



.NET FastTrade API

Programming Manual

Version 1.3.0.1

Table of Contents

Introduction	1
ListGroup & FastTrade Contacts	1
Related Documentation	1
Overview	2
FastTrade Overview	2
Data Distribution	4
Api Details	5
ftapi dnet.Namespace	8
Connection Class	14
Connection.MonitorEventArgs Class	35
Connection.OpenEventArgs Class	46
ConnectionActivity Class	59
ConnectionContext Class	63
Disposable Class	69
Entity Class	81
EntityClass Class	153
EntityField Class	169
FastTrackEventArgs Class	181
FastTrackException Class	188
Filter Class	203
Filter.CloseEventArgs Class	221
Filter.OpenEventArgs Class	225
Filter.SetEventArgs Class	229
KeyValue Class	233
Lib Class	240
Lib.CommunicationExceptionHandler Delegate	265
Lib.CommunicationIdleHandler Delegate	267
Lib.TraceEventArgs Class	269
Mask Class	275
Query Class	285
Query.CloseEventArgs Class	308
Query.NotifyEventArgs Class	312
Query.OpenEventArgs Class	321
Query.RowsEventArgs Class	329
Revision Class	333
Subscription Class	345
Subscription.CloseEventArgs Class	381
Subscription.NotifyEventArgs Class	385
Subscription.OpenEventArgs Class	393
TimeStamp Class	399
Transaction Class	410
Transaction.QueryEventArgs Class	432
Transaction.SendEventArgs Class	434
Transaction.TransactionEventArgs Class	436
TransactionID Class	446
IDump Interface	458
Connection.BrokenEventHandler Delegate	462
Connection.CloseEventHandler Delegate	463

Table of Contents

ftapi_dnet Namespace

<u>Connection.MonitorEventHandler Delegate</u>	464
<u>Connection.OpenEventHandler Delegate</u>	465
<u>Filter.CloseEventHandler Delegate</u>	466
<u>Filter.OpenEventHandler Delegate</u>	467
<u>Filter.SetEventHandler Delegate</u>	468
<u>Lib.TraceEventHandler Delegate</u>	469
<u>Query.CloseEventHandler Delegate</u>	470
<u>Query.NotifyEventHandler Delegate</u>	471
<u>Query.OpenEventHandler Delegate</u>	472
<u>Query.RowsEventHandler Delegate</u>	473
<u>Subscription.BrokenEventHandler Delegate</u>	474
<u>Subscription.CloseEventHandler Delegate</u>	475
<u>Subscription.IdleEventHandler Delegate</u>	476
<u>Subscription.NotifyEventHandler Delegate</u>	477
<u>Subscription.OpenEventHandler Delegate</u>	478
<u>Transaction.QueryEventHandler Delegate</u>	479
<u>Transaction.SendEventHandler Delegate</u>	480
<u>ContextAttribute Enumeration</u>	481
<u>EntityAction Enumeration</u>	483
<u>EntityClassType Enumeration</u>	485
<u>EntityFieldType Enumeration</u>	486
<u>Error Enumeration</u>	490
<u>FilterClose Enumeration</u>	492
<u>FilterOpen Enumeration</u>	493
<u>FilterSet Enumeration</u>	494
<u>QueryClose Enumeration</u>	495
<u>QueryOpen Enumeration</u>	496
<u>QueryRows Enumeration</u>	497
<u>QueryType Enumeration</u>	498
<u>Status Enumeration</u>	499
<u>SubscribeFlow Enumeration</u>	502
<u>ThreadingMode Enumeration</u>	503
<u>TraceLevel Enumeration</u>	504
<u>TraceOptions Enumeration</u>	505
<u>TraceSource Enumeration</u>	506
<u>TransactionNamStatus Enumeration</u>	507
<u>TransactionStatus Enumeration</u>	508
<u>UserType Enumeration</u>	510

Introduction

This manual describes version 1.3.0.1 of the *ftapi_dnet*, the .NET FastTrade Application Program Interface, developed by LIST within FastTrade to access electronic markets and other services handled by FastTrade.

To start to use this library please read the [Overview](#).

ListGroup & FastTrade Contacts

Requests for clarifications, comments and suggestions to improve the quality of the product are welcome. Please contact us:

LIST SpA
Via Pietrasantina, 123
56122 PISA – ITALY

Or contact us by email:

Marketing:	marketing@list.it
General Support:	helpdesk@list.it
FT/API Programming Support:	ftapi@list.it

Or visit us at www.listgroup.it

Related Documentation

FastTrade FTAPI.pdf – FT/API Programmer's Guide – V 3.5.1

The traditional C interface to FastTrade.

FastTrade White Paper

Introduction to FastTrade technology.

Overview

This manual describes version 1.3.0.1 of the *ftapi_dnet*, the .NET FastTrade Application Program Interface, developed by LIST within FastTrade to access electronic markets and other services handled by FastTrade.

Data structure and functions of *ftapi_dnet* are described, along with the main concepts regarding access to the FastTrade server: connections, data subscriptions, transactions, queries, etc...

The first section [FastTrade Overview](#) introduces the main basic concepts of FastTrade with a short overview of system architecture.

The second section [Data Distribution](#) explains how data are distributed, searched and retrieved inside FastTrade.

Section three [Api Details](#) describes the main peculiarities of this Access Point to FastTrade: lifecycle, communication model, exception handling, threads synchronization, etc...

Following chapters contain the effective and detailed descriptions of the Api available in the [ftapi_dnet Namespace](#).

Note – ASIA –

Applications using FTApi are now able to connect to a FastTrack platform that supports ASIA technology. With version 3.5.1 of the FTApi, users can exploit both ASIA and non- ASIA connections (R2 and R3) within the same application, without worrying about the management of the dual access connection, public and private, typical of ASIA. The dual ASIA connection is presented in terms of a normal conversation, so that users can use the conversation-Id obtained during the connection and achieve their needs with conventional FTApi methods. To route to the appropriate access, FTApi uses a text file that contains the rules by which a given class-id is sent via one connection rather than on the other one (eg: PUBLMETAMARKET-PRIVMETAMARKET.routing.properties). An ASIA connection can be made via FTApi exclusively by the service name and never directly with an IP-address and a port. The service name string has two tokens "FIRSTSERV|SECONDSERV" separated by a pipe. In the working directory of the user's application, there must be a file named FIRSTSERV-SECONDSERV.routing.properties that defines class-id routing rules. For example, the connection to the MetaMarket service can be done setting the "PUBLMETAMARKET|PRIVMETAMARKET" string to the service name param of the FTOpenConversationExt().

FastTrade Overview

This chapter explains the basic concepts of FastTrade that are fundamental in developing Java applications that access FastTrade services using this Api.

System Architecture Overview

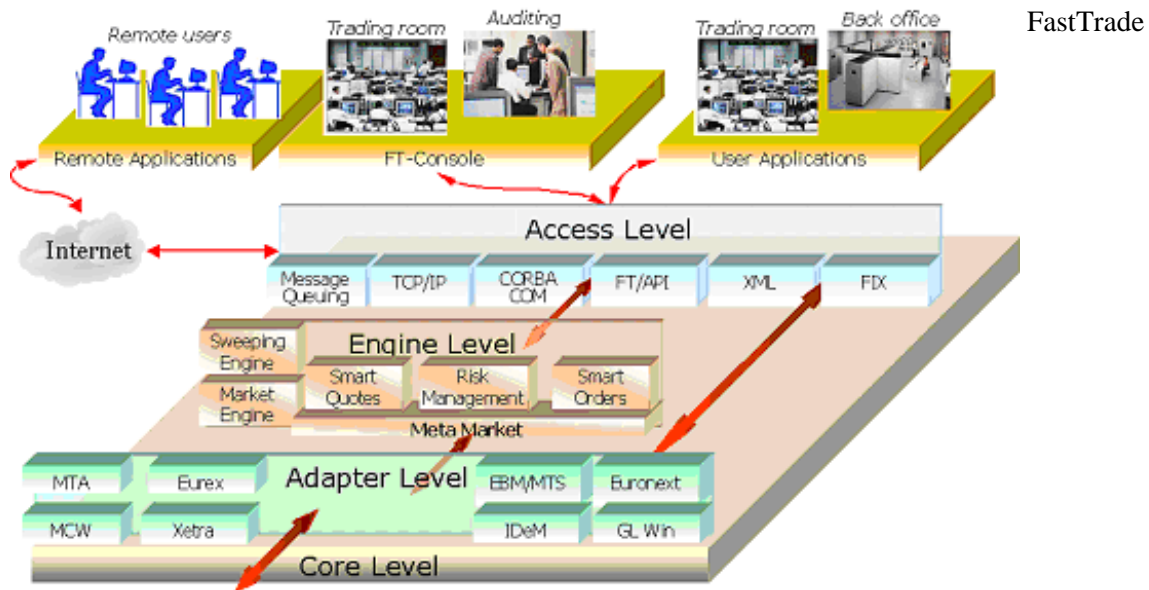
FastTrade's architecture is **modular** and **distributed**.

There are three different architectural levels for the various components, depending on the service they implement:

- Core
- Basic
- Enterprise

The **core level** is the heart of the system, the foundation of FastTrade's modular architecture.

The set of FastTrade modules managing communication with the external world makes up the **basic level**. These are organized into two categories: **Adapters**, through which FastTrade can access other systems or electronic markets, and **Access points**, which give access to FastTrade from the outside. Finally, the **enterprise level** hosts the functional applications in FastTrade (Engine level).



's components interact both in a synchronous and asynchronous way – the latter uses a **publish–subscribe** paradigm.

FastTrade is a distributed system. The computing process is subdivided into several steps. It is not performed by one individual component, but by a series of elements (Application Servers or **Services**) each designed to carry out its particular part of the process.

Access points

The Access Point Level is the only interface FastTrade offers to use its services from the outside.

Queries made to the FastTrade platform and answers to these queries both go through the Access Point Level.

The Access Point Level:

- provides access to external applications which use communication protocols which are very different from one another, thus guaranteeing their complete access to FastTrade's functionalities;
- manages connections coming from external applications;
- manages in a centralized way the sessions that have been opened on FastTrade, by users connected via proprietary applications, or the FastTrade Console, or a Web browser, independently of the communication protocol;
- provides all functionalities needed by the external application to utilize the services offered by internal engines, hence:
 - ◆ transactions to send requests;
 - ◆ real–time data distribution mechanisms (for example push on HTML);
 - ◆ mechanisms to perform point to point requests (e.g. Query).

.NET FastTrade Access Point

This *ftapi_dnet* library is just a FastTrade Access Point that offers a .NET interface to be used in order to connect FastTrade servers via TCP/IP connections.

Using this library it is possible to construct applications (written in a .NET language: C#, C++, Visual Basic, JScript, etc...) that may communicate with one or more FastTrade servers and/or services.

Obviously the effective access to these services is managed/controlled/verified in relation to various credentials (user names, passwords, authorization keys, etc...) that an external application must presents in order to be properly authorized to enter in the system.

Data Distribution

Data are distributed through a publish/subscribe protocol, in which producers and consumers exchange messages to access the data. There are many data structure exchanged within FastTrade. Each data structure is called [EntityClass](#) and it is structured in many fields of many different types (numbers, strings, etc...). Each set of values that corresponds to these fields is an instance of the EntityClass. This instance is called [Entity](#). Applications may modify or access an entire Entity (all the fields of an instance of an EntityClass) or a subset of these fields defined using a mask.

From now on we call:

- *client* any .NET application that use *ftapi_dnet*,
- *server* any FastTrade service that is connected to the *client*.

As we will see below, normally a client is a subscriber, and the server is a producer, of a set of data which are exchanged between them.

Publish

The producer (typically a FastTrade service) notifies the availability of new data with a publish message. These messages are sent to all connected components (other Fasttrack services and/or *ftapi_dnet* applications) that expressed interest on these type of data.

Example:

a FastTrade order-manager publishes all records that describes new received or changed orders.

Only certain EntityClasses of a FastTrade server may be published and then subscribed. The documentation of each FastTrade server clearly says which EntityClasses can be published/subscribed.

Subscribe

A request for data by consumers is done by a subscribe message. This message typically says in which EntityClass the consumer is interested.

Example:

a customer subscribes to the orders handled by FastTrade.

Subscriptions in *ftapi_dnet* are modelled by [Subscription](#). Among other things within subscriptions it's possible to have:

- [Incremental Subscriptions](#), in which the server is only required to send updated contents of an EntityClass, rather than sending all its records;
- [Partial Subscriptions](#), in which the server is requested to send only those entities in a EntityClass that satisfy certain constraints;
- an optional [Filter](#) to restrict (at the server level) the set of entities that will be notified;
- an optional [Mask](#) to restrict (at the server level) the set of fields of entities that will be notified.

Queries

In addition to the publish/subscribe mechanism a client has the possibility to obtain from the server a specific set of entities. This is done with the [Query](#) metaphor that mimics the homonymous facility of DBMS. Normally each type of query (identified by a specific number) has an argument that specify the particular request.

Example:

A client queries for all orders sent by a specific operator.

Only certain queries (each identified by a unique number) are permitted with a FastTrade server. The documentation of each FastTrade server clearly lists which queries (i.e. numbers) are permitted and with which arguments.

Transactions

The client may request a server to make some actions that may result in the update of one or more entities. The server evaluates the request and accepts or refuses it. Accepting the modification often implies application specific check actions by the server with the aim of controlling the access rights and action consistency while interpreting the semantics of the request arrived from the client. This is modeled in *ftapi_dnet* with the [Transaction](#) metaphor.

Example:

the client asks the server to issue an order on a specific product.

This operation may take a long period to be completed by the server and so the client have to monitor it until a good or bad (committed vs aborted) final result. This monitoring must be always done by the client, even after a client–restart on previous initiated (past) transactions. Only when the client see the final committed result it may assume that the transaction was succesfully.

Connections

All the above described capabilities are communicated between the client .NET application (that use this *ftapi_dnet* library) and the FastTrade server (a specific service of a FastTrade server) using a TCP/IP channel modelled with a [Connection](#) . The main attributes of such Connections are the TCP/IP host and port on which a named service reside.

Example:

an order–manager service may reside on port 1234 of host `myFTserver.myDomain.com` (or something like 10.91.195.33) and beeing named `OrderManager`

Api Details

This *ftapi_dnet* library may be used with .NET SDK version 1.1 and following.

Required Libraries

The actual implementation of this *ftapi_dnet* library is based on *The traditional C interface to FastTrade* so, in order to successfully compile and run any application based on *ftapi_dnet* the following libraries are needed:

- **FTApi30.dll Version >= 3.5.1**: The traditional C interface to FastTrade.
- **ftapi_dnet.dll**: this .NET FastTrade API.
- a set of **FastTrade market libraries**: a library for each different FastTrade market/service to link.

Please see the *Prerequisites* sections of C# Application Examples to have a full description about to compile and execute an application based on the *ftapi_dnet* library.

Which Languages?

A main .NET feature is the *Common Language Infrastructure* CLI, i.e. a specification describing how applications written in *multiple* high-level languages may be executed in different system environments without the need to rewrite the applications to take into consideration the unique characteristics of those environments.

As an important .NET CLI side-effect this *ftapi_dnet* library may be linked and used to write FastTrade application programs in any language supported by .NET: e.g. C#, C++, Visual Basic, JScript, etc...

All examples given in this manual are actually written in C# but they may be easily re-written in C++ (or in other languages) without the need to have a different *ftapi_dnet* library:

The *same* ftapi_dnet.dll library may be used from different applications written in different languages.

Asynchronous Communication Model

The implementation of *ftapi_dnet* functionalities is based on an asynchronous communication model. A functionality (such as connection opening or a subscription for a set of data) is requested to the library via a method invocation. When the data or answers arrive from the server an appropriate event is raised and then the .NET delegates (i.e. notification methods) are automatically invoked on the appropriate objects.

LifeCycle

Many objects exposed by the *ftapi_dnet* library share a common life cycle metaphor: once they are created, their life goes through well defined steps depending on their internal status.

See the [Disposable LifeCycle](#) to understand how this behaviour is controlled and regulated.

Other Peculiarities

Some other details of the *ftapi_dnet* implementation are referenced here in order to use at the best the library:

- [Implementation Threads](#) describes how the library use its own background threads,
- [Segments](#), [Keys](#) and [KeyValues](#) introduce some common definitions used in [Subscription](#) and [Mask](#).

Entry Point

Last but not least:

Where is the first initial entry point to use this library?

It's available in the singleton that implements the [Lib](#). Using that singleton, referenced by the [lib constant](#), every programmer may starts to use the library accessing all its functionalities.

ftapi_dnet Namespace

This is the set of types available in the .NET FastTrack API.

To start to use this library please read the [Overview](#) or watch the source-code examples available in FTApi .Net Application Examples Manual

[Namespace hierarchy](#)

Classes

Class	Description
Connection	Logical bidirectional channel with a server.
Connection.MonitorEventArgs	Nam transaction status data.
Connection.OpenEventArgs	Connection server answer to open .
ConnectionActivity	Super-class common to all objects of a given Connection .
ConnectionContext	Container for additional Connection attributes.
Disposable	Common super-class for all objects that have a life cycle.
Entity	Value of an EntityClass .
EntityClass	Entity 's and EntityField 's type.
EntityField	A field of an EntityClass .
FastTrackEventArgs	FastTrackEventArgs is the FastTrack base class for classes containing event data.
FastTrackException	The exception that is thrown when a non-fatal FastTrack application error occurs.
Filter	A manner to restrict the set of values notified by a Subscription .
Filter.CloseEventArgs	Filter server answer to Close .
Filter.OpenEventArgs	Filter server answer to open .
Filter.SetEventArgs	Filter server answer to set .
KeyValue	A (partial or full) value of a key of an EntityClass.
Lib	Main entry-point to use .NET FastTrack API.
Lib.TraceEventArgs	Data to be traced by the application.
Mask	A set of segments of a EntityClass .
Query	A client's request to a server to obtain a set of entities (or rows) from its own Data Base.
Query.CloseEventArgs	Query server answer to Close .
Query.NotifyEventArgs	Query data coming from the server as result of a open or queryRows .
Query.OpenEventArgs	Query server answer to open .
Query.RowsEventArgs	Query server answer to queryRows .
Revision	A market/service version.
Subscription	An arrangement with the server for receiving a continuing set of interesting entities of the same EntityClass .

Subscription.CloseEventArgs	Subscription server answer to Close .
Subscription.NotifyEventArgs	Subscription data coming from the server as result of a open or refreshEntity
Subscription.OpenEventArgs	Subscription server answer to open .
TimeStamp	A temporal indicator.
Transaction	A client's request to the server to make an operation and/or to add/remove/modify an entity in its own Data Base.
Transaction.QueryEventArgs	Transaction server answer to queryStatus .
Transaction.SendEventArgs	Transaction server answer to send .
Transaction.TransactionEventArgs	Transaction server answer to send/queryStatus .
TransactionID	An unique Transaction identifier.

Interfaces

Interface	Description
IDump	Defines a method to retrieve a multi-line string representation of an object.

Delegates

Delegate	Description
Connection.BrokenEventHandler	Connection callback method to be called when the connection with the server crashed in an unrecoverable manner.
Connection.CloseEventHandler	Connection callback method to be called when the server answer to Close arrives.
Connection.MonitorEventHandler	Connection callback method to be called when there is a status-change on a NAM-market transaction monitored by monitorNamTransactions .
Connection.OpenEventHandler	Connection callback method to be called when the server answer to open arrives.
Filter.CloseEventHandler	Filter callback method to be called when the server answer to Close arrives.
Filter.OpenEventHandler	Filter callback method to be called when the server answer to open arrives.
Filter.SetEventHandler	Filter callback method to be called when the server answer to set arrives.
Lib.CommunicationExceptionHandler	Lib callback delegate method called when communication reach an exception. Operative in multipleThreads mode only.
Lib.CommunicationIdleHandler	Lib callback delegate method called when communication is in idle. Operative in multipleThreads mode only.
Lib.TraceEventHandler	Lib callback method to be called when a trace message must be programmatically traced as requested by setTraceMode .
Query.CloseEventHandler	Query callback method to be called when the server answer to Close arrives.
Query.NotifyEventHandler	Query callback method to be called when a new data comes from the server as result of a open or queryRows .

Query.OpenEventHandler	Query callback method to be called when the server answer to open arrives.
Query.RowsEventHandler	Query callback method to be called when the server answer to queryRows arrives.
Subscription.BrokenEventHandler	Subscription callback method to be called when the connection with the server crashed or when the server choose to terminate the connection.
Subscription.CloseEventHandler	Subscription callback method to be called when the server answer to Close arrives.
Subscription.IdleEventHandler	Subscription callback method to be called when the flow of historical data is finished and the flow of actual data is starting.
Subscription.NotifyEventHandler	Subscription callback method to be called when a new data comes from the server as result of a open or refreshEntity .
Subscription.OpenEventHandler	Subscription callback method to be called when the server answer to open arrives.
Transaction.QueryEventHandler	Transaction callback method to be called when the server answer to queryStatus arrives.
Transaction.SendEventHandler	Transaction callback method to be called when the server answer to send arrives.

Enumerations

Enumeration	Description
ContextAttribute	Additional attributes to associate to a new Connection.
EntityAction	Request/Notification about an Entity operation.
EntityClassType	EntityClass type.
EntityFieldType	EntityField type.
Error	Failure result–code.
FilterClose	Server filter–destruction failure–code.
FilterOpen	Server filter–creation failure–code.
FilterSet	Server filter–extension failure–code.
QueryClose	Server query–destruction failure–code.
QueryOpen	Server query–creation failure–code.
QueryRows	Server query–rows failure–code.
QueryType	Subscription query–selection criteria.
Status	Status of every Disposable object.
SubscribeFlow	Subscription data–transmission policy.
ThreadingMode	Library threading behaviour.
TraceLevel	Trace message level.
TraceOptions	Trace message options.
TraceSource	Trace message category.
TransactionNamStatus	Status of a transaction sent to a NAM market.
TransactionStatus	Status of a transaction sent to a server.
UserType	Activities that can be made on a Connection .

ftapi_dnet Hierarchy

System.Object

[ftapi_dnet.Disposable](#) ----- [System.IDisposable](#)

[ftapi_dnet.Connection](#)

[ftapi_dnet.ConnectionActivity](#)

[ftapi_dnet.Filter](#)

[ftapi_dnet.Query](#)

[ftapi_dnet.Subscription](#)

[ftapi_dnet.Transaction](#)

[ftapi_dnet.ConnectionContext](#)

[ftapi_dnet.KeyValue](#)

[ftapi_dnet.Lib](#)

[ftapi_dnet.Mask](#)

[ftapi_dnet.Entity](#) ----- [ftapi_dnet.IDump](#), [System.ICloneable](#)

[ftapi_dnet.EntityClass](#)

[ftapi_dnet.EntityField](#)

[ftapi_dnet.IDump](#)

[ftapi_dnet.Revision](#) ----- [System.IComparable](#)

[ftapi_dnet.TimeStamp](#) ----- [System.IComparable](#)

[ftapi_dnet.TransactionID](#)

[System.Delegate](#) ----- [System.ICloneable](#), [System.Runtime.Serialization.ISerializable](#)

[System.MulticastDelegate](#)

[ftapi_dnet.Connection.BrokenEventHandler](#)

[ftapi_dnet.Connection.CloseEventHandler](#)

[ftapi_dnet.Connection.MonitorEventHandler](#)

[ftapi_dnet.Connection.OpenEventHandler](#)

[ftapi_dnet.Filter.CloseEventHandler](#)

[ftapi_dnet.Filter.OpenEventHandler](#)

[ftapi_dnet.Filter.SetEventHandler](#)

[ftapi_dnet.Lib.CommunicationExceptionHandler](#)

[ftapi_dnet.Lib.CommunicationIdleHandler](#)

[ftapi_dnet.Lib.TraceEventHandler](#)

[ftapi_dnet.Query.CloseEventHandler](#)

[ftapi_dnet.Query.NotifyEventHandler](#)

[ftapi_dnet.Query.OpenEventHandler](#)

[ftapi_dnet.Query.RowsEventHandler](#)

[ftapi_dnet.Subscription.BrokenEventHandler](#)

[ftapi_dnet.Subscription.CloseEventHandler](#)

[ftapi_dnet.Subscription.IdleEventHandler](#)

[ftapi_dnet.Subscription.NotifyEventHandler](#)

[ftapi_dnet.Subscription.OpenEventHandler](#)

[ftapi_dnet.Transaction.QueryEventHandler](#)

[ftapi_dnet.Transaction.SendEventHandler](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#) ----- [ftapi_dnet.IDump](#)

[ftapi_dnet.Connection.MonitorEventArgs](#)

[ftapi_dnet.Connection.OpenEventArgs](#)

[ftapi_dnet.Filter.CloseEventArgs](#)

[ftapi_dnet.Filter.OpenEventArgs](#)

[ftapi_dnet.Filter.SetEventArgs](#)
[ftapi_dnet.Lib.TraceEventArgs](#)
[ftapi_dnet.Query.CloseEventArgs](#)
[ftapi_dnet.Query.NotifyEventArgs](#)
[ftapi_dnet.Query.OpenEventArgs](#)
[ftapi_dnet.Query.RowsEventArgs](#)
[ftapi_dnet.Subscription.CloseEventArgs](#)
[ftapi_dnet.Subscription.NotifyEventArgs](#)
[ftapi_dnet.Subscription.OpenEventArgs](#)
[ftapi_dnet.Transaction.TransactionEventArgs](#)
[ftapi_dnet.Transaction.QueryEventArgs](#)
[ftapi_dnet.Transaction.SendEventArgs](#)
System.Exception ----- System.Runtime.Serialization.ISerializable
System.ApplicationException
[ftapi_dnet.FastTrackException](#)
System.ValueType
System.Enum ----- System.IComparable, System.IConvertible, System.IFormattable
[ftapi_dnet.ContextAttribute](#)
[ftapi_dnet.EntityAction](#)
[ftapi_dnet.EntityClassType](#)
[ftapi_dnet.EntityFieldType](#)
[ftapi_dnet.Error](#)
[ftapi_dnet.FilterClose](#)
[ftapi_dnet.FilterOpen](#)
[ftapi_dnet.FilterSet](#)
[ftapi_dnet.QueryClose](#)
[ftapi_dnet.QueryOpen](#)
[ftapi_dnet.QueryRows](#)
[ftapi_dnet.QueryType](#)
[ftapi_dnet.Status](#)
[ftapi_dnet.SubscribeFlow](#)
[ftapi_dnet.ThreadingMode](#)
[ftapi_dnet.TraceLevel](#)
[ftapi_dnet.TraceOptions](#)
[ftapi_dnet.TraceSource](#)
[ftapi_dnet.TransactionNamStatus](#)
[ftapi_dnet.TransactionStatus](#)
[ftapi_dnet.UserType](#)

See Also

[ftapi_dnet Namespace](#)

Connection Class

Logical bidirectional channel with a server.

For a list of all members of this type, see [Connection Members](#).

System.Object

[ftapi_dnet.Disposable](#)

ftapi_dnet.Connection

public class Connection : [Disposable](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

Within the same application several Connections can be [created](#) and then [opened](#) with the same Server or with several Servers.

Every connection is described/defined by/with the **open** parameters:

- **(ipAddress,ipPort) pair** – used as principal server to which the client must talk,
- **clientID** – used to identify the client,
- **(userName, userPassword) pair** – used to describe the user associated to the connection,
- **userType** – used to [identify which activities](#) will be done on the connection,
- **connectionContext** – used as [additional attributes](#) associated to the Connection,
- **applRevision** – optionally used as client [Revision](#),
- **applSignature** – optionally used as client signature by some FastTrack service/market,
- **service** – optionally used to specify a particular FastTrack service.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[Connection Members](#) | [ftapi_dnet Namespace](#)




Connection Members

[Connection overview](#)













Public Instance Constructors

 Connection Constructor	Initializes a new instance of the Connection class.
--	---




Public Instance Properties


 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

Public Instance Methods

 Close (inherited from Disposable)	Logically destroy the current object, making it again available.
 Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 monitorNamTransactions	Starts the monitoring of all transactions sent to NAM markets on this connection.
 onBroken	Method automatically invoked when the connection with the server crashed in an unrecoverable manner.
 onClose	Method automatically invoked when the server answer to Close arrives.
 onMonitor	Method automatically invoked when there is a status-change on a NAM-market transaction monitored by monitorNamTransactions .
 onOpen	Method automatically invoked when the server answer to open arrives.
 open	Overloaded. Opens this connection with a given server.
 ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

Public Instance Events

 brokenEvent	Unrecoverable event asynchronously raised when the connection with the server crashed.
 closeEvent	Event asynchronously raised when the server answer to Close arrives.
 monitorEvent	

	Event asynchronously raised when there is a status-change on a NAM-market transaction monitored by monitorNamTransactions .
 openEvent	Event asynchronously raised when the server answer to open arrives.

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#)

Connection Constructor

Initializes a new instance of the Connection class.

```
public Connection();
```

Remarks

Status

- on successfully Exit – [init](#)

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#)

Connection Methods

The methods of the **Connection** class are listed below. For a complete list of **Connection** class members, see the [Connection Members](#) topic.

Public Instance Methods

✦ Close (inherited from Disposable)	Logically destroy the current object, making it again available.
✦ Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
✦ Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
✦ GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
✦ GetType (inherited from Object)	Gets the Type of the current instance.
✦ monitorNamTransactions	Starts the monitoring of all transactions sent to NAM markets on this connection.
✦ onBroken	Method automatically invoked when the connection with the server crashed in an unrecoverable manner.
✦ onClose	Method automatically invoked when the server answer to Close arrives.
✦ onMonitor	Method automatically invoked when there is a status-change on a NAM-market transaction monitored by monitorNamTransactions .
✦ onOpen	Method automatically invoked when the server answer to open arrives.
✦ open	Overloaded. Opens this connection with a given server.
✦ ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[Connection Class](#) | [ftapi dnet Namespace](#)

Connection.monitorNamTransactions Method

Starts the monitoring of all transactions sent to NAM markets on this connection.

```
public virtual void monitorNamTransactions();
```

Remarks

This method must be used to check the final status of all transactions sent to NAM markets because the [OK](#) indication eventually returned in the [transactionStatus](#) field inside a [Transaction.TransactionEventArgs](#) only means a flying status and **not** a committed/aborted status.

If this method invocation completed successfully, then

- the request was accepted by the client,
- when a new transaction status will be available the [monitorEvent](#) will be raised.

otherwise

- the client rejected the request,
- no **monitorEvent** will be raised,
- an appropriate [FastTrackException](#) will be thrown.

In both cases the current [Status](#) remains unchanged.

Status

- required on Entry – [running](#)
- on successfully Exit – running

Events

Event Type	Reason
monitorEvent	Event asynchronously raised when there is a status-change on a NAM-market transaction monitored by <code>monitorNamTransactions</code> .

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#) | [onMonitor](#)

Connection.onBroken Method

Method automatically invoked when the connection with the server crashed in an unrecoverable manner.

```
public virtual void onBroken();
```

Remarks

The current [Status](#) changed to [init](#).

