want to reset all modifications for all FIDs and periods of a symbol, use the **Restore Default Hist. Data** from the ribbon bar (see section 9.3.1).

### 9.2.6 Saving price data

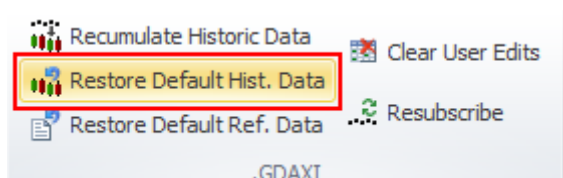
Click **Save Changes** to commit your data changes to the data cache. Changed data will be checked for plausibility; if this validation fails, the data will be rejected. When candles are tagged as to be deleted, you will be asked what action to perform (delete from data cache or tag as disabled, see above).



## 9.3 Clearing and restoring data

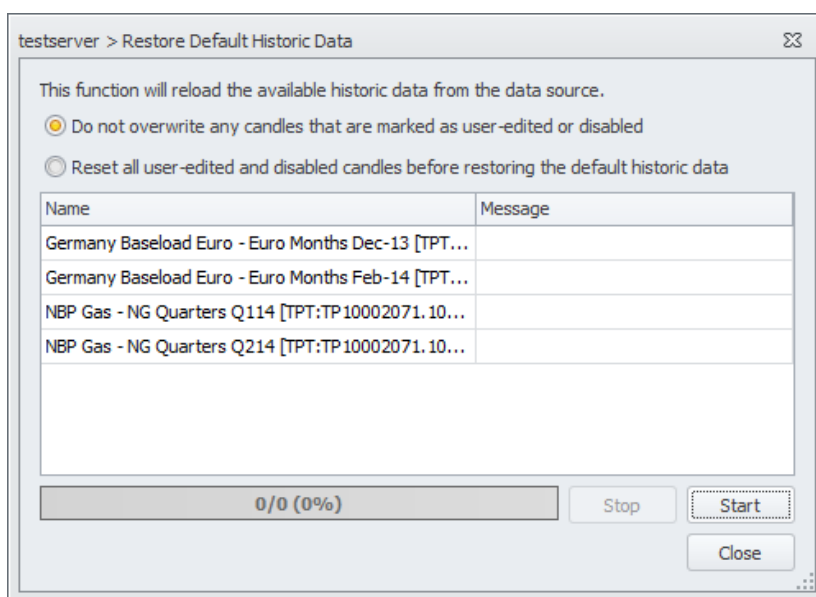
### 9.3.1 Restoring default historic data

This function will restore the historic data as currently provided from the datafeed.



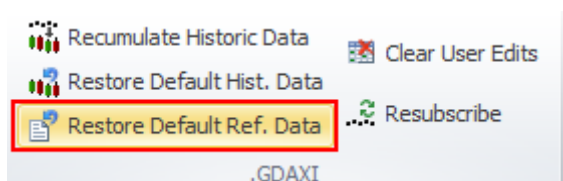
**Click Restore Default Historic Data to restore all available historic data for the currently selected symbols from the datafeed.**

The retrieved history will not overwrite bars that were marked as user-edited or disabled, unless the option to reset all user-edited and disabled candles is selected.



### 9.3.2 Restore default reference data

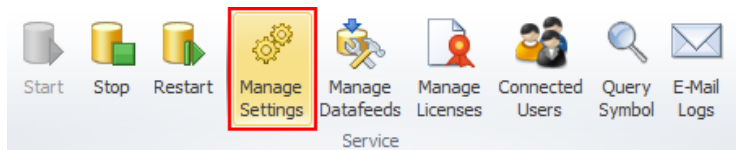
This function will restore the reference data of the symbol as received from the datafeed.



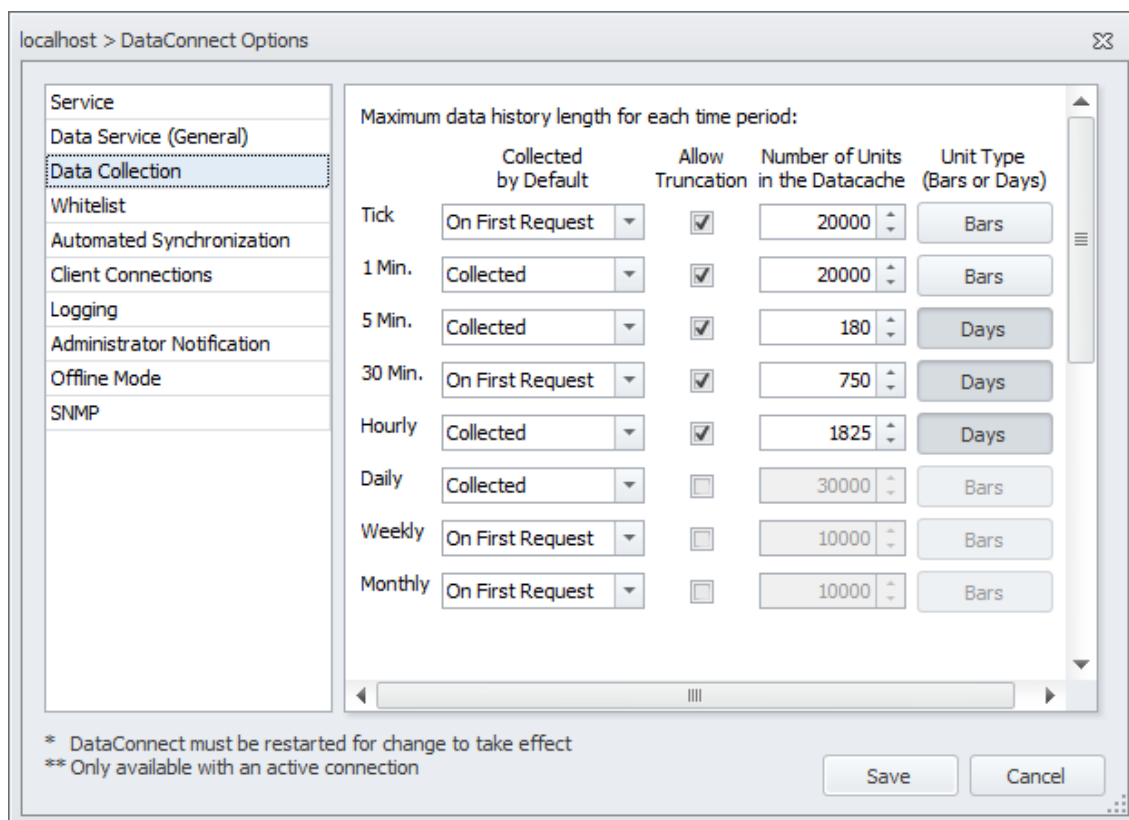
**Click Restore Ref. Data to restore the reference data for the selected symbol from the datafeed.**



## 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, DataConnect 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**.

DataConnect Rolling Forward symbol names are constructed following a specific syntax:  
`<root code><shift>.<back adjustment>.<candle calculation>.<period offset>.<months>`

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

## 9.6 Pass-Through Mode vs. Persistent Mode

Before version 5.3, all symbols were created on DataConnect in a so-called *persistent* fashion, which means that reference data and historic data for that symbol were permanently collected in the local data cache and topped up with incoming realtime ticks. Symbols remained in the data cache until explicitly deleted (either manually using the DataConnect Console's symbol band or automatically via the symbol cleanup for securities which have not been accessed for a longer time).

Starting with version 5.3, DataConnect implements the so-called *Pass-Through Mode*, available for the Bloomberg and Open Connect data feeds. In Pass-Through Mode, all data coming from the upstream data source is forwarded directly to the connected Tradesignal clients, without any intermediate storage in the DataConnect data cache. This means that complete data histories will always be fetched directly from the data source, so that the most recent upstream version is guaranteed to be forwarded to clients.

Note that feeds that require tick data storage to perform trade corrections via Trade IDs (e.g. Trayport) or feeds that do not provide intraday histories (e.g. Thomson Reuters) cannot be put into Pass-Through mode so collection needs to be persistent for this class of feeds.



The following table compares advantages and disadvantages of both modes in order to aid in deciding which mode fits best for a given data feed:

Persistent Mode	Pass-Through Mode
Typically faster delivery of long histories to connected Tradesignal clients due to local caching (for symbols that already exist in the cache)	Charts containing long histories may take longer to load, depending on how fast the upstream data source is able to supply the data. The history download needs to be repeated for every access (no caching available)
Historic corrections in the upstream database will not be reflected in DataConnect data until manually refreshed (esp. Bloomberg)	A freshly open chart in Tradesignal always reflects the most current state of the data directly from the source
Over time, persistent data collection leads to longer histories than the upstream feed usually supplies. So even when the data source supplies only 50 days of intraday data, DataConnect can supply 100 days after 50 days of realtime collection	Only that amount of historic candles are available which the data source currently supplies (e.g. exactly 50 days of intraday data)
Considerably higher memory, disk, and CPU footprint as symbols will stay present and get updated until removed from the data cache	Very light-weight in terms of memory and disk impact as only those symbols need resources which are currently subscribed from Tradesignal
Data feed vendors may restrict local storage of provided data, so Persistent Mode may not be applicable	No data is persistently stored so Pass-Through Mode is appropriate for restricted data sources
Support for trade corrections available (Open Connect)	Trade corrections not supported

### 9.6.1 Pass-Through Mode for Bloomberg data feed

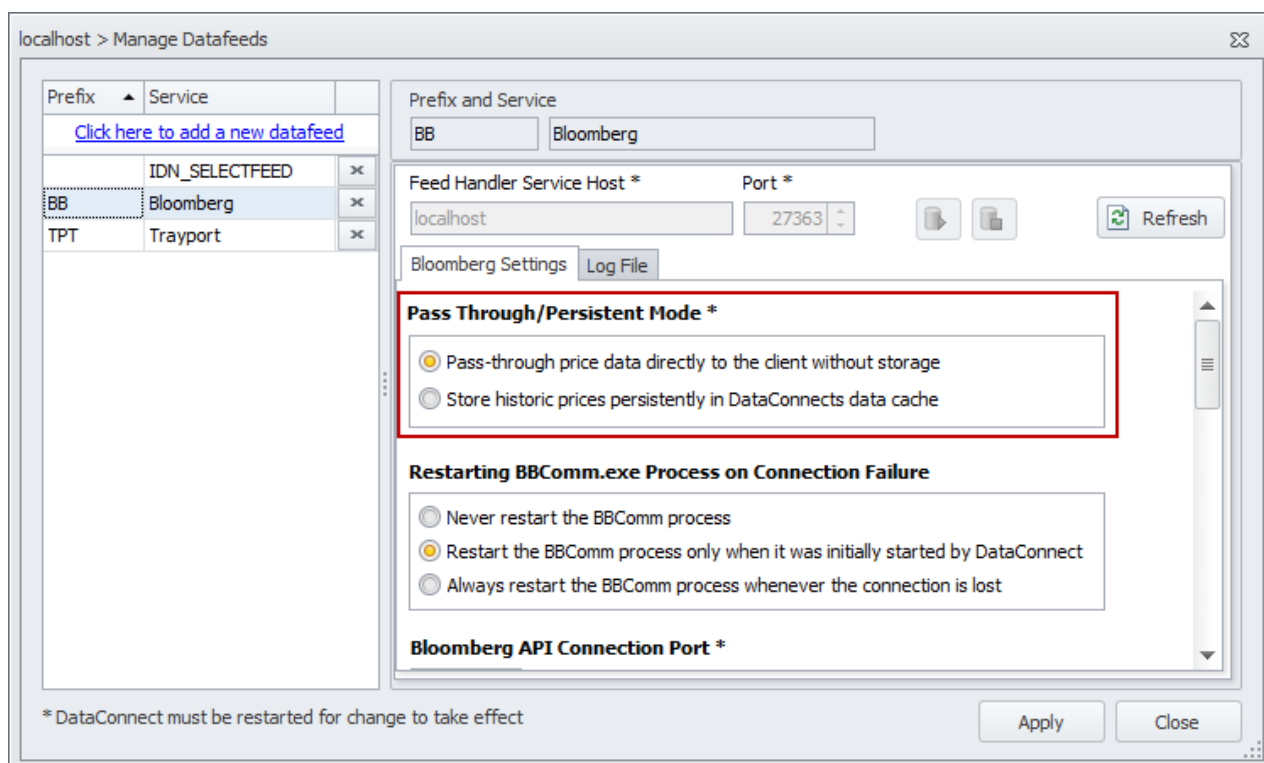
Starting with version 5.3, new DataConnect installations using the Bloomberg data feed will by default use Pass-Through Mode.

When the DataConnect Console detects an existing Bloomberg installation that was recently upgraded to version 5.3, a wizard will guide the user through the process of optionally converting to Pass-Through Mode. If necessary, the Bloomberg data feed can still be used in a Persistent Mode.

*Note: When deciding to convert to Pass-Through Mode, DataConnect will remove all currently present Bloomberg symbols from the data cache.*

Transition from one mode into the other is always available using the DataConnect Console's *Manage Datafeeds* page:





### 9.6.2 Pass-Through Mode for Eikon data feed

By default, Thomson Reuters Eikon data feeds are set up to store historic prices persistently in DataConnect's data cache. See section 6.1.2.1 (Advanced Eikon data feed configuration) for setting up Pass-Through Mode for this data feed.

### 9.6.3 Pass-Through Mode for OpenConnect data feeds

Open Connect based data feeds are persistent by default. If you want to switch your Open Connect data feed to Pass-Through Mode, you will have to add the following setting to the [General] section in the settings file (accessible via the Manage Datafeeds page for the respective Open Connect service):

```
[General]
PersistentSymbols=0
```

Changing this setting requires a restart of DataConnect and the OpenConnect data feed.

Please be aware that feeds publishing trade corrections cannot support pass-through mode because DataConnect needs access to already collected trades via Trade IDs stored in the local data cache in order to apply corrections to the correct candles.



## 9.7 Automatic copying of historic data for Thomson Reuters Expired Contracts symbols

The naming of currently traded realtime future contracts for Thomson-Reuters symbols (RICs) basically follows this pattern:

{Future Root Code}{Month Code}{Last Digit Year}

For example, at the time of writing in September 2013, the RIC “FDXZ3” denotes the December contract ({Month Code} = Z) in 2013 ({Last Digit Year} = 3) for the FDAX future ({Future Root Code} = FDX). Once the contract expires in end of December, 2013, FDXZ3’s data state will become closed. However, once trading for the FDAX December 2023 commences, FDXZ3 will be reactivated (with a then different meaning).

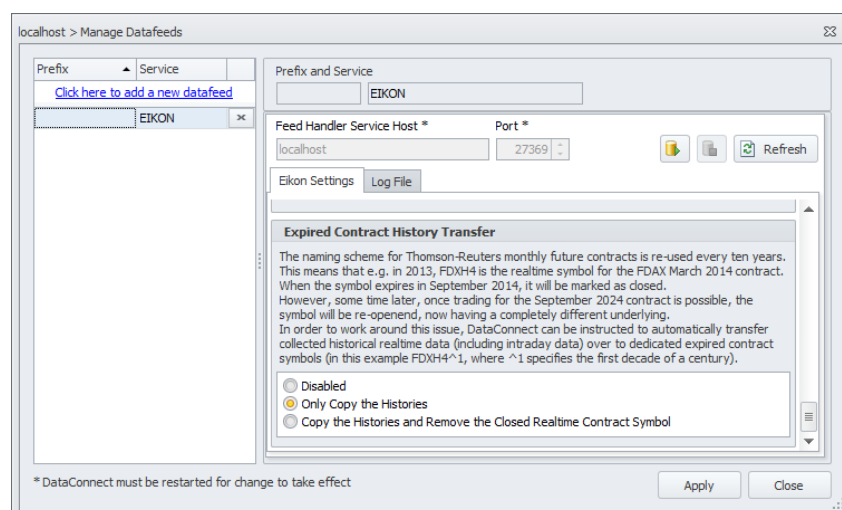
Thomson-Reuters also supplies special RICs for expired contracts that show a longer life span (up to 100 years as opposed to up to 10 years for current contracts). The pattern for an expired contract symbols is

{Future Root Code}{Month Code}{Last Digit Year}^{Decade}

So once the FDXZ3 expires, historic (daily, weekly, monthly) data will be made available under the expired contract code FDXZ3^1 (^1 denotes decade one of the current century).

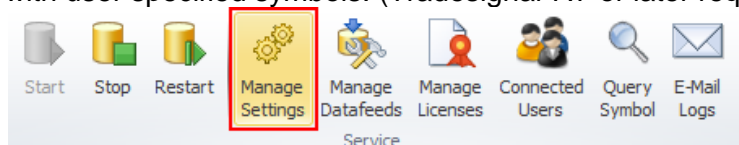
In order to avoid inconsistent data (as in the example above, mixed up histories containing data both for December contracts from 2013 and 2023), DataConnect allows to copy all collected data (including intraday histories) from the freshly expired realtime contract symbol over to the dedicated expired contract symbol. If wanted, the closed realtime symbol can be automatically removed from the data cache. This will be performed as part of the nightly servicing.

Configuration of this feature is available for both TREP and EIKON based Thomson-Reuters data feeds via the Manage Datafeeds dialog.

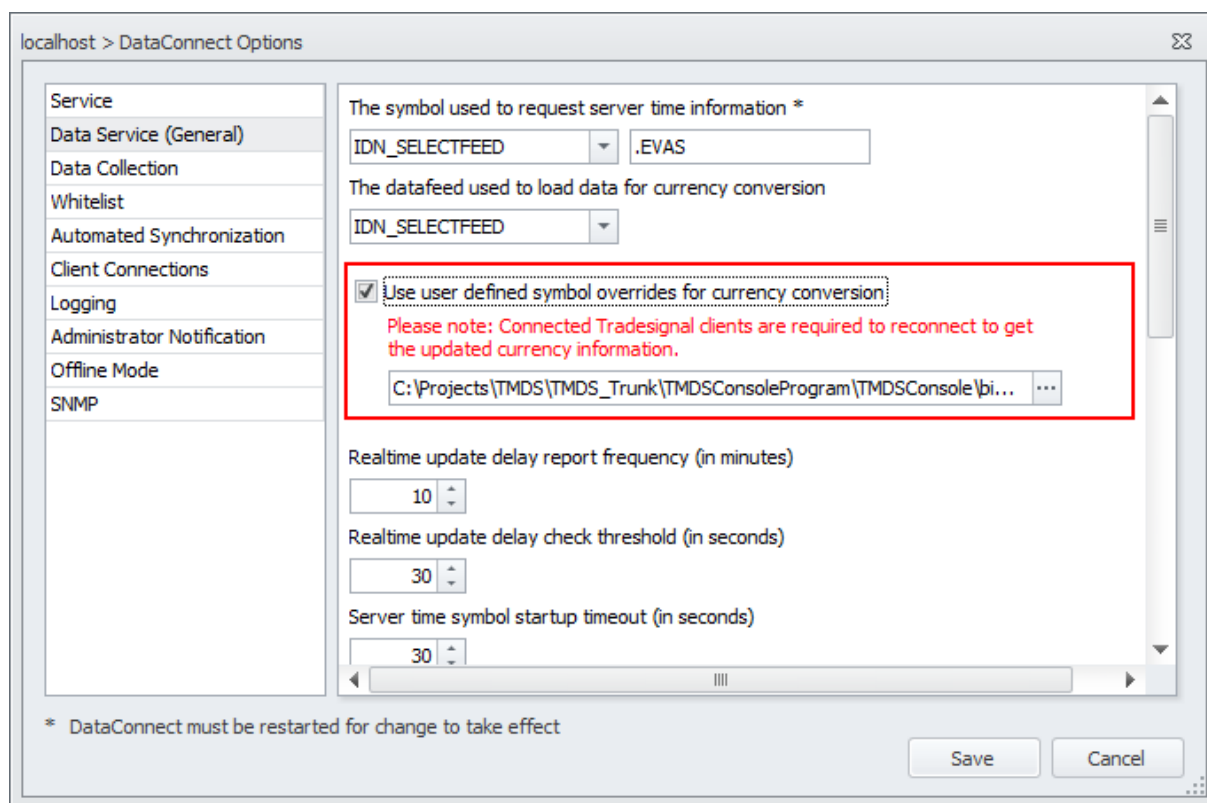


## 9.8 Specifying Overrides for Currency Symbols

Starting with version 6.0 DataConnect allows overriding provider supplied currency instruments with user specified symbols. (Tradesignal 7.7 or later required)



Click Manage Settings →  
Data Services (General)



To enable this feature set the “Use user defined...” checkmark as shown in the screenshot and pick the override definition file with the file picker. Once you specified the file click the “Save” button. In order to adopt the new settings currently running Tradesignal applications need to be **restarted or reconnected** using the “Reconnect”-Feature from the connection menu. Currency override files use the Microsoft “.ini”-File format. A template for this file is installed within the DataConnect-Console directory. (CurrencyOverrides.ini) Each override consists of a pairing of currency code and provider symbol. Every currency is mapped to exactly one provider symbol. The currency codes used are specified by the ISO 4217 standard. ([http://en.wikipedia.org/wiki/ISO\\_4217](http://en.wikipedia.org/wiki/ISO_4217))

A sample file using ThomsonReuters symbols could look like this:

```
[Currencies]
EUR = EUR=X
CHF = "CHF=X"
```

On the left of the equal sign is the ISO-Code (EUR, CHF) and on the right side is the provider symbol (EUR=X, CHF=X). The quotation marks are optional and will be ignored.