The [brokenEvent](#) (which causes the automatic invocation of this method) may be raised only the connection with the server crashed in an unrecoverable manner

This method must be properly extended in Connection sub-classes since it does nothing in the Connection class.

See Also

[Connection Class](#) | [ftapi dnet Namespace](#) | [Connection.BrokenEventHandler](#)

Connection.onClose Method

Method automatically invoked when the server answer to [Close](#) arrives.

```
public virtual void onClose();
```

Remarks

The current [Status](#) changed to [init](#).

This method must be properly extended in Connection sub-classes since it does nothing in the Connection class.

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#) | [Close](#) | [Connection.CloseEventHandler](#)

Connection.onMonitor Method

Method automatically invoked when there is a status-change on a NAM-market transaction monitored by [monitorNamTransactions](#).

```
public virtual void onMonitor(  
    MonitorEventArgs me  
);
```

Parameters

me
Nam transaction status data.

Remarks

The current [Status](#) remains [running](#).

This method must be properly extended in Connection sub-classes since it does nothing in the Connection class.

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#) | [monitorNamTransactions](#) | [Connection.MonitorEventHandler](#)

Connection.onOpen Method

Method automatically invoked when the server answer to [open](#) arrives.

```
public virtual void onOpen(  
    OpenEventArgs oe  
);
```

Parameters

oe
Server answer to [open](#).

Remarks

If *oe.result* is [OK](#) then

- the server accepted the connection creation,
- the current [Status](#) changed to [running](#),
- the server may now accept subscription/query/filter/transaction creation on this connection.

otherwise

- the server rejected the connection creation (see **result** to understand why),
- the server does not accept any subscription/query/filter/transaction creation on this connection.

This method must be properly extended in Connection sub-classes since it does nothing in the Connection class.

Example

```
public override void onOpen(OpenEventArgs oe)  
{  
    base.onOpen(oe);  
    if(oe.result == Error.OK)  
    {  
        Subscription oSub = new SubscriptionOrd(this);  
        Subscription mSub = new SubscriptionMrk(this);  
        oSub.open( "FT_C_ORDER" );  
        mSub.open( "FT_C_MARKET" );  
    }  
}
```

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#) | [open](#) | [Connection.OpenEventHandler](#)

Connection.open Method

Opens this connection with a given server.

Overload List

Shorthand method to open this connection with a given server using some defaults values.

[public virtual void open\(string,ushort,uint,string,string,UserType,ConnectionContext\);](#)

Full method to open this connection with a given server.

[public virtual void open\(string,ushort,uint,string,string,UserType,ConnectionContext,Revision,uint,string\);](#)

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#)

Connection.open Method (String, UInt16, UInt32, String, String, UserType, ConnectionContext)

Shorthand method to open this connection with a given server using some defaults values.

```
public virtual void open(  
    string ipAddress,  
    ushort ipPort,  
    uint clientID,  
    string userName,  
    string userPassword,  
    UserType userType,  
    ConnectionContext connectionContext  
);
```

Parameters

ipAddress

Server host.

ipPort

Server port.

clientID

Client identifier.

userName

User name.

userPassword

User password.

userType

[Identify which activities](#) will be done on this connection.

connectionContext

[Additional attributes](#) associated to this Connection.

Remarks

The invocation

```
open(ipAddress, ipPort, clientID, userName, userPassword,  
    userType, connectionContext)
```

is equivalent to

```
open(ipAddress, ipPort, clientID, userName, userPassword,  
    userType, connectionContext, null, 0, "")
```

so *please see* the [full open](#) method for a full detailed description of it.

Status & Events

Please see the **full open** method for a full detailed description of them.

Example

```
public ConnectionFromArgs(string[] args)
{
    using(ConnectionContext cc = new ConnectionContext())
    {
        cc.setAttribute(ContextAttribute.applAuthFile,    LIC_FILE);
        cc.setAttribute(ContextAttribute.alternativeHost1, args[4]);
        cc.setAttribute(ContextAttribute.alternativePort1, args[5]);
        open(args[2], UInt16.Parse(args[3]), clientID++, args[0], args[1], UserType.view, cc
            , null, 0, "METAMARKET" // to access a service via a specific ServiceName
        );
    }
}
```

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#) | [Connection.open Overload List](#)

Connection.open Method (String, UInt16, UInt32, String, String, UserType, ConnectionContext, Revision, UInt32, String)

Full method to open this connection with a given server.

```
public virtual void open(  
    string ipAddress,  
    ushort ipPort,  
    uint clientID,  
    string userName,  
    string userPassword,  
    UserType userType,  
    ConnectionContext connectionContext,  
    Revision applRevision,  
    uint applSignature,  
    string service  
);
```

Parameters

ipAddress

Server host.

ipPort

Server port.

clientID

Client identifier.

This number uniquely identifies the client connected to this connection.

This number is automatically used on [TransactionIDs](#) of all [Transactions](#) sent on this connection.

The server may subsequently return a [invalidCID](#) failure-code if it does not comply with this value.

userName

User name.

The server may subsequently return a [invalidUserName](#) failure-code if it does not comply with this value.

userPassword

User password.

The server may subsequently return a [invalidPassword](#) failure-code if it does not comply with this value.

userType

[Identify which activities](#) will be done on this connection.

The server may subsequently return a [invalidUserType](#) failure-code if it does not comply with this value.

connectionContext

[Additional attributes](#) associated to this Connection.

applRevision

Optional value indicating the client [Revision](#).

The server may subsequently return a [invalidRevision](#) failure-code if it does not comply with this value.

applSignature

Optional value used as client signature by some FastTrack service/market.

service

Optional value used to specify a particular FastTrack service.

Remarks

If this method invocation completed successfully, then

- the open request was sent to server,
- the current [Status](#) changes to [opening](#),
- when the server-answer will be available the [openEvent](#) will be raised.

otherwise

- the client rejected the open,
- the open request was **not** sent to server,
- the current **Status** remains unchanged,
- no **openEvent** will be raised,
- an appropriate [FastTrackException](#) will be thrown.

Status

- required on Entry – [init](#)
- on successfully Exit – [opening](#)

Events

Event Type	Reason
openEvent	Event asynchronously raised when the server answer to a successfully open arrives. Following this event the following event may be asynchronously raised:
brokenEvent	Event asynchronously raised when the connection with the server crashed or when the server choose to terminate the connection.

Example

The following code-fragment shows a [shorthand open](#) invocation.

```
public ConnectionFromArgs(string[] args)
{
    using(ConnectionContext cc = new ConnectionContext())
    {
        cc.setAttribute(ContextAttribute.applAuthFile,    LIC_FILE);
        cc.setAttribute(ContextAttribute.alternativeHost1, args[4]);
        cc.setAttribute(ContextAttribute.alternativePort1, args[5]);
        open(args[2], UInt16.Parse(args[3]), clientID++, args[0], args[1], UserType.view, cc,
            null, 0, "METAMARKET" // to access a service via a specific ServiceName
        );
    }
}
```





See Also

[Connection Class](#) | [ftapi_dnet Namespace](#) | [Connection.open Overload List](#) | [_onOpen](#) | [_onBroken](#)

Connection Events

The events of the **Connection** class are listed below. For a complete list of **Connection** class members, see the [Connection Members](#) topic.

Public Instance Events

 brokenEvent	Unrecoverable event asynchronously raised when the connection with the server crashed.
 closeEvent	Event asynchronously raised when the server answer to Close arrives.
 monitorEvent	Event asynchronously raised when there is a status-change on a NAM-market transaction monitored by monitorNamTransactions .
 openEvent	Event asynchronously raised when the server answer to open arrives.

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#)

Connection.brokenEvent Event

Unrecoverable event asynchronously raised when the connection with the server crashed.

public event [BrokenEventHandler](#) brokenEvent;

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#) | [onBroken](#)

Connection.closeEvent Event

Event asynchronously raised when the server answer to [Close](#) arrives.

```
public event CloseEventHandler closeEvent;
```

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#) | [onClose](#)

Connection.monitorEvent Event

Event asynchronously raised when there is a status-change on a NAM-market transaction monitored by [monitorNamTransactions](#).

public event [MonitorEventHandler](#) monitorEvent;

See Also

[Connection Class](#) | [ftapi_dnet Namespace](#) | [onMonitor](#)

Connection.openEvent Event

Event asynchronously raised when the server answer to [open](#) arrives.

```
public event OpenEventHandler openEvent;
```

Example

```
public OrderThread(Connection connectionOrder, bool disposeAtEnd)
{
    connection = connectionOrder;
    callDisposeAtEnd = disposeAtEnd;
    connection.openEvent += new Connection.OpenEventHandler(myOpenEventHandler);
}
```

See Also

[Connection Class](#) | [ftapi dnet Namespace](#) | [onOpen](#)

Connection.MonitorEventArgs Class

Nam transaction status data.

For a list of all members of this type, see [Connection.MonitorEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Connection.MonitorEventArgs

public class Connection.MonitorEventArgs : [FastTrackEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)









See Also

[Connection.MonitorEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onMonitor](#)





Connection.MonitorEventArgs Members

[Connection.MonitorEventArgs overview](#)

Public Instance Fields

 classID	ClassID on which the transaction was sent.
 entityAction	Requested EntityAction.
 keyID	KeyID on which the transaction was sent.
 localError	Local-error failure-code.
 marketError	Market-error failure-code.
 namError	NAM-error failure-code.
 status	Status of a transaction sent to a NAM market.
 transactionID	Associated TransactionID.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.









See Also

[Connection.MonitorEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onMonitor](#)

Connection.MonitorEventArgs Fields

The fields of the **Connection.MonitorEventArgs** class are listed below. For a complete list of **Connection.MonitorEventArgs** class members, see the [Connection.MonitorEventArgs Members](#) topic.

Public Instance Fields

 classID	ClassID on which the transaction was sent.
 entityAction	Requested EntityAction.
 keyID	KeyID on which the transaction was sent.
 localError	Local-error failure-code.
 marketError	Market-error failure-code.
 namError	NAM-error failure-code.
 status	Status of a transaction sent to a NAM market.
 transactionID	Associated TransactionID.

See Also

[Connection.MonitorEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onMonitor](#)

Connection.MonitorEventArgs.classID Field

ClassID on which the transaction was sent.

```
public readonly uint classID;
```

See Also

[Connection.MonitorEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.MonitorEventArgs.entityAction Field

Requested EntityAction.

```
public readonly EntityAction entityAction;
```

See Also

[Connection.MonitorEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.MonitorEventArgs.keyID Field

KeyID on which the transaction was sent.

```
public readonly uint keyID;
```

See Also

[Connection.MonitorEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.MonitorEventArgs.LocalError Field

Local-error failure-code.

```
public readonly uint LocalError;
```

Remarks

This field is meaning only if [status](#) is [aborted](#).

See Also

[Connection.MonitorEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.MonitorEventArgs.marketError Field

Market-error failure-code.

```
public readonly uint marketError;
```

Remarks

This field is meaning only if [status](#) is [Haborted](#).

See Also

[Connection.MonitorEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.MonitorEventArgs.NameError Field

NAM-error failure-code.

```
public readonly uint NameError;
```

Remarks

This field is meaning only if [status](#) is [Aborted](#).

See Also

[Connection.MonitorEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.MonitorEventArgs.status Field

Status of a transaction sent to a NAM market.

```
public readonly TransactionNamStatus status;
```

See Also

[Connection.MonitorEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.MonitorEventArgs.transactionID Field

Associated TransactionID.

```
public readonly TransactionID transactionID;
```

See Also

[Connection.MonitorEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.OpenEventArgs Class

Connection server answer to [open](#).

For a list of all members of this type, see [Connection.OpenEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Connection.OpenEventArgs

public class Connection.OpenEventArgs : [FastTrackEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)











See Also

[Connection.OpenEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onOpen](#)





Connection.OpenEventArgs Members

[Connection.OpenEventArgs overview](#)

Public Instance Fields

 businessServiceID	Business service ID associated to this connection.
 clientID	ClientID associated to this connection.
 clientServiceID	Client service ID associated to this connection.
 environment	An indication of the FastTrack server environment (e.g.: Production, Testing, etc...).
 ftID	FastTrack Server ID.
 marketRevision	Server market version.
 result	Server answer failure-code.
 systemDate	Server system date.
 systemTime	Server system time.
 userName	Same open userName parameter.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.











See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Connection.OpenEventArgs Fields

The fields of the **Connection.OpenEventArgs** class are listed below. For a complete list of **Connection.OpenEventArgs** class members, see the [Connection.OpenEventArgs Members](#) topic.

Public Instance Fields

 businessServiceID	Business service ID associated to this connection.
 clientID	ClientID associated to this connection.
 clientServiceID	Client service ID associated to this connection.
 environment	An indication of the FastTrack server environment (e.g.: Production, Testing, etc...).
 ftID	FastTrack Server ID.
 marketRevision	Server market version.
 result	Server answer failure-code.
 systemDate	Server system date.
 systemTime	Server system time.
 userName	Same open userName parameter.

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Connection.OpenEventArgs.businessServiceID Field

Business service ID associated to this connection.

```
public readonly uint businessServiceID;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

The triplet (**businessServiceID**, [clientServiceID](#), [clientID](#)) identifies the connection between the client (see the [open](#) `clientID` parameter) and the server (the first two elements of the triplet).

The same triplet is also used by [TransactionID](#) to identify the transactions sent to a server on a given connection.

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.OpenEventArgs.clientID Field

ClientID associated to this connection.

```
public readonly uint clientID;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

The triplet ([businessServiceID](#), [clientServiceID](#), **clientID**) identifies the connection between the client (see the [open](#) `clientID` parameter) and the server (the first two elements of the triplet).

The same triplet is also used by [TransactionID](#) to identify the transactions sent to a server on a given connection.

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.OpenEventArgs.clientServiceID Field

Client service ID associated to this connection.

```
public readonly uint clientServiceID;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

The triplet ([businessServiceID](#), **clientServiceID**, [clientID](#)) identifies the connection between the client (see the [open](#) `clientID` parameter) and the server (the first two elements of the triplet).

The same triplet is also used by [TransactionID](#) to identify the transactions sent to a server on a given connection.

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.OpenEventArgs.environment Field

An indication of the FastTrack server environment (e.g.: Production, Testing, etc...).

```
public readonly uint environment;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

The precise meaning of this value depends on the particular FastTrack server and it is documented in the corresponding manual.

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.OpenEventArgs.ftID Field

FastTrack Server ID.

```
public readonly uint ftID;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

Each FastTrack server in the world is identified by an unique FastTrack Server ID.

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.OpenEventArgs.marketRevision Field

Server market version.

```
public readonly Revision marketRevision;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.OpenEventArgs.result Field

Server answer failure—code.

public readonly [Error](#) result;

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.OpenEventArgs.systemDate Field

Server system date.

```
public readonly uint systemDate;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

The returned value reflects the date in which the server opened the connection.

The returned value is an integer whose decimal representation is: YYYYMMDD (see [LDATE](#)).

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.OpenEventArgs.systemTime Field

Server system time.

```
public readonly uint systemTime;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

The returned value reflects the time in which the server opened the connection.

The returned value is an integer whose decimal representation is: HHMMSScc (see [LTIME](#)).

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Connection.OpenEventArgs.userName Field

Same [open](#) userName parameter.

```
public readonly string userName;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

See Also

[Connection.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

ConnectionActivity Class

Super-class common to all objects of a given [Connection](#).

For a list of all members of this type, see [ConnectionActivity Members](#).

[System.Object](#)

[ftapi_dnet.Disposable](#)

ftapi_dnet.ConnectionActivity

[ftapi_dnet.Filter](#)

[ftapi_dnet.Query](#)

[ftapi_dnet.Subscription](#)

[ftapi_dnet.Transaction](#)

public abstract class ConnectionActivity : [Disposable](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

All sub-classes ([Subscription](#), [Transaction](#), [Query](#) and [Filter](#)) share:

- a common property to get the [associated connection](#),
- a common behaviour on [Close/Dispose](#) of the associated connection (in particular they will be automatically and respectively closed/disposed),
- a common requirement on they normal [running](#) activities (in particular these activities are allowed only if the associated connection [Status](#) is **running**).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)





See Also

[ConnectionActivity Members](#) | [ftapi_dnet Namespace](#)







ConnectionActivity Members

[ConnectionActivity overview](#)

Public Instance Properties

 connection	Gets the associated Connection.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

Public Instance Methods

 Close (inherited from Disposable)	Logically destroy the current object, making it again available.
 Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from Disposable)	Returns a single-line String that represents the current object.





See Also

[ConnectionActivity Class](#) | [ftapi_dnet Namespace](#)

ConnectionActivity Properties

The properties of the **ConnectionActivity** class are listed below. For a complete list of **ConnectionActivity** class members, see the [ConnectionActivity Members](#) topic.

Public Instance Properties

 connection	Gets the associated Connection.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

See Also

[ConnectionActivity Class](#) | [ftapi_dnet Namespace](#)

ConnectionActivity.connection Property

Gets the associated Connection.

```
public virtual Connection connection {get;}
```

Property Value

The associated Connection, sometime called *parent Connection*, is a parameter of every [ConnectionActivity](#) subclass constructor.

See Also

[ConnectionActivity Class](#) | [ftapi dnet Namespace](#)

ConnectionContext Class

Container for additional [Connection](#) attributes.

For a list of all members of this type, see [ConnectionContext Members](#).

[System.Object](#)

[ftapi_dnet.Disposable](#)

ftapi_dnet.ConnectionContext

public class ConnectionContext : [Disposable](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Remarks

A ConnectionContext is used in [Connection](#) (connectionContext parameter of Connection.[open](#)) to specify additional attributes to associate to a Connection.

One of most used additional attributes is [applAuthFile](#).

Container (where all attributes have their specific default values) are created by the [constructor](#) and then attribute values may be changed with [setAttribute](#).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ConnectionContext Members](#) | [ftapi_dnet Namespace](#) | [Connection](#)




ConnectionContext Members

[ConnectionContext overview](#)








Public Instance Constructors

 ConnectionContext Constructor	Creates a new ConnectionContext.
---	----------------------------------

Public Instance Properties

 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

Public Instance Methods

 Close (inherited from Disposable)	Logically destroy the current object, making it again available.
 Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 setAttribute	Set a new value for a given attribute.
 ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[ConnectionContext Class](#) | [ftapi_dnet Namespace](#) | [Connection](#)

ConnectionContext Constructor

Creates a new ConnectionContext.

```
public ConnectionContext();
```

Remarks

The new created ConnectionContext has all attributes with they specific default values.

Status

- on successfully Exit – [running](#)

See Also

[ConnectionContext Class](#) | [ftapi_dnet Namespace](#)

ConnectionContext Methods

The methods of the **ConnectionContext** class are listed below. For a complete list of **ConnectionContext** class members, see the [ConnectionContext Members](#) topic.

Public Instance Methods

✦ Close (inherited from Disposable)	Logically destroy the current object, making it again available.
✦ Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
✦ Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
✦ GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
✦ GetType (inherited from Object)	Gets the Type of the current instance.
✦ setAttribute	Set a new value for a given attribute.
✦ ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[ConnectionContext Class](#) | [ftapi_dnet Namespace](#) | [Connection](#)

ConnectionContext.setAttribute Method

Set a new value for a given attribute.

```
public virtual void setAttribute(  
    ContextAttribute contextAttribute,  
    string value  
);
```

Parameters

contextAttribute

[ContextAttribute](#) to change.

value

New value (represented as a string).

Remarks

If this method invocation completed successfully, then

- the request was accepted by the client,

otherwise

- the client rejected the request,
- an appropriate [FastTrackException](#) will be thrown.

In both cases the current [Status](#) remains unchanged.

Status

- required on Entry – [running](#)
- on successfully Exit – running

Example

```
public ConnectionFromArgs(string[] args)  
{  
    using(ConnectionContext cc = new ConnectionContext())  
    {  
        cc.setAttribute(ContextAttribute.applAuthFile,    LIC_FILE);  
        cc.setAttribute(ContextAttribute.alternativeHost1, args[4]);  
        cc.setAttribute(ContextAttribute.alternativePort1, args[5]);  
        open(args[2], UInt16.Parse(args[3]), clientID++, args[0], args[1], UserType.view, cc  
            , null, 0, "METAMARKET" // to access a service via a specific ServiceName  
        );  
    }  
}
```


See Also

[ConnectionContext Class](#) | [ftapi_dnet Namespace](#)

Disposable Class

Common super–class for all objects that have a life cycle.

For a list of all members of this type, see [Disposable Members](#).

[System.Object](#)

ftapi_dnet.Disposable

[Derived types](#)

public abstract class Disposable : IDisposable

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Remarks

All Disposable objects share:

- **A common life cycle** – See the state–diagram below.

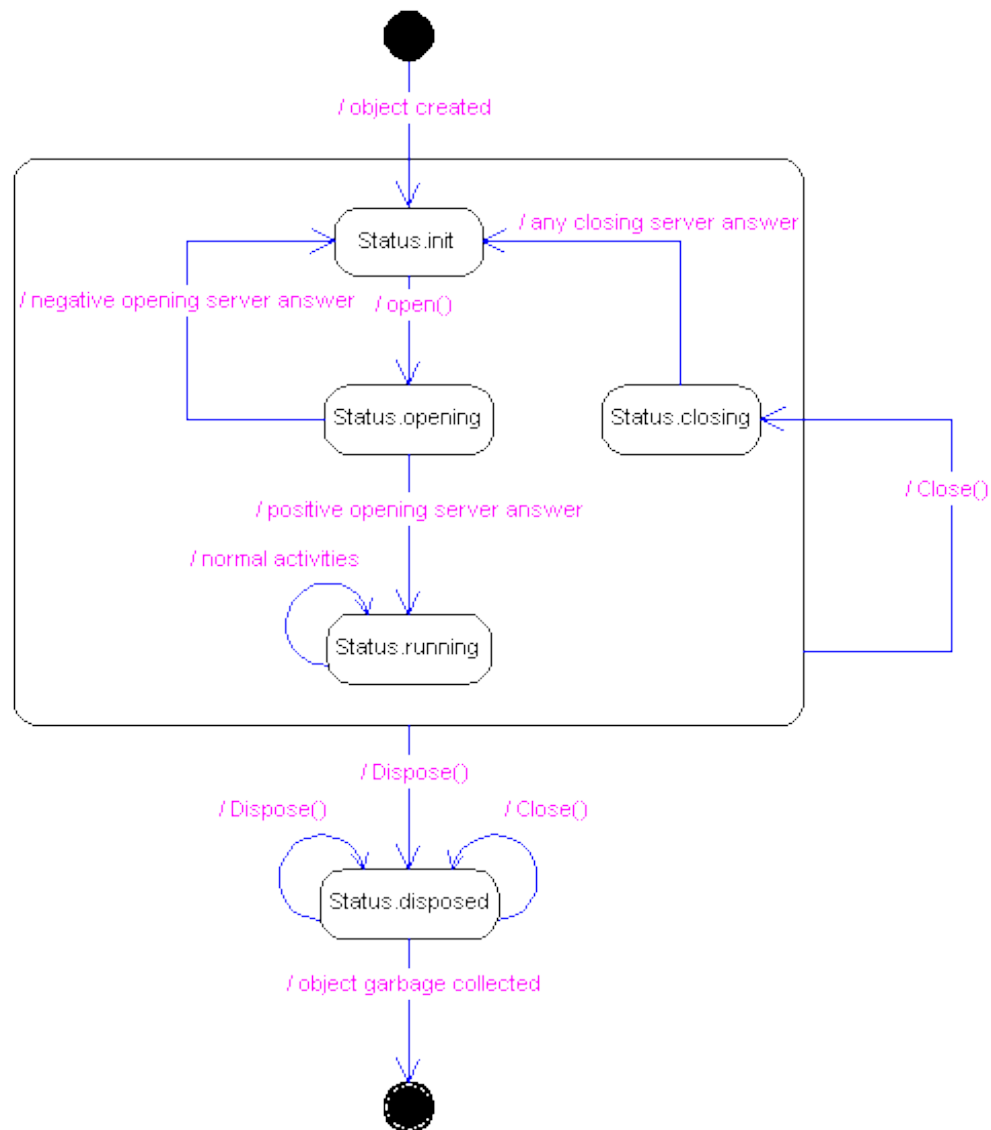
Please note that some Disposable sub–classes have a slightly different state–diagram: e.g. at the exit of [Transaction](#) constructor the Status may be [opening](#) or [running](#) instead of [init](#).

The constructor/method documentation of every Disposable sub–class clearly indicates the *Status required on Entry* and the *Status on successfully Exit*.

- A corresponding common [status](#) property
- **A common [Dispose/Close](#) couple of methods** – They may be invoked (as shown in the state–diagram below) to force a final–state or a re–use of the Disposable object.
- Other [common properties](#)

Lifecycle

The following state–diagram, taken from [Status](#) documentation, shows the Disposable objects generic life cycle.



Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[Disposable Members](#) | [ftapi_dnet Namespace](#)

Disposable Hierarchy

[System.Object](#)

Disposable

[Connection](#)

[ConnectionActivity](#)

[ConnectionContext](#)

[KeyValue](#)

[Lib](#)

[Mask](#)




See Also

[Disposable Class](#) | [Disposable Members](#) | [ftapi_dnet Namespace](#)







Disposable Members

[Disposable overview](#)

Public Instance Properties

 name	Gets or sets the name used by ToString to represent the current object.
 status	Gets the current Status of this object.
 suppressFinalizeForbidden	Gets or set an indication about the automatic invocation of Finalize .

Public Instance Methods

 Close	Logically destroy the current object, making it again available.
 Dispose	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current object.




See Also

[Disposable Class](#) | [ftapi_dnet Namespace](#)

Disposable Properties

The properties of the **Disposable** class are listed below. For a complete list of **Disposable** class members, see the [Disposable Members](#) topic.

Public Instance Properties

 name	Gets or sets the name used by ToString to represent the current object.
 status	Gets the current Status of this object.
 suppressFinalizeForbidden	Gets or set an indication about the automatic invocation of Finalize .

See Also

[Disposable Class](#) | [ftapi_dnet Namespace](#)

Disposable.name Property

Gets or sets the name used by [ToString](#) to represent the current object.

```
public virtual string name { get; set; }
```

Property Value

If this value is `null` then an hidden appropriate meaningful internal name will be used by [ToString](#).

The initial default value for this property is `null`.

Example

```
public TransactionOrd(Connection conn, Entity order, string newName) : base(conn)
{
    name = newName;
    Lib.lib.trace(TraceLevel.full, "Over " + this
        + " sending order: " + order.ToString(true,0));
    send(order, FT_C_ORDERKey);
}
```

See Also

[Disposable Class](#) | [ftapi_dnet Namespace](#)

Disposable.status Property

Gets the current [Status](#) of this object.

```
public virtual Status status {get;}
```

Property Value

The initial value of this property for new created object is [init](#) then the value may change as generally explained in [Status](#) or specifically explained in every method of every [Disposable](#) class.

See Also

[Disposable Class](#) | [ftapi_dnet Namespace](#)

Disposable.SuppressFinalizeForbidden Property

Gets or set an indication about the automatic invocation of [Finalize](#).

```
public virtual bool SuppressFinalizeForbidden { get; set; }
```

Property Value

The actual library implementation is sometime *interested* and sometime *not interested* about the inaccessibility of managed [Disposable](#) objects.

If the library is no more interested about the inaccessibility of a specified object then it will be invoke the [SuppressFinalize](#) method unless the value of this property is true.

An application may explicitly set this property equals *true* when it make a subclass of a **Disposable** (e.g. a `MySubscription` subclass of [Subscription](#)) with proprietary data (e.g. fields of `MySubscription`) that are sensibles to the garbage-collection.

The initial default value of this property is *false*. In all normal cases this is the suggested value for this property.







See Also

[Disposable Class](#) | [ftapi_dnet Namespace](#) | [Finalize](#) | [SuppressFinalize](#)

Disposable Methods

The methods of the **Disposable** class are listed below. For a complete list of **Disposable** class members, see the [Disposable Members](#) topic.

Public Instance Methods

 Close	Logically destroy the current object, making it again available.
 Dispose	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current object.

See Also

[Disposable Class](#) | [ftapi_dnet Namespace](#)

Disposable.Close Method

Logically destroy the current object, making it again available.

```
public virtual void Close();
```

Remarks

If the current Status is [disposed](#) this method does nothing otherwise

- The current status is changed to [closing](#).
- If the status (before the previous change) was [running](#) then
 - This method is recursively invoked on the childs of the current object: e.g. a `Close` of a [Connection](#) will cause the automatic `Close` of all its [ConnectionActivity](#) (i.e. all its Subscriptions, Transactions, Query and Filters will be automatically closed).
 - A destroy request is sent to the server.
 - When the server-answer will be available the current status will change to [init](#) and an appropriate `closeEvent` will be raised.

This method, differently from [Dispose](#), allow the re-use of the current object because the Status may returns **init**.

Please note that in few [Disposable](#) subclasses this method is implemented as **Dispose**: in these cases the object will never returns to **init** because *no `closeEvent` of any sort* will ever be raised.

Status

- **required on Entry** – any [Status](#)
- **on successfully Exit (if Entry Status was disposed)** – **disposed**
- **on successfully Exit (if Entry Status was not disposed)** – **closing**

See Also

[Disposable Class](#) | [ftapi_dnet Namespace](#) | [Dispose](#)

Disposable.Dispose Method

Logically destroy the current object, making it un-available.

```
public virtual void Dispose();
```

Implements

[IDisposable.Dispose](#)

Remarks

If the current Status is [disposed](#) this method does nothing otherwise

- The current status is changed to **disposed**.
- If the status (before the previous change) was [running](#) then
 - This method is recursively invoked on the childs of the current object: e.g. a `Dispose` of a [Connection](#) will cause the automatic `Dispose` of all its [ConnectionActivity](#) (i.e. all its Subscriptions, Transactions, Query and Filters will be automatically disposed).
 - A destroy request is sent to the server.

This method, differently from [Close](#), does not allow the re-use of the current object because the Status remains forever **disposed**.

Status

- **required on Entry** – any [Status](#)
- **on successfully Exit** – **disposed**

Example

```
public static void abort(string message)
{
    Console.WriteLine();
    Console.WriteLine("**** Program abort: " + message);
    Console.WriteLine();
    Lib.lib.trace(TraceLevel.error, message);
    Lib.lib.Dispose();
    Environment.Exit(0);
}
```

In the previous code the `Lib.lib.Dispose()` invocation well-close all [Connection/ConnectionActivity](#) opened by the application.

See Also

[Disposable Class](#) | [ftapi_dnet Namespace](#) | [Close](#)

Disposable.ToString Method

Returns a single-line [String](#) that represents the current object.

```
public override string ToString();
```

Return Value

A single-line [String](#) that represents the current object.

See Also

[Disposable Class](#) | [ftapi_dnet Namespace](#)

Entity Class

Value of an [EntityClass](#).

For a list of all members of this type, see [Entity Members](#).

[System.Object](#)

[ftapi_dnet.Entity](#)

```
public class Entity : ICloneable, IDump
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

Entity instances are value of [STRUCT](#) or [UNION](#) EntityClasses. EntityClass with `type = ENUM` have not values represented with Entities instances: instead they have values accessible through [EntityFields](#).