## 10 Filtering Ticks

There are a number of configuration options to instruct DataConnect

- 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 71.
- on a per instrument class basis, see "Setting up tick filtering for " on page 72.

*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 DataConnect 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 Setting up tick filtering for symbol groupings

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

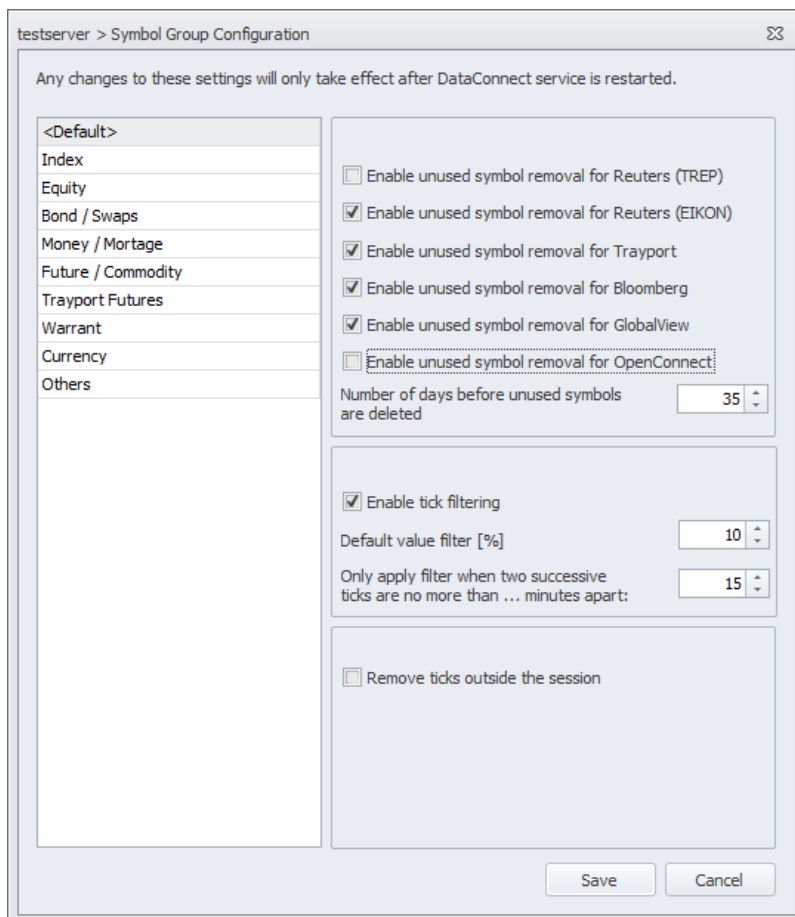
If no explicit price filter configuration had been performed for an instrument, a global filter will apply. The global filter is a relative value filter that will ignore a tick if it is greater or less than the preceding tick by a specified percentile amount. It is set in Tick Filter Settings.



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

**The Symbol Group Configuration dialog opens.**

The **<Default>** values apply to all instrument classes unless overwritten by a specifically configured symbol group.



Tick filtering is based on two settings: 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, DataConnect 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 DataConnect-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 DataConnect log.

To get a list of the actually filtered ticks printed to the DataConnect 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.

## 10.5 Automatic removal of symbols based on usage

In order to avoid unrestricted growth of the data cache and thereby to help keeping performance at high levels, DataConnect allows to clean out symbols from the data cache that have not been accessed by Tradesignal clients for a longer duration of time. This can be configured in the setting **Number of days before unused symbols are deleted** and will be used in the nightly service to determine if a symbol can be safely cleaned from the data cache.

Starting with version 5.7, configuration of **unused symbol removal** can be achieved on a per symbol group basis like the bad tick filter settings above. Additionally, DataConnect allows targeting symbols provided by the following data feeds: Thomson-Reuters (TREP), Thomson-Reuters (EIKON), Trayport, Bloomberg, GlobalView, and OpenConnect. This allows fine-tuning automatic symbol removal taking into account feed-specific attributes like availability of intraday histories.

Note that certain symbol types are excluded from the automatic clean-up, including DataConnect usage metrics (statistics symbols starting with '#STAT') and user-defined symbols (starting with '\$').

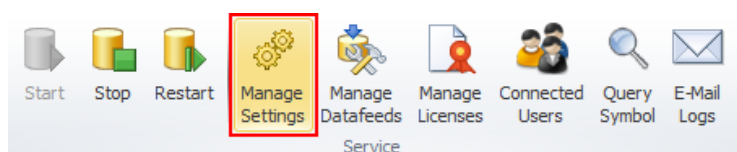


## 11 Filtering Instruments (Whitelisting)

In order to handle a large number of instruments, multiple DataConnects 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 11*

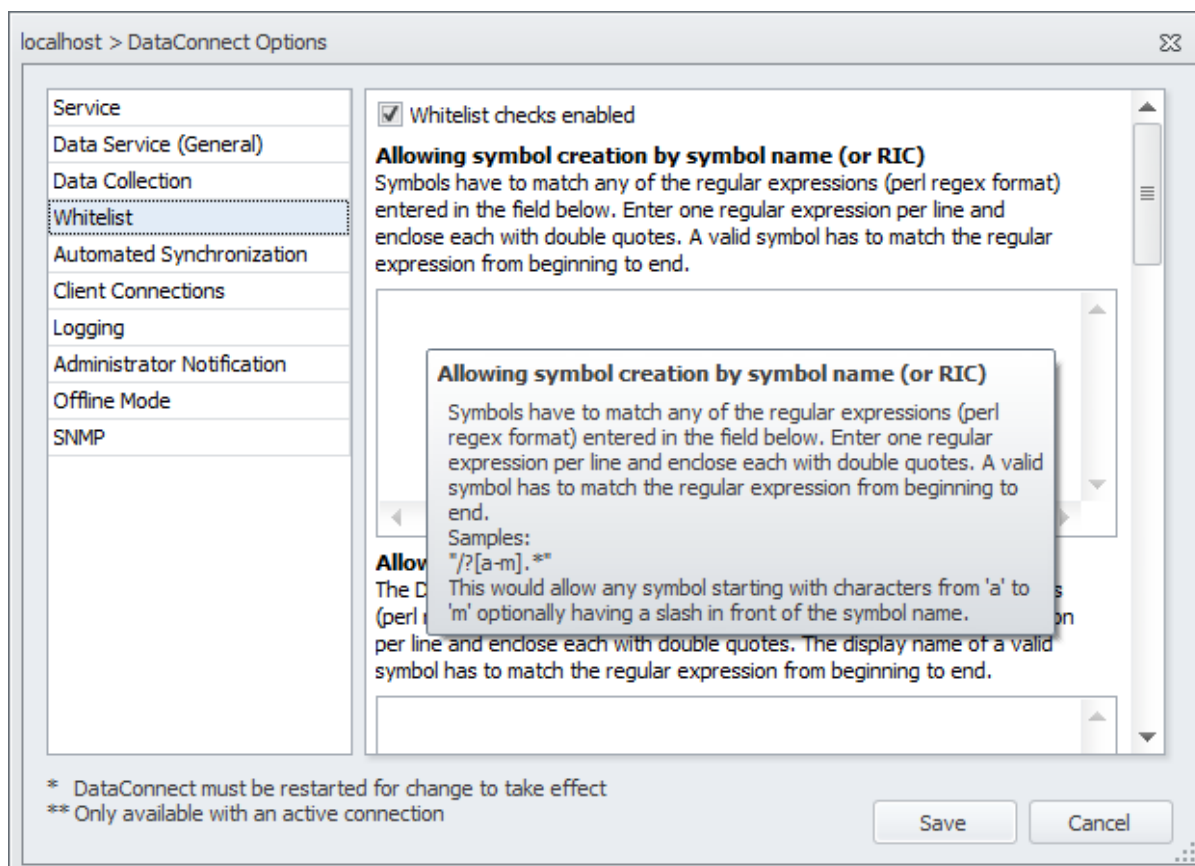
DataConnect will match each new symbol against every whitelist section. The symbol will be created by DataConnect as soon as one matching whitelist entry is found. If no whitelist filtering should take place, the section(s) need to be left blank or any existing entries have to be commented out.



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

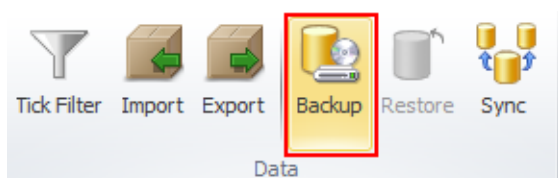
This section offers information on manual backup and restore of the data cache file. For automatic backups, see "Nightly Maintenance" on page 96.

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

Backing up a DataConnect data cache file will create a *tmds.dat.bak* file, which can be copied to another DataConnect 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) DataConnect



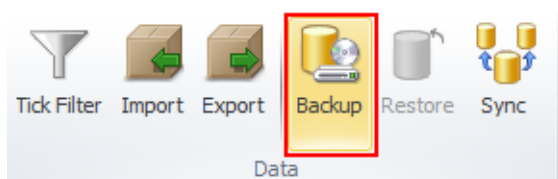
Click Backup and confirm.

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 DataConnect



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 DataConnect, the backup is performed by the *gbak* utility. DataConnect Console will start this utility via the *backup.cmd* script in the installation folder. You can also start this script directly.

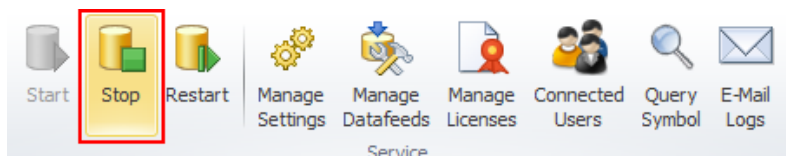
Resulting files in the installation folder:

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



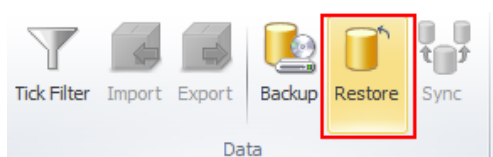
## 12.3 Restoring a DataConnect data cache

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



Click **Stop** to stop the DataConnect 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. DataConnect 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

DataConnect 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*

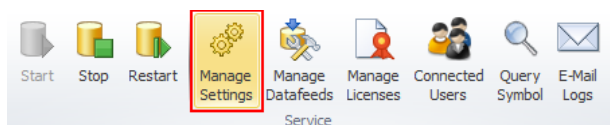
For both synchronization methods, a working connection to the source system must be available, see “DataConnect Connections” on page 57.

*Automated synchronization is an important feature for deployment configurations, see “Deployment: Common Installation Configurations” on page 9.*



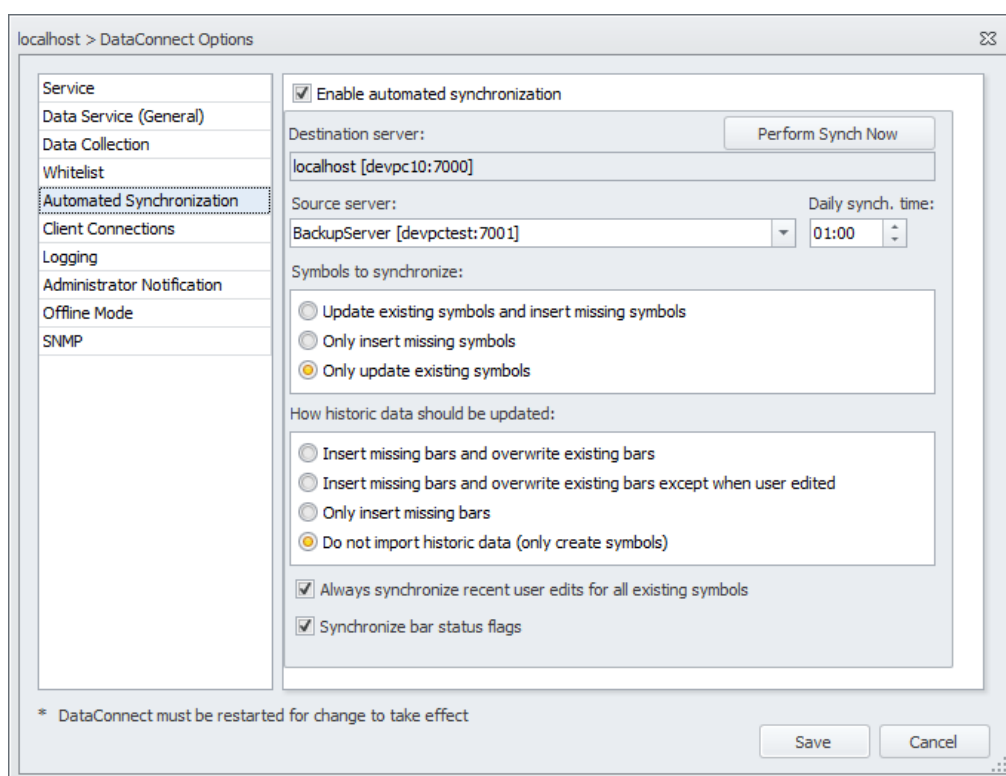
### 12.4.1 Enabling automated synchronization

Automated synchronization requires identical datafeed setups (including all prefixes).



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

The destination server is preset with the currently selected DataConnect instance. Choose the appropriate options and synch time (in local computer time) and save.

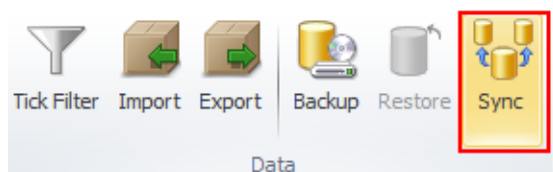


Selecting “*Always synchronize recent user edits for all existing symbols*” ensures that all data that has been modified or permanently deleted (cf. section 9.2) on the source server will be transmitted to the destination server. A typical use case of this feature would be to set “*Symbols to synchronize*” to “*Only insert missing symbols*” in conjunction with this option. This would ensure that the same set of symbols from the source server will be present on the destination server and all user edits performed on the source will be transmitted to the destination as well. Manual user-edits would therefore need to be performed on the source system only.

*Please note:* DataConnect will keep track of all performed automated synchronizations in order to reduce the amount of user-edited data to be transferred. This works based on recording the IP address of the requesting (target) server. Therefore, when planning to use this feature it has to be ensured that the network environment is set up in a way that guarantees unique IP addresses for every host DataConnect is running on.



## 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.

Synchronization

Source server: testserver [stephantest:7000] Source datafeed: IDN\_SELECTFEED

Destination server: localhost [ingoseven2:7000] Destination datafeed: IDN\_SELECTFEED

Select how preexisting data should be updated (for the synchronized time span):

Insert new bars and replace existing bars

Select Missing Symbols Select Existing Symbols

	Fields	Synch Message
Insert new bars and replace existing bars	LAST, VOL	
Insert new bars, replace existing bars and delete missing bars	LAST, VOL	
Insert new bars and replace existing bars unless marked as user-edited	LAST, VOL	
Insert new bars only	LAST, VOL	
Create new symbols without importing data	LAST, VOL	
<input type="checkbox"/> .GDAXI XETRA DAX PF [.GDAXI]	LAST, VOL	
<input type="checkbox"/> .MDAX MDAX INDEX PF [.MDAX]	LAST, VOL	
<input type="checkbox"/> .MDAXI MDAX INDEX PF [.MDAXI]	LAST, VOL	
<input type="checkbox"/> .N225 NIKKEI 225 INDEX [.N225]	LAST, VOL	
<input type="checkbox"/> /.AEX AEX-Index/d [/.AEX]	LAST, VOL	

Stop Synchronize Close

*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, DataConnect 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.

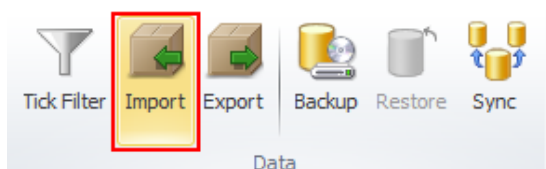
- For the export of user groups, see “Exporting / importing user groups” on page 53.
- For the export of the data cache files, see “Backup / Restore” on page 76.
- For the export of DataConnect logs and information, see “Preparing information for sending by e-mail” on page 94.

### 13.1 Importing symbols

For importing symbols, files have to be available in one of the supported formats:

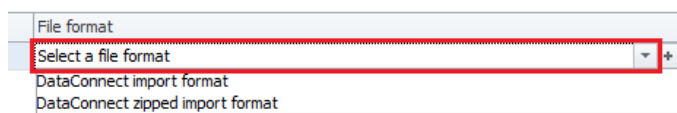
Binary: \*.zdbu, DataConnect format

CSV: \*.csv, comma separated values (single or multiple symbols), compatible with Thomson Reuters DBU format or in a format that is described by a template (see below; note that template based generic CSV file import is available since DataConnect 5.7.0)



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

The wizard lets you select one or more files for import. In the next step, the file formats for the selected files have to be assigned. In case a selected file has a not yet described format, a new CSV file template can be defined for this purpose by clicking on the **plus** button next to the dropdown.



A dialog window will be opened allowing the format of the import file to be defined, including a preview of the data according to the currently picked selections. Parameters such as date formats, rows to be skipped, column headers, and delimiters can be specified here.

*Note: Only one file format can be specified per file. If the data in the file is made up of more than one format then it should be manually split into different files and templates should be specified separately for each unique format.*

Selecting a different **Delimiter** will alter the preview according to the selected delimiter. The **Date format** defines the order in which Day (D), Month (M) and Year(Y) appears in the data. *Specifying a wrong date format will lead to unsuccessful import of data.*

*Note: a **DateTime** or **Date** column and a **Close** column have to be specified in order for the file format to be valid.*



The correct **Time Zone** should be picked based on the date/time values specified in the file. This is necessary to allow DataConnect to automatically convert the data to the internally used UTC format and is especially important for intraday data.

**Skip first...row** can be used to ignore a number of rows present in a file, e.g. to skip unused header information.

Selecting the **Ignore zero value** will cause all the data points with zero as value for **Open**, **High**, **Low**, **Close**, **Volume**, or **OpenInterest** columns to be skipped while importing. The same applies to selecting **Ignore negative value**, where data points with negative values will be skipped.

Columns can be assigned with their corresponding titles by picking an adequate value from the list that will appear when clicking the column header. All columns that have to be ignored should be tagged with **<Skip>**.

CSV file format creator

CSV file format name: FileFormatName

Column Delimiter

☐ TAB

☐ Semicolon

☐ Space

☒ Comma

☐ Other

Time Zone: UTC

Date format: MDY

Skip first ... rows: 2

Decimal delimiter:

☐ Parse file name

File name: GOOG.OQ.csv

Pattern:

☐ Ignore zero values

☐ Ignore negative values

DateTime	Open	High	Low	Close	Volume
DATE	OPEN	HIGH	LOW	CLOSE	VOLUME
12/17/2012	705.5	738.28	704.02	721.07	6041900
12/10/2012	685.39	716.48	683.79	701.96	12058000
12/3/2012	702.24	705.89	682.33	684.21	9429900
11/26/2012	666.44	699.22	658	698.37	13702000
11/19/2012	655.7	678	655.53	667.97	7494200
11/12/2012	663.75	669.8	636	647.18	9956200
11/5/2012	684.5	686.86	650.3	663.03	11164000
10/29/2012	679.86	695.55	675	687.92	5912000
10/22/2012	681.01	687.33	669.7	675.15	13825800
10/15/2012	741.94	759.43	672	681.79	31297100
10/8/2012	761	763.58	738.29	744.75	11794400
10/1/2012	759.05	774.38	750.27	767.65	13359700
9/24/2012	731	764.89	730.25	754.5	22017500

Save Cancel

Selecting **Parse file name** is an advanced feature which will allow the user to specify a regular expression in the **Pattern** section. The importer will parse the file name according to



this regular expression while reading lines from a file. The destination symbol, fid, and period fields can be specified with this option (by way of named subexpressions `?<symbol>`, `?<fid>`, `?<period>`).

☒ Parse file name

File name:

Pattern:

After saving the newly created file format, it can be reused later for other files. In the next step of the wizard one has to specify target symbols, fids, and periods. If the target symbol, fid, and period are already available from the file format definition or from the parsed file name, it will be pre-set in the dedicated grid column.