An Entity is logically defined by/with its properties:

- [entityClass](#) – EntityClass of the Entity.
- [bytes](#) – Bytes array containing the representation of all **EntityFields** of the Entity.
- [isShared](#) – Indicates that the **bytes** representation is shared with other Entity instances.

If many different Entities share the same data representation then new field values written through a `setFieldX` method will be visible by all Entities.

Each single field of an Entity (and also every single component of vector-fields) may be read (its value returned) or written (its value changed) through many several `getFieldX` or `setFieldX` methods where X may be

- [Long](#) or [Double](#) – To get/set a number from/to any non-**STRUCT** and non-**UNION** field.

If the corresponding [EntityFieldType](#) is not numeric then the value will be automatically converted: e.g. it's possible to set a 123.45 value into a [String](#).

- [String](#) – To get/set a string from/to any non-**STRUCT** and non-**UNION** field.

If the corresponding **EntityFieldType** is not a **String** then the value will be automatically converted: e.g. it's possible to get a "Buy" value from an **ENUM** field containing the value 0 corresponding to the "Buy" String.

- [Entity](#) – To get/set another Entity from/to a **STRUCT** or **UNION** field.

There are many ways to create new Entity instances:

- Through one of the seven available constructors (see [Entity Constructor Overload List](#)).
- Through one of the eight `getEntityField` methods (see [Entity.getFieldEntity Overload List](#)).
- Through the [Clone](#) method.

With all these ways it's possible to create new Entity instances that are

- **uninitialized** – their **bytes** representation is automatically [zero-filled](#).
- **initialized and unshared** – their **bytes** representation is copied from another existing entity or from another generic byte-array.
- **initialized and shared** – their **bytes** representation is shared with another existing entity-representation or with another generic byte-array.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[Entity Members](#) | [ftapi_dnet Namespace](#) | [EntityClass](#) | [EntityField](#)




Entity Members

[Entity overview](#)















Public Instance Constructors

 Entity	Overloaded. Initializes a new instance of the Entity class.
--	---

Public Instance Properties

 bytes	Gets <i>a copy</i> or sets a new value for the bytes representation of this Entity.
 entityClass	Gets the EntityClass type of this Entity.
 isShared	Gets an indication about the sharing of the bytes representation.

Public Instance Methods

 clear	Zero-fills the bytes representation.
 Clone	Creates a new Entity that is a copy of the current instance.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 getFieldDouble	Overloaded. Returns a field value as a double .
 getFieldEntity	Overloaded. Returns a field value as a new Entity .
 getFieldLong	Overloaded. Returns a field value as a long .
 getFieldString	Overloaded. Returns a field value as a String .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 setFieldDouble	Overloaded. Sets a field value given a double argument.
 setFieldEntity	Overloaded. Sets a field value given a Entity argument.
 setFieldLong	Overloaded. Sets a field value given a long argument.
 setFieldString	Overloaded. Sets a field value given a String argument.
 ToString	Overloaded. Returns a single/multiple line String that represents the current Entity.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [EntityClass](#) | [EntityField](#)

Entity Constructor

Initializes a new instance of the Entity class.

Overload List

Alternative *copy-constructor* to create a copy of a given Entity with the same sharing as the original Entity.

[public Entity\(Entity\):](#)

Main copy-constructor to create a copy of a given Entity with an explicated sharing indication.

[public Entity\(Entity,bool\):](#)

Main traditional-constructor to create an unshared Entity of a given EntityClass.

[public Entity\(EntityClass\):](#)

Alternative *traditional-constructor* to create an unshared Entity of a given EntityClass.

[public Entity\(string\):](#)

Alternative *traditional-constructor* to create an unshared Entity of a given EntityClass.

[public Entity\(uint\):](#)

Alternative *traditional-constructor* to create an unshared Entity of a given EntityClass.

[public Entity\(uint,string\):](#)

Creates an unshared Entity with a given initial [data representation](#).

[public Entity\(uint,string,byte\[\],uint\):](#)

Remarks

There are many kind of Entity constructors

- **Traditional-Constructors** – They have as parameter an [EntityClass](#) (or a `classID` or a `className`).
They are able to create *only unshared* Entities initially [zero-filled](#).
- **Copy-Constructors** – They take as first parameter an Entity object to copy.
They are able to create new Entities with *both shared and non-shared* [data representation](#).
- **Constructors specifying an initial data representation** – They have as last parameters a couple (`initialData`, `start`) describing a byte slice-array from which the initial Entity data-representation must be copied.

They are able to create *only unshared* Entities.

Please note that there are others manners to create new Entities



Entity Class

- [Clone](#)
- [getFieldEntity](#)

See Also

[Entity Class](#) | [ftapi_dnet.Namespace](#)

Entity Constructor (EntityClass)

Main traditional–constructor to create an unshared Entity of a given EntityClass.

```
public Entity(  
    EntityClass entityClass  
);
```

Parameters

entityClass
Requested EntityClass.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity Constructor Overload List](#)

Entity Constructor (String)

Alternative *traditional–constructor* to create an unshared Entity of a given EntityClass.

```
public Entity(  
    string className  
);
```

Parameters

className

Name of the requested EntityClass.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity Constructor Overload List](#)

Entity Constructor (UInt32)

Alternative *traditional-constructor* to create an unshared Entity of a given EntityClass.

```
public Entity(  
    uint classId  
);
```

Parameters

classId
Id of the requested EntityClass.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity Constructor Overload List](#)

Entity Constructor (UInt32, String)

Alternative *traditional-constructor* to create an unshared Entity of a given EntityClass.

```
public Entity(  
    uint classId,  
    string className  
);
```

Parameters

classId

Id of the requested EntityClass.

className

Name of the requested EntityClass.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity Constructor Overload List](#)

Entity Constructor (Entity, Boolean)

Main copy–constructor to create a copy of a given Entity with an explicated sharing indication.

```
public Entity(  
    Entity entity,  
    bool shared  
);
```

Parameters

entity Entity to copy.
shared Sharing indication

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity Constructor Overload List](#)

Entity Constructor (Entity)

Alternative *copy-~~constructor~~* to create a copy of a given Entity with the same sharing as the original Entity.

```
public Entity(  
    Entity entity  
);
```

Parameters

entity
Entity to copy.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity Constructor Overload List](#) | [Clone](#)

Entity Constructor (UInt32, String, Byte[], UInt32)

Creates an unshared Entity with a given initial [data representation](#).

```
public Entity(  
    uint classId,  
    string className,  
    byte[] initialData,  
    uint start  
);
```

Parameters

classId

Id of the requested EntityClass.

className

Name of the requested EntityClass.

initialData

Initial requested data representation.

start

Index of the first byte of *initialData*




See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity Constructor Overload List](#)

Entity Properties

The properties of the **Entity** class are listed below. For a complete list of **Entity** class members, see the [Entity Members](#) topic.

Public Instance Properties

 bytes	Gets <i>a copy</i> or sets a new value for the bytes representation of this Entity.
 entityClass	Gets the EntityClass type of this Entity.
 isShared	Gets an indication about the sharing of the bytes representation.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [EntityClass](#) | [EntityField](#)

Entity.bytes Property

Gets *a copy* or sets a new value for the bytes representation of this Entity.

```
public virtual byte[] bytes { get; set; }
```

Property Value

Please note that the Length of the array is always equal to the [dataSize](#) field of [entityClass](#) property even if this Entity was created with a `start` parameter not equal to zero.

For the same reason **please note** that all bytes are meaningful.

Even if the returned (i.e. `get`) value is always a copy **please note** that setting a new value to a shared Entity does *always* propagate the new representation to all shared Entities.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.entityClass Property

Gets the [EntityClass](#) type of this Entity.

```
public virtual EntityClass entityClass {get;}
```

Property Value

This value is never null.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.isShared Property

Gets an indication about the sharing of the [bytes](#) representation.

```
public virtual bool isShared {get;}
```

Property Value

Please note that this value just reflects how the Entity was created, and **not** the actual sharing of Entity representation: e.g. an Entity X may have this value as **false** because it was created with an

```
X = new Entity("className")
```

constructor even if a new Entity Y shares their representation with X because Y was created with

```
Y = new Entity(X, true)
```

constructor.

Remarks

If many different Entities share the same [bytes](#) representation then new field values written through a `setFieldX` method will affect all Entities.















See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity Methods

The methods of the **Entity** class are listed below. For a complete list of **Entity** class members, see the [Entity Members](#) topic.

Public Instance Methods

 clear	Zero-fills the bytes representation.
 Clone	Creates a new Entity that is a copy of the current instance.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 getFieldDouble	Overloaded. Returns a field value as a double .
 getFieldEntity	Overloaded. Returns a field value as a new Entity .
 getFieldLong	Overloaded. Returns a field value as a long .
 getFieldString	Overloaded. Returns a field value as a String .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 setFieldDouble	Overloaded. Sets a field value given a double argument.
 setFieldEntity	Overloaded. Sets a field value given a Entity argument.
 setFieldLong	Overloaded. Sets a field value given a long argument.
 setFieldString	Overloaded. Sets a field value given a String argument.
 ToString	Overloaded. Returns a single/multiple line String that represents the current Entity.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [EntityClass](#) | [EntityField](#)

Entity.clear Method

Zero-fills the [bytes](#) representation.

```
public virtual void clear();
```

Remarks

Zero-filling is made through [Clear](#).

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Clear](#)

Entity.Clone Method

Creates a new Entity that is a copy of the current instance.

```
public virtual object Clone();
```

Return Value

A new Entity that is a copy of this instance.

Implements

[ICloneable.Clone](#)

Remarks

The new returned Entity will have the same [isShared](#) property as the current instance.

So a call to

```
X.Clone( )
```

is implemented as

```
new Entity\(X, X.isShared\)
```

which is equivalent to

```
new Entity\(X\)
```

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity](#) | [Entity](#)

Entity.getFieldDouble Method

Returns a field value as a [double](#).

Overload List

Returns an EntityField value as a [double](#).

[public virtual double getFieldDouble\(EntityField\);](#)

Full method to return an EntityField (or component) value as a [double](#).

[public virtual double getFieldDouble\(EntityField,int\);](#)

Returns a field value as a [double](#).

[public virtual double getFieldDouble\(string\);](#)

Returns a field (or component) value as a [double](#).

[public virtual double getFieldDouble\(string,int\);](#)

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.getFieldDouble Method (EntityField)

Returns an EntityField value as a [double](#).

```
public virtual double getFieldDouble(  
    EntityField field  
);
```

Parameters

field
Interesting field.

Return Value

The [double](#) representation of the requested field.

Remarks

The invocation

```
getFieldDouble(entityField)
```

is equivalent to

```
getFieldDouble(entityField, -1)
```

so *please see* the [full getFieldDouble](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldDouble Overload List](#)

Entity.getFieldDouble Method (EntityField, Int32)

Full method to return an EntityField (or component) value as a [double](#).

```
public virtual double getFieldDouble(  
    EntityField field,  
    int index  
);
```

Parameters

field

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

Return Value

The [double](#) representation of the requested field.

Remarks

This method may be used to get any

- any floating numeric field or component,
- any [STRING](#) field parsable as a floating numeric value.

This invocation throw an appropriate [FastTrackException](#) on

- any integer numeric field or component,
- any [STRUCT](#) or [UNION](#) or [ENUM](#) field or component,
- any **STRING** field not parsable as a floating numeric value,
- any unexisting *field* and/or *index*.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldDouble Overload List](#)

Entity.getFieldDouble Method (String)

Returns a field value as a [double](#).

```
public virtual double getFieldDouble(  
    string fieldName  
);
```

Parameters

fieldName
Interesting field.

Return Value

The [double](#) representation of the requested field.

Remarks

The invocation

```
X.getFieldDouble(fieldName)
```

is equivalent to

```
X.getFieldDouble(X.entityClass.getEntityField(fieldName), -1)
```

so *please see* the [full getFieldDouble](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldDouble Overload List](#)

Entity.getFieldDouble Method (String, Int32)

Returns a field (or component) value as a **double**.

```
public virtual double getFieldDouble(  
    string fieldName,  
    int index  
);
```

Parameters

fieldName

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

Return Value

The **double** representation of the requested field.

Remarks

The invocation

```
X.getFieldDouble(fieldName, index)
```

is equivalent to

```
X.getFieldDouble(X.entityClass.getEntityField(fieldName), index)
```

so *please see* the [full getFieldDouble](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldDouble Overload List](#)

Entity.getFieldEntity Method

Returns a field value as a new [Entity](#).

Overload List

Returns an EntityField value as a new shared [Entity](#).

[public virtual Entity getFieldEntity\(EntityField\);](#)

Returns an EntityField value as a new [Entity](#).

[public virtual Entity getFieldEntity\(EntityField,bool\);](#)

Returns an EntityField (or component) value as a new shared [Entity](#).

[public virtual Entity getFieldEntity\(EntityField,int\);](#)

Full method to return an EntityField (or component) value as a new [Entity](#).

[public virtual Entity getFieldEntity\(EntityField,int,bool\);](#)

Returns a field value as a new shared [Entity](#).

[public virtual Entity getFieldEntity\(string\);](#)

Returns a field value as a new [Entity](#).

[public virtual Entity getFieldEntity\(string,bool\);](#)

Returns a field (or component) value as a new shared [Entity](#).

[public virtual Entity getFieldEntity\(string,int\);](#)

Returns a field (or component) value as a new [Entity](#).

[public virtual Entity getFieldEntity\(string,int,bool\);](#)

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.getFieldEntity Method (EntityField)

Returns an EntityField value as a new shared [Entity](#).

```
public virtual Entity getFieldEntity(  
    EntityField field  
);
```

Parameters

field
Interesting field.

Return Value

The new [Entity](#) representation of the requested field.

Remarks

The invocation

```
getFieldEntity(field)
```

is equivalent to

```
getFieldEntity(field, -1, true)
```

so *please see* the [full getFieldEntity](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldEntity Overload List](#)

Entity.getFieldEntity Method (EntityField, Boolean)

Returns an EntityField value as a new [Entity](#).

```
public virtual Entity getFieldEntity(  
    EntityField field,  
    bool shared  
);
```

Parameters

field

Interesting field.

shared

An indication about the [sharing](#) data [representation](#) of the returned new Entity.

Return Value

The new [Entity](#) representation of the requested field.

Remarks

The invocation

```
getFieldEntity(field, shared)
```

is equivalent to

```
getFieldEntity(field, -1, shared)
```

so *please see* the [full getFieldEntity](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldEntity Overload List](#)

Entity.getFieldEntity Method (EntityField, Int32)

Returns an EntityField (or component) value as a new shared [Entity](#).

```
public virtual Entity getFieldEntity(  
    EntityField field,  
    int index  
);
```

Parameters

field

Interesting field.

index

Specific component-index for a vector field, or -1 for a non-vector field.

Return Value

The new [Entity](#) representation of the requested field.

Remarks

The invocation

```
getFieldEntity(field, index)
```

is equivalent to

```
getFieldEntity(field, index, true)
```

so *please see* the [full getFieldEntity](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldEntity Overload List](#)

Entity.getFieldEntity Method (EntityField, Int32, Boolean)

Full method to return an EntityField (or component) value as a new [Entity](#).

```
public virtual Entity getFieldEntity(  
    EntityField field,  
    int index,  
    bool shared  
);
```

Parameters

field

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

shared

An indication about the [sharing](#) data [representation](#) of the returned new Entity.

Return Value

The new [Entity](#) representation of the requested field.

Remarks

This method may be used to get any

- [STRUCT](#) or [UNION](#) field or component,

This invocation throw an appropriate [FastTrackException](#) on

- any non–**STRUCT** and non–**UNION** field or component,
- any unexisting *field* and/or *index*.

If this method is invoked with a **true** value for the *shared* parameter then any E.setFieldX invocation made on the returned Entity value E will automatically affects the [bytes](#) representation of this Entity.

Example

The following code–fragment shows a [alternative getFieldEntity](#) invocation.

```
public Entity readNextOrder()  
{  
    Entity result = null;  
    string line   = sr.ReadLine();  
    if(line == null)  
        sr.Close();  
    else  
    {  
        Lib.lib.trace(TraceLevel.full, "Read order line \"" + line + "\"");  
        MatchCollection matchCollection = regex.Matches(line);  
        result = new Entity("FT_C_ORDER");  
    }  
}
```

```
result.setFieldString("Verb", matchColletion[0].Value.Trim());
result.setFieldString("FTSecID",matchColletion[1].Value.Trim());
result.setFieldString("Price", matchColletion[2].Value.Trim());
result.setFieldString("Qty", matchColletion[3].Value.Trim());
Entity client = result.getFieldEntity("Client");
client.setFieldString("ClientAccount", matchColletion[4].Value.Trim());
client.setFieldString("Origin", "Client");
result.setFieldString("TimeInForce", "Good till day");
}
return result;
}
```

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldEntity Overload List](#)

Entity.getFieldEntity Method (String)

Returns a field value as a new shared [Entity](#).

```
public virtual Entity getFieldEntity(  
    string fieldName  
);
```

Parameters

fieldName
Interesting field.

Return Value

The new [Entity](#) representation of the requested field.

Remarks

The invocation

```
X.getFieldEntity(fieldName)
```

is equivalent to

```
X.getFieldEntity(X.entityClass.getEntityField(fieldName), -1, true)
```

so *please see* the [full getFieldEntity](#) method for a full detailed description of it.

Example

```
public Entity readNextOrder()  
{  
    Entity result = null;  
    string line    = sr.ReadLine();  
    if(line == null)  
        sr.Close();  
    else  
    {  
        Lib.lib.trace(TraceLevel.full, "Read order line \"" + line + "\"");  
        MatchCollection matchCollection = regex.Matches(line);  
        result = new Entity("FT_C_ORDER");  
        result.setFieldString("Verb",    matchCollection[0].Value.Trim());  
        result.setFieldString("FTSecID", matchCollection[1].Value.Trim());  
        result.setFieldString("Price",   matchCollection[2].Value.Trim());  
        result.setFieldString("Qty",     matchCollection[3].Value.Trim());  
        Entity client = result.getFieldEntity("Client");  
        client.setFieldString("ClientAccount", matchCollection[4].Value.Trim());  
        client.setFieldString("Origin",       "Client");  
        result.setFieldString("TimeInForce",  "Good till day");  
    }  
    return result;  
}
```

See Also

[Entity Class](#) | [ftapi_dnet.Namespace](#) | [Entity.getFieldEntity Overload List](#)

Entity.getFieldEntity Method (String, Boolean)

Returns a field value as a new [Entity](#).

```
public virtual Entity getFieldEntity(  
    string fieldName,  
    bool shared  
);
```

Parameters

fieldName

Interesting field.

shared

An indication about the [sharing](#) data [representation](#) of the returned new Entity.

Return Value

The new [Entity](#) representation of the requested field.

Remarks

The invocation

```
X.getFieldEntity(fieldName, shared)
```

is equivalent to

```
X.getFieldEntity(X.entityClass.getEntityField(fieldName), -1, shared)
```

so *please see* the [full getFieldEntity](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldEntity Overload List](#)

Entity.getFieldEntity Method (String, Int32)

Returns a field (or component) value as a new shared [Entity](#).

```
public virtual Entity getFieldEntity(  
    string fieldName,  
    int index  
);
```

Parameters

fieldName

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

Return Value

The new [Entity](#) representation of the requested field.

Remarks

The invocation

```
X.getFieldEntity(fieldName, index)
```

is equivalent to

```
X.getFieldEntity(X.entityClass.getEntityField(fieldName), index, true)
```

so *please see* the [full getFieldEntity](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldEntity Overload List](#)

Entity.getFieldEntity Method (String, Int32, Boolean)

Returns a field (or component) value as a new [Entity](#).

```
public virtual Entity getFieldEntity(  
    string fieldName,  
    int index,  
    bool shared  
);
```

Parameters

fieldName

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

shared

An indication about the [sharing](#) data [representation](#) of the returned new Entity.

Return Value

The new [Entity](#) representation of the requested field.

Remarks

The invocation

```
X.getFieldEntity(fieldName, index, shared)
```

is equivalent to

```
X.getFieldEntity(X.entityClass.getEntityField(fieldName), index, shared)
```

so *please see* the [full getFieldEntity](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi dnet Namespace](#) | [Entity.getFieldEntity Overload List](#)

Entity.getFieldLong Method

Returns a field value as a [long](#).

Overload List

Returns an EntityField value as a [long](#).

[public virtual long getFieldLong\(EntityField\);](#)

Full method to return an EntityField (or component) value as a [long](#).

[public virtual long getFieldLong\(EntityField,int\);](#)

Returns a field value as a [long](#).

[public virtual long getFieldLong\(string\);](#)

Returns a field (or component) value as a [long](#).

[public virtual long getFieldLong\(string,int\);](#)

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.getFieldLong Method (EntityField)

Returns an EntityField value as a [long](#).

```
public virtual long getFieldLong(  
    EntityField field  
);
```

Parameters

field
Interesting field.

Return Value

The [long](#) representation of the requested field.

Remarks

The invocation

```
getFieldLong(entityField)
```

is equivalent to

```
getFieldLong(entityField, -1)
```

so *please see* the [full getFieldLong](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldLong Overload List](#)

Entity.getFieldLong Method (EntityField, Int32)

Full method to return an EntityField (or component) value as a [long](#).

```
public virtual long getFieldLong(
    EntityField field,
    int index
);
```

Parameters

field

Interesting field.

index

Specific component-index for a vector field, or -1 for a non-vector field.

Return Value

The [long](#) representation of the requested field.

Remarks

This method may be used to get any

- any integer numeric field or component,
- any [ENUM](#) field or component,
- any byte-[encoded](#) component of a [STRING](#),
- any **STRING** field parsable as an integer numeric value.

This invocation throw an appropriate [FastTrackException](#) on

- any floating numeric field or component,
- any [STRUCT](#) or [UNION](#) field or component,
- any **STRING** field not parsable as an integer numeric value,
- any unexisting *field* and/or *index*.

Example

The following code-fragment shows a [alternative getFieldLong](#) invocation.

```
private void display(uint entityId, string s)
{
    if(entityClassId != param.getFieldLong("EntityID"))
        ExampleGeneric.abort(
            "Unexpected class referring EntityClassID = "
            + entityId + " != " + param.getFieldLong("EntityID"));
    print(entityClassId, s);
}
```

See Also

[Entity Class](#) | [ftapi_dnet.Namespace](#) | [Entity.getFieldLong Overload List](#)

Entity.getFieldLong Method (String)

Returns a field value as a [long](#).

```
public virtual long getFieldLong(  
    string fieldName  
);
```

Parameters

fieldName
Interesting field.

Return Value

The [long](#) representation of the requested field.

Remarks

The invocation

```
X.getFieldLong(fieldName)
```

is equivalent to

```
X.getFieldLong(X.entityClass.getEntityField(fieldName), -1)
```

so *please see* the [full getFieldLong](#) method for a full detailed description of it.

Example

```
private void display(uint entityId, string s)  
{  
    if(entityClassId != param.getFieldLong("EntityID"))  
        ExampleGeneric.abort(  
            "Unexpected class referring EntityClassID = "  
            + entityId + " != " + param.getFieldLong("EntityID"));  
    print(entityClassId, s);  
}
```

See Also

[Entity Class](#) | [ftapi dnet Namespace](#) | [Entity.getFieldLong Overload List](#)

Entity.getFieldLong Method (String, Int32)

Returns a field (or component) value as a [long](#).

```
public virtual long getFieldLong(  
    string fieldName,  
    int index  
);
```

Parameters

fieldName

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

Return Value

The [long](#) representation of the requested field.

Remarks

The invocation

```
X.getFieldLong(fieldName, index)
```

is equivalent to

```
X.getFieldLong(X.entityClass.getEntityField(fieldName), index)
```

so *please see* the [full getFieldLong](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldLong Overload List](#)

Entity.getFieldString Method

Returns a field value as a [String](#).

Overload List

Returns an EntityField value as a [String](#).

[public virtual string getFieldString\(EntityField\);](#)

Full method to return an EntityField (or component) value as a [String](#).

[public virtual string getFieldString\(EntityField.int\);](#)

Returns a field value as a [String](#).

[public virtual string getFieldString\(string\);](#)

Returns a field (or component) value as a [String](#).

[public virtual string getFieldString\(string.int\);](#)

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.getFieldString Method (EntityField)

Returns an EntityField value as a [String](#).

```
public virtual string getFieldString(  
    EntityField field  
);
```

Parameters

field
Interesting field.

Return Value

The [String](#) representation of the requested field.

Remarks

The invocation

```
getFieldString(entityField)
```

is equivalent to

```
getFieldString(entityField, -1)
```

so *please see* the [full getFieldString](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldString Overload List](#)

Entity.getFieldString Method (EntityField, Int32)

Full method to return an EntityField (or component) value as a [String](#).

```
public virtual string getFieldString(
    EntityField field,
    int index
);
```

Parameters

field

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

Return Value

The [String](#) representation of the requested field.

Remarks

This method may be used to get any

- any numeric field or component,
- any [ENUM](#) field or component,
- any byte–[encoded](#) component of a [STRING](#),
- any **STRING** field.

This invocation throw an appropriate [FastTrackException](#) on

- any [STRUCT](#) or [UNION](#) field or component,
- any unexisting *field* and/or *index*.

Example

The following code–fragment shows a [alternative getFieldString](#) invocation.

```
public override void onNotify(NotifyEventArgs ne)
{
    // base.onNotify(ne);
    #if DEBUG
        Lib.lib.trace(TraceLevel.full,
            "Incoming market " + ne.entity.ToString(true, 0));
    #else
        Lib.lib.trace(TraceLevel.normal,
            "Incoming market " + ne.entity.getFieldString("MarketID"));
    #endif
}
```

See Also

[Entity Class](#) | [ftapi_dnet.Namespace](#) | [Entity.getFieldString Overload List](#)

Entity.getFieldString Method (String)

Returns a field value as a [String](#).

```
public virtual string getFieldString(  
    string fieldName  
);
```

Parameters

fieldName
Interesting field.

Return Value

The [String](#) representation of the requested field.

Remarks

The invocation

```
X.getFieldString(fieldName)
```

is equivalent to

```
X.getFieldString(X.entityClass.getEntityField(fieldName), -1)
```

so *please see* the [full getFieldString](#) method for a full detailed description of it.

Example

```
public override void onNotify(NotifyEventArgs ne)  
{  
    // base.onNotify(ne);  
#if DEBUG  
    Lib.lib.trace(TraceLevel.full,  
        "Incoming market " + ne.entity.ToString(true, 0));  
#else  
    Lib.lib.trace(TraceLevel.normal,  
        "Incoming market " + ne.entity.getFieldString("MarketID"));  
#endif  
}
```

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldString Overload List](#)

Entity.getFieldString Method (String, Int32)

Returns a field (or component) value as a [String](#).

```
public virtual string getFieldString(  
    string fieldName,  
    int index  
);
```

Parameters

fieldName

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

Return Value

The [String](#) representation of the requested field.

Remarks

The invocation

```
X.getFieldString(fieldName, index)
```

is equivalent to

```
X.getFieldString(X.entityClass.getEntityField(fieldName), index)
```

so *please see* the [full getFieldString](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.getFieldString Overload List](#)

Entity.setFieldDouble Method

Sets a field value given a [double](#) argument.

Overload List

Sets an EntityField value given a [double](#) argument.

[public virtual void setFieldDouble\(EntityField,double\);](#)

Full method to set an EntityField (or component) value given a [double](#) argument.

[public virtual void setFieldDouble\(EntityField,int,double\);](#)

Sets a field value given a [double](#) argument.

[public virtual void setFieldDouble\(string,double\);](#)

Sets a field (or component) value given a [double](#) argument.

[public virtual void setFieldDouble\(string,int,double\);](#)

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.setFieldDouble Method (EntityField, Double)

Sets an EntityField value given a [double](#) argument.

```
public virtual void setFieldDouble(  
    EntityField field,  
    double value  
);
```

Parameters

field

Interesting field.

value

[double](#) new value for the requested field.

Remarks

The invocation

```
setFieldDouble(field, value)
```

is equivalent to

```
setFieldDouble(field, -1, value)
```

so *please see* the [full setFieldDouble](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldDouble Overload List](#)

Entity.setFieldDouble Method (EntityField, Int32, Double)

Full method to set an EntityField (or component) value given a [double](#) argument.

```
public virtual void setFieldDouble(  
    EntityField field,  
    int index,  
    double value  
);
```

Parameters

field

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

value

[double](#) new value for the requested field.

Remarks

This method may be used to set any

- any floating numeric field or component,
- any [STRING](#) field.

This invocation throw an appropriate [FastTrackException](#) on

- any integer numeric field or component,
- any [STRUCT](#) or [UNION](#) or [ENUM](#) field or component,
- any unexisting *field* and/or *index*.

If the [bytes](#) representation [isShared](#) then this invocation affects all other representations.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldDouble Overload List](#)

Entity.setFieldDouble Method (String, Double)

Sets a field value given a `double` argument.

```
public virtual void setFieldDouble(  
    string fieldName,  
    double value  
);
```

Parameters

fieldName

Interesting field.

value

`double` new value for the requested field.

Remarks

The invocation

```
X.setFieldDouble(fieldName, value)
```

is equivalent to

```
X.setFieldDouble(X.entityClass.getEntityField(fieldName), -1, value)
```

so *please see* the [full setFieldDouble](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldDouble Overload List](#)

Entity.setFieldDouble Method (String, Int32, Double)

Sets a field (or component) value given a [double](#) argument.

```
public virtual void setFieldDouble(  
    string fieldName,  
    int index,  
    double value  
);
```

Parameters

fieldName

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

value

[double](#) new value for the requested field.

Remarks

The invocation

```
X.setFieldDouble(fieldName, index, value)
```

is equivalent to

```
X.setFieldDouble(X.entityClass.getEntityField(fieldName), index, value)
```

so *please see* the [full setFieldDouble](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldDouble Overload List](#)

Entity.setFieldEntity Method

Sets a field value given a [Entity](#) argument.

Overload List

Sets an EntityField value given a [Entity](#) argument.

[public virtual void setFieldEntity\(EntityField,Entity\):](#)

Full method to set an EntityField (or component) value given a [Entity](#) argument.

[public virtual void setFieldEntity\(EntityField,int,Entity\):](#)

Sets a field value given a [Entity](#) argument.

[public virtual void setFieldEntity\(string,Entity\):](#)

Sets a field (or component) value given a [Entity](#) argument.

[public virtual void setFieldEntity\(string,int,Entity\):](#)

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.setFieldEntity Method (EntityField, Entity)

Sets an EntityField value given a [Entity](#) argument.

```
public virtual void setFieldEntity(  
    EntityField field,  
    Entity value  
);
```

Parameters

field

Interesting field.

value

[Entity](#) new value for the requested field.