*Note: The destination symbol, fid and period can be changed even if the imported was able to auto-assign them. For DataConnect zipped import formats this is not possible.*

stephantest > Data Import Wizard

### Analyzing File Content

File Analyzing Progress

	<input checked="" type="checkbox"/> File Name	Source Data...	Symbol Field P...	Dest. Symbol	Dest. FID	Dest. Period
▶	<input checked="" type="checkbox"/> C:\sample_CSV_files\GOOG.OQ.csv		GOOG.OQ	GOOG.OQ	<select>	<select>

Next > Cancel

The destination Fid and Period have to be assigned in this step, along with specifying which symbols from the file are to be imported. Fid and the Period option will be updated according to the selected symbols.

Dest. Symbol	Dest. FID	Dest. Period
GOOG.OQ	<select>	<select>

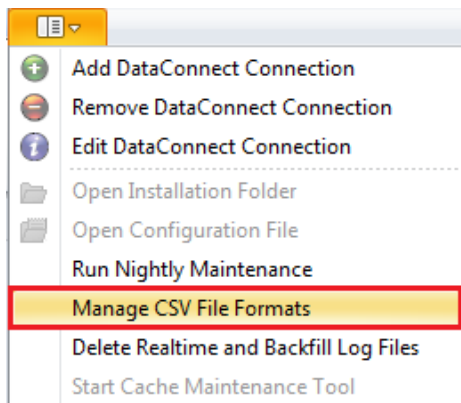
LAST  
BID  
ASK

The remaining steps in the wizard will let you select the datafeed (when the selected import files contain DataConnect import formats) and the overwrite mode.



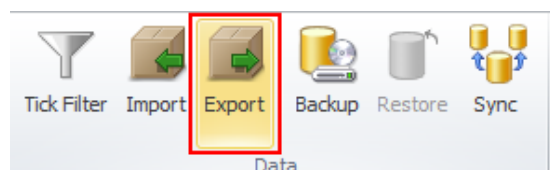
### 13.1.1 CSV file formats

The user created CSV file formats can be accessed via the **Manage CSV file formats** in the application menu in DataConnect console.



This option will allow you to edit, delete and create new CSV file formats. While editing a file format you can rearrange the column order or change the parameters defined in it.

## 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 DataConnect
- 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.

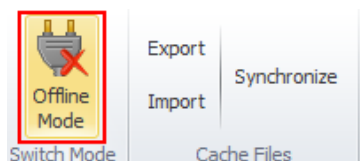


## 14 Offline Mode

Since version 5.3.0, DataConnect may get switched into the so-called *offline mode*. In this mode all real time data feed connections will get disconnected and the data access is limited to users connecting locally. All existing symbols may then be accessed as if DataConnect would still be online. Note that there is an exception with Bloomberg symbols, where data access is generally not allowed when not connected to the Bloomberg data feed.

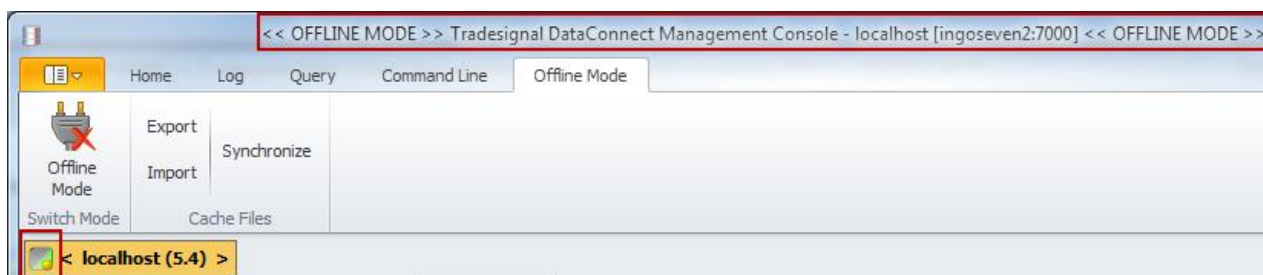
The offline mode is designed to allow temporary access to the data, while not being able to connect to the upstream feed, e.g. when travelling. When using an offline DataConnect everything that has been done with DataConnect while being online will still work as expected. This includes normal charts, forward curves, seasonal charts and other available features. This is possible because DataConnect is caching to disk symbol list requests performed during real-time mode. This means that the contents of symbol lists will reflect the state when last accessed in online mode. Therefore any changes to the lists since populated (e.g. future symbols being expired in the meantime) will not be available.

The DataConnect Console allows the user to directly switch a running DataConnect into offline mode and back to real time processing via the Offline Mode tab.



Toggle **Offline Mode** to enable or disable the offline mode via the DataConnect Console.

Switching to offline mode is also accessible via Tradesignal 6.3.0 or newer using the right-click connection status context menu. Note that Tradesignal has to be connected locally. Offline mode can also be activated using Manage Settings (Offline Mode section), see below.

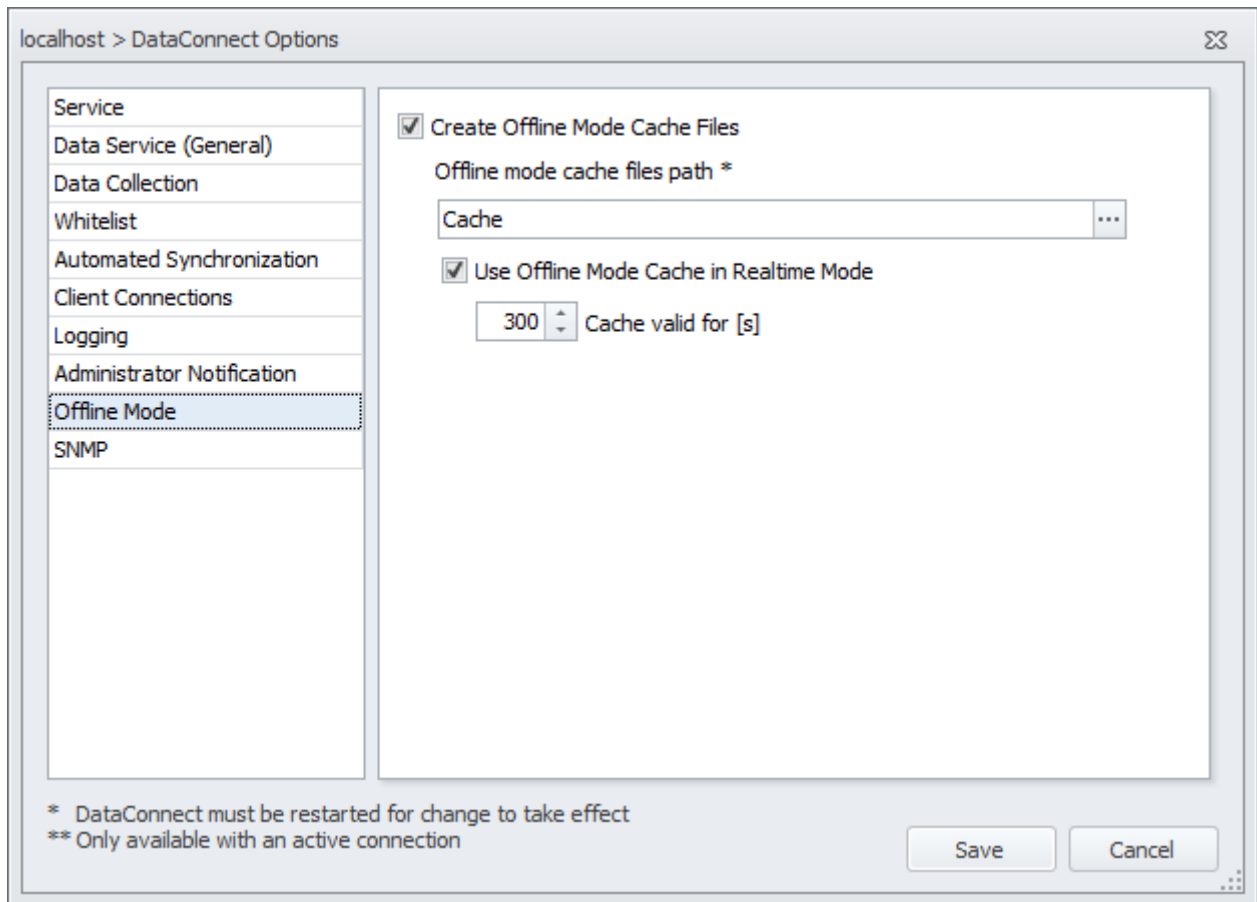


When DataConnect is put into offline mode, the connection button icon will be colored grey-green, the offline mode toggle button will change to reflect the current mode and the application title will indicate that DataConnect is running in offline mode.



## 14.1 Offline Mode Options

The available settings may get changed via the options dialog reached by selecting “Manage Settings” in the ribbon bar. Click on “Offline Mode” in the topics list to see available offline mode options.



The Offline Mode can only work correctly if the DataConnect previously cached enough information while being fully online and connected. This background information gathering may get switched off, if not necessary by un-checking the setting “Create Offline Mode Cache Files”.

When necessary, the location for cache files can be changed. The default location is the “Cache” subdirectory in the DataConnect installation path.

DataConnect may use cached lists in online mode as well in order to avoid additional requests to fetch list contents from the upstream feed. By default, DataConnect will re-use cache contents for successive list requests within the time span of 300 seconds. If it is not desirable to use cached symbol lists, this feature can be deactivated.

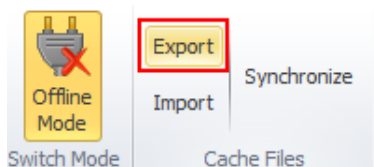
In case of OpenConnect data feeds serving their symbol list data on a user based mechanism, cache files for each user will be generated.



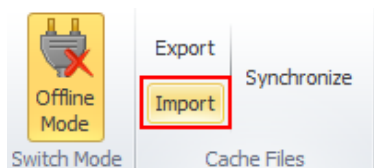
## 14.2 Offline Mode Cache Files

The DataConnect will cache all symbol list and test chain request results to show the content of a predefined symbol list or to create requested forward curves in offline mode. It is possible to share these offline mode cache files with other DataConnect instances.

There are two ways to share cache file data. One is to export the content to a binary file and import it later on the target machine:



Click **Export** to export cache files to a binary file.



Click **Import** to import cache files from a binary file.

The second way is to directly synchronize those files. To use this feature the DataConnect Console needs a connection to both DataConnect instances.

*Note: A DataConnect in offline mode may only get accessed locally; a remote connection to such a DataConnect is not possible.*



Click **Synchronize** to open the synchronization wizard for the cache files of two DataConnect connections.



## **15 Simple Network Management Protocol (SNMP) Integration**

Simple Network Management Protocol (SNMP) is a standardized protocol for managing devices on IP networks. Wide-spread SNMP support is available for hardware devices like printers or network routers, but software programs, e.g. specific Windows Services can also be implemented with an SNMP aware interface.

Since version 5.3.0, DataConnect is capable to act as an SNMP agent in order to provide SNMP Management Applications access to DataConnect metrics and also to receive important push events in form of so-called SNMP Traps.

### **15.1 Configuring the SNMP Integration for DataConnect**

The DataConnect SNMP integration relies on the Microsoft Windows SNMP Service which has to be installed on the same machine on which DataConnect resides. The Windows SNMP Service acts as an SNMP agent which is able to integrate multiple sub-agents. Each sub-agent is registered with a dedicated extension DLL that is loaded into the Windows SNMP Service on service startup.

Whenever SNMP Management Applications enquire information from the Windows SNMP service, the latter will propagate the request to the correct sub-agent. The DataConnect sub-agent DLL (called TMDSSNMP.dll) will establish a permanent connection to the DataConnect in order to propagate requests to DataConnect and to receive the results as well as asynchronous events (traps) from DataConnect.

The Windows SNMP Service is not installed by default. The way to install this service differs slightly depending on which Windows operating system version is used.

#### **15.1.1 Installing Windows SNMP Service on Windows XP or Windows Server 2003 R2**

Open Control Panel → Add or Remove Programs → Add/Remove Windows Components, select “Management and Monitoring Tools”, press the button titled “Details” and check “Simple Network Management Protocol”, then press OK.

#### **15.1.2 Installing Windows SNMP Service on Windows 7 or Vista**

Open Control Panel → Programs → Programs and Features → Turn Windows features on or off, then check “Simple Network Management Protocol (SNMP)” and press OK.

#### **15.1.3 Installing Windows SNMP Service on Windows Server 2008 R2**

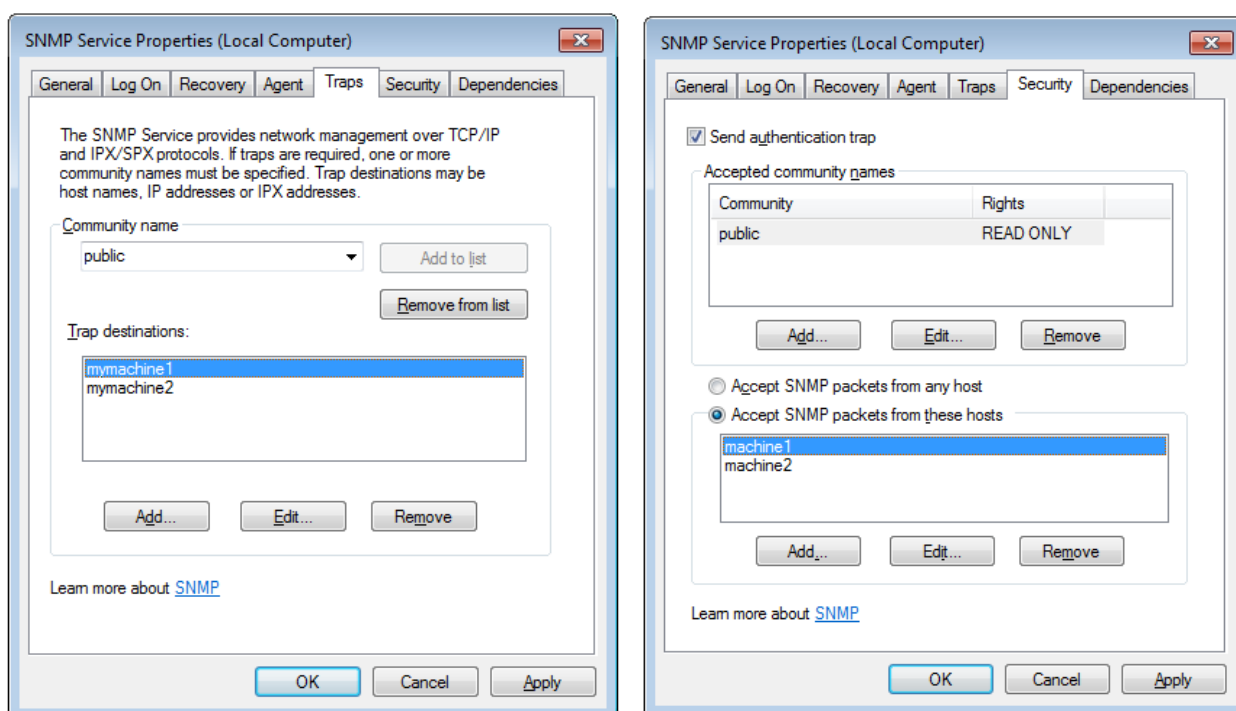
Open Server Manager → Roles → Features → Add Features, then check SNMP Service.



### 15.1.4 Configuring Windows SNMP Service

The SNMP Service has to be configured in order to accept requests from remote SNMP Management Applications and in order to propagate traps to the hosts where the Management Applications reside. The following configuration is identical for all operating systems mentioned above.

Open the Windows Services snap-in listing all services installed on the local system and locate “SNMP Service”, then right-click to open the service properties. Select the Traps page and add a new Community Name called *public* to the list (if not yet present). Add all machines that host an SNMP Management Application you plan to access DataConnect with as Trap destinations.



Now select the Security page and add *public* as an accepted community name for read-only access. Add your SNMP Management hosts to let the SNMP service accept their requests in the list by selecting “Accept SNMP packets from these hosts”.

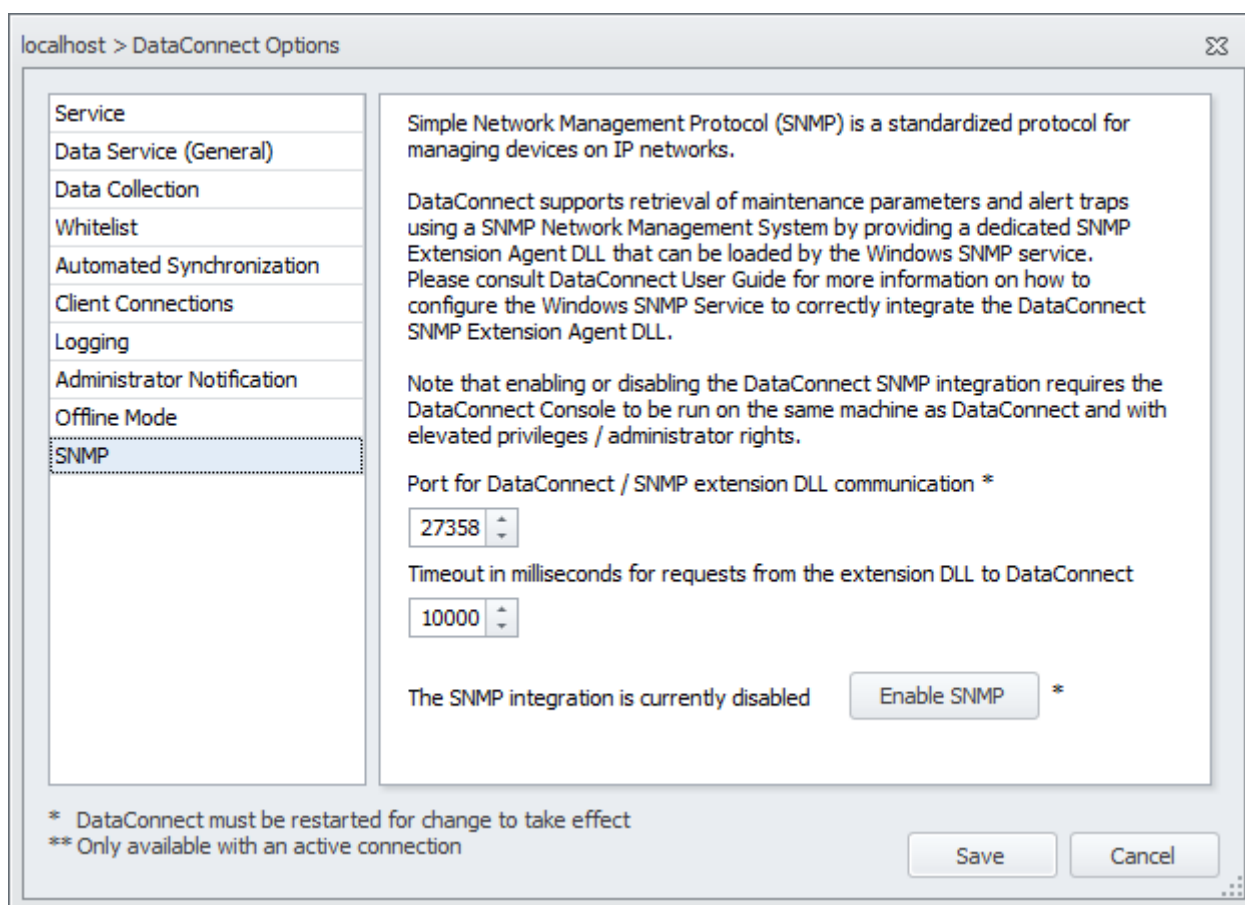
### 15.1.5 Configuring DataConnect to Provide SNMP Integration

By default, SNMP integration is switched off in DataConnect. It can be activated using the DataConnect Console. The Console will need to be equipped with elevated permissions (Administrator rights) in order to write to the registry and to stop and start services. This can be achieved by right-clicking the DataConnect Console application icon and selecting “Run as administrator”.

In DataConnect Console, open the “Manage Settings” dialog and select the SNMP tab. The DataConnect SNMP Extension DLL will use a dedicated port to connect to DataConnect via TCP/IP. That port (27358 by default) can be changed here if necessary. To register the DataConnect SNMP extension DLL with the SNMP service, click on “Enable SNMP...”. After the registration, a restart of the Windows SNMP service will be performed so that the DataConnect



SNMP extension DLL is loaded by the SNMP service and can start to service requests directed to DataConnect by SNMP Management Applications.



### 15.1.6 Disabling DataConnect SNMP Integration

SNMP integration can simply be disabled by clicking on “Disable SNMP...” (once it was enabled) in the SNMP tab of the DataConnect Options dialog. DataConnect Console will remove the registry entries necessary to register an SNMP sub-agent and will restart the Windows SNMP service in order to unload the DataConnect SNMP Extension DLL.