Remarks

The invocation

```
setFieldEntity(field, value)
```

is equivalent to

```
setFieldEntity(field, -1, value)
```

so *please see* the [full setFieldEntity](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldEntity Overload List](#)

Entity.setFieldEntity Method (EntityField, Int32, Entity)

Full method to set an EntityField (or component) value given a [Entity](#) argument.

```
public virtual void setFieldEntity(  
    EntityField field,  
    int index,  
    Entity value  
);
```

Parameters

field

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

value

[Entity](#) new value for the requested field.

Remarks

This method may be used to set any

- [STRUCT](#) or [UNION](#) field or component,

This invocation throw an appropriate [FastTrackException](#) on

- any non–**STRUCT** and non–**UNION** field or component,
- null or bad EntityClass *value*,
- any unexisting *field* and/or *index*.

This method is typically used if the corresponding [getFieldEntity](#) was invoked with the shared parameter equal to *false*.

If the [bytes](#) representation [isShared](#) then this invocation affects all other representations.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldEntity Overload List](#)

Entity.setFieldEntity Method (String, Entity)

Sets a field value given a [Entity](#) argument.

```
public virtual void setFieldEntity(  
    string fieldName,  
    Entity value  
);
```

Parameters

fieldName

Interesting field.

value

[Entity](#) new value for the requested field.

Remarks

The invocation

```
X.setFieldEntity(fieldName, value)
```

is equivalent to

```
X.setFieldEntity(X.entityClass.getEntityField(fieldName), -1, value)
```

so *please see* the [full setFieldEntity](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldEntity Overload List](#)

Entity.setFieldEntity Method (String, Int32, Entity)

Sets a field (or component) value given a [Entity](#) argument.

```
public virtual void setFieldEntity(  
    string fieldName,  
    int index,  
    Entity value  
);
```

Parameters

fieldName

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

value

[Entity](#) new value for the requested field.

Remarks

The invocation

```
X.setFieldEntity(fieldName, index, value)
```

is equivalent to

```
X.setFieldEntity(X.entityClass.getEntityField(fieldName), index, value)
```

so *please see* the [full setFieldEntity](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldEntity Overload List](#)

Entity.setFieldLong Method

Sets a field value given a [long](#) argument.

Overload List

Full method to set an EntityField (or component) value given a [long](#) argument.

[public virtual void setFieldLong\(EntityField,int,long\):](#)

Sets an EntityField value given a [long](#) argument.

[public virtual void setFieldLong\(EntityField,long\):](#)

Sets a field (or component) value given a [long](#) argument.

[public virtual void setFieldLong\(string,int,long\):](#)

Sets a field value given a [long](#) argument.

[public virtual void setFieldLong\(string,long\):](#)

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.setFieldLong Method (EntityField, Int32, Int64)

Full method to set an EntityField (or component) value given a [long](#) argument.

```
public virtual void setFieldLong(  
    EntityField field,  
    int index,  
    long value  
);
```

Parameters

field

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

value

[long](#) new value for the requested field.

Remarks

This method may be used to set any

- any integer numeric field or component,
- any [ENUM](#) field or component,
- any well byte–[encoded](#) component of a [STRING](#),
- any **STRING** field.

This invocation throw an appropriate [FastTrackException](#) on

- any integer numeric *value* not representable with the corresponding [type](#) field (e.g. a negative value for an unsigned numeric type, a too big value for a short numeric type, etc...),
- any floating numeric field or component,
- any [STRUCT](#) or [UNION](#) field or component,
- any not well byte–**encoded** component of a **STRING** (i.e. any *value* not in [0..255] range),
- any unexisting *field* and/or *index*.

If the [bytes](#) representation [isShared](#) then this invocation affects all other representations.

Example

The following code–fragment shows a [alternative setFieldLong](#) invocation.

```
public QuerySpecific(  
    Connection connection,  
    int          entityID,  
    string       entityName,  
    string       entityDesc) : base(connection)  
{  
    name = entityDesc;  
    param = new Entity(SIB_DESC_CLASS_PARAMS_ID);  
    param.setFieldLong( "EntityID",  entityID);  
}
```



```
param.setFieldString("EntityName", entityName);  
open(SIB_DESC_CLASS_QUERY_ID, param);  
counter++;  
}
```

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldLong Overload List](#)

Entity.setFieldLong Method (EntityField, Int64)

Sets an EntityField value given a [long](#) argument.

```
public virtual void setFieldLong(  
    EntityField field,  
    long value  
);
```

Parameters

field
Interesting field.

value
[long](#) new value for the requested field.

Remarks

The invocation

```
setFieldLong(field, value)
```

is equivalent to

```
setFieldLong(field, -1, value)
```

so *please see* the [full setFieldLong](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldLong Overload List](#)

Entity.setFieldLong Method (String, Int32, Int64)

Sets a field (or component) value given a [long](#) argument.

```
public virtual void setFieldLong(  
    string fieldName,  
    int index,  
    long value  
);
```

Parameters

fieldName

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

value

[long](#) new value for the requested field.

Remarks

The invocation

```
X.setFieldLong(fieldName, index, value)
```

is equivalent to

```
X.setFieldLong(X.entityClass.getEntityField(fieldName), index, value)
```

so *please see* the [full setFieldLong](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldLong Overload List](#)

Entity.setFieldLong Method (String, Int64)

Sets a field value given a [long](#) argument.

```
public virtual void setFieldLong(  
    string fieldName,  
    long value  
);
```

Parameters

fieldName

Interesting field.

value

[long](#) new value for the requested field.

Remarks

The invocation

```
X.setFieldLong(fieldName, value)
```

is equivalent to

```
X.setFieldLong(X.entityClass.getEntityField(fieldName), -1, value)
```

so *please see* the [full setFieldLong](#) method for a full detailed description of it.

Example

```
public QuerySpecific(  
    Connection connection,  
    int          entityID,  
    string       entityName,  
    string       entityDesc) : base(connection)  
{  
    name = entityDesc;  
    param = new Entity(SIB_DESC_CLASS_PARAMS_ID);  
    param.setFieldLong( "EntityID",  entityID);  
    param.setFieldString("EntityName", entityName);  
    open(SIB_DESC_CLASS_QUERY_ID, param);  
    counter++;  
}
```

See Also

[Entity Class](#) | [ftapi dnet Namespace](#) | [Entity.setFieldLong Overload List](#)

Entity.setFieldString Method

Sets a field value given a [String](#) argument.

Overload List

Full method to set an EntityField (or component) value given a [String](#) argument.

[public virtual void setFieldString\(EntityField,int,string\);](#)

Sets an EntityField value given a [String](#) argument.

[public virtual void setFieldString\(EntityField,string\);](#)

Sets a field (or component) value given a [String](#) argument.

[public virtual void setFieldString\(string,int,string\);](#)

Sets a field value given a [String](#) argument.

[public virtual void setFieldString\(string,string\);](#)

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.setFieldString Method (EntityField, Int32, String)

Full method to set an EntityField (or component) value given a [String](#) argument.

```
public virtual void setFieldString(  
    EntityField field,  
    int index,  
    string value  
);
```

Parameters

field

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

value

[String](#) new value for the requested field.

Remarks

This method may be used to set any

- any numeric field or component,
- any [ENUM](#) field or component,
- any well byte–[encoded](#) component of a [STRING](#),
- any **STRING** field.

This invocation throw an appropriate [FastTrackException](#) on

- any string not correctly parsable for the corresponding numeric field with a specific [type](#) (e.g. a non–numeric parsable string as "Hello", a numeric parsable string corresponding to a negative value for an unsigned numeric type, a numeric parsable string corresponding to a too big value for a short numeric type, etc...),
- any [STRUCT](#) or [UNION](#) field or component,
- any not well byte–**encoded** component of a **STRING** (i.e. any *value* not parsable as an integer in range [0..255]),
- any unexisting *field* and/or *index*.

If the [bytes](#) representation [isShared](#) then this invocation affects all other representations.

Example

The following code–fragment shows a [alternative setFieldString](#) invocation.

```
public QuerySpecific(  
    Connection connection,  
    int          entityID,  
    string       entityName,  
    string       entityDesc) : base(connection)  
{  
    name = entityDesc;  
    param = new Entity(SIB_DESC_CLASS_PARAMS_ID);
```

```

param.setFieldLong( "EntityID",  entityID);
param.setFieldString("EntityName", entityName);
open(SIB_DESC_CLASS_QUERY_ID, param);
counter++;
}

```

The following code–fragment shows how to set fields of *different* types with `setFieldString`.

```

public Entity readNextOrder()
{
    Entity result = null;
    string line   = sr.ReadLine();
    if(line == null)
        sr.Close();
    else
    {
        Lib.lib.trace(TraceLevel.full, "Read order line \"" + line + "\"");
        MatchCollection matchCollection = regex.Matches(line);
        result = new Entity("FT_C_ORDER");
        result.setFieldString("Verb",    matchCollection[0].Value.Trim());
        result.setFieldString("FTSecID",matchCollection[1].Value.Trim());
        result.setFieldString("Price",   matchCollection[2].Value.Trim());
        result.setFieldString("Qty",     matchCollection[3].Value.Trim());
        Entity client = result.getFieldEntity("Client");
        client.setFieldString("ClientAccount", matchCollection[4].Value.Trim());
        client.setFieldString("Origin",      "Client");
        result.setFieldString("TimeInForce", "Good till day");
    }
    return result;
}

```

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldString Overload List](#)

Entity.setFieldString Method (EntityField, String)

Sets an EntityField value given a [String](#) argument.

```
public virtual void setFieldString(  
    EntityField field,  
    string value  
);
```

Parameters

field

Interesting field.

value

[String](#) new value for the requested field.

Remarks

The invocation

```
setFieldString(field, value)
```

is equivalent to

```
setFieldString(field, -1, value)
```

so *please see* the [full setFieldString](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldString Overload List](#)

Entity.setFieldString Method (String, Int32, String)

Sets a field (or component) value given a [String](#) argument.

```
public virtual void setFieldString(  
    string fieldName,  
    int index,  
    string value  
);
```

Parameters

fieldName

Interesting field.

index

Specific component–index for a vector field, or –1 for a non–vector field.

value

[String](#) new value for the requested field.

Remarks

The invocation

```
X.setFieldString(fieldName, index, value)
```

is equivalent to

```
X.setFieldString(X.entityClass.getEntityField(fieldName), index, value)
```

so *please see* the [full setFieldString](#) method for a full detailed description of it.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.setFieldString Overload List](#)

Entity.setFieldString Method (String, String)

Sets a field value given a [String](#) argument.

```
public virtual void setFieldString(  
    string fieldName,  
    string value  
);
```

Parameters

fieldName

Interesting field.

value

[String](#) new value for the requested field.

Remarks

The invocation

```
X.setFieldString(fieldName, value)
```

is equivalent to

```
X.setFieldString(X.entityClass.getEntityField(fieldName), -1, value)
```

so *please see* the [full setFieldString](#) method for a full detailed description of it.

Example

```
public QuerySpecific(  
    Connection connection,  
    int          entityID,  
    string       entityName,  
    string       entityDesc) : base(connection)  
{  
    name = entityDesc;  
    param = new Entity(SIB_DESC_CLASS_PARAMS_ID);  
    param.setFieldLong( "EntityID",  entityID);  
    param.setFieldString("EntityName", entityName);  
    open(SIB_DESC_CLASS_QUERY_ID, param);  
    counter++;  
}
```

See Also

[Entity Class](#) | [ftapi dnet Namespace](#) | [Entity.setFieldString Overload List](#)

Entity.ToString Method

Returns a single/multiple line [String](#) that represents the current Entity.

Overload List

Returns a single-line [String](#) that represents the current Entity.

[public override string ToString\(\);](#)

Returns a multi-line [String](#) that represents the current EntityFields values.

[public virtual string ToString\(bool.unt\);](#)

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#)

Entity.ToString Method ()

Returns a single-line [String](#) that represents the current Entity.

```
public override string ToString();
```

Return Value

A single-line [String](#) that represents the current Entity.

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.ToString Overload List](#)

Entity.ToString Method (Boolean, UInt32)

Returns a multi-line [String](#) that represents the current EntityFields values.

```
public virtual string ToString(  
    bool withHeader,  
    uint offset  
);
```

Parameters

withHeader

First line header request.

offset

Indentation value.

Return Value

A multi-line [String](#) that represents the current EntityFields values.

Implements

[IDump.ToString](#)

See Also

[Entity Class](#) | [ftapi_dnet Namespace](#) | [Entity.ToString Overload List](#)

EntityClass Class

[Entity](#)'s and [EntityField](#)'s type.

For a list of all members of this type, see [EntityClass Members](#).

[System.Object](#)

ftapi_dnet.EntityClass

public sealed class EntityClass

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Remarks

An EntityClass is logically defined by/with its properties:

- [classId](#) – A `uint` identifier.

If different from zero then this value is a unique identifier (i.e. a key) for the EntityClass.

- [className](#) – A `string` identifier.

This value is a unique identifier (i.e. a key) for the EntityClass.

- [type](#) – Its type.
- [entityFields](#) – The array of [EntityField](#) members of the EntityClass.
- [dataSize](#) – Size (in bytes) occupied by a C representation of a [Entity](#) of the EntityClass.

EntityClasses are only created through [loadEntityClasses](#).

The only two manners to find an EntityClass are:

- 1. – via the static [getEntityClass](#) method.
- 2. – via the [entityClass](#) property of an **Entity**.

It's guaranteed that if two EntityClass references X e Y are different then the two corresponding EntityClasses are different.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[EntityClass Members](#) | [ftapi_dnet Namespace](#) | [EntityClassType](#) | [EntityField](#) | [EntityFieldType](#)






EntityClass Members

[EntityClass overview](#)






Public Static Methods

 getEntityClass	Overloaded. Returns a reference to an EntityClass denoted by a given classId and/or className.
--	--

Public Instance Properties

 classId	Gets the <code>uint</code> identifier of this EntityClass.
 className	Gets the <code>string</code> identifier of this EntityClass.
 dataSize	Gets the size (in bytes) occupied by a C representation of a Entity of this EntityClass.
 entityFields	Gets the array of EntityField members of this EntityClass.
 type	Gets the type of this EntityClass.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 getEntityField	Returns the EntityField describing a given field.
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current EntityClass.






See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#) | [EntityClassType](#) | [EntityField](#) | [EntityFieldType](#)

EntityClass Properties

The properties of the **EntityClass** class are listed below. For a complete list of **EntityClass** class members, see the [EntityClass Members](#) topic.

Public Instance Properties

 classId	Gets the <code>uint</code> identifier of this EntityClass.
 className	Gets the <code>string</code> identifier of this EntityClass.
 dataSize	Gets the size (in bytes) occupied by a C representation of a Entity of this EntityClass.
 entityFields	Gets the array of EntityField members of this EntityClass.
 type	Gets the type of this EntityClass.

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#) | [EntityClassType](#) | [EntityField](#) | [EntityFieldType](#)

EntityClass.classId Property

Gets the `uint` identifier of this EntityClass.

```
public uint classId {get;}
```

Property Value

If different from zero then this value is a unique identifier (i.e. a key) for this EntityClass.

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#)

EntityClass.className Property

Gets the `string` identifier of this EntityClass.

```
public string className {get;}
```

Property Value

This value is a unique identifier (i.e. a key) for the EntityClass.

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#)

EntityClass.dataSize Property

Gets the size (in bytes) occupied by a C representation of a [Entity](#) of this EntityClass.

```
public uint dataSize {get;}
```

Property Value

This value is *not* the space occupied by an [Entity](#) instance of this EntityClass.

This value is *not* the space occupied by a transmission of an **Entity** of this EntityClass to/from a server.

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#)

EntityClass.entityFields Property

Gets the array of [EntityField](#) members of this EntityClass.

```
public EntityField[] entityFields {get;}
```

Property Value

`null` is never returned: if this EntityClass has no members then an empty array (i.e. an array with `Length` field = 0) is returned.

Depending on [EntityClassType](#)

- **STRUCT** or **UNION** – Returned EntityFields are the their corresponding fields.
- **ENUM** – Returned EntityFields are the corresponding [String](#) values associated to enum values in range `[0..Length-1]`.

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#)

EntityClass.type Property

Gets the type of this EntityClass.

```
public EntityClassType type {get;}
```

Property Value

No [NONE](#) value may be get from this property.


See Also

[EntityClass Class](#) | [ftapi dnet Namespace](#)






EntityClass Methods

The methods of the **EntityClass** class are listed below. For a complete list of **EntityClass** class members, see the [EntityClass Members](#) topic.

Public Static Methods

 getEntityClass	Overloaded. Returns a reference to an EntityClass denoted by a given classId and/or className.
--	--

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 getEntityField	Returns the EntityField describing a given field.
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current EntityClass.

See Also

[EntityClass Class](#) |
 [ftapi dnet Namespace](#) |
 [EntityClassType](#) |
 [EntityField](#) |
 [EntityFieldType](#)

EntityClass.getEntityClass Method

Returns a reference to an EntityClass denoted by a given classId and/or className.

Overload List

Returns a reference to an EntityClass denoted by a given className.

[public static EntityClass getEntityClass\(string\);](#)

Returns a reference to an EntityClass denoted by a given classId.

[public static EntityClass getEntityClass\(uint\);](#)

Returns a reference to an EntityClass denoted by a given pair (classId, className).

[public static EntityClass getEntityClass\(uint,string\);](#)

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#) | [loadEntityClasses](#)

EntityClass.getEntityClass Method (String)

Returns a reference to an EntityClass denoted by a given className.

```
public static EntityClass getEntityClass(  
    string className  
);
```

Parameters

className
null or the [String](#) key denoting the interested EntityClass.

Return Value

A reference to an EntityClass denoted by the given parameters.

null is returned if *className* parameter does not refer any existing loaded EntityClass.

Remarks

EntityClasses are searched through all classes previously loaded by [loadEntityClasses](#).

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#) | [EntityClass.getEntityClass Overload List](#) | [loadEntityClasses](#)

EntityClass.getEntityClass Method (UInt32)

Returns a reference to an EntityClass denoted by a given classId.

```
public static EntityClass getEntityClass(  
    uint classId  
);
```

Parameters

classId
Zero or the uint key denoting the interested EntityClass.

Return Value

A reference to an EntityClass denoted by the given parameters.

null is returned if *classId* parameter does not refer any existing loaded EntityClass.

Remarks

EntityClasses are searched through all classes previously loaded by [loadEntityClasses](#).

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#) | [EntityClass.getEntityClass Overload List](#) | [loadEntityClasses](#)

EntityClass.getEntityClass Method (UInt32, String)

Returns a reference to an EntityClass denoted by a given pair (classId, className).

```
public static EntityClass getEntityClass(  
    uint classId,  
    string className  
);
```

Parameters

classId

Zero or the `uint` key denoting the interested EntityClass.

className

null or the [String](#) key denoting the interested EntityClass.

Return Value

A reference to an EntityClass denoted by the given parameters.

null is returned

- if *classId* and *className* parameters does not refer any existing loaded EntityClass: e.g. it does not exist any EntityClass with the given *classId* and *className* values.
- if *classId* and *className* parameters are contradictories: e.g. there are two different EntityClasses X and Y with `X.classId == classId` and `Y.className == className`.

Remarks

EntityClasses are searched through all classes previously loaded by [loadEntityClasses](#).

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#) | [EntityClass.getEntityClass Overload List](#) | [loadEntityClasses](#)

EntityClass.getEntityField Method

Returns the [EntityField](#) describing a given field.

```
public EntityField getEntityField(  
    string field  
);
```

Parameters

field
Interested field name.

Return Value

A [EntityField](#) describing a given field.

`null` is returned if *field* does not exist in the current EntityClass.

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#)

EntityClass.ToString Method

Returns a single-line [String](#) that represents the current EntityClass.

```
public override string ToString();
```

Return Value

A single-line [String](#) that represents the current EntityClass.

See Also

[EntityClass Class](#) | [ftapi_dnet Namespace](#)

EntityField Class

A field of an [EntityClass](#).

For a list of all members of this type, see [EntityField Members](#).

[System.Object](#)

ftapi_dnet.EntityField

public sealed class EntityField

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

An EntityField is logically defined by/with its properties:

- **type** – [Type](#) of the field.
- **name** – Name of the field.
- **numElements** – Number of components when the field is a vector.
- **entityClass** – EntityClass of the field.
- **entityClassParent** – EntityClass in which the field is defined.
- **dataSize** – Size (in bytes) occupied by a C representation of a value of the field.
- **offset** – Displacement (in bytes) relative to the C representation of the field.

EntityFields are only created through [loadEntityClasses](#).

The only two manners to find an EntityField are:

- **1.** – via the [entityFields](#) property of an [EntityClass](#).
- **2.** – via the [getEntityField](#) method of an **EntityClass**.

It's guaranteed that if two EntityFields references X e Y are different then the two corresponding EntityFields are different.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)








See Also

[EntityField Members](#) | [ftapi_dnet Namespace](#) | [EntityFieldType](#) | [EntityClass](#) | [EntityClassType](#)





EntityField Members

[EntityField overview](#)

Public Instance Properties

 dataSize	Gets the size (in bytes) occupied by a C representation of a value of this EntityField.
 entityClass	Gets the EntityClass of this EntityField.
 entityClassParent	Gets the EntityClass in which this EntityField is defined.
 name	Gets the name of this EntityField.
 numElements	Gets the number of components for an EntityField vector.
 offset	Gets the displacement (in bytes) relative to the C representation of this EntityField.
 type	Gets the type of this EntityField.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current EntityField.








See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#) | [EntityFieldType](#) | [EntityClass](#) | [EntityClassType](#)

EntityField Properties

The properties of the **EntityField** class are listed below. For a complete list of **EntityField** class members, see the [EntityField Members](#) topic.

Public Instance Properties

 dataSize	Gets the size (in bytes) occupied by a C representation of a value of this EntityField.
 entityClass	Gets the EntityClass of this EntityField.
 entityClassParent	Gets the EntityClass in which this EntityField is defined.
 name	Gets the name of this EntityField.
 numElements	Gets the number of components for an EntityField vector.
 offset	Gets the displacement (in bytes) relative to the C representation of this EntityField.
 type	Gets the type of this EntityField.

See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#) | [EntityFieldType](#) | [EntityClass](#) | [EntityClassType](#)

EntityField.dataSize Property

Gets the size (in bytes) occupied by a C representation of a value of this EntityField.

```
public uint dataSize {get;}
```

Property Value

This value is *not* the space occupied, at run-time, by a value of this EntityField.

This value is *not* the space occupied by a transmission of a value of this EntityField to/from a server.

Remarks

See [EntityFieldType Compendium](#) for a quick view.

In case of vector EntityField (i.e. when [numElements](#) > 1) the returned values represents the full space occupied for the full vector and *not* the space occupied for a single component.

See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#)

EntityField.entityClass Property

Gets the EntityClass of this EntityField.

```
public EntityClass entityClass {get;}
```

Property Value

This value is null if this EntityField is not an EntityClass, i.e. when the [type](#) property does not evaluate null.

See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#)

EntityField.entityClassParent Property

Gets the EntityClass in which this EntityField is defined.

```
public EntityClass entityClassParent { get; }
```

Property Value

This value is never null.

See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#)

EntityField.name Property

Gets the name of this EntityField.

```
public string name {get;}
```

Property Value

Depending on [entityClassParent.type](#) this value may be

- [STRUCT](#) or [UNION](#) – Name of the field.
- [ENUM](#) – The [String](#) value associated to the corresponding enum value.

See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#)

EntityField.numElements Property

Gets the number of components for an EntityField vector.

```
public uint numElements {get;}
```

Property Value

A value = 1 indicates a non-vector EntityField.

A value > 1 indicates a vector EntityField.

[STRING](#) always retruns a value > 1.

Remarks

Every single component of an EntityField vector may be accessed (get/set) via a [Entity](#).getFieldX/setFieldX method invocation using an `index` parameter with a value in range `[0..numElements-1]`.

A non-vector EntityField may be accessed (get/set) via a **Entity**.getFieldX/setFieldX method invocation using an `index` parameter with value = -1.

See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#)

EntityField.offset Property

Gets the displacement (in bytes) relative to the C representation of this EntityField.

```
public uint offset {get;}
```

Property Value

This value has little interest in a .NET application.

See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#)

EntityField.type Property

Gets the type of this EntityField.

```
public EntityFieldType type {get;}
```

Property Value

A [NONE](#) value indicates an [EntityClass](#) field.





See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#)

EntityField Methods

The methods of the **EntityField** class are listed below. For a complete list of **EntityField** class members, see the [EntityField Members](#) topic.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current EntityField.

See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#) | [EntityFieldType](#) | [EntityClass](#) | [EntityClassType](#)

EntityField.ToString Method

Returns a single-line [String](#) that represents the current EntityField.

```
public override string ToString();
```

Return Value

A single-line [String](#) that represents the current EntityField.

See Also

[EntityField Class](#) | [ftapi_dnet Namespace](#)

FastTrackEventArgs Class

FastTrackEventArgs is the FastTrack base class for classes containing event data.

For a list of all members of this type, see [FastTrackEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

ftapi_dnet.FastTrackEventArgs

[Derived types](#)

```
public class FastTrackEventArgs : EventArgs, IDump
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[FastTrackEventArgs Members](#) | [ftapi_dnet Namespace](#)





FastTrackEventArgs Members

[FastTrackEventArgs overview](#)

Public Instance Constructors

 FastTrackEventArgs Constructor	Initializes a new instance of the FastTrackEventArgs class.
--	---

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Overloaded. Returns a single/multiple line String that represents the current public fields values.

See Also

[FastTrackEventArgs Class](#) | [ftapi_dnet Namespace](#)

FastTrackEventArgs Constructor

Initializes a new instance of the [FastTrackEventArgs](#) class.

```
public FastTrackEventArgs();
```





See Also

[FastTrackEventArgs Class](#) | [ftapi_dnet Namespace](#)

FastTrackEventArgs Methods

The methods of the **FastTrackEventArgs** class are listed below. For a complete list of **FastTrackEventArgs** class members, see the [FastTrackEventArgs Members](#) topic.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Overloaded. Returns a single/multiple line String that represents the current public fields values.

See Also

[FastTrackEventArgs Class](#) | [ftapi_dnet Namespace](#)

FastTrackEventArgs.ToString Method

Returns a single/multiple line [String](#) that represents the current public fields values.

Overload List

Returns a single–line [String](#) that represents the current public fields values.

[public override string ToString\(\);](#)

Returns a multi–line [String](#) that represents the current public fields values.

[public virtual string ToString\(bool.unt\);](#)

See Also

[FastTrackEventArgs Class](#) | [ftapi_dnet Namespace](#)

FastTrackEventArgs.ToString Method ()

Returns a single-line [String](#) that represents the current public fields values.

```
public override string ToString();
```

Return Value

A single-line [String](#) that represents the current public fields values.

Remarks

All fields are written on a single line with a format like:

```
FastTrackEvenetArgsSubClass[fieldName1=fieldValue1,...,fieldNameN=fieldValueN]
```

Example

A typically generated string will be:

```
ftapi_dnet.Subscription+NotifyEventArgs[entityAction=add,classID=30007,keyID=1,
timeStamp=753179494.16,isMasked=False,entity=entity 33 of class FT_C_MARKET(30007)]
```

The generated string is always in one single-line. The previuos cited output example in represented in two lines for didactic reason.

See Also

[FastTrackEventArgs Class](#) | [ftapi_dnet Namespace](#) | [FastTrackEventArgs.ToString Overload List](#)

FastTrackEventArgs.ToString Method (Boolean, UInt32)

Returns a multi-line [String](#) that represents the current public fields values.

```
public virtual string ToString(  
    bool withHeader,  
    uint offset  
);
```

Parameters

withHeader

First line header request.

offset

Indentation value.

Return Value

A multi-line [String](#) that represents the current public fields values.

Implements

[IDump.ToString](#)

See Also

[FastTrackEventArgs Class](#) | [ftapi_dnet Namespace](#) | [FastTrackEventArgs.ToString Overload List](#)

FastTrackException Class

The exception that is thrown when a non–fatal FastTrack application error occurs.

For a list of all members of this type, see [FastTrackException Members](#).

[System.Object](#)

[System.Exception](#)

[System.ApplicationException](#)

ftapi_dnet.FastTrackException

public class FastTrackException : ApplicationException

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Remarks

A FastTrackException is logically defined by/with the 5 properties

- [Message](#) – A full message describing the current FastTrackException.

The generic message format is:

```
Error E: I (in O) [M].
```

where E, I, O, M are the following 4 properties:

- **E** = [error](#) – An [Error](#) failure–code describing the current FastTrackException.
- **I** = [errorInfo](#) – A string describing the specific cause of the current FastTrack Exception.
- **O** = [relatedObject](#) – An [Object](#) related to the current FastTrackException.
- **M** = [method](#) – A string describing the method in which the current FastTrack Exception was created.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[FastTrackException Members](#) | [ftapi_dnet Namespace](#) | [Error](#)











FastTrackException Members

[FastTrackException overview](#)







Public Instance Constructors

 FastTrackException	Overloaded. Initializes a new instance of the FastTrackException class.
--	---

Public Instance Properties

 error	Gets an Error failure-code describing the current FastTrackException.
 errorInfo	Gets a string describing the specific cause of the current FastTrack Exception.
 HelpLink (inherited from Exception)	Gets or sets a link to the help file associated with this exception.
 InnerException (inherited from Exception)	Gets the Exception instance that caused the current exception.
 Message	Gets a full message describing the current FastTrackException.
 method	Gets a string describing the method in which the current FastTrack Exception was created.
 relatedObject	Gets an Object related to the current FastTrackException.
 Source (inherited from Exception)	Gets or sets the name of the application or the object that causes the error.
 StackTrace (inherited from Exception)	Gets a string representation of the frames on the call stack at the time the current exception was thrown.
 TargetSite (inherited from Exception)	Gets the method that throws the current exception.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetBaseException (inherited from Exception)	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions.
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetObjectData (inherited from Exception)	When overridden in a derived class, sets the SerializationInfo with information about the exception.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from Exception)	Creates and returns a string representation of the current exception.

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#) | [Error](#)

FastTrackException Constructor

Initializes a new instance of the FastTrackException class.

Overload List

Shorthand constructor to initialize a new instance of the FastTrackException class using some defaults values.

[public FastTrackException\(Error\):](#)

Shorthand constructor to initialize a new instance of the FastTrackException class using some defaults values.

[public FastTrackException\(Error,string\):](#)

Shorthand constructor to initialize a new instance of the FastTrackException class using some defaults values.

[public FastTrackException\(object>Error\):](#)

Shorthand constructor to initialize a new instance of the FastTrackException class using some defaults values.

[public FastTrackException\(object>Error,string\):](#)

Full constructor to initialize a new instance of the FastTrackException class.

[public FastTrackException\(object>Error,string,Exception\):](#)

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#)

FastTrackException Constructor (Object, Error, String, Exception)

Full constructor to initialize a new instance of the FastTrackException class.