## 15.2 Working with the DataConnect SNMP Integration

### 15.2.1 DataConnect Management Information Base (MIB)

DataConnect comes with a dedicated Management Information Base (MIB) file that can be loaded into SNMP Management Applications in order to provide a hierarchic structure of all the information accessible from DataConnect via the SNMP interface. The `TMDS-MIB.mib` file is located in the DataConnect installation directory.

Management Applications usually contain a so-called MIB viewer which displays all information nodes in a tree-like structure, together with human-readable explanation for each node (read from



the MIB file). Therefore SNMP Management Applications will automatically “know” which pieces of information are available from DataConnect via SNMP in a fully documented way.

In the MIB hierarchy, DataConnect nodes can be found under the branch `iso.org.dod.internet.private.enterprises.tradesignal.tmds (1.3.6.1.4.1.38556.1)`.

### 15.2.2 Information retrievable from DataConnect via SNMP

DataConnect supplies three main sections of information retrievable via SNMP. The following listing displays an excerpt of the data that can be displayed using a SNMP Management Application:

General Information:

- Is there currently a valid connection between the DataConnect SNMP extension and DataConnect?
- Which version number has DataConnect?
- How long has the server been running for?
- Who is currently connected to DataConnect from which host and what application?
- Which feed services are available and what state do they have?

License Information:

- State of the license server (if one is configured).
- Number of licenses currently available for different key features.

Statistics:

- Metrics on all kinds of DataConnect related operations and resources, e.g. CPU load, memory consumption, data cache file size, number of advises symbols, internal processing queue sizes and more. This mirrors the information available from DataConnect statistics symbols (`#STAT_...`) ; also obtainable in the DataConnect Console’s statistics section.

### 15.2.3 SNMP Traps

The following (usually exceptional) situations will trigger an alert (called a Trap in SNMP parlance):

- The connection state of the SNMP sub-agent has changed (connection to DataConnect is lost or established).
- A message of severity *warning* or *critical* has been written to the DataConnect log file. The Trap will include the message, the severity, and the timestamp of the warning.
- A configured TDMS license is about to expire shortly (this trap has to be explicitly activated in the DataConnect options, cf. section 7.12).



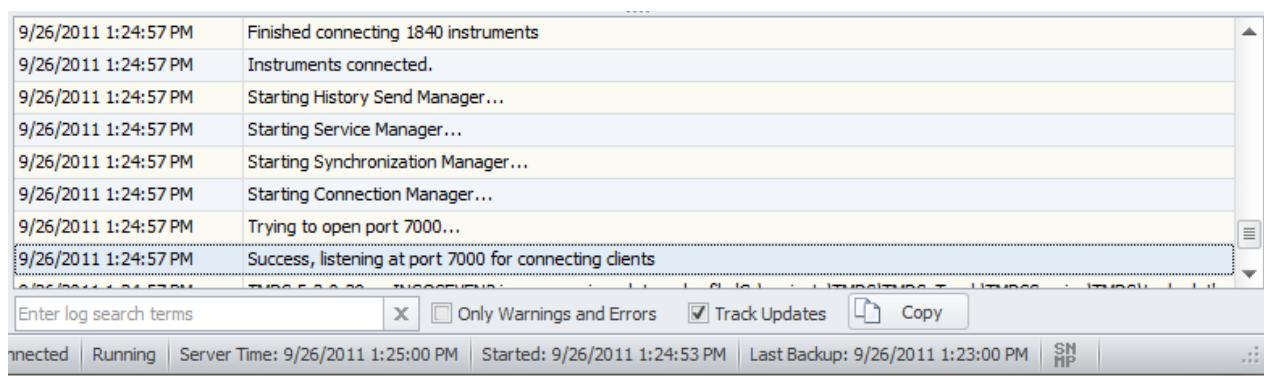
## 16 Logs, Alerts & Troubleshooting Tools

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

### 16.1 Logging console and log file

The logging console displays the contents of the DataConnect log file (*mds.log*), which is located in the installation folder's sub-folder `WD/`.

By default, only significant events on a DataConnect 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

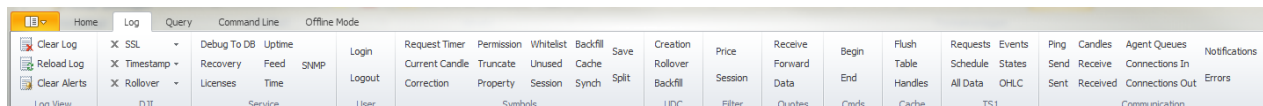
Options can be found in the context menu, for example to clear the log or the alerts (errors and warnings).



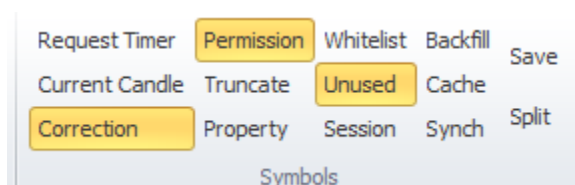
## 16.2 Log

*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 DataConnect. 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 DataConnect is restarted.

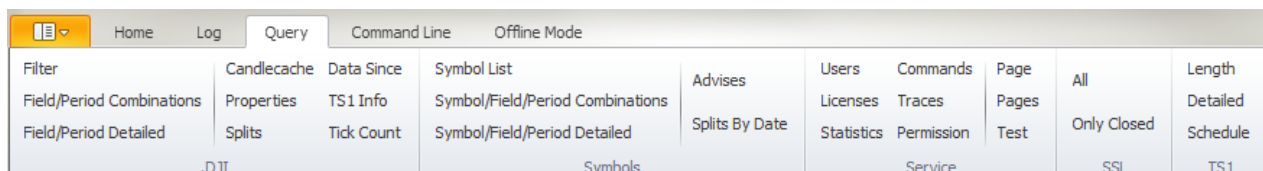
*Note: Logging can also be enabled manually by entering **trace** commands in the command line. Manually entered trace commands are not persistent, i.e., will not be re-activated when DataConnect restarts.*

## 16.3 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 DataConnect 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.*

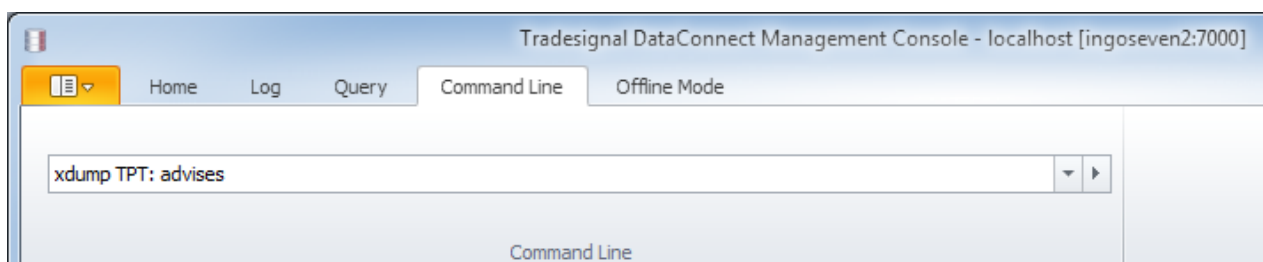


## 16.4 Command line

*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.*

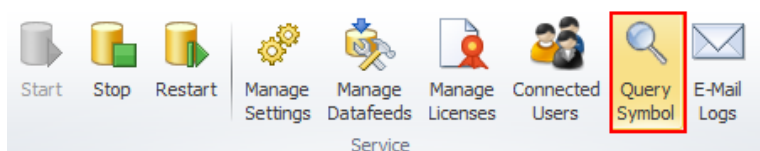
With the command line, script commands can be issued directly to any connected DataConnect. For a complete list of suitable commands, see “Command Reference” on page 97.

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



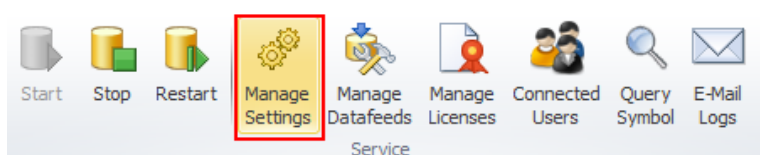
*Note: Commands can only be executed if the DataConnect Service is running and the DataConnect Console has a valid connection to it. Both of these states can be validated in the status bar, see “Status bar” on page 19.*

## 16.5 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.

## 16.6 Sending automatic e-mail messages to administrators



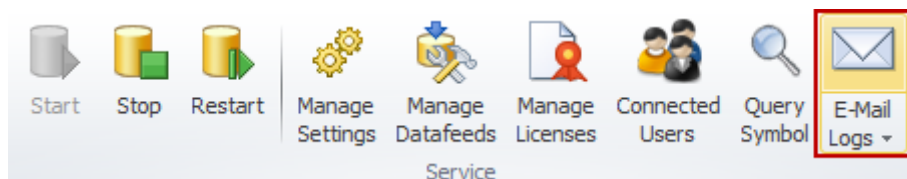
Click **Manage Settings** → **Administrator Notification**.

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

*Notifications will be sent for critical events only.*



## 16.7 Preparing information for sending by e-mail



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

The E-Mail Logs function creates a zip package containing relevant files and additional information that helps the Tradesignal support pin-pointing possible problems. If demanded, DataConnect can be instructed to include statistics data in the support package.

The resulting *.zip* file is written to the folder **Console/export** in the installation folder. The naming convention is *[host name]\_MDSLogs\_YYYY-mm-dd-hh-mm-ss.zip*.

Preparing the support package usually takes less than half a minute, however when statistics are included, preparation time can take a few minutes. After the support package creation has finished, a Windows Explorer window will open showing the location of the support package. The most recent *.zip* file could now be sent to Tradesignal support by selecting **Send to** from the context menu, for example.



## 17 Scripting DataConnect

The command line in DataConnect offers the ability to enter single commands, see “Command line” on page 93. These commands can also be used for more thorough scripting of the DataConnect software with batch files.

Every aspect of DataConnect can be scripted. For a full list of all possible scripting commands, see “Command Reference” on page 97.

### 17.1 Using batch/script files

Batch files must be in text format.

#### 17.1.1 Automatic execution of script files



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

The service settings open.