```
public FastTrackException(  
    object relatedObject,  
    Error error,  
    string info,  
    Exception innerException  
);
```

Parameters

relatedObject

The object that caused the exception, or null.

This value will be available in the [relatedObject](#) property.

error

The [Error](#) that caused the exception.

This value will be available in the [error](#) property.

info

The specific information message describing the cause of the exception, or null or the empty string "".

This value will be available in the [errorInfo](#) property.

innerException

The exception that is the cause of the current exception.

This value will be available in the [InnerException](#) property.

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#) | [FastTrackException Constructor Overload List](#)

FastTrackException Constructor (Object, Error, String)

Shorthand constructor to initialize a new instance of the FastTrackException class using some defaults values.

```
public FastTrackException(  
    object relatedObject,  
    Error error,  
    string info  
);
```

Parameters

relatedObject

The object that caused the exception, or null.

error

The [Error](#) that caused the exception.

info

The specific information message describing the cause of the exception, or null or the empty string "".

Remarks

The invocation

```
new FastTrackException(relatedObject, error, info)
```

is equivalent to

```
new FastTrackException(relatedObject, error, info, null)
```

so *please see* the [full constructor](#) for a full detailed description of it.

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#) | [FastTrackException Constructor Overload List](#)

FastTrackException Constructor (Object, Error)

Shorthand constructor to initialize a new instance of the FastTrackException class using some defaults values.

```
public FastTrackException(  
    object relatedObject,  
    Error error  
);
```

Parameters

relatedObject

The object that caused the exception, or null.

error

The [Error](#) that caused the exception.

Remarks

The invocation

```
new FastTrackException(relatedObject, error)
```

is equivalent to

```
new FastTrackException(relatedObject, error, null, null)
```

so *please see* the [full constructor](#) for a full detailed description of it.

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#) | [FastTrackException Constructor Overload List](#)

FastTrackException Constructor (Error, String)

Shorthand constructor to initialize a new instance of the FastTrackException class using some defaults values.

```
public FastTrackException(  
    Error error,  
    string info  
);
```

Parameters

error

The [Error](#) that caused the exception.

info

The specific information message describing the cause of the exception, or `null` or the empty string `""`.

Remarks

The invocation

```
new FastTrackException(error, info)
```

is equivalent to

```
new FastTrackException(null, error, info, null)
```

so *please see* the [full constructor](#) for a full detailed description of it.

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#) | [FastTrackException Constructor Overload List](#)

FastTrackException Constructor (Error)

Shorthand constructor to initialize a new instance of the FastTrackException class using some defaults values.

```
public FastTrackException(  
    Error error  
);
```

Parameters

error

The [Error](#) that caused the exception.

Remarks

The invocation

```
new FastTrackException(error)
```

is equivalent to

```
new FastTrackException(null, error, null, null)
```

so *please see* the [full constructor](#) for a full detailed description of it.











See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#) | [FastTrackException Constructor Overload List](#)

FastTrackException Properties

The properties of the **FastTrackException** class are listed below. For a complete list of **FastTrackException** class members, see the [FastTrackException Members](#) topic.

Public Instance Properties

 error	Gets an Error failure-code describing the current FastTrackException.
 errorInfo	Gets a string describing the specific cause of the current FastTrack Exception.
 HelpLink (inherited from Exception)	Gets or sets a link to the help file associated with this exception.
 InnerException (inherited from Exception)	Gets the Exception instance that caused the current exception.
 Message	Gets a full message describing the current FastTrackException.
 method	Gets a string describing the method in which the current FastTrack Exception was created.
 relatedObject	Gets an Object related to the current FastTrackException.
 Source (inherited from Exception)	Gets or sets the name of the application or the object that causes the error.
 StackTrace (inherited from Exception)	Gets a string representation of the frames on the call stack at the time the current exception was thrown.
 TargetSite (inherited from Exception)	Gets the method that throws the current exception.

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#) | [Error](#)

FastTrackException.error Property

Gets an [Error](#) failure-code describing the current FastTrackException.

```
public virtual Error error {get;}
```

Property Value

The Error failure-code that categorize the reason for the exception.

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#)

FastTrackException.errorInfo Property

Gets a string describing the specific cause of the current FastTrack Exception.

```
public virtual string errorInfo {get;}
```

Property Value

The string that explains the specific cause for the exception.

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#)

FastTrackException.Message Property

Gets a full message describing the current FastTrackException.

```
public override string Message {get;}
```

Property Value

The error message that explains the reason for the exception.

The generic format of the message is:

```
Error E: I (in O) [M].
```

where

- **E** – is [error](#).
- **I** – is [errorInfo](#).
- **O** – is [relatedObject](#).
- **M** – is [method](#).

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#)

FastTrackException.method Property

Gets a string describing the method in which the current FastTrack Exception was created.

```
public virtual string method {get;}
```

Property Value

The method name where the exception was created.

Please note that this is *not* the method where the exception was thrown, even if, typically, the two methods are coinciding.

A full stack–trace is always generated using the trace sub–system with [error](#) level and [ftapi_dnet](#) category.

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#)

FastTrackException.relatedObject Property

Gets an [Object](#) related to the current FastTrackException.

```
public virtual object relatedObject {get;}
```

Property Value

The Object that caused the exception, or the null value.

See Also

[FastTrackException Class](#) | [ftapi_dnet Namespace](#)

Filter Class

A manner to restrict the set of values notified by a [Subscription](#).

For a list of all members of this type, see [Filter Members](#).

System.Object

[ftapi_dnet.Disposable](#)

[ftapi_dnet.ConnectionActivity](#)

ftapi_dnet.Filter

public class Filter : [ConnectionActivity](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Remarks

A filter may be used in a subscription to restrict (at the server level) the set of entities that will be [notified](#) to the client application. See [Partial Subscriptions](#) to understand how to use this class.

A filter is logically defined by/with the [open](#) parameters:

- **entityClass** – the associated [EntityClass](#),
- **type** – the filter type,
- **definition** – the filter definition.

The precise meaning of these 3 things depends from the particular filter and they are specified by the server.

Once defined, via **open**, a filter may also be extended via [set](#).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[Filter Members](#) | [ftapi_dnet Namespace](#) | [Partial Subscriptions](#)





Filter Members

[Filter overview](#)












Public Instance Constructors

 Filter Constructor	Initializes a new instance of the Filter class.
--	---




Public Instance Properties

 connection (inherited from ConnectionActivity)	Gets the associated Connection.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

Public Instance Methods

 Close (inherited from Disposable)	Logically destroy the current object, making it again available.
 Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 onClose	Method automatically invoked when the server answer to Close arrives.
 onOpen	Method automatically invoked when the server answer to open arrives.
 onSet	Method automatically invoked when the server answer to set arrives.
 open	Overloaded. Creates this filter on the server.
 set	Extends, with a given new definition, this filter on the server.
 ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

Public Instance Events

 closeEvent	Event asynchronously raised when the server answer to Close arrives.
 openEvent	Event asynchronously raised when the server answer to open arrives.
 setEvent	Event asynchronously raised when the server answer to set arrives.

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [Partial Subscriptions](#)

Filter Constructor

Initializes a new instance of the Filter class.

```
public Filter(  
    Connection connection  
);
```

Parameters

connection
the parent Connection.

Remarks

Status

- on successfully Exit – [init](#)

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#)

Filter Methods

The methods of the **Filter** class are listed below. For a complete list of **Filter** class members, see the [Filter Members](#) topic.

Public Instance Methods

✦ Close (inherited from Disposable)	Logically destroy the current object, making it again available.
✦ Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
✦ Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
✦ GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
✦ GetType (inherited from Object)	Gets the Type of the current instance.
✦ onClose	Method automatically invoked when the server answer to Close arrives.
✦ onOpen	Method automatically invoked when the server answer to open arrives.
✦ onSet	Method automatically invoked when the server answer to set arrives.
✦ open	Overloaded. Creates this filter on the server.
✦ set	Extends, with a given new definition, this filter on the server.
✦ ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [Partial Subscriptions](#)

Filter.onClose Method

Method automatically invoked when the server answer to [Close](#) arrives.

```
public virtual void onClose(  
    CloseEventArgs ce  
);
```

Parameters

ce
Server answer to [Close](#).

Remarks

If *ce.result* is [OK](#) then the server has destroyed the filter otherwise some unknown error occurred.

In both cases the current [Status](#) changed to [init](#).

This method must be properly extended in Filter sub-classes since it does nothing in the Filter class.

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [Close](#) | [Filter.CloseEventHandler](#)

Filter.onOpen Method

Method automatically invoked when the server answer to [open](#) arrives.

```
public virtual void onOpen(  
    OpenEventArgs oe  
);
```

Parameters

oe
Server answer to [open](#).

Remarks

If *oe.result* is [OK](#) then

- the server accepted the filter creation,
- the current [Status](#) changed to [running](#),
- the server may now accept [set](#) on this filter,
- the server may now accept subscription creation based on this filter.

otherwise

- the server rejected the filter creation (see **result** to understand why),
- the current **Status** changed to [init](#),
- the server does not accept any **set** on this filter,
- the server does not accept any subscription creation based on this filter.

This method must be properly extended in Filter sub-classes since it does nothing in the Filter class.

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [open](#) | [Filter.OpenEventHandler](#)

Filter.onSet Method

Method automatically invoked when the server answer to [set](#) arrives.

```
public virtual void onSet(  
    SetEventArgs se  
);
```

Parameters

se
Server answer to [set](#).

Remarks

If *oe.result* is [OK](#) then the server has accepted the filter setting otherwise some error occurred (see *oe.result* to understand why).

In both cases:

- the current [Status](#) remains [running](#),
- the server may accept additional [set](#) on this filter,
- the server may accept subscription creation based on this filter.

This method must be properly extended in Filter sub-classes since it does nothing in the Filter class.

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [set](#) | [Filter.SetEventHandler](#)

Filter.open Method

Creates this filter on the server.

Overload List

Full method to create this filter on the server.

[public virtual void open\(EntityClass,uint,string\):](#)

Alternative method to create this filter on the server.

[public virtual void open\(string,uint,string\):](#)

Alternative method to create this filter on the server.

[public virtual void open\(uint,uint,string\):](#)

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Filter.open Method (EntityClass, UInt32, String)

Full method to create this filter on the server.

```
public virtual void open(
    EntityClass entityClass,
    uint type,
    string definition
);
```

Parameters

entityClass

The associated [EntityClass](#) of the filter.

The server may subsequently returns a [invalidClass](#) failure-code if it does not comply with this value.

type

The associated type of the filter.

The server may subsequently returns a [invalidFilterType](#) failure-code if it does not comply with this value.

definition

The optional associated definition of the filter.

The server may subsequently returns [syntaxError](#) or [invalidFilterLen](#) failure-codes if it does not comply with this value.

This value may be subsequently changed via a [set](#) invocation.

Remarks

Three parameters represent the 3 things (*entityClass*, *type*, optional *definition*) that define a filter. The precise meaning of these 3 things depends from the particular filter and they are specified by the server.

If this method invocation completed successfully, then

- the filter create request was sent to server,
- the current [Status](#) changes to [opening](#),
- when the server-answer will be available the [openEvent](#) will be raised.

otherwise

- the client rejected the filter create request,
- the filter create request was **not** sent to server,
- the current **Status** remains unchanged,
- no **openEvent** will be raised,
- an appropriate [FastTrackException](#) will be thrown.

Status

- **required on Entry** – [init](#)
- **on successfully Exit** – [opening](#)

Events

Event Type	Reason
openEvent	Event asynchronously raised when the server answer to a successfully open arrives.

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [Filter.open Overload List](#) | [onOpen](#)

Filter.open Method (String, UInt32, String)

Alternative method to create this filter on the server.

```
public virtual void open(  
    string className,  
    uint type,  
    string definition  
);
```

Parameters

className

Name of the associated [EntityClass](#) of the filter.

type

The associated type of the filter.

definition

The optional associated definition of the filter.

Remarks

The invocation

```
open(className, type, definition)
```

is equivalent to

```
open(EntityClass.getEntityClass(0, className), type, definition)
```

so *please see* the [full open](#) method for a full detailed description of it.

Status & Events

Please see the **full open** method for a full detailed description of them.

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [Filter.open Overload List](#)

Filter.open Method (UInt32, UInt32, String)

Alternative method to create this filter on the server.

```
public virtual void open(  
    uint classID,  
    uint type,  
    string definition  
);
```

Parameters

classID

Identifier of the associated [EntityClass](#) of the filter.

type

The associated type of the filter.

definition

The optional associated definition of the filter.

Remarks

The invocation

```
open(classID, type, definition)
```

is equivalent to

```
open(EntityClass.getEntityClass(classID, null), type, definition)
```

so *please see* the [full open](#) method for a full detailed description of it.

Status & Events

Please see the **full open** method for a full detailed description of them.

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [Filter.open Overload List](#)

Filter.set Method

Extends, with a given new definition, this filter on the server.

```
public virtual void set(
    string definition
);
```

Parameters

definition

The new associated definition to associate to this filter.

The server may subsequently returns [syntaxError](#) or [invalidFilterLen](#) failure-codes if it does not comply with this value.

Remarks

If this method invocation completed successfully, then

- the request was sent to server,
- when the server-answer will be available the [setEvent](#) will be raised.

otherwise

- the client rejected the request,
- the request was **not** sent to server,
- no **setEvent** will be raised,
- an appropriate [FastTrackException](#) will be thrown.

In both cases the current [Status](#) remains unchanged.

Status

- required on Entry – [running](#)
- on successfully Exit – **running**

Events

Event Type	Reason
setEvent	Event asynchronously raised when the server answer to a successfully set arrives.




See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [onSet](#)

Filter Events

The events of the **Filter** class are listed below. For a complete list of **Filter** class members, see the [Filter Members](#) topic.

Public Instance Events

 closeEvent	Event asynchronously raised when the server answer to Close arrives.
 openEvent	Event asynchronously raised when the server answer to open arrives.
 setEvent	Event asynchronously raised when the server answer to set arrives.

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [Partial Subscriptions](#)

Filter.closeEvent Event

Event asynchronously raised when the server answer to [Close](#) arrives.

```
public event CloseEventHandler closeEvent;
```

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [onClose](#)

Filter.openEvent Event

Event asynchronously raised when the server answer to [open](#) arrives.

```
public event OpenEventHandler openEvent;
```

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Filter.setEvent Event

Event asynchronously raised when the server answer to [set](#) arrives.

```
public event SetEventHandler setEvent;
```

See Also

[Filter Class](#) | [ftapi_dnet Namespace](#) | [onSet](#)

Filter.CloseEventArgs Class

Filter server answer to [Close](#).

For a list of all members of this type, see [Filter.CloseEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Filter.CloseEventArgs

public class Filter.CloseEventArgs : [FastTrackEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[Filter.CloseEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onClose](#)





Filter.CloseEventArgs Members

[Filter.CloseEventArgs overview](#)

Public Instance Fields

 result	Server filter–destruction failure–code.
--	---

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.


See Also

[Filter.CloseEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onClose](#)

Filter.CloseEventArgs Fields

The fields of the **Filter.CloseEventArgs** class are listed below. For a complete list of **Filter.CloseEventArgs** class members, see the [Filter.CloseEventArgs Members](#) topic.

Public Instance Fields

 result	Server filter–destruction failure–code.
--	---

See Also

[Filter.CloseEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onClose](#)

Filter.CloseEventArgs.result Field

Server filter–destruction failure–code.

```
public readonly FilterClose result;
```

See Also

[Filter.CloseEventArgs Class](#) | [ftapi_dnet Namespace](#)

Filter.OpenEventArgs Class

Filter server answer to [open](#).

For a list of all members of this type, see [Filter.OpenEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Filter.OpenEventArgs

public class Filter.OpenEventArgs : [FastTrackEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[Filter.OpenEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onOpen](#)





Filter.OpenEventArgs Members

[Filter.OpenEventArgs overview](#)

Public Instance Fields

 result	Server filter–creation failure–code.
--	--------------------------------------

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.


See Also

[Filter.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Filter.OpenEventArgs Fields

The fields of the **Filter.OpenEventArgs** class are listed below. For a complete list of **Filter.OpenEventArgs** class members, see the [Filter.OpenEventArgs Members](#) topic.

Public Instance Fields

 result	Server filter-creation failure-code.
--	--------------------------------------

See Also

[Filter.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Filter.OpenEventArgs.result Field

Server filter—creation failure—code.

```
public readonly FilterOpen result;
```

See Also

[Filter.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Filter.SetEventArgs Class

Filter server answer to [set](#).

For a list of all members of this type, see [Filter.SetEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Filter.SetEventArgs

public class Filter.SetEventArgs : [FastTrackEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[Filter.SetEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onSet](#)





Filter.SetEventArgs Members

[Filter.SetEventArgs overview](#)

Public Instance Fields

 result	Server filter–extension failure–code.
--	---------------------------------------

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.


See Also

[Filter.SetEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onSet](#)

Filter.SetEventArgs Fields

The fields of the **Filter.SetEventArgs** class are listed below. For a complete list of **Filter.SetEventArgs** class members, see the [Filter.SetEventArgs Members](#) topic.

Public Instance Fields

 result	Server filter-extension failure-code.
--	---------------------------------------

See Also

[Filter.SetEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onSet](#)

Filter.SetEventArgs.result Field

Server filter—extension failure—code.

```
public readonly FilterSet result;
```

See Also

[Filter.SetEventArgs Class](#) | [ftapi_dnet Namespace](#)

KeyValue Class

A (partial or full) value of a key of an EntityClass.

For a list of all members of this type, see [KeyValue Members](#).

System.Object

[ftapi_dnet.Disposable](#)

ftapi_dnet.KeyValue

public class KeyValue : [Disposable](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

A KeyValue is an ordered sequence of K values corresponding to the ordered sequence of N ($0 < K \leq N$) [segments](#) that describe a [key](#) of an EntityClass.

KeyValues may be

- **partial** – if $K < N$
- **full** – if $K = N$

Partial KeyValues are used in [Partial Subscriptions](#).

KeyValues are made extracting the ordered sequence of K values from a given Entity.

Example

Given the following EntityClasses (described in an invented meta-language)

```

ENUM VERB(0)                                // an ENUM EntityClass with name="VERB" and id=0
    "Buy"                                     // first ENUM constant with value 0 = "Buy"
    "Sell"                                   // second ENUM constant with value 1 = "Sell"
ENDENUM

STRUCT SEQUENCE(0)                           // a STRUCT EntityClass with name="SEQUENCE" and id=0
    LONG     SequenceNumber                 // a single LONG
    CHAR     SequenceType                   // a single CHAR
ENDSTRUCT

STRUCT PROPOSAL(2468)                        // a STRUCT EntityClass with name="PROPOSAL" and id=2468
    SEQUENCE Sequence                       // a single STRUCT SEQUENCE(0)
    SEQUENCE LeftRight[2]                   // a vector of 2 STRUCT SEQUENCE(0)
    VERB     Verb                           // a single ENUM VERB(0)
    STRING   Operator(4)                    // a single STRING with a max of 4 characters
    LONG     BondCode                       // a single LONG
    INT      PriceQuantity[2]               // a vector of 2 INT
    BOOLEAN  Flags[3]                       // a vector of 3 BOOLEAN
ENDSTRUCT
  
```

where the following key (KeyID = 2 on PROPOSAL (2486) EntityClass) is defined

```
KEY( PROPOSAL, 2)
  Verb
  PriceQuantity[0]
  PriceQuantity[1]
  Sequence.SequenceNumber
  BondCode
ENDKEY
```

and with an Entity X with the following value

```
Entity 567 of class PROPOSAL(2468)
  Sequence =
    SequenceNumber = 1234
    SequenceType = 65
  LeftRight[0] =
    SequenceNumber = 1001
    SequenceType = 66
  LeftRight[1] =
    SequenceNumber = 1002
    SequenceType = 67
  Verb = 1 = Sell
  Operator = ABC
  BondCode = 9876
  PriceQuantity[0] = 50
  PriceQuantity[1] = 90
```

then

- this is an example of a full (5/5 segments) key value

```
Key 2 of class PROPOSAL(2468)
  Verb = 1 = Sell
  PriceQuantity[0] = 50
  PriceQuantity[1] = 90
  Sequence.SequenceNumber = 1234
  BondCode = 9876
```

generated by

```
Entity X;
...
KeyValue kv = new KeyValue();
kv.set(X, 2, 0); // or else kv.set(X, 2, 5)
```

- this is an example of a partial (3/5 segments) key value

```
Key 2 of class PROPOSAL(2468)
  Verb = 1 = Sell
  PriceQuantity[0] = 50
  PriceQuantity[1] = 90
```

generated by

```
Entity X;
...
KeyValue kv = new KeyValue();
kv.set(X, 2, 3);
```

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[KeyValue Members](#) | [ftapi_dnet Namespace](#) | [Subscription.open](#) | [Partial Subscriptions](#)




KeyValue Members

[KeyValue overview](#)








Public Instance Constructors

 KeyValue Constructor	Initializes a new instance of the KeyValue class.
--	---

Public Instance Properties

 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

Public Instance Methods

 Close (inherited from Disposable)	Logically destroy the current object, making it again available.
 Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 set	Make the (partial or full) KeyValue deriving it from a given Entity.
 ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[KeyValue Class](#) | [ftapi_dnet Namespace](#) | [Subscription.open](#) | [Partial Subscriptions](#)

KeyValue Constructor

Initializes a new instance of the KeyValue class.

```
public KeyValue();
```

Remarks

Status

- on successfully Exit – [opening](#)

See Also

[KeyValue Class](#) | [ftapi_dnet Namespace](#)

KeyValue Methods

The methods of the **KeyValue** class are listed below. For a complete list of **KeyValue** class members, see the [KeyValue Members](#) topic.

Public Instance Methods

◆ Close (inherited from Disposable)	Logically destroy the current object, making it again available.
◆ Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
◆ Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
◆ GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
◆ GetType (inherited from Object)	Gets the Type of the current instance.
◆ set	Make the (partial or full) KeyValue deriving it from a given Entity.
◆ ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[KeyValue Class](#) | [ftapi_dnet Namespace](#) | [Subscription.open](#) | [Partial Subscriptions](#)

KeyValue.set Method

Make the (partial or full) KeyValue deriving it from a given Entity.

```
public virtual void set(  
    Entity entity,  
    int keyID,  
    uint numSegments  
);
```

Parameters

entity

Entity value from which this KeyValue must be derived.

keyID

A KeyID of the *entity*'s EntityClass.

numSegments

Num. of segments to be set.

If *keyID* has N segments then *numSegments* may be:

- $0 < \text{numSegments} < N$ – to define a partial KeyValue.
- $\text{numSegments} = 0$ or $\text{numSegments} = N$ – to define a full KeyValue.

Remarks

Please see [KeyValues](#) to have a formal detailed description of KeyValues.

Status

- required on Entry – [init](#)
- on successfully Exit – [running](#)

Example

```
Entity X;  
...  
KeyValue kv = new KeyValue();  
kv.set(X, 2, 3);
```

See Also

[KeyValue Class](#) | [ftapi_dnet Namespace](#) | [KeyValues](#)

Lib Class

Main entry–point to use .NET FastTrack API.

For a list of all members of this type, see [Lib Members](#).

System.Object
[ftapi_dnet.Disposable](#)
ftapi_dnet.Lib

public sealed class Lib : [Disposable](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Remarks

To start to use this library please read the [Overview](#) or watch the source–code examples available in FTApi .Net Application Examples Manual

A singleton of this class is available in the [lib](#) static field of this class. Using this field it's possible to access any functionality exposed by this library.

This class is typically used to start–up any application program in two phases

- **EntityClasses Loading Phase** – This phase consists of:
 - a Lib.lib.[open](#) invocation.
 - an optional trace customization.
 - a set of Lib.lib.[loadEntityClasses](#) invocations.
- **Application Activities Phase** – This phase consists of:
 - a Lib.lib.[start](#) invocation.
 - all the specific application activities needed.

Here there is a syntetic meta–code summary:

```
Lib.lib.open(ThreadingMode.multipleThreads);
Lib.lib.traceEvent += new Lib.TraceEventHandler(myTraceEventHandler);
Lib.lib.setTraceMode(<my trace options>);
Lib.lib.loadEntityClasses(null, myMarketLibrary);
Lib.lib.start();
```

Implementation Threads

The underlying implementation of .NET FastTrack API library need to make continuously some activities:

- read/write data from/to the server, and
- raise the appropriate asynchronous events to the client application when data/answers are arriving from the server.

In the .NET platform the natural preferred way to do this is via a set of one or more implementation background threads but the ultimate choice on what to do is left to the application.

The application may use one of the three [ThreadingMode](#) constants during the **library initialization** to specify which will be the library behaviour:

- [multipleThreads](#) – This is the *suggested* value.

With this value the .NET FastTrack API library implementation will use one or more background threads that will be automatically started by the **start** method invocation.

The behaviour of these threads is controlled by [millisecondsTimeout](#), [stopIfNoConnections](#) and [maxInactivityInterval](#) properties.

Event raising: the various events will be raised in one of the background threads.

- [singleThread](#) and [singleThreadSafe](#) – With these values *no* thread will be used and the responsibility for the previous cited activities is left to the application that must periodically and quickly invoke the Lib.[Select](#) and Lib.[step](#) methods as shown in the **Remarks** section of **Select** method documentation.

Event raising: the various events will be raised in the same thread that invoked Lib.**step** and exactly during this invocation.

The difference among `singleThread` and `singleThreadSafe` is

- **singleThread** – The application *cannot* use its own threads because the various .NET FastTrack API calls are *not* thread-safe.
- **singleThreadSafe** – The application is free to use its own threads because the various .NET FastTrack API calls are thread-safe.

If the client application use its own threads (this is possible only with `multipleThreads` or `singleThreadSafe`) then it must pay attention in order to have a proper synchronization amongs all its own threads. In the .NET platform this is done through an appropriate use of the [Monitor](#) class (or, in C#, through the `lock` statement).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[Lib Members](#) | [ftapi_dnet Namespace](#) |









Lib Members

[Lib overview](#)













Public Static Fields

 lib	Reference to the Lib singleton.
---	---

Public Instance Properties




 encoding	Gets or sets the Encoding to be used in any String ? byte[] transformation.
 maxInactivityInterval	Gets or sets a maximum inactivity TimeSpan controlling the automatic stop of background threads.
 millisecondsTimeout	Gets or sets the internal timeout used by background threads.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 stopIfNoConnections	Gets or sets the indication about an automatic stop of background threads.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .
 version	Gets the current library version.

Public Instance Methods

 Close (inherited from Disposable)	Logically destroy the current object, making it again available.
 Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 loadEntityClasses	Extends the set of known EntityClasses.
 open	Initializes the library, starting the EntityClasses loading phase.
 Select	Determines the status of one or more application sockets.
 setTraceMode	Customize the trace sub-system.
 start	Starts the library usage, ending the EntityClasses loading phase.
 step	In non- multipleThreads mode operates the communication in two directions: Client ? any connected Server.
 ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

 trace	Generates applicative trace events.
---	-------------------------------------

Public Instance Events

 CommunicationExceptionEvent	Event raised when iteration of the main communication loop stops or fails. in multipleThreads mode only.
 CommunicationIdleEvent	Event raised every iteration of the main communication loop. in multipleThreads mode only.
 traceEvent	Event raised when a trace message must be programmatically traced as requested by setTraceMode .


See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) |

Lib Fields

The fields of the **Lib** class are listed below. For a complete list of **Lib** class members, see the [Lib Members](#) topic.

Public Static Fields

 lib	Reference to the Lib singleton.
---	---

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) |

Lib.lib Field

Reference to the [Lib](#) singleton.

```
public static readonly Lib lib;
```

Remarks

Every access to any [Lib](#) method/property/event must be made via this reference automatically created, at start-up time, with a [init Status](#).









See Also

[Lib Class](#) | [ftapi_dnet Namespace](#)

Lib Properties

The properties of the **Lib** class are listed below. For a complete list of **Lib** class members, see the [Lib Members](#) topic.

Public Instance Properties

 encoding	Gets or sets the Encoding to be used in any String ? <code>byte[]</code> transformation.
 maxInactivityInterval	Gets or sets a maximum inactivity TimeSpan controlling the automatic stop of background threads.
 millisecondsTimeout	Gets or sets the internal timeout used by background threads.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 stopIfNoConnections	Gets or sets the indication about an automatic stop of background threads.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .
 version	Gets the current library version.

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) |

Lib.encoding Property

Gets or sets the [Encoding](#) to be used in any [String](#) ? byte[] transformation.

```
public System.Text.Encoding encoding {get; set;}
```

Property Value

To understand what is a [String](#) ? byte[] transformation see the [STRING](#) documentation where an example regarding the "A?B" string is full explained.

The initial default value of this property is the first runtime-available encoding, chosen from the following ordered list

- [Encoding.GetEncoding\("iso-8859-15"\)](#) – This encoding uses a 1 character ? 1 byte correspondence. It's the mostly used encoding in the EU countries.
- [Encoding.GetEncoding\("iso-8859-1"\)](#) – This encoding uses a 1 character ? 1 byte correspondence but it encode some characters in different manner with respect to the most used "iso-8859-15" encoding: e.g. "iso-8859-1" uses the byte values 0xA4 (164) to represent the currency sign ¤ but it is the EUR sign € in "iso-8859-15".
- [new ASCIIEncoding\(\)](#) – This encoding uses a 1 character ? 1 byte correspondence but it encodes only bytes and characters in range [0..127]: e.g. the accented letters and the EUR sign € cannot be encoded.
- [Encoding.GetEncoding\("windows-1252"\)](#) – This encoding uses a 1 character ? 1 byte correspondence but it's Windows-specific and it encode some characters in different manner with respect to the most used "iso-8859-15" encoding: e.g. it encodes EUR sign € with the 0x80 (128) byte value which is different from byte values 0xA4 (164) used by "iso-8859-15".
- [new UTF8Encoding\(false, true\)](#) – It may encode a character with one to four bytes: actually many FastTrack market/services typically require a 1 character ? 1 byte correspondence.
- [Encoding.Default](#) – Its value depends on particular host where the application runs.

Remarks

Please note that, as a limitation of the current library implementation, this value is a unique globally value used by all [Entity](#) transmitted/received over all [Connection](#). In a future implementation there will be an `encoding` property associated to every Entity and every Connection in a manner that the [String](#) ? byte[] transformation will be automatically performed (if necessary) on the basis of the Connection `encoding` property.

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [STRING](#)

Lib.maxInactivityInterval Property

Gets or sets a maximum inactivity [TimeSpan](#) controlling the automatic stop of background threads.

```
public System.TimeSpan maxInactivityInterval { get; set; }
```

Property Value

This value is useful only when the library runs in [multipleThreads](#) mode and it have the following meaning

- **MaxValue** – background threads runs forever, until an explicit `Lib.Close`/`Lib.Dispose` is made or another constraint (e.g. see [stopIfNoConnections](#)) will occur.
- **any other value** – background threads automatically die when they does anything for at least the indicated interval.

Example

```
#if DEBUG
    Lib.lib.maxInactivityInterval = new TimeSpan(0, 0, 30);
#endif
Lib.lib.start();
run(args);
```

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [Implementation Threads](#)

Lib.millisecondsTimeout Property

Gets or sets the internal timeout used by background threads.

```
public uint millisecondsTimeout { get; set; }
```

Property Value

This value is useful only when the library runs in [multipleThreads](#) mode and it represents the maximum timeout spent by background threads in an internal proprietary `Select`.

The initial default value for this property is 1000 (i.e. 1 second).

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [Implementation Threads](#)

Lib.stopIfNoConnections Property

Gets or sets the indication about an automatic stop of background threads.

```
public bool stopIfNoConnections {get; set;}
```

Property Value

This value is useful only when the library runs in [multipleThreads](#) mode and it have the following meaning

- **true** – background threads automatically die when there are no more [running Connection](#).
- **false** – background threads runs forever, until an explicit Lib.[Close](#)/Lib.[Dispose](#) is made or another constraint (e.g. see [maxInactivityInterval](#)) will occur.

The initial default value for this property is *true*.

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [Implementation Threads](#)

Lib.version Property

Gets the current library version.

```
public string version {get;}
```

Property Value

The returned value identify both

- the ftapi_dnet library version, and
- the FTApi30 library version.

A typical returned string may be the following:

```
version 1.0.1403 (build on 04/11/2003, based on FTApi.V.3.1.1.Develop.1.Date.08/10/2003-FT-NTMT)
```

Remarks

Please note that, for historical reasons, all dates representation returned are in format DD/MM/YYYY.

Example

```
if(marketLibrary != null && marketLibrary != "")
{
    marletLibraryVersion = Lib.lib.loadEntityClasses(null, marketLibrary);
    Lib.lib.trace(TraceLevel.normal, "Using market library " + marketLibrary +
        " at version " + marletLibraryVersion);
}
```

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#)

Lib Methods

The methods of the **Lib** class are listed below. For a complete list of **Lib** class members, see the [Lib Members](#) topic.

Public Instance Methods

✦ Close (inherited from Disposable)	Logically destroy the current object, making it again available.
✦ Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
✦ Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
✦ GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
✦ GetType (inherited from Object)	Gets the Type of the current instance.
✦ loadEntityClasses	Extends the set of known EntityClasses.
✦ open	Initializes the library, starting the EntityClasses loading phase.
✦ Select	Determines the status of one or more application sockets.
✦ setTraceMode	Customize the trace sub-system.
✦ start	Starts the library usage, ending the EntityClasses loading phase.
✦ step	In non- multipleThreads mode operates the communication in two directions: Client ? any connected Server.
✦ ToString (inherited from Disposable)	Returns a single-line String that represents the current object.
✦ trace	Generates applicative trace events.

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) |

Lib.loadEntityClasses Method

Extends the set of known EntityClasses.

```
public string loadEntityClasses(  
    string path,  
    string libraryName  
);
```

Parameters

path

Optional folder-name to add to the standard places where the Operating System search for the shared libraries.

This value, if different from null and empty-string "", allows to place *libraryName* in an unusual folder.

libraryName

Market-library name from which load the [EntityClass](#) set.

The shared library identified by this value must have a name like "ftl<something>" and it must be present (at run-time) in one of the standard places where the Operating System search for the shared libraries (eventually extended with the folder denoted by the *path* parameter).

E.g. in the Windows operating system the library for the FastTrack MetaMarket service is called "ftlmetamarket.dll" and it may be moved/copied in the same place where ftapi_dnet.dll and FTApi30.dll libraries reside (i.e. the current directory, or the "C:\WINNT\system32" directory, etc...).

Return Value

version of the loaded *libraryName*.

Remarks

The initially empty-set of known [EntityClass](#) (i.e. EntityClasses for which the [getEntityClass](#) invocation does *not* return null) may be extended by one or more invocations of this method.

Typically this method is invoked once for each FastTrack market/service used in the rest of the application.

Status

- required on Entry – [opening](#)
- on successfully Exit – opening

Example

```
if(marketLibrary != null && marketLibrary != "")  
{  
    marletLibraryVersion = Lib.lib.loadEntityClasses(null, marketLibrary);  
    Lib.lib.trace(TraceLevel.normal, "Using market library " + marketLibrary +  
        " at version " + marletLibraryVersion);  
}
```

```
}
```

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [.getEntityClass](#)

Lib.open Method

Initializes the library, starting the EntityClasses loading phase.

```
public void open(  
    ThreadingMode threadingMode  
);
```

Parameters

threadingMode

Library threading behaviour.

This value controls the [start](#) behaviour where additional background threads may be created and launched as explained in [Implementation Threads](#).

The typical and suggested value for this parameter is [multipleThreads](#).

Remarks

This phase (i.e. between this and [start](#) methods invocation) is devoted to the market/services EntityClasses loading (see [loadEntityClasses](#)).

Status

- required on Entry – [init](#)
- on successfully Exit – [opening](#)

Example

The following code–fragment shows the typical .NET library initialization

```
Lib.lib.open(ThreadingMode.multipleThreads);
```

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [Implementation Threads](#)

Lib.Select Method

Determines the status of one or more application sockets.

```
public void Select(
    IList iDescriptors,
    IList oDescriptors,
    IList eDescriptors,
    uint microSeconds
);
```

Parameters

iDescriptors

An [IList](#) of [uint](#) application socket descriptors to check for readability.

This value may be null or it may be an empty-list if there are any socket descriptors to check.

oDescriptors

An [IList](#) of [uint](#) application socket descriptors to check for writeability.

This value may be null or it may be an empty-list if there are any socket descriptors to check.

eDescriptors

An [IList](#) of [uint](#) application socket descriptors to check for errors.

This value may be null or it may be an empty-list if there are any socket descriptors to check.

microSeconds

The time to wait for a response, in microseconds.

Remarks

This method is mainly useful when the library runs in single-thread mode i.e. when the [open](#) `threadingMode` parameter was not equal to the suggested [multipleThreads](#) value.

In this case, as explained in [Implementation Threads](#), the application must continuously and quickly run in a loop like the following:

```
ArrayList iDesc = new ArrayList();
ArrayList oDesc = new ArrayList();
ArrayList eDesc = new ArrayList();
...
while(Lib.lib.status == Status.running) {
    Lib.lib.step();

    iDesc.Clear();
    iDesc.Add(<my first uint socket descriptor to check for readability>);
    ...
    iDesc.Add(<my last uint socket descriptor to check for readability>);

    oDesc.Clear();
    oDesc.Add(<my first uint socket descriptor to check for writeability>);
    ...
    oDesc.Add(<my last uint socket descriptor to check for writeability>);

    eDesc.Clear();
```

```
eDesc.Add(<my first uint socket descriptor to check for errors>);  
...  
eDesc.Add(<my last uint socket descriptor to check for errors>);  
  
Lib.lib.Select(iDesc, oDesc, eDesc, 1000*1000);  
  
foreach(uint iSocketDescriptor in iDesc)  
    <manage input over my iSocketDescriptor>;  
  
foreach(uint oSocketDescriptor in oDesc)  
    <manage output over my oSocketDescriptor>;  
  
foreach(uint eSocketDescriptor in eDesc)  
    <manage error over my eSocketDescriptor>;  
}
```

This method mimics [Select](#) but here the *microSeconds* parameter is `uint` instead of `int`.

Status

- required on Entry – [running](#)
- on successfully Exit – running

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [Implementation Threads](#)

Lib.setTraceMode Method

Customize the trace sub-system.

```
public void setTraceMode(
    bool traceOn,
    TraceLevel traceLevel,
    TraceOptions traceOptions,
    bool writeFile,
    int oldLogs,
    int flushLines
);
```

Parameters

traceOn

Controls the [traceEvent](#) raising.

- **true** – events will be raised if they satisfy the *traceLevel* and *traceOptions* constraints.
- **false** – events will *never* be raised.

traceLevel

Minimum [TraceLevel](#) for which the [traceEvent](#) will be generated (if *traceOn* is **true**) and the corresponding trace-line will be stored into the file (if *writeFile* is **true**).

traceOptions

Mask of [TraceOptions](#) values controlling which events must be traced.

writeFile

Controls the storing of trace-lines into a file

- **true** – Trace-lines will be stored if they satisfy the *traceLevel* and *traceOptions* constraints.

Trace-lines are stored in files in the LOGS subdirectory of the current working directory and with names having the following format

```
code_YYYYMMDD.log
```

where code refers to a specific [TraceSource](#) and YYYYMMDD is the generation date.

E.g. the name `dnet_20031105.log` refers to a trace logfile regarding [ftapi_dnet](#) generated November 5, 2003.

- **false** – Trace-lines will *never* be stored.

oldLogs

Indicates the number of old daily logfiles kept in the LOGS subdirectory.

This value is meaning only if *writeFile* is **true**.

flushLines

Indicates the logfile flushing interval, expressed in number of lines.

This value is meaning only if *writeFile* is **true**.

Remarks

Every *unusual event*, occurring/regarding the application, is categorized by a couple ([TraceSource](#), [TraceLevel](#)) and it may be automatically traced by the application and/or stored in a file.

This method allow the customization of these activities.

Example

```
Lib.lib.traceEvent += new Lib.TraceEventHandler(myTraceEventHandler);  
Lib.lib.setTraceMode(true, traceLevel, traceOptions, true, 1, 1);
```

See Also

[Lib Class](#) | [ftapi dnet Namespace](#) | [Lib.TraceEventHandler](#)

Lib.start Method

Starts the library usage, ending the EntityClasses loading phase.

```
public void start();
```

Remarks

The phase between [open](#) and this methods invocation was devoted to the market/services EntityClasses loading (see [loadEntityClasses](#)).

Now it's possible to use the library for all normal activities: create [Connections](#) and then create and handle the various [ConnectionActivity](#) (i.e. [Subscription](#), [Transaction](#), [Query](#) and [Filter](#)).

Please note that this invocation (depending on **open** threadingMode parameter) may automatically starts some background threads as explained in [Implementation Threads](#).

Status

- required on Entry – [opening](#)
- on successfully Exit – [running](#)

Example

```
#if DEBUG
    Lib.lib.maxInactivityInterval = new TimeSpan(0, 0, 30);
#endif
Lib.lib.start();
run(args);
```

See Also

[Lib Class](#) | [ftapi dnet Namespace](#) | [Implementation Threads](#)

Lib.step Method

In non-[multipleThreads](#) mode operates the communication in two directions: Client ? any connected Server.

```
public void step();
```

Remarks

This method is *only* useful when the library runs in single-thread mode i.e. when the [open](#) threadingMode parameter was not equal to the suggested [multipleThreads](#) value.

In this case this method:

- sends operation requests to the servers (such as a request for connection/subscription opening, etc...);
- raise the appropriate events when information arrives from the servers.

This method has to be systematically invoked in the main cycle of the client application. Not calling it for a period of time longer than the alive timeout defined by the FastTrack protocol communication leads to the server connection being closed. A useful prototype example of this request behaviour is depicted in the *Remarks* section of [Select](#).

Status

- **required on Entry** – [running](#)
- **on successfully Exit** – **running**

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [Implementation Threads](#) | [Select](#)

Lib.trace Method

Generates applicative trace events.

```
public void trace(
    TraceLevel traceLevel,
    string message
);
```

Parameters

traceLevel

[TraceLevel](#) associated to *message*.

message

Trace message.

Remarks

These generated trace events may be automatically raised as [traceEvent](#) and/or stored in an appropriate trace logfile as customized by the most recent invocation of [setTraceMode](#).

Events

Event Type	Reason
traceEvent	Event synchronously raised if allowed by the most recent invocation of setTraceMode .

Example

```
public override void onNotify(NotifyEventArgs ne)
{
    // base.onNotify(ne);
#if DEBUG
    Lib.lib.trace(TraceLevel.full,
        "Incoming order " + ne.entity.ToString(true, 0));
#else
    Lib.lib.trace(TraceLevel.normal,
        "Incoming order " + ne.entity.getFieldString("OrderID"));
#endif
}
```




See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [setTraceMode](#)

Lib Events

The events of the **Lib** class are listed below. For a complete list of **Lib** class members, see the [Lib Members](#) topic.

Public Instance Events

 CommunicationExceptionEvent	Event raised when iteration of the main communication loop stops or fails. in multipleThreads mode only.
 CommunicationIdleEvent	Event raised every iteration of the main communication loop. in multipleThreads mode only.
 traceEvent	Event raised when a trace message must be programmatically traced as requested by setTraceMode .

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) |

Lib.CommunicationExceptionEvent Event

Event raised when iteration of the main communication loop stops or fails. in [multipleThreads](#) mode only.

```
public event CommunicationExceptionHandler CommunicationExceptionEvent;
```

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [Lib.CommunicationIdleHandler](#)

Lib.CommunicationExceptionHandler Delegate

Lib callback delegate method called when communication reach an exception. Operative in [multipleThreads](#) mode only.

```
public delegate void Lib.CommunicationExceptionHandler(  
    object lib  
);
```

Parameters

lib
Involved [Lib](#).

Remarks

This method is **only** useful when the library runs in multi-thread mode i.e. when the [open](#) threadingMode parameter was equal to the suggested [multipleThreads](#) value.

This method is invoked when main cycle of the client application stops for exception. The method implementation can only notify and shutdown the application components

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [Implementation Threads](#)

Lib.CommunicationIdleEvent Event

Event raised every iteration of the main communication loop. in [multipleThreads](#) mode only.

public event [CommunicationIdleHandler](#) CommunicationIdleEvent;

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [Lib.CommunicationIdleHandler](#)

Lib.CommunicationIdleHandler Delegate

Lib callback delegate method called when communication is in idle. Operative in [multipleThreads](#) mode only.

```
public delegate void Lib.CommunicationIdleHandler(  
    object lib  
);
```

Parameters

lib
Involved [Lib](#).

Remarks

This method is *only* useful when the library runs in multi-thread mode i.e. when the [open](#) threadingMode parameter was equal to the suggested [multipleThreads](#) value.

This method is systematically invoked in the main cycle of the client application. The method implementation have to no too much 'heavy' in order to return soon the control to the communication thread.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [Implementation Threads](#)

Lib.traceEvent Event

Event raised when a trace message must be programmatically traced as requested by [setTraceMode](#).

```
public event TraceEventHandler traceEvent;
```

Example

```
Lib.lib.traceEvent += new Lib.TraceEventHandler(myTraceEventHandler);  
Lib.lib.setTraceMode(true, traceLevel, traceOptions, true, 1, 1);
```

See Also

[Lib Class](#) | [ftapi_dnet Namespace](#) | [Lib.TraceEventHandler](#)

Lib.TraceEventArgs Class

Data to be traced by the application.

For a list of all members of this type, see [Lib.TraceEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Lib.TraceEventArgs

public class Lib.TraceEventArgs : [FastTrackEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Remarks

These data are generated as result of

- a synchronous [trace](#) method invocation,
- or another synchronous method invocation (e.g. see [ftapi_dnetInOut](#)), or
- or an asynchronous event/answer coming from the server (e.g. see [dataIn](#)).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)




See Also

[Lib.TraceEventArgs Members](#) | [ftapi_dnet Namespace](#)





Lib.TraceEventArgs Members

[Lib.TraceEventArgs overview](#)

Public Instance Fields

 message	The message itself.
 traceLevel	Trace message level.
 traceSource	Trace message category.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.




See Also

[Lib.TraceEventArgs Class](#) | [ftapi_dnet Namespace](#)

Lib.TraceEventArgs Fields

The fields of the **Lib.TraceEventArgs** class are listed below. For a complete list of **Lib.TraceEventArgs** class members, see the [Lib.TraceEventArgs Members](#) topic.

Public Instance Fields

 message	The message itself.
 traceLevel	Trace message level.
 traceSource	Trace message category.

See Also

[Lib.TraceEventArgs Class](#) | [ftapi_dnet Namespace](#)

Lib.TraceEventArgs.message Field

The message itself.

```
public readonly string message;
```

See Also

[Lib.TraceEventArgs Class](#) | [ftapi_dnet Namespace](#)

Lib.TraceEventArgs.traceLevel Field

Trace message level.

```
public readonly TraceLevel traceLevel;
```

See Also

[Lib.TraceEventArgs Class](#) | [ftapi_dnet Namespace](#)

Lib.TraceEventArgs.traceSource Field

Trace message category.

```
public readonly TraceSource traceSource;
```

See Also

[Lib.TraceEventArgs Class](#) | [ftapi_dnet Namespace](#)

Mask Class

A set of [segments](#) of a [EntityClass](#).

For a list of all members of this type, see [Mask Members](#).

System.Object

[ftapi_dnet.Disposable](#)

ftapi_dnet.Mask

public class Mask : [Disposable](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

A mask may be used in [Subscription](#) (subscrMask parameter of Subscription.[open](#)) or [Transaction](#) (entityMask parameter or Transaction.[send](#)) to restrict (at the server level) the set of fields of entities that will be notified.

Empty masks are created by a [Mask constructor](#) and then filled with [addSegment](#).

Given the following EntityClasses (described in an invented meta-language)

```
ENUM VERB(0)                                // an ENUM EntityClass with name="VERB" and id=0
    "Buy"                                    // first ENUM constant with value 0 = "Buy"
    "Sell"                                  // second ENUM constant with value 1 = "Sell"
ENDENUM

STRUCT SEQUENCE(0)                          // a STRUCT EntityClass with name="SEQUENCE" and id=0
    LONG    SequenceNumber                 // a single LONG
    CHAR    SequenceType                  // a single CHAR
ENDSTRUCT

STRUCT PROPOSAL(2468)                      // a STRUCT EntityClass with name="PROPOSAL" and id=2468
    SEQUENCE Sequence                     // a single STRUCT SEQUENCE(0)
    SEQUENCE LeftRight[2]                 // a vector of 2 STRUCT SEQUENCE(0)
    VERB    Verb                          // a single ENUM VERB(0)
    STRING  Operator(4)                   // a single STRING with a max of 4 characters
    LONG    BondCode                      // a single LONG
    INT     PriceQuantity[2]              // a vector of 2 INT
    BOOLEAN Flags[3]                     // a vector of 3 BOOLEAN
ENDSTRUCT
```

these are all the valid available segments for the PROPOSAL (2468) EntityClass

```
Sequence.SequenceNumber    // definition 3.
Sequence.SequenceType      // definition 3.
LeftRight[0].SequenceNumber // definition 4.
LeftRight[0].SequenceType  // definition 4.
LeftRight[1].SequenceNumber // definition 4.
LeftRight[1].SequenceType  // definition 4.
```



```
Verb                // definition 1.  
Operator            // definition 1.  
BondCode            // definition 1.  
PriceQuantity[0]    // definition 2.  
PriceQuantity[1]    // definition 2.
```

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[Mask Members](#) | [ftapi_dnet Namespace](#) | [Subscription](#) | [Transaction](#)




Mask Members

[Mask overview](#)








Public Instance Constructors

 Mask	Overloaded. Creates a new empty Mask for a given EntityClass.
--	---

Public Instance Properties

 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

Public Instance Methods

 addSegment	Add a segment to this mask.
 Close (inherited from Disposable)	Logically destroy the current object, making it again available.
 Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[Mask Class](#) | [ftapi_dnet Namespace](#) | [Subscription](#) | [Transaction](#)

Mask Constructor

Creates a new empty Mask for a given EntityClass.

Overload List

Full constructor to create an empty Mask for a given EntityClass.

[public Mask\(EntityClass\);](#)

Alternative constructor to create an empty Mask for a given EntityClass.

[public Mask\(string\);](#)

Alternative constructor to create an empty Mask for a given EntityClass.

[public Mask\(uint\);](#)

See Also

[Mask Class](#) | [ftapi_dnet Namespace](#)

Mask Constructor (EntityClass)

Full constructor to create an empty Mask for a given EntityClass.

```
public Mask(  
    EntityClass entityClass  
);
```

Parameters

entityClass
associated EntityClass.

Remarks

Status

- on successfully Exit – [running](#)

See Also

[Mask Class](#) | [ftapi_dnet Namespace](#) | [Mask Constructor Overload List](#)

Mask Constructor (UInt32)

Alternative constructor to create an empty Mask for a given EntityClass.

```
public Mask(  
    uint classID  
);
```

Parameters

classID

Identifier of the associated EntityClass.

Remarks

The invocation

```
new Mask(classID)
```

is equivalent to

```
new Mask(EntityClass.getEntityClass(classID, null))
```

Status

- on successfully Exit – [running](#)

See Also

[Mask Class](#) | [ftapi_dnet Namespace](#) | [Mask Constructor Overload List](#)

Mask Constructor (String)

Alternative constructor to create an empty Mask for a given EntityClass.

```
public Mask(  
    string className  
);
```

Parameters

className

Name of the associated EntityClass.

Remarks

The invocation

```
new Mask(className)
```

is equivalent to

```
new Mask(EntityClass.getEntityClass(0, className))
```

Status

- on successfully Exit – [running](#)

See Also

[Mask Class](#) | [ftapi_dnet Namespace](#) | [Mask Constructor Overload List](#)

Mask Methods

The methods of the **Mask** class are listed below. For a complete list of **Mask** class members, see the [Mask Members](#) topic.

Public Instance Methods

◆ addSegment	Add a segment to this mask.
◆ Close (inherited from Disposable)	Logically destroy the current object, making it again available.
◆ Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
◆ Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
◆ GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
◆ GetType (inherited from Object)	Gets the Type of the current instance.
◆ ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[Mask Class](#) | [ftapi_dnet Namespace](#) | [Subscription](#) | [Transaction](#)

Mask.addSegment Method

Add a segment to this mask.

```
public virtual void addSegment(  
    string maskedSegment  
);
```

Parameters

maskedSegment

Description of a [segment](#) to be added to this mask.

Remarks

If this method invocation completed successfully, then

- the request was accepted by the client,

otherwise

- the client rejected the request,
- an appropriate [FastTrackException](#) will be thrown.

In both cases the current [Status](#) remains unchanged.

Status

- required on Entry – [running](#)
- on successfully Exit – **running**

Example

Given the following EntityClasses

```
ENUM VERB(0)                                // an ENUM EntityClass with name="VERB" and id=0  
    "Buy"                                    // first ENUM constant with value 0 = "Buy"  
    "Sell"                                  // second ENUM constant with value 1 = "Sell"  
ENDENUM  
  
STRUCT SEQUENCE(0)                          // a STRUCT EntityClass with name="SEQUENCE" and id=0  
    LONG    SequenceNumber                 // a single LONG  
    CHAR    SequenceType                  // a single CHAR  
ENDSTRUCT  
  
STRUCT PROPOSAL(2468)                      // a STRUCT EntityClass with name="PROPOSAL" and id=2468  
    SEQUENCE Sequence                     // a single STRUCT SEQUENCE(0)  
    SEQUENCE LeftRight[2]                 // a vector of 2 STRUCT SEQUENCE(0)  
    VERB    Verb                          // a single ENUM VERB(0)  
    STRING  Operator(4)                   // a single STRING with a max of 4 characters  
    LONG    BondCode                      // a single LONG  
    INT     PriceQuantity[2]              // a vector of 2 INT  
    BOOLEAN Flags[3]                      // a vector of 3 BOOLEAN  
ENDSTRUCT
```


these are all the valid available segments for the PROPOSAL (2468) EntityClass

```
Sequence.SequenceNumber      // definition 3.
Sequence.SequenceType        // definition 3.
LeftRight[0].SequenceNumber  // definition 4.
LeftRight[0].SequenceType    // definition 4.
LeftRight[1].SequenceNumber  // definition 4.
LeftRight[1].SequenceType    // definition 4.
Verb                          // definition 1.
Operator                      // definition 1.
BondCode                     // definition 1.
PriceQuantity[0]             // definition 2.
PriceQuantity[1]             // definition 2.
```

and these are *a few* of the valid addSegment invocation:

```
myMask.addSegment ( "Verb" );
myMask.addSegment ( "Operator" );
myMask.addSegment ( "LeftRight[0].SequenceNumber" );
myMask.addSegment ( "PriceQuantity[1]" );
```

See Also

[Mask Class](#) | [ftapi_dnet Namespace](#)

Query Class

A client's request to a server to obtain a set of entities (or rows) from its own Data Base.

For a list of all members of this type, see [Query Members](#).

System.Object

[ftapi_dnet.Disposable](#)

[ftapi_dnet.ConnectionActivity](#)

ftapi_dnet.Query

public class Query : [ConnectionActivity](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

An entire result-set (or a part of it, as eventually requested by [queryRows](#)) is returned to the client application, one row at time.

A query is defined by

- a QueryID identifier that identifies the query in the server,
- an optional [Entity](#), i.e. the optional argument of the query.

The precise meaning of the QueryID identifier is specified by the particular server.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[Query Members](#) | [ftapi_dnet Namespace](#)






Query Members

[Query overview](#)













Public Instance Constructors

 Query Constructor	Initializes a new instance of the Query class.
---	--



Public Instance Properties



 connection (inherited from ConnectionActivity)	Gets the associated Connection.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 queryRowsAllowed	Indicates that a queryRows is allowed.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

Public Instance Methods

 Close (inherited from Disposable)	Logically destroy the current object, making it again available.
 Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 onClose	Method automatically invoked when the server answer to Close arrives.
 onNotify	Method automatically invoked when a new data comes from the server as result of a open or queryRows .
 onOpen	Method automatically invoked when the server answer to open arrives.
 onRows	Method automatically invoked when the server answer to queryRows arrives.
 open	Overloaded. Creates this query on the server.
 queryRows	Retrieves a subset of the result-set of this query from the server.
 ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

Public Instance Events

 closeEvent	Event asynchronously raised when the server answer to Close arrives.
 notifyEvent	Event asynchronously raised when a new data comes from the server as result of a open or queryRows .

 openEvent	Event asynchronously raised when the server answer to open arrives.
 rowsEvent	Event asynchronously raised when the server answer to queryRows arrives.

See Also

[Query Class](#) | [ftapi_dnet Namespace](#)

Query Constructor

Initializes a new instance of the Query class.

```
public Query(  
    Connection connection  
);
```

Parameters

connection
the parent Connection.

Remarks

Status

- on successfully Exit – [init](#)






See Also

[Query Class](#) | [ftapi_dnet Namespace](#)

Query Properties

The properties of the **Query** class are listed below. For a complete list of **Query** class members, see the [Query Members](#) topic.

Public Instance Properties

 connection (inherited from ConnectionActivity)	Gets the associated Connection.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 queryRowsAllowed	Indicates that a queryRows is allowed.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

See Also

[Query Class](#) | [ftapi_dnet Namespace](#)

Query.queryRowsAllowed Property

Indicates that a [queryRows](#) is allowed.

```
public virtual bool queryRowsAllowed { get; }
```

Property Value

This property evaluates *true* only when an [openEvent](#) is raised with [resultSetFollows](#) = *false* and [canComputeRowNumber](#) = *true*.

Example

```
public override void onOpen(OpenEventArgs oe)
{
    // base.onOpen(oe);
    if(oe.result != QueryOpen.OK)
        ExampleGeneric.abort("open of " + this + " returned " + oe.result);
    else
        if(queryRowsAllowed)
            queryRows(1, oe.rowNumber);
}
```

See Also

[Query Class](#) | [ftapi_dnet Namespace](#)

Query Methods

The methods of the **Query** class are listed below. For a complete list of **Query** class members, see the [Query Members](#) topic.

Public Instance Methods

✦ Close (inherited from Disposable)	Logically destroy the current object, making it again available.
✦ Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
✦ Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
✦ GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
✦ GetType (inherited from Object)	Gets the Type of the current instance.
✦ onClose	Method automatically invoked when the server answer to Close arrives.
✦ onNotify	Method automatically invoked when a new data comes from the server as result of a open or queryRows .
✦ onOpen	Method automatically invoked when the server answer to open arrives.
✦ onRows	Method automatically invoked when the server answer to queryRows arrives.
✦ open	Overloaded. Creates this query on the server.
✦ queryRows	Retrieves a subset of the result-set of this query from the server.
✦ ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[Query Class](#) | [ftapi_dnet Namespace](#)

Query.onClose Method

Method automatically invoked when the server answer to [Close](#) arrives.

```
public virtual void onClose(  
    CloseEventArgs ce  
);
```

Parameters

ce
Server answer to [Close](#).

Remarks

If *ce.result* is [OK](#) then the server has destroyed the query otherwise some unknow error occurred.

In both cases the current [Status](#) changed to [init](#).

This method must be properly extended in Query sub-classes since it does nothing in the Query class.

See Also

[Query Class](#) | [ftapi_dnet Namespace](#) | [Close](#) | [Query.CloseEventHandler](#)

Query.onNotify Method

Method automatically invoked when a new data comes from the server as result of a [open](#) or [queryRows](#).

```
public virtual void onNotify(  
    NotifyEventArgs ne  
);
```

Parameters

ne

Data coming from the server as result of a [open](#) or [queryRows](#).

Remarks

If the query result-set computed by the server, as an answer to a [open](#) or [queryRows](#), is composed by N entities then [notifyEvent](#) will be raised (N+1) times (and correspondingly this method and all [Query.NotifyEventHandler](#) will be invoked (N+1) times):

- N times with each of the N entities and with the [eoq](#) = *false* indication.
- and one more time with the [eoq](#) = *true* indication.

In any case the current [Status](#) remains [running](#).

This method must be properly extended in Query sub-classes since it does nothing in the Query class.

Example

```
public override void onNotify(NotifyEventArgs ne)  
{  
    // base.onNotify(ne);  
    if(ne.eoq)  
        checkEnd();  
    else  
        switch(ne.entity.entityClass.classId)  
        {  
            case SIB_DESC_CLASS_ID:  
                displaySIB_DESC_CLASS(ne.entity);  
                break;  
            case SIB_DESC_CLASS_FIELD_ID:  
                displaySIB_DESC_CLASS_FIELD(ne.entity);  
                break;  
            case SIB_DESC_CLASS_KEY_ID:  
                displaySIB_DESC_CLASS_KEY(ne.entity);  
                break;  
            case SIB_DESC_CLASS_SEGKEY_ID:  
                displaySIB_DESC_CLASS_SEGKEY(ne.entity);  
                break;  
        }  
}
```

See Also

[Query Class](#) | [ftapi_dnet Namespace](#) | [open](#) | [_queryRows](#) | [_Query.NotifyEventHandler](#)

Query.onOpen Method

Method automatically invoked when the server answer to [open](#) arrives.

```
public virtual void onOpen(  
    OpenEventArgs oe  
);
```

Parameters

oe
Server answer to [open](#).

Remarks

If *oe.result* is [OK](#) then

- the server accepted the query creation,
- the current [Status](#) changed to [running](#),
- if *oe.resultSetFollows* is *true* then the server raise many [notifyEvent](#) to notify the query result,
- if *oe.resultSetFollows* is *false* then the server may now accept [queryRows](#) on this query.

otherwise

- the server rejected the query creation (see **result** to understand why),
- the current **Status** changed to [init](#),
- the server does not raise any **notifyEvent**,
- the server does not accept any **queryRows** on this query.

This method must be properly extended in Query sub-classes since it does nothing in the Query class.

Example

```
public override void onOpen(OpenEventArgs oe)  
{  
    // base.onOpen(oe);  
    if(oe.result != QueryOpen.OK)  
        ExampleGeneric.abort("open of " + this + " returned " + oe.result);  
    else  
        if(queryRowsAllowed)  
            queryRows(1, oe.rowNumber);  
}
```

See Also

[Query Class](#) | [ftapi dnet Namespace](#) | [open](#) | [Query.OpenEventHandler](#)

Query.onRows Method

Method automatically invoked when the server answer to [queryRows](#) arrives.