Two kinds of scripts can be added:

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

#### 17.1.2 Manually starting a script

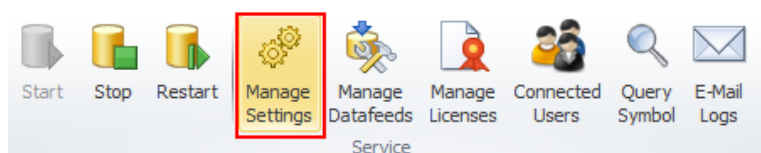
A script can be started directly via **Application button** → **Execute Script**.

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



## 17.2 Nightly Maintenance

DataConnect 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 76.

**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]**).



## 17.3 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**

### 17.3.1 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 DataConnect data cache file. For more information, see "Backup / Restore" on page 76.

#### **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 DataConnect 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.



### 17.3.2 TS1 commands

#### **clearts1**

Removes all current TS1 requests.

#### **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 DataConnect 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*

### 17.3.3 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 DataConnect 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 DataConnect 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 DataConnect 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 DataConnect 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 DataConnect 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.



### 17.3.4 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 DataConnect statistic counters.

#### **dump traces**

Displays all traces and their current values.

#### **dump users**

Displays information about any connected users.

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

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

#### **getstarttime**

Displays the time that the DataConnect 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 DataConnect on or off. If no RIC is specified, this applies to all DataConnect 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.



### 17.3.5 Recovery commands

#### **clearrecovery**

If issued of a *master* DataConnect, this command removes all outstanding items received from the recovery system pending insertion into the data file. If executed of a recovery DataConnect, 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* DataConnect 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.



## 18 General Troubleshooting

### 18.1 FAQ

***I have started the DataConnect but I do not appear to be getting any data***

This is normally because of a problem configuring DataConnect to your specific Reuters infrastructure, please refer to “Advanced configuration and troubleshooting of DataConnect for Thomson Reuters Data” on page 106 for a list of common installation problems and their solutions for each of the main Reuters infrastructures.

***DataConnect 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 DataConnect 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 DataConnect 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 DataConnect 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, DataConnect 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 DataConnect:

```
*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 DataConnect running in an RWSHi environment where the VPN tunnel gets cut every 16 hours. For some reasons, DataConnect sometimes does not automatically re-establish the connection so that all my instruments are not updating anymore meaning I have to restart DataConnect. 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 DataConnect 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. DataConnect 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 DataConnect 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 DataConnect data cache file for another type of file. If you have a file in your DataConnect installation directory called `mdsdb.gdb`, simply stop DataConnect, 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 DataConnect.

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

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

When the machine starts up, DataConnect 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 DataConnect.

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

***Since I have upgraded my DataConnect from version 5.x to 5.3, connected Tradesignal clients cannot access symbols from the Reuters data feed anymore***

Since version 5.2, DataConnect performs DACS permission checks on behalf of connected Tradesignal clients. Since version 5.3, DataConnect assumes that all configured Reuters feeds can perform DACS checks using DACS daemons present on the configured sink distributor hosts. However, if your DACS daemon is configured on another server, you will have to change this in the DataConnect console's data feed settings. See sections 6.1 (Setting up Thomson Reuters) and 18.2.4 for further details.

***Since I have upgraded my DataConnect using the Trayport data feed from version 5.x to 5.5, I get warning messages in the Log that certain trader information was not found***

You have probably configured a customized SQL statement for ODBC backfill retrieval in the TMDSTRPT.INI file (section [Backfill], setting SQLDatedTickQueryVoice).

Starting from version 5.5, DataConnect is able to provide information about the participants of a trade which can then be charted using Tradesignal. The required data fields can be retrieved from Aggressor and Initiator User Ids available in the SQL database as follows:

Querying information for BYID (Buyer Id):

```
CASE WHEN [DealTreeNodes].[BuySell] = 1 THEN [DealsDone].[InitUserId]
ELSE [DealsDone].[AggUserId] END AS BYID
```

Querying information for SLID (Seller Id):

```
CASE WHEN [DealTreeNodes].[BuySell] = 1 THEN [DealsDone].[AggUserId]
ELSE [DealsDone].[InitUserId] END AS SLID
```

These SQL fragments have to be inserted in the *SELECT* section of your customized SQL query statement.



## 18.2 Advanced configuration and troubleshooting of DataConnect for Thomson Reuters Data

### 18.2.1 Troubleshooting failed connections to Triarch/RWS (SSL4)

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

Should connecting to the feed infrastructure fail for some reason, the DataConnect 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*.

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

*The DataConnect Console allows configuring datafeeds in a simple and straight-forward manner. A manual configuration is seldom necessary. DataConnect 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 DataConnect to a RMDS (SASS3) infrastructure using a configuration file (usually called *ss/rv.cnf* for RMDS-based infrastructures). As with all feed infrastructures, you must have correctly configured your DACS settings in the UI.

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

In the Manage Datafeeds pane, pick your *ss/rv.cnf* file in the **Non-default SSL configuration file** field.



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

Should start up fail for some reason, DataConnect 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 `sslr.cnf` 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 `sslr.cnf` 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 `sslr.cnf` 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
```



### 18.2.3 Manually configuring TREP-RT RSSL/OMM connections

*The DataConnect Console allows configuring datafeeds in a simple and straight-forward manner. A manual configuration is seldom necessary. DataConnect nevertheless supports manual configuration of an RSSL/OMM data feed using a user-provided **RSSL configuration file**.*

This section describes the steps necessary to manually set up a connection of DataConnect to an RSSL (OMM) infrastructure using a configuration file. As with all feed infrastructures, you must have correctly configured your DACS settings in the UI. In the datafeed wizard (when setting up a new RSSL feed) choose “*Configuration via a user-provided RSSL configuration file*” instead of “*Automatic Configuration*”. Enter the name of the configuration file residing on the same machine as DataConnect. When the Console is installed locally as well, you can use the file picker dialog (click ... next to the edit box).

The configuration file entries must adhere to the Thomson Reuters specifications as defined in the RFA configuration guide document. DataConnect automatically creates a minimal configuration file when the Automatic Configuration option is activated. This file can be used as a basis for additional configuration, if necessary.

Sample configuration file:

```
\Sessions\Session_IDN_SELECTFEED\connectionList = "Connection_RSSL_IDN_SELECTFEED"
\Connections\Connection_RSSL_IDN_SELECTFEED\connectionType = "RSSL"
\Connections\Connection_RSSL_IDN_SELECTFEED\serverList = "sink_dist1"
\Logger\AppLogger\windowsLoggerEnabled = false
```

The configuration must define a so-called Session (called `Session_IDN_SELECTFEED` above) referring to a connection (named `Connection_RSSL_IDN_SELECTFEED` in the example) which then has to be specified further. The Session name has to be specified in the wizard dialog so that DataConnect is able to pick the correct connection.

For a full list of configuration options contact your Thomson Reuters account manager.

### 18.2.4 Entitlement Management with DACS

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

*DataConnect 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 DataConnect. 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 DataConnect. This service is called *DataConnect Entitlement Service* and its lifetime is managed transparently by DataConnect without the need for user intervention. DataConnect talks to the Entitlement Service 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 DataConnect Console’s *Manage Datafeeds* dialog in the Reuters shared settings section (local DataConnect Entitlement Service Port).



The Entitlement Service will establish permanent connections to all configured DACS daemons. Should a connection fail, DataConnect 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 DataConnect log and alerts are sent out to connected clients using these feeds.

When a user connected to DataConnect from a remote host requests accessing a symbol's data (for example a Tradesignal user opening a chart), DataConnect will forward the symbol request to the Entitlement Service which in turn will interact with the DACS infrastructure to request a DACS lock on this symbol. When the DACS rejects accessing this symbol, DataConnect 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 DataConnect console (the default is set to 30 seconds).

#### **18.2.4.2 Connecting Tradesignal 6.1 or earlier to DataConnect 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. DataConnect 5.2 and above will detect that an older Tradesignal client is connected and will therefore skip the entitlement checks for those clients.



### 18.2.4.3 Connecting Tradesignal 6.2 or later to DataConnect 5.1 or earlier

DataConnect 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 the Tradesignal directory (under *Programs\Tradesignal*).

The following SFC components and configuration files are needed:

- *dacs4w32.dll*
- *sipc3290.dll*
- *ssl48w3290.dll*
- *appendix\_a*
- *enumtype.def*
- *sslapi.cnf*

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

Note that Tradesignal does not need to perform client-side permissioning when connected to a DataConnect 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 DataConnect.

### 18.2.5 How to fix the “Thomson Reuters Excel Add-In not activated” error when using the Eikon data feed.

Sometimes Microsoft Excel automatically deactivates Add-Ins if it detects that it crashed while an Add-In was running. If this happens to the Thomson Reuters Excel Add-In the data feed will not be able to download instrument histories anymore. When this problem is detected an error notification will be raised to inform the user. In order to reactivate the Add-In the following steps should be executed.

1. **Stop the Eikon data feed.**  
This can be done by running the “Stop EikonConnect” from the start menu or using the “Stop Feed Handler” button of the DataConnect Console’s Eikon data feed settings dialog.
2. **Close all Excel instances.**  
Make sure all running instances of Microsoft Excel are closed. You can use the Windows task manager to check if there are any *excel.exe* instances running in the background.
3. **Run “Thomson Reuters Eikon Excel” from the start menu.**  
Starting Excel via the start menu entry will forcefully activate the Eikon Add-In.
4. **Close Excel**  
This will save the activated state of the Eikon Add-In for future Excel instances.
5. **Start the Eikon data feed.**  
Run the “Restart EikonConnect” from the start menu or use the “Start Feed Handler” button of the DataConnect Console’s Eikon data feed settings dialog.



## Appendix A: Supported Trade Units

Following is a list of trade units supported by DataConnect for unit conversion purposes:

Trade Unit Name	Trade Unit Key
Tonne	Ton
Kilo Tonne	kt
Mega Tonne	Mt
Kilogram	Kg
Gram	Grm
Pound	Lbs
Troy Ounce	Ozs
Hundred Wt	Cwt
Short Ton	STn
Long Ton	LTn
30kg Bag	30B
50kg Bag	50B
60kg Bag	60B
70kg Bag	70B
100kg Bag	100B
53.5kg Bag	53B
Bushel	Bsh
Barrel	Bbl
Kilo Barrel	kBbl
US Gallon	USG
UK Gallon	UKG
Litre	Lt
Hecto Litre	HLt
Cubic Metre	M3
Cubic Feet	Ft3
British Thermal Units	BTU
Million British Thermal Units	MMBTU
Therm	Thm
Kilo Therm	kThm
Giga Joule	GJ
Tera Joule	TJ
Mega Watt Hour	MWh
Giga Watt Hour	GWh
Percentage	%