```
public virtual void onRows(  
    RowsEventArgs re  
);
```

Parameters

re
Server answer to [queryRows](#).

Remarks

If *re.result* is [OK](#) then

- the server accepted the query rows request,
- the server raise many [notifyEvent](#) to notify the query result.

otherwise

- the server rejected the query rows request (see **result** to understand why),
- the server does not raise any **notifyEvent**.

In both cases the current [Status](#) remains [running](#).

This method must be properly extended in Query sub-classes since it does nothing in the Query class.

Example

```
public override void onRows(RowsEventArgs re)  
{  
    // base.onRows(re);  
    if(re.result != QueryRows.OK)  
        ExampleGeneric.abort("queryRows of " + this + " returned " + re.result);  
}
```

See Also

[Query Class](#) | [ftapi dnet Namespace](#) | [queryRows](#) | [Query.RowsEventHandler](#)

Query.open Method

Creates this query on the server.

Overload List

Shorthand method to create this query on the server using some defaults values.

[public virtual void open\(uint\);](#)

Full method to create this query on the server.

[public virtual void open\(uint,Entity\);](#)

See Also

[Query Class](#) | [ftapi_dnet Namespace](#)

Query.open Method (UInt32)

Shorthand method to create this query on the server using some defaults values.

```
public virtual void open(  
    uint queryID  
);
```

Parameters

queryID

The query identifier on the server.

The meaning of this value is defined by the server. The server may subsequently returns a [wrongQueryID](#) failure-code if it does not comply with this value.

Remarks

The invocation

```
open(queryID)
```

is equivalent to

```
open(queryID, null)
```

so *please see* the [full open](#) method for a full detailed description of it.

Status & Events

Please see the **full open** method for a full detailed description of them.

See Also

[Query Class](#) | [ftapi_dnet Namespace](#) | [Query.open Overload List](#)

Query.open Method (UInt32, Entity)

Full method to create this query on the server.

```
public virtual void open(
    uint queryID,
    Entity entity
);
```

Parameters

queryID

The query identifier on the server.

The meaning of this value is defined by the server. The server may subsequently returns a [wrongQueryID](#) failure-code if it does not comply with this value.

entity

The optional query [Entity](#) parameter.

Is the argument of the query and it is given to the server and so its meaning must comply with the server specification. The server may subsequently returns a [badParameters](#) failure-code if it does not comply with this value.

Remarks

If this method invocation completed successfully, then

- the query create request was sent to server,
- the current [Status](#) changes to [opening](#),
- when the server-answer will be available the [openEvent](#) will be raised.

otherwise

- the client rejected the query create request,
- the query create request was **not** sent to server,
- the current **Status** remains unchanged,
- no **openEvent** will be raised,
- an appropriate [FastTrackException](#) will be thrown.

Status

- required on Entry – [init](#)
- on successfully Exit – [opening](#)

Events

Event Type	Reason
openEvent	Event asynchronously raised when the server answer to a successfully open arrives.

Example

```
public QuerySpecific(
    Connection connection,
    int          entityID,
    string       entityName,
    string       entityDesc) : base(connection)
{
    name = entityDesc;
    param = new Entity(SIB_DESC_CLASS_PARAMS_ID);
    param.setFieldLong( "EntityID",    entityID);
    param.setFieldString("EntityName", entityName);
    open(SIB_DESC_CLASS_QUERY_ID, param);
    counter++;
}
```

See Also

[Query Class](#) | [ftapi_dnet Namespace](#) | [Query.open Overload List](#) | [onOpen](#)

Query.queryRows Method

Retrieves a subset of the result-set of this query from the server.

```
public virtual void queryRows(
    uint firstRow,
    uint rowNumber
);
```

Parameters

firstRow

index (1-based) of the first row to be retrieved.

rowNumber

number of rows to be retrieved.

Remarks

This method may be invoked only when [queryRowsAllowed](#) evaluates *true*.

If this method invocation completed successfully, then

- the request was sent to server,
- when the server-answer will be available the [rowsEvent](#) will be raised.

otherwise

- the client rejected the request,
- the request was *not* sent to server,
- no **rowsEvent** will be raised,
- an appropriate [FastTrackException](#) will be thrown.

In both cases the current [Status](#) remains unchanged.

Status

- required on Entry – [running](#)
- on successfully Exit – running

Events

Event Type	Reason
rowsEvent	Event asynchronously raised when the server answer to a successfully queryRows arrives.

Example

```
public override void onOpen(OpenEventArgs oe)
{
    // base.onOpen(oe);
    if(oe.result != QueryOpen.OK)
        ExampleGeneric.abort("open of " + this + " returned " + oe.result);
}
```

```
else
    if (queryRowsAllowed)
        queryRows(1, oe.rowNumber);
}
```





See Also

[Query Class](#) | [ftapi_dnet Namespace](#) | [onRows](#) | [queryRowsAllowed](#)

Query Events

The events of the **Query** class are listed below. For a complete list of **Query** class members, see the [Query Members](#) topic.

Public Instance Events

 closeEvent	Event asynchronously raised when the server answer to Close arrives.
 notifyEvent	Event asynchronously raised when a new data comes from the server as result of a open or queryRows .
 openEvent	Event asynchronously raised when the server answer to open arrives.
 rowsEvent	Event asynchronously raised when the server answer to queryRows arrives.

See Also

[Query Class](#) | [ftapi_dnet Namespace](#)

Query.closeEvent Event

Event asynchronously raised when the server answer to [Close](#) arrives.

```
public event CloseEventHandler closeEvent;
```

See Also

[Query Class](#) | [ftapi dnet Namespace](#) | [onClose](#)

Query.notifyEvent Event

Event asynchronously raised when a new data comes from the server as result of a [open](#) or [queryRows](#).

```
public event NotifyEventHandler notifyEvent;
```

See Also

[Query Class](#) | [ftapi_dnet Namespace](#) | [onNotify](#)

Query.openEvent Event

Event asynchronously raised when the server answer to [open](#) arrives.

```
public event OpenEventHandler openEvent;
```

See Also

[Query Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Query.rowsEvent Event

Event asynchronously raised when the server answer to [queryRows](#) arrives.

```
public event RowsEventHandler rowsEvent;
```

See Also

[Query Class](#) | [ftapi dnet Namespace](#) | [onRows](#)

Query.CloseEventArgs Class

Query server answer to [Close](#).

For a list of all members of this type, see [Query.CloseEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Query.CloseEventArgs

```
public class Query.CloseEventArgs : FastTrackEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[Query.CloseEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onClose](#)





Query.CloseEventArgs Members

[Query.CloseEventArgs overview](#)

Public Instance Fields

 result	Server query–destruction failure–code.
--	--

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.


See Also

[Query.CloseEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onClose](#)

Query.CloseEventArgs Fields

The fields of the **Query.CloseEventArgs** class are listed below. For a complete list of **Query.CloseEventArgs** class members, see the [Query.CloseEventArgs Members](#) topic.

Public Instance Fields

 result	Server query–destruction failure–code.
--	--

See Also

[Query.CloseEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onClose](#)

Query.CloseEventArgs.result Field

Server query–destruction failure–code.

public readonly [QueryClose](#) result;

See Also

[Query.CloseEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.NotifyEventArgs Class

Query data coming from the server as result of a [open](#) or [queryRows](#).

For a list of all members of this type, see [Query.NotifyEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Query.NotifyEventArgs

public class Query.NotifyEventArgs : [FastTrackEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)








See Also

[Query.NotifyEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onNotify](#)





Query.NotifyEventArgs Members

[Query.NotifyEventArgs overview](#)

Public Instance Fields

 classVersion	EntityClass version of last entity modification.
 Delete	Delete flag of the entity .
 entity	Entity of the current row in the result-set.
 eqq	Indication that the result-set is ended.
 info	Query-specific result.
 rowNumber	Index (1-based) of the current row in the result-set.
 timeStamp	TimeStamp of last entity modification.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.








See Also

[Query.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onNotify](#)

Query.NotifyEventArgs Fields

The fields of the **Query.NotifyEventArgs** class are listed below. For a complete list of **Query.NotifyEventArgs** class members, see the [Query.NotifyEventArgs Members](#) topic.

Public Instance Fields

 classVersion	EntityClass version of last entity modification.
 Delete	Delete flag of the entity .
 entity	Entity of the current row in the result-set.
 eoq	Indication that the result-set is ended.
 info	Query-specific result.
 rowNumber	Index (1-based) of the current row in the result-set.
 timeStamp	TimeStamp of last entity modification.

See Also

[Query.NotifyEventArgs Class](#) | [ftapi dnet Namespace](#) | [onNotify](#)

Query.NotifyEventArgs.classVersion Field

EntityClass version of last [entity](#) modification.

```
public readonly uint classVersion;
```

Remarks

This field is meaning only when [eoq](#) is *false*.

This field, together with [timeStamp](#), uniquely identifies the [entity](#) time.

See Also

[Query.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.NotifyEventArgs.entity Field

Entity of the current row in the result-set.

```
public readonly Entity entity;
```

Remarks

This field is meaning only when [eoq](#) is *false*.

See Also

[Query.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.NotifyEventArgs.eoq Field

Indication that the result-set is ended.

```
public readonly bool eoq;
```

Remarks

If the query result-set computed by the server, as an answer to a [open](#) or [queryRows](#), is composed by N entities then [notifyEvent](#) will be raised (N+1) times:

- N times with each of the N entities and with this EOQ indication = *false* indication.
- and one more time with this EOQ indication = *true* indication.

See Also

[Query.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.NotifyEventArgs.info Field

Query-specific result.

```
public readonly byte[] info;
```

Remarks

This field is meaning only when [eoq](#) is *false*.

The meaning of this field depends on the specific query made. Generally (on the most of queries) this field is unused.

See Also

[Query.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.NotifyEventArgs.rowNumber Field

Index (1-based) of the current row in the result-set.

```
public readonly uint rowNumber;
```

Remarks

This field is meaning only when [eoq](#) is *false*.

The index of the first row of a result-set returned by [open](#) is 1.

The index of the first row of a result-set returned by [queryRows](#) is `firstRow`.

See Also

[Query.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.NotifyEventArgs.timeStamp Field

TimeStamp of last [entity](#) modification.

```
public readonly TimeStamp timeStamp;
```

Remarks

This field is meaning only when [eoq](#) is *false*.

This field, together with [classVersion](#), uniquely identifies the [entity](#) time.

See Also

[Query.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.OpenEventArgs Class

Query server answer to [open](#).

For a list of all members of this type, see [Query.OpenEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Query.OpenEventArgs

public class Query.OpenEventArgs : [FastTrackEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)






See Also

[Query.OpenEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onOpen](#)





Query.OpenEventArgs Members

[Query.OpenEventArgs overview](#)

Public Instance Fields

 canComputeRowNumber	Validity indication about the rowNumber field.
 result	Server query–creation failure–code.
 resultSetFollows	Indication that the result–set is immediately available.
 rowNumber	Number of rows in the result–set as computed by the server.
 timeToLive	Interval time (in seconds) during which the server cache the result–set.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.






See Also

[Query.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Query.OpenEventArgs Fields

The fields of the **Query.OpenEventArgs** class are listed below. For a complete list of **Query.OpenEventArgs** class members, see the [Query.OpenEventArgs Members](#) topic.

Public Instance Fields

 canComputeRowNumber	Validity indication about the rowNumber field.
 result	Server query–creation failure–code.
 resultSetFollows	Indication that the result–set is immediately available.
 rowNumber	Number of rows in the result–set as computed by the server.
 timeToLive	Interval time (in seconds) during which the server cache the result–set.

See Also

[Query.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Query.OpenEventArgs.canComputeRowNumber Field

Validity indication about the [rowNumber](#) field.

```
public readonly bool canComputeRowNumber;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

See Also

[Query.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.OpenEventArgs.result Field

Server query–creation failure–code.

```
public readonly QueryOpen result;
```

See Also

[Query.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.OpenEventArgs.resultSetFollows Field

Indication that the result-set is immediately available.

```
public readonly bool resultSetFollows;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

The meaning is

<i>true</i>	notifyEvent will be raised (N+1) times: <ul style="list-style-type: none"> • N times with each of the N entities and with the eoq = <i>false</i> indication. • and one more time with the eoq = <i>true</i> indication.
<i>false</i>	notifyEvent will not be raised and the application must issue a specific queryRows to obtains a subset of the result-set.

In any case the [queryRowsAllowed](#) property may be inspect to see if a **queryRows** method invocation is allowed.

See Also

[Query.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.OpenEventArgs.rowNumber Field

Number of rows in the result-set as computed by the server.

```
public readonly uint rowNumber;
```

Remarks

This field is meaning only when [result](#) is [OK](#) and [canComputeRowNumber](#) is *true*.

See Also

[Query.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.OpenEventArgs.timeToLive Field

Interval time (in seconds) during which the server cache the result-set.

```
public readonly uint timeToLive;
```

Remarks

This field is meaning only when [result](#) is [OK](#) and [resultSetFollows](#) is *false* and this value is greater than zero.

During this interval the client may invoke [queryRows](#) to obtains the various parts of the result-set.

See Also

[Query.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Query.RowsEventArgs Class

Query server answer to [queryRows](#).

For a list of all members of this type, see [Query.RowsEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Query.RowsEventArgs

```
public class Query.RowsEventArgs : FastTrackEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[Query.RowsEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onRows](#)





Query.RowsEventArgs Members

[Query.RowsEventArgs overview](#)

Public Instance Fields

 result	Server query-rows failure-code.
--	---------------------------------

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.


See Also

[Query.RowsEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onRows](#)

Query.RowsEventArgs Fields

The fields of the **Query.RowsEventArgs** class are listed below. For a complete list of **Query.RowsEventArgs** class members, see the [Query.RowsEventArgs Members](#) topic.

Public Instance Fields

 result	Server query-rows failure-code.
--	---------------------------------

See Also

[Query.RowsEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onRows](#)

Query.RowsEventArgs.result Field

Server query—rows failure—code.

```
public readonly QueryRows result;
```

See Also

[Query.RowsEventArgs Class](#) | [ftapi_dnet Namespace](#)

Revision Class

A market/service version.

For a list of all members of this type, see [Revision Members](#).

[System.Object](#)

ftapi_dnet.Revision

public class Revision : IComparable

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Remarks

Each version is identified by a triplet of numbers (H, M, L) usually represented as "H.M.L".

This class implements the [IComparable](#) interface in order to allow the use of [CompareTo](#) method when it will be necessary.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[Revision Members](#) | [ftapi_dnet Namespace](#)




Revision Members

[Revision overview](#)






Public Instance Constructors

 Revision Constructor	Initializes a new Revision instance given the triplet (H, M, L).
--	--

Public Instance Fields

 H	The High version number.
 L	The Low version number.
 M	The Medium version number.

Public Instance Methods

 CompareTo	Compares this Revision with another Revision.
 Equals	Determines whether another Revision is equal to this Revision.
 GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current Revision.

See Also

[Revision Class](#) | [ftapi_dnet Namespace](#)

Revision Constructor

Initializes a new Revision instance given the triplet (H, M, L).

```
public Revision(  
    uint h,  
    uint m,  
    uint l  
);
```

Parameters

h
H value.

m
M value.

l
L value.




See Also

[Revision Class](#) | [ftapi_dnet Namespace](#)

Revision Fields

The fields of the **Revision** class are listed below. For a complete list of **Revision** class members, see the [Revision Members](#) topic.

Public Instance Fields

 H	The High version number.
 L	The Low version number.
 M	The Medium version number.

See Also

[Revision Class](#) | [ftapi_dnet Namespace](#)

Revision.H Field

The High version number.

```
public readonly uint H;
```

See Also

[Revision Class](#) | [ftapi_dnet Namespace](#)

Revision.L Field

The Low version number.

```
public readonly uint L;
```

See Also

[Revision Class](#) | [ftapi_dnet Namespace](#)

Revision.M Field

The Medium version number.

```
public readonly uint M;
```






See Also

[Revision Class](#) | [ftapi_dnet Namespace](#)

Revision Methods

The methods of the **Revision** class are listed below. For a complete list of **Revision** class members, see the [Revision Members](#) topic.

Public Instance Methods

 CompareTo	Compares this Revision with another Revision.
 Equals	Determines whether another Revision is equal to this Revision.
 GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current Revision.

See Also

[Revision Class](#) | [ftapi_dnet Namespace](#)

Revision.CompareTo Method

Compares this Revision with another Revision.

```
public int CompareTo(  
    object anotherRevision  
);
```

Parameters

anotherRevision

The other Revision to compare with this Revision.

Return Value

A 32-bit signed integer that indicates the relative order of the comparands. The return value has these meanings:

Value	Meaning
Less than zero	This revision is less than <i>anotherRevision</i> .
Zero	This revision is equal to <i>anotherRevision</i> .
Greater than zero	This revision is greater than <i>anotherRevision</i> .

Implements

[IComparable.CompareTo](#)

Exceptions

Exception Type	Condition
ArgumentException	<i>anotherRevision</i> is null or it is not a Revision.

See Also

[Revision Class](#) | [ftapi_dnet Namespace](#)

Revision.Equals Method

Determines whether another Revision is equal to this Revision.

```
public override bool Equals(  
    object anotherRevision  
);
```

Parameters

anotherRevision

The other Revision to compare with this Revision.

Return Value

true if the other Revision is equal to this Revision; otherwise, *false*.

See Also

[Revision Class](#) | [ftapi_dnet Namespace](#)

Revision.GetHashCode Method

Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.

```
public override int GetHashCode();
```

Return Value

A hash code for the current Revision.

See Also

[Revision Class](#) | [ftapi dnet Namespace](#)

Revision.ToString Method

Returns a single-line [String](#) that represents the current Revision.

```
public override string ToString();
```

Return Value

A single-line [String](#) that represents the current Revision.

See Also

[Revision Class](#) | [ftapi dnet Namespace](#)

Subscription Class

An arrangement with the server for receiving a continuing set of interesting entities of the same [EntityClass](#).

For a list of all members of this type, see [Subscription Members](#).

[System.Object](#)

[ftapi_dnet.Disposable](#)

[ftapi_dnet.ConnectionActivity](#)

ftapi_dnet.Subscription

public class Subscription : [ConnectionActivity](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

Opening a subscription on a EntityClass of a Server entails both the initial acquisition of the [Entity](#) of that EntityClass from the Server and the subsequent notification of any additions or cancellations executed by the Server on that EntityClass.

Special subscription modalities enable the acquisition of just a subset of the Entities of a EntityClass from the Server: see documentation below.

A subscription is logically defined by/with the [open](#) parameters:

- **entityClass** – The [EntityClass](#) on which the Subscription is made.
- **subscribeFlow** – The [data transmission policy](#) used by the server to eventually adapt the sending server speed with the reception client speed.
- **the (classVersion,classTimeStamp) couple** – This refers the last past notification received by the client (see [Partial Subscriptions](#) below).
- **queryType** – The [query selection criteria](#) used by the server to chose whose Entities must be received.
- **an optional (keyID,keyValue) couple** – It describes the [partial KeyValue](#) to be used in case the queryType is [set](#) indicating a [Partial Subscription](#).
- **an optional filter** – It is used to restrict (at the server level) the set of entities that will be notified.
- **an optional subscrMask** – It is used to restrict (at the server level) the set of fields of entities that will be notified.

Incremental Subscriptions

This section outlines how to manage incremental subscriptions, in which the server is only required to send updated Entities of an EntityClass, rather than sending all its Entities.

A client application can consequently keep a local data base aligned with the market server's, or more generally avoid processing data twice, by minimizing, at the same time, the volume of data to transfer and the needed time.

In particular, the subscription **open** allows you to specify the parameters couple (classVersion,

`classTimeStamp`) i.e., respectively, the last version index of the `EntityClass` and the `timeStamp` of the last Entity received of that class during the previous subscription.

Thus, if the version of the `EntityClass` maintained in the server coincides with the one supplied, only those entities with a time stamp that is later than the one supplied will be sent.

On the other hand if the version of the class is earlier than the server's current one, then `resetClass` field will be *true*. This indicates a general invalidation of any entities from that `EntityClass` that have been archived until that moment. In the latter case it will not be possible to proceed to an incremental acquisition, but all the entities in the `EntityClass` will have to be received.

In order to be able to make this data available, the client application has to maintain for each `EntityClass` the following information:

- **current timeStamp** – It can be maintained by updating local `TimeStamp` data whenever `add/rwt/del` operations are made, on the basis of the `timeStamp` field value of `Subscription.NotifyEventArgs`.
- **current version** – It can be maintained by acquiring the value of the version at the opening of the subscription by `classVersion` field of `Subscription.OpenEventArgs`, and updating it on every version variation of the server class, this basically means at each notification of a `kill` operation, by setting it to the `seconds` subfield of the `timeStamp` field of `Subscription.NotifyEventArgs`.

Remarks

`del` and `kill` operations with `keyID` field of `Subscription.NotifyEventArgs` > 0 , should be intended as notifications of (logical or physical) cancellation of an individual entity at a server level. In this case the Entity available in the `entity` field of `Subscription.NotifyEventArgs` is generally undefined on any fields apart from those associated with `keyID`.

`kill` operation with `keyID` field of `Subscription.NotifyEventArgs` ≤ 0 should be intended as the physical cancellation of every entity in the specified `EntityClass` that has been acquired beforehand. In this case the `entity` field of `Subscription.NotifyEventArgs` is `null`.

Partial Subscriptions

This section outlines how to manage partial subscriptions, in which the server is requested to send only those entities in a `EntityClass` that satisfy certain constraints.

Partial subscriptions can be formulated in two manners:

- Setting the `open queryType` parameter to the `set` value and setting the (`keyID`, `keyValue`) couple parameters to an appropriate `partial KeyValue`.
- Or setting the `open filter` parameter to a non-`null` value representing an appropriate `Filter` in `running` status.

In the first case the server will send only Entities whose full key matches the given partial `KeyValue`.

In the latter case the server will send only Entities that satisfy the given filter.

Both manners cannot coexist.

Segments

Given the following definitions:

- **single field** – an EntityField which is not an array. **vector field** – an EntityField which is an array.
- **primitive field** – an EntityField with EntityFieldType != [NONE](#).
- **enum field** – an EntityField with EntityFieldType = **NONE** and the corresponding EntityClassType = [ENUM](#).
- **struct field** – an EntityField with EntityFieldType = **NONE** and the corresponding EntityClassType = [STRUCT](#).
- **numeric field** – an EntityField with EntityFieldType != **NONE** and != [BYTE](#) and != [CHAR](#) and != [UCHAR](#) and != [BOOLEAN](#) and != [STRING](#).

a segment may be defined as:

With reference to a given EntityClass C a segment may be	and it's denoted by
1. An enum or primitive single field X of C	"X"
2. An existing component of a numeric vector field X of C	"X[number]"
3. A segment (<i>recursive definition!</i>) of a struct single field X of C	"X.segment"
4. A segment (<i>recursive definition!</i>) of an existing component of a struct vector field X of C	"X[number].segment"

Segment Example

Given the following EntityClasses (described in an invented meta-language)

```

ENUM VERB(0)                                // an ENUM EntityClass with name="VERB" and id=0
    "Buy"                                    // first ENUM constant with value 0 = "Buy"
    "Sell"                                  // second ENUM constant with value 1 = "Sell"
ENDENUM

STRUCT SEQUENCE(0)                           // a STRUCT EntityClass with name="SEQUENCE" and id=0
    LONG    SequenceNumber                  // a single LONG
    CHAR    SequenceType                   // a single CHAR
ENDSTRUCT

STRUCT PROPOSAL(2468)                        // a STRUCT EntityClass with name="PROPOSAL" and id=2468
    SEQUENCE Sequence                      // a single STRUCT SEQUENCE(0)
    SEQUENCE LeftRight[2]                  // a vector of 2 STRUCT SEQUENCE(0)
    VERB    Verb                          // a single ENUM VERB(0)
    STRING  Operator(4)                   // a single STRING with a max of 4 characters
    LONG    BondCode                      // a single LONG
    INT     PriceQuantity[2]               // a vector of 2 INT
    BOOLEAN Flags[3]                      // a vector of 3 BOOLEAN
ENDSTRUCT

```

this is the full list of all valid segments

```

Sequence.SequenceNumber    // definition 3.
Sequence.SequenceType      // definition 3.
LeftRight[0].SequenceNumber // definition 4.
LeftRight[0].SequenceType  // definition 4.
LeftRight[1].SequenceNumber // definition 4.

```



```

LeftRight[1].SequenceType    // definition 4.
Verb                         // definition 1.
Operator                     // definition 1.
BondCode                     // definition 1.
PriceQuantity[0]             // definition 2.
PriceQuantity[1]             // definition 2.

```

and this is a partial list of invalid segments

```

Sequence                     // 1. Sequence is not an enum or primitive single field
PriceQuantity[2]             // 2. PriceQuantity[2] is not an existing component
Verb[2]                      // 2. Verb is not a vector
Flags[1]                     // 2. Flags is not numeric
Operator[2]                  // 2. Operator is not a vector
LeftRight.SequenceNumber     // 3. LeftRight is not a segment
LeftRight[0].Number          // 4. Number is not a segment of the LeftRight[0] component
Verba                        // Verba is not an EntityField of PROPOSAL(2468)
PriceQuantity[+1]            // Variuos syntactic errors
PriceQuantity[ 1]            // .....
PriceQuantity [1]            // .....
Price Quantity[1]            // .....
PriceQuantity(1)             // .....

```

Keys

Every **STRUCT EntityClass** may have zero, one or more keys.

A key is an ordered list of N ($N > 0$) [segments](#) of that class.

Each key is identified by a number, called keyID.

Key Example

Given the following EntityClasses (described in an invented meta-language)

```

ENUM VERB(0)                 // an ENUM EntityClass with name="VERB" and id=0
    "Buy"                     // first ENUM constant with value 0 = "Buy"
    "Sell"                    // second ENUM constant with value 1 = "Sell"
ENDENUM

STRUCT SEQUENCE(0)           // a STRUCT EntityClass with name="SEQUENCE" and id=0
    LONG    SequenceNumber    // a single LONG
    CHAR    SequenceType      // a single CHAR
ENDSTRUCT

STRUCT PROPOSAL(2468)        // a STRUCT EntityClass with name="PROPOSAL" and id=2468
    SEQUENCE Sequence         // a single STRUCT SEQUENCE(0)
    SEQUENCE LeftRight[2]     // a vector of 2 STRUCT SEQUENCE(0)
    VERB     Verb              // a single ENUM VERB(0)
    STRING   Operator(4)       // a single STRING with a max of 4 characters
    LONG     BondCode          // a single LONG
    INT      PriceQuantity[2]  // a vector of 2 INT
    BOOLEAN  Flags[3]          // a vector of 3 BOOLEAN
ENDSTRUCT

```

the following is a key definition (with keyID = 2) for the PROPOSAL (2468) EntityClass:

```

KEY( PROPOSAL , 2 )
    Verb

```

```

PriceQuantity[0]
PriceQuantity[1]
Sequence.SequenceNumber
BondCode
ENDKEY

```

KeyValues

A [Key Value](#) is an ordered sequence of K values corresponding to the ordered sequence of N ($0 < K \leq N$) [segments](#) that describe a [key](#) of an EntityClass.

KeyValues may be

- **partial** – if $K < N$
- **full** – if $K = N$

Partial KeyValues are used in [Partial Subscriptions](#).

KeyValue Example

Given the following EntityClasses (described in an invented meta-language)

```

ENUM VERB(0)                                // an ENUM EntityClass with name="VERB" and id=0
    "Buy"                                     // first ENUM constant with value 0 = "Buy"
    "Sell"                                    // second ENUM constant with value 1 = "Sell"
ENDENUM

STRUCT SEQUENCE(0)                          // a STRUCT EntityClass with name="SEQUENCE" and id=0
    LONG    SequenceNumber                  // a single LONG
    CHAR    SequenceType                   // a single CHAR
ENDSTRUCT

STRUCT PROPOSAL(2468)                       // a STRUCT EntityClass with name="PROPOSAL" and id=2468
    SEQUENCE Sequence                      // a single STRUCT SEQUENCE(0)
    SEQUENCE LeftRight[2]                  // a vector of 2 STRUCT SEQUENCE(0)
    VERB    Verb                          // a single ENUM VERB(0)
    STRING  Operator(4)                    // a single STRING with a max of 4 characters
    LONG    BondCode                       // a single LONG
    INT     PriceQuantity[2]               // a vector of 2 INT
    BOOLEAN Flags[3]                       // a vector of 3 BOOLEAN
ENDSTRUCT

```

where the following key (KeyID = 2 on PROPOSAL (2486) EntityClass) is defined

```

KEY( PROPOSAL , 2 )
    Verb
    PriceQuantity[0]
    PriceQuantity[1]
    Sequence.SequenceNumber
    BondCode
ENDKEY

```

these are examples of

- **a full (5/5 segments) KeyValue** –

```

Key 2 of class PROPOSAL(2468)
    Verb = 1 = Sell

```

```
PriceQuantity[0] = 50  
PriceQuantity[1] = 90  
Sequence.SequenceNumber = 1234  
BondCode = 9876
```

- a partial (3/5 segments) Key Value –

```
Key 2 of class PROPOSAL(2468)  
Verb = 1 = Sell  
PriceQuantity[0] = 50  
PriceQuantity[1] = 90
```

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[Subscription Members](#) | [ftapi_dnet Namespace](#)






Subscription Members

[Subscription overview](#)














Public Instance Constructors

 Subscription Constructor	Initializes a new instance of the Subscription class.
--	---






Public Instance Properties

 autoDisposeOnIdle	Gets or set an indication about the automatic invocation of Dispose after the Subscription.IdleEventHandler invocation.
 connection (inherited from ConnectionActivity)	Gets the associated Connection.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

Public Instance Methods

 Close (inherited from Disposable)	Logically destroy the current object, making it again available.
 Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 onBroken	Method automatically invoked when the connection with the server crashed or when the server choose to terminate the connection.
 onClose	Method automatically invoked when the server answer to Close arrives.
 onIdle	Method automatically invoked when the flow of historical data is finished and the flow of actual data is starting.
 onNotify	Method automatically invoked when a new data comes from the server as result of a open or refreshEntity .
 onOpen	Method automatically invoked when the server answer to open arrives.
 open	Overloaded. Starts this subscription with the server.
 refreshEntity	Requests the server to re-publish a single full (not masked) Entity.
 ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

Public Instance Events

 brokenEvent	Event asynchronously raised when the connection with the server crashed or when the server choose to terminate the connection.
 closeEvent	Event asynchronously raised when the server answer to Close arrives.
 idleEvent	Event asynchronously raised when the flow of historical data is finished and the flow of actual data is starting.
 notifyEvent	Event asynchronously raised when a new data comes from the server as result of a open or refreshEntity .
 openEvent	Event asynchronously raised when the server answer to open arrives.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#)

Subscription Constructor

Initializes a new instance of the Subscription class.

```
public Subscription(  
    Connection connection  
);
```

Parameters

connection
the parent Connection.

Remarks

Status

- on successfully Exit – [init](#)






See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#)

Subscription Properties

The properties of the **Subscription** class are listed below. For a complete list of **Subscription** class members, see the [Subscription Members](#) topic.

Public Instance Properties

 autoDisposeOnIdle	Gets or set an indication about the automatic invocation of Dispose after the Subscription.IdleEventHandler invocation.
 connection (inherited from ConnectionActivity)	Gets the associated Connection.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#)

Subscription.autoDisposeOnIdle Property

Gets or set an indication about the automatic invocation of [Dispose](#) after the [Subscription.IdleEventHandler](#) invocation.

```
public virtual bool autoDisposeOnIdle {get; set;}
```

Property Value

As explained in [onIdle](#), if this property evaluates *true* after all [Subscription.IdleEventHandler](#) invocations then the subscription will be automatically [Disposed](#).

The initial default value of this property is *false* but it will be automatically set as *true* during an [open](#) with [QueryType](#) equals to [past](#).

Example

```
public SubscriptionMrk(Connection connection) : base(connection)
{
    autoDisposeOnIdle = true; // even if we do not open with QueryType.past
}
```

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#)

Subscription Methods

The methods of the **Subscription** class are listed below. For a complete list of **Subscription** class members, see the [Subscription Members](#) topic.

Public Instance Methods

✦ Close (inherited from Disposable)	Logically destroy the current object, making it again available.
✦ Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
✦ Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
✦ GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
✦ GetType (inherited from Object)	Gets the Type of the current instance.
✦ onBroken	Method automatically invoked when the connection with the server crashed or when the server choose to terminate the connection.
✦ onClose	Method automatically invoked when the server answer to Close arrives.
✦ onIdle	Method automatically invoked when the flow of historical data is finished and the flow of actual data is starting.
✦ onNotify	Method automatically invoked when a new data comes from the server as result of a open or refreshEntity .
✦ onOpen	Method automatically invoked when the server answer to open arrives.
✦ open	Overloaded. Starts this subscription with the server.
✦ refreshEntity	Requests the server to re-publish a single full (not masked) Entity.
✦ ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#)

Subscription.onBroken Method

Method automatically invoked when the connection with the server crashed or when the server choose to terminate the connection.

```
public virtual void onBroken();
```

Remarks

The current [Status](#) changed to [init](#).

This method must be properly extended in Subscription sub-classes since it does nothing in the Subscription class.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [Subscription.BrokenEventHandler](#)

Subscription.onClose Method

Method automatically invoked when the server answer to [Close](#) arrives.

```
public virtual void onClose(  
    CloseEventArgs ce  
);
```

Parameters

ce
Server answer to [Close](#).

Remarks

If *ce.result* is [OK](#) then the server has destroyed the subscription otherwise some unknow error occurred.

In both cases the current [Status](#) changed to [init](#).

This method must be properly extended in Subscription sub-classes since it does nothing in the Subscription class.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [Close](#) | [Subscription.CloseEventHandler](#)

Subscription.onIdle Method

Method automatically invoked when the flow of historical data is finished and the flow of actual data is starting.

```
public virtual void onIdle();
```

Remarks

The current [Status](#) remains [running](#).

The [idleEvent](#) will be raised (and correspondingly this method and all [Subscription.IdleEventHandler](#) will be invoked) only if subscription [QueryType](#) is not [onTime](#).

After this (and all related **Subscription.IdleEventHandler**) method invocation the subscription will be automatically disposed

- if subscription **QueryType** is [past](#),
- or if subscription [autoDisposeOnIdle](#) property is *true*.

This method must be properly extended in Subscription sub-classes since it does nothing in the Subscription class.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [Subscription.IdleEventHandler](#)

Subscription.onNotify Method

Method automatically invoked when a new data comes from the server as result of a [open](#) or [refreshEntity](#).

```
public virtual void onNotify(  
    NotifyEventArgs ne  
);
```

Parameters

ne

Data coming from the server as result of a [open](#) or [refreshEntity](#).

Remarks

The current [Status](#) remains [running](#).

If this is the server-answer to a [refreshEntity](#) then the data available through *ne*.[entity](#) is always complete even if the subscription was opened with a non-null [Mask](#).

This method must be properly extended in Subscription sub-classes since it does nothing in the Subscription class.

Example

```
public override void onNotify(NotifyEventArgs ne)  
{  
    // base.onNotify(ne);  
#if DEBUG  
    Lib.lib.trace(TraceLevel.full,  
        "Incoming market " + ne.entity.ToString(true, 0));  
#else  
    Lib.lib.trace(TraceLevel.normal,  
        "Incoming market " + ne.entity.getFieldString("MarketID"));  
#endif  
}
```

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [open](#) | [refreshEntity](#) | [Subscription.NotifyEventHandler](#)

Subscription.onOpen Method

Method automatically invoked when the server answer to [open](#) arrives.

```
public virtual void onOpen(  
    OpenEventArgs oe  
);
```

Parameters

oe
Server answer to [open](#).

Remarks

If *oe.result* is [OK](#) then

- the server accepted the subscription creation,
- the current [Status](#) changed to [running](#),
- the server start raising many [notifyEvent](#) and/or [idleEvent](#) to notify the subscription result as described in [QueryType](#).

otherwise

- the server rejected the subscription creation,
- the current **Status** changed to [init](#),
- the server does not raise any **notifyEvent** nor **idleEvent**,
- the server does not accept any [refreshEntity](#) on this subscription.

This method must be properly extended in Subscription sub-classes since it does nothing in the Subscription class.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [open](#) | [Subscription.OpenEventHandler](#)

Subscription.open Method

Starts this subscription with the server.

Overload List

Shorthand method to start this subscription with the server using some defaults values.

[public virtual void open\(EntityClass\);](#)

Full method to start this subscription with the server.

[public virtual void open\(EntityClass,SubscribeFlow,uint,TimeStamp,QueryType,uint,KeyValue,Filter,Mask\);](#)

Shorthand method to start this subscription with the server using some defaults values.

[public virtual void open\(string\);](#)

Shorthand method to start this subscription with the server using some defaults values.

[public virtual void open\(string,SubscribeFlow,uint,TimeStamp,QueryType,uint,KeyValue,Filter,Mask\);](#)

Shorthand method to start this subscription with the server using some defaults values.

[public virtual void open\(uint\);](#)

Shorthand method to start this subscription with the server using some defaults values.

[public virtual void open\(uint,SubscribeFlow,uint,TimeStamp,QueryType,uint,KeyValue,Filter,Mask\);](#)

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#)

Subscription.open Method (EntityClass)

Shorthand method to start this subscription with the server using some defaults values.

```
public virtual void open(  
    EntityClass entityClass  
);
```

Parameters

entityClass
Requested EntityClass.

Remarks

The invocation

```
open(entityClass)
```

is equivalent to

```
open(entityClass, SubscribeFlow.all, 0, null, QueryType.all, 0, null, null, null)
```

so *please see* the [full open](#) method for a full detailed description of it.

Status & Events

Please see the **full open** method for a full detailed description of them.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [Subscription.open Overload List](#)

Subscription.open Method (EntityClass, SubscribeFlow, UInt32, TimeStamp, QueryType, UInt32, KeyValue, Filter, Mask)

Full method to start this subscription with the server.

```
public virtual void open(
    EntityClass entityClass,
    SubscribeFlow subscribeFlow,
    uint classVersion,
    TimeStamp classTimeStamp,
    QueryType queryType,
    uint keyID,
    KeyValue keyValue,
    Filter filter,
    Mask subscrMask
);
```

Parameters

entityClass

Requested EntityClass.

subscribeFlow

Requested [data transmission policy](#).

classVersion

Zero or the EntityClass version of the last past notification received by the client.

See [Incremental Subscriptions](#) to understand how to use this value.

classTimeStamp

Zero or the EntityClass TimeStamp of the last past notification received by the client.

See [Incremental Subscriptions](#) to understand how to use this value.

queryType

Requested [query selection criteria](#).

keyID

Requested key identifier.

See [Partial Subscriptions](#) to understand how to use this value.

keyValue

Null or the requested partial [KeyValue](#) associated to *keyID*.

filter

Optional [Filter](#).

It is used to restrict (at the server level) the set of entities that will be notified. See [Partial Subscriptions](#) to understand how to use this value.

subscrMask

Optional [Mask](#).

It is used to restrict (at the server level) the set of fields of entities that will be notified. If this value is not-null then entities returned in [entity](#) will contain only the fields specified by this mask.

Remarks

If this method invocation completed successfully, then

- the start request was sent to server,
- the current [Status](#) changes to [opening](#),
- when the server–answer will be available the [openEvent](#) will be raised.

otherwise

- the client rejected the send,
- the open request was **not** sent to server,
- the current **Status** remains unchanged,
- no **openEvent** will be raised,
- an appropriate [FastTrackException](#) will be thrown.

Status

- **required on Entry** – [init](#)
- **on successfully Exit** – [opening](#)

Events

Event Type	Reason
openEvent	Event asynchronously raised when the server answer to a successfully open arrives. Following this event the following events may be asynchronously raised:
notifyEvent	Event asynchronously raised when a new data comes from the server.
idleEvent	Event asynchronously raised when the flow of historical data is finished and the flow of actual data is starting.
brokenEvent	Event asynchronously raised when the connection with the server crashed or when the server choose to terminate the connection.

Example

The following code–fragment shows a [shorthand open](#) invocation.

```
public override void onOpen(OpenEventArgs oe)
{
    base.onOpen(oe);
    if(oe.result == Error.OK)
    {
        Subscription oSub = new SubscriptionOrd(this);
        Subscription mSub = new SubscriptionMrk(this);
        oSub.open( "FT_C_ORDER" );
        mSub.open( "FT_C_MARKET" );
    }
}
```

See Also

[Subscription Class](#) | [_ftapi_dnet Namespace](#) | [_Subscription.open Overload List](#) | [_onOpen](#)

Subscription.open Method (String)

Shorthand method to start this subscription with the server using some defaults values.

```
public virtual void open(  
    string className  
);
```

Parameters

className

Requested EntityClass name.

Remarks

The invocation

```
open(className)
```

is equivalent to

```
open(EntityClass.getEntityClass(0, className), SubscribeFlow.all, 0, null, QueryType.all, 0, null, null)
```

so *please see* the [full open](#) method for a full detailed description of it.

Status & Events

Please see the **full open** method for a full detailed description of them.

Example

```
public override void onOpen(OpenEventArgs oe)  
{  
    base.onOpen(oe);  
    if(oe.result == Error.OK)  
    {  
        Subscription oSub = new SubscriptionOrd(this);  
        Subscription mSub = new SubscriptionMrk(this);  
        oSub.open("FT_C_ORDER");  
        mSub.open("FT_C_MARKET");  
    }  
}
```

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [Subscription.open Overload List](#)

Subscription.open Method (String, SubscribeFlow, UInt32, TimeStamp, QueryType, UInt32, KeyValue, Filter, Mask)

Shorthand method to start this subscription with the server using some defaults values.

```
public virtual void open(
    string className,
    SubscribeFlow subscribeFlow,
    uint classVersion,
    TimeStamp classTimeStamp,
    QueryType queryType,
    uint keyID,
    KeyValue keyValue,
    Filter filter,
    Mask subscrMask
);
```

Parameters

className

Requested EntityClass name.

subscribeFlow

Requested [data transimission policy](#).

classVersion

Zero or the EntityClass version of the last past notification received by the client.

See [Incremental Subscriptions](#) to understand how to use this value.

classTimeStamp

Zero or the EntityClass TimeStamp of the last past notification received by the client.

See [Incremental Subscriptions](#) to understand how to use this value.

queryType

Requested [query selection criteria](#).

keyID

Requested key identifier.

See [Partial Subscriptions](#) to understand how to use this value.

keyValue

Null or the requested partial [KeyValue](#) associated to *keyID*.

filter

Optional [Filter](#).

It is used to restrict (at the server level) the set of entities that will be notified. See [Partial Subscriptions](#) to understand how to use this value.

subscrMask

Optional [Mask](#).

It is used to restrict (at the server level) the set of fields of entities that will be notified. If this value is not-null then entities returned in [entity](#) will contain only the fields specified by this mask.

Remarks

The invocation

```
open(className, subscribeFlow, classVersion,  
      classTimeStamp, queryType, keyID, keyValue, filter, subscrMask)
```

is equivalent to

```
open(EntityClass.getEntityClass(0, className), subscribeFlow, classVersion,  
      classTimeStamp, queryType, keyID, keyValue, filter, subscrMask)
```

so *please see* the [full open](#) method for a full detailed description of it.

Status & Events

Please see the **full open** method for a full detailed description of them.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [Subscription.open Overload List](#)

Subscription.open Method (UInt32)

Shorthand method to start this subscription with the server using some defaults values.

```
public virtual void open(  
    uint classID  
);
```

Parameters

classID
Requested EntityClassID.

Remarks

The invocation

```
open(classID)
```

is equivalent to

```
open(EntityClass.getEntityClass(classID, null), SubscribeFlow.all, 0, null, QueryType.all, 0, null, null)
```

so *please see* the [full open](#) method for a full detailed description of it.

Status & Events

Please see the **full open** method for a full detailed description of them.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [Subscription.open Overload List](#)

Subscription.open Method (UInt32, SubscribeFlow, UInt32, TimeStamp, QueryType, UInt32, KeyValue, Filter, Mask)

Shorthand method to start this subscription with the server using some defaults values.

```
public virtual void open(  
    uint classID,  
    SubscribeFlow subscribeFlow,  
    uint classVersion,  
    TimeStamp classTimeStamp,  
    QueryType queryType,  
    uint keyID,  
    KeyValue keyValue,  
    Filter filter,  
    Mask subscrMask  
);
```

Parameters

classID

Requested EntityClassID.

subscribeFlow

Requested [data transimission policy](#).

classVersion

Zero or the EntityClass version of the last past notification received by the client.

See [Incremental Subscriptions](#) to understand how to use this value.

classTimeStamp

Zero or the EntityClass TimeStamp of the last past notification received by the client.

See [Incremental Subscriptions](#) to understand how to use this value.

queryType

Requested [query selection criteria](#).

keyID

Requested key identifier.

See [Partial Subscriptions](#) to understand how to use this value.

keyValue

Null or the requested partial [KeyValue](#) associated to *keyID*.

filter

Optional [Filter](#).

It is used to restrict (at the server level) the set of entities that will be notified. See [Partial Subscriptions](#) to understand how to use this value.

subscrMask

Optional [Mask](#).

It is used to restrict (at the server level) the set of fields of entities that will be notified. If this value is not-null then entities returned in [entity](#) will contain only the fields specified by this mask.

Remarks

The invocation

```
open(classID, subscribeFlow, classVersion,  
      classTimeStamp, queryType, keyID, keyValue, filter, subscrMask)
```

is equivalent to

```
open(EntityClass.getEntityClass(classID, null), subscribeFlow, classVersion,  
      classTimeStamp, queryType, keyID, keyValue, filter, subscrMask)
```

so *please see* the [full open](#) method for a full detailed description of it.

Status & Events

Please see the **full open** method for a full detailed description of them.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [Subscription.open Overload List](#)

Subscription.refreshEntity Method

Requests the server to re-publish a single full (not masked) Entity.

```
public virtual void refreshEntity(  
    Entity entity,  
    uint keyID  
);
```

Parameters

entity

the requested [Entity](#).

keyID

the primary key identifier of the requested *entity*. (typically the same keyID parameter used in the [open](#) method).

All the keyID [segments](#) of the Entity must be properly filled.

Remarks

If this method invocation completed successfully, then

- the re-publish request was sent to server,
- when the server-answer will be available the [notifyEvent](#) will be raised.

Please note that a server-answer will be returned if and only if the requested *entity* exists in the server, otherwise there will not be any failure indication of any sort!

otherwise

- the client has rejected the re-publish request,
- the re-publish request was **not** sent to the server,
- no **notifyEvent** will be raised,
- an appropriate [FastTrackException](#) will be thrown.

In both cases the current [Status](#) remains unchanged.

Status

- required on Entry – [running](#)
- on successfully Exit – running

Events

Event Type	Reason
notifyEvent	Event asynchronously raised when the server answer to a successfully refreshEntity arrives.

All fields of [entity](#) will be available, even if this subscription is masked: the [isMasked](#) field may be used to discriminate between masked and not-masked entities.

Please note that this event will be raised if and only if the requested *entity* exists in the server, otherwise there will not be any failure indication of any sort!






See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [onNotify](#)

Subscription Events

The events of the **Subscription** class are listed below. For a complete list of **Subscription** class members, see the [Subscription Members](#) topic.

Public Instance Events

 brokenEvent	Event asynchronously raised when the connection with the server crashed or when the server choose to terminate the connection.
 closeEvent	Event asynchronously raised when the server answer to Close arrives.
 idleEvent	Event asynchronously raised when the flow of historical data is finished and the flow of actual data is starting.
 notifyEvent	Event asynchronously raised when a new data comes from the server as result of a open or refreshEntity .
 openEvent	Event asynchronously raised when the server answer to open arrives.

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#)

Subscription.brokenEvent Event

Event asynchronously raised when the connection with the server crashed or when the server choose to terminate the connection.

```
public event BrokenEventHandler brokenEvent;
```

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [onBroken](#)

Subscription.closeEvent Event

Event asynchronously raised when the server answer to [Close](#) arrives.

```
public event CloseEventHandler closeEvent;
```

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [onClose](#)

Subscription.idleEvent Event

Event asynchronously raised when the flow of historical data is finished and the flow of actual data is starting.

public event [IdleEventHandler](#) idleEvent;

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [onIdle](#)

Subscription.notifyEvent Event

Event asynchronously raised when a new data comes from the server as result of a [open](#) or [refreshEntity](#).

```
public event NotifyEventHandler notifyEvent;
```

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [onNotify](#)

Subscription.openEvent Event

Event asynchronously raised when the server answer to [open](#) arrives.

```
public event OpenEventHandler openEvent;
```

See Also

[Subscription Class](#) | [ftapi_dnet Namespace](#) | [_onOpen](#)

Subscription.CloseEventArgs Class

Subscription server answer to [Close](#).

For a list of all members of this type, see [Subscription.CloseEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Subscription.CloseEventArgs

```
public class Subscription.CloseEventArgs : FastTrackEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[Subscription.CloseEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onClose](#)





Subscription.CloseEventArgs Members

[Subscription.CloseEventArgs overview](#)

Public Instance Fields

 result	Server answer failure—code.
--	-----------------------------

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.


See Also

[Subscription.CloseEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onClose](#)

Subscription.CloseEventArgs Fields

The fields of the **Subscription.CloseEventArgs** class are listed below. For a complete list of **Subscription.CloseEventArgs** class members, see the [Subscription.CloseEventArgs Members](#) topic.

Public Instance Fields

 result	Server answer failure-code.
--	-----------------------------

See Also

[Subscription.CloseEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onClose](#)

Subscription.CloseEventArgs.result Field

Server answer failure-code.

public readonly [Error](#) result;

See Also

[Subscription.CloseEventArgs Class](#) | [ftapi_dnet Namespace](#)

Subscription.NotifyEventArgs Class

Subscription data coming from the server as result of a [open](#) or [refreshEntity](#)

For a list of all members of this type, see [Subscription.NotifyEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

[ftapi_dnet.Subscription.NotifyEventArgs](#)

public class Subscription.NotifyEventArgs : [FastTrackEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)






See Also

[Subscription.NotifyEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onNotify](#)





Subscription.NotifyEventArgs Members

[Subscription.NotifyEventArgs overview](#)

Public Instance Fields

 entity	Entity received from the server.
 entityAction	Server action associated with the received entity.
 isMasked	Indication that some EntityFields may be meaningless.
 keyID	Index of the key on the basis of which the server has carried out the entityAction
 timeStamp	Entity timestamp.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.






See Also

[Subscription.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onNotify](#)

Subscription.NotifyEventArgs Fields

The fields of the **Subscription.NotifyEventArgs** class are listed below. For a complete list of **Subscription.NotifyEventArgs** class members, see the [Subscription.NotifyEventArgs Members](#) topic.

Public Instance Fields

 entity	Entity received from the server.
 entityAction	Server action associated with the received entity.
 isMasked	Indication that some EntityFields may be meaningless.
 keyID	Index of the key on the basis of which the server has carried out the entityAction
 timeStamp	Entity timestamp.

See Also

[Subscription.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onNotify](#)

Subscription.NotifyEventArgs.entity Field

Entity received from the server.

public readonly [Entity](#) entity;

Remarks

Depending on [keyID](#) or entity values

- `keyID <= 0 || entity == null` –

All entities was physically removed from the server.

In this case [entityAction](#) value may only be [kil](#).

- `keyID > 0 || entity != null` –

entityAction value	meaning
del or kil	Only EntityFields associated to keyID are meaningful.
add or rwt	<p>Depending on isMasked value:</p> <ul style="list-style-type: none"> • <i>true</i> – <p>This NotifyEventArgs instance refers a masked subscription <i>and</i> it is not an answer to a refreshEntity.</p> <p>Only EntityFields associated to the mask are meaningful.</p> <ul style="list-style-type: none"> • <i>false</i> – <p>This NotifyEventArgs instance refers a not masked subscription <i>or</i> it is an answer to a refreshEntity.</p> <p>All EntityFields are meaningful.</p>

Meaningless EntityFields are evaluated as

- **zero** – by [getFieldLong](#)
- **empty-string** – by [getFieldString](#)

See Also

[Subscription.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#)

Subscription.NotifyEventArgs.entityAction Field

Server action associated with the received entity.

```
public readonly EntityAction entityAction;
```

Remarks

If this NotifyEventArgs instance is the server answer to a [refreshEntity](#) then this field evaluates [add](#).

When this field is [del](#) or [kil](#) only [entity](#) EntityFields associated to [keyID](#) are meaningful.

Meaningless EntityFields are evaluated as

- **zero** – by [getFieldLong](#)
- **empty-string** – by [getFieldString](#)

For a full description see **entity**.

See Also

[Subscription.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#) | [entity](#)

Subscription.NotifyEventArgs.isMasked Field

Indication that some EntityFields may be meaningless.

```
public readonly bool isMasked;
```

Remarks

When [entityAction](#) is [add](#) or [rwt](#) then this indication may be:

<i>true</i>	This NotifyEventArgs instance refers a masked subscription <i>and</i> it is not an answer to a refreshEntity . Only EntityFields associated to the mask are meaningful.
<i>false</i>	This NotifyEventArgs instance refers a not masked subscription <i>or</i> it is an answer to a refreshEntity . All entity EntityFields are meaningful.

Please note that the behaviour of this field does not depends on the values [lastZeroMasked/allZeroMasked](#) given to [open](#).

Meaningless EntityFields are evaluated as

- **zero** – by [getFieldLong](#)
- **empty-string** – by [getFieldString](#)

For a full description see **entity**.

See Also

[Subscription.NotifyEventArgs Class](#) | [ftapi dnet Namespace](#) | [entity](#)

Subscription.NotifyEventArgs.keyID Field

Index of the key on the basis of which the server has carried out the [entityAction](#)

```
public readonly int keyID;
```

Remarks

When [entityAction](#) is [del](#) or [kil](#) only [entity](#) EntityFields associated to this keyID are meaningful.

Meaningless EntityFields are evaluated as

- **zero** – by [getFieldLong](#)
- **empty-string** – by [getFieldString](#)

For a full description see **entity**.

See Also

[Subscription.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#) | [entity](#)

Subscription.NotifyEventArgs.timeStamp Field

Entity timestamp.

```
public readonly TimeStamp timeStamp;
```

See Also

[Subscription.NotifyEventArgs Class](#) | [ftapi_dnet Namespace](#)

Subscription.OpenEventArgs Class

Subscription server answer to [open](#).

For a list of all members of this type, see [Subscription.OpenEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Subscription.OpenEventArgs

```
public class Subscription.OpenEventArgs : FastTrackEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)




See Also

[Subscription.OpenEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onOpen](#)





Subscription.OpenEventArgs Members

[Subscription.OpenEventArgs overview](#)

Public Instance Fields

 classVersion	Server EntityClass version.
 resetClass	A reset–class indication.
 result	Server answer failure–code.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.




See Also

[Subscription.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Subscription.OpenEventArgs Fields

The fields of the **Subscription.OpenEventArgs** class are listed below. For a complete list of **Subscription.OpenEventArgs** class members, see the [Subscription.OpenEventArgs Members](#) topic.

Public Instance Fields

 classVersion	Server EntityClass version.
 resetClass	A reset–class indication.
 result	Server answer failure–code.

See Also

[Subscription.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onOpen](#)

Subscription.OpenEventArgs.classVersion Field

Server EntityClass version.

```
public readonly uint classVersion;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

This value may be different from the `classVersion` parameter requested during the [open](#) invocation. In this case the [resetClass](#) field evaluates *true*.

See Also

[Subscription.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Subscription.OpenEventArgs.resetClass Field

A reset-class indication.

```
public readonly bool resetClass;
```

Remarks

This field is meaning only when [result](#) is [OK](#).

A *true* value indicates that the required EntityClass version (see `classVersion` parameter of [open](#)) is different from the server [classVersion](#). In this case the historical data coming with [notifyEvent](#) are full and not (as usual) restricted to data following a given timestamp.

See Also

[Subscription.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

Subscription.OpenEventArgs.result Field

Server answer failure—code.

public readonly [Error](#) result;

See Also

[Subscription.OpenEventArgs Class](#) | [ftapi_dnet Namespace](#)

TimeStamp Class

A temporal indicator.

For a list of all members of this type, see [TimeStamp Members](#).

[System.Object](#)

ftapi_dnet.TimeStamp

[Serializable]

public sealed class TimeStamp : IComparable

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

Each timestamp is represented by a couple of integers:

- [seconds](#) – the number of seconds since the standard base time known as "the epoch", namely January 1, 1970, 00:00:00 GMT.
- [counter](#) – an unique counter used to make the time stamps univocal within the same second.

This class is marked with [SerializableAttribute](#) in order to be able to save and then re-create TimeStamp objects.

As an alternative an application may save the two fields couple ([seconds](#), [counter](#)) and then re-create the same TimeStamp object using the given constructor.

This class implements the [IComparable](#) interface in order to allow the use of [CompareTo](#) method when it will be necessary.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[TimeStamp Members](#) | [ftapi_dnet Namespace](#)



TimeStamp Members

[TimeStamp overview](#)






Public Instance Constructors

 TimeStamp Constructor	Initializes a new TimeStamp instance given the couple (seconds, counter).
---	---

Public Instance Fields

 counter	An associate unique counter.
 seconds	Number of seconds since January 1, 1970, 00:00:00 GMT.

Public Instance Methods

 CompareTo	Compares this TimeStamp with another TimeStamp.
 Equals	Determines whether another TimeStamp is equal to this TimeStamp.
 GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current TimeStamp.

See Also

[TimeStamp Class](#) | [ftapi_dnet Namespace](#)

TimeStamp Constructor

Initializes a new TimeStamp instance given the couple (seconds, counter).

```
public TimeStamp(  
    uint s,  
    uint c  
);
```

Parameters

s
seconds value.

c
counter value.

Remarks

This convenience constructor may be used to re-create a timestamp previously saved as a two-fields couple ([seconds](#), [counter](#)).



See Also

[TimeStamp Class](#) | [ftapi_dnet Namespace](#)

TimeStamp Fields

The fields of the **TimeStamp** class are listed below. For a complete list of **TimeStamp** class members, see the [TimeStamp Members](#) topic.

Public Instance Fields

 counter	An associate unique counter.
 seconds	Number of seconds since January 1, 1970, 00:00:00 GMT.

See Also

[TimeStamp Class](#) | [ftapi_dnet Namespace](#)

TimeStamp.counter Field

An associate unique counter.

```
public readonly uint counter;
```

Remarks

This unique counter is used to make the time stamp univocal when it was generated within the same [seconds](#).

See Also

[TimeStamp Class](#) | [ftapi_dnet Namespace](#)

TimeStamp.seconds Field

Number of seconds since January 1, 1970, 00:00:00 GMT.

```
public readonly uint seconds;
```






See Also

[TimeStamp Class](#) | [ftapi_dnet Namespace](#)

TimeStamp Methods

The methods of the **TimeStamp** class are listed below. For a complete list of **TimeStamp** class members, see the [TimeStamp Members](#) topic.

Public Instance Methods

 CompareTo	Compares this TimeStamp with another TimeStamp.
 Equals	Determines whether another TimeStamp is equal to this TimeStamp.
 GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current TimeStamp.

See Also

[TimeStamp Class](#) | [ftapi_dnet Namespace](#)

TimeStamp.CompareTo Method

Compares this TimeStamp with another TimeStamp.

```
public int CompareTo(  
    object anotherTimeStamp  
);
```

Parameters

anotherTimeStamp

The other TimeStamp to compare with this TimeStamp.

Return Value

A 32-bit signed integer that indicates the relative order of the comparands. The return value has these meanings:

Value	Meaning
Less than zero	This timeStamp is less than <i>anotherTimeStamp</i> .
Zero	This timeStamp is equal to <i>anotherTimeStamp</i> .
Greater than zero	This timeStamp is greater than <i>anotherTimeStamp</i> .

Implements

[IComparable.CompareTo](#)

Exceptions

Exception Type	Condition
ArgumentException	<i>anotherTimeStamp</i> is null or it is not a TimeStamp.

See Also

[TimeStamp Class](#) | [ftapi_dnet Namespace](#)

TimeStamp.Equals Method

Determines whether another TimeStamp is equal to this TimeStamp.

```
public override bool Equals(  
    object anotherTimeStamp  
);
```

Parameters

anotherTimeStamp

The other TimeStamp to compare with this TimeStamp.

Return Value

true if the other TimeStamp is equal to this TimeStamp; otherwise, *false*.

See Also

[TimeStamp Class](#) | [ftapi_dnet Namespace](#)

TimeStamp.GetHashCode Method

Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.

```
public override int GetHashCode();
```

Return Value

A hash code for the current TimeStamp.

See Also

[TimeStamp Class](#) | [ftapi dnet Namespace](#)

TimeStamp.ToString Method

Returns a single-line [String](#) that represents the current TimeStamp.

```
public override string ToString();
```

Return Value

A single-line [String](#) that represents the current TimeStamp.

See Also

[TimeStamp Class](#) | [ftapi_dnet Namespace](#)

Transaction Class

A client's request to the server to make an operation and/or to add/remove/modify an entity in its own Data Base.

For a list of all members of this type, see [Transaction Members](#).

System.Object

[ftapi_dnet.Disposable](#)

[ftapi_dnet.ConnectionActivity](#)

ftapi_dnet.Transaction

public class Transaction : [ConnectionActivity](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

A new transaction (i.e. a new client's request to the server) is logically defined by/with the [send](#) parameters:

- **entityAction** – the [EntityAction](#) (add/modify/remove/kill) to be performed,
- **entity** – the [Entity](#) on which the action must be done,
- **keyID** – a primary key identifier of the EntityClass of that Entity (all the KeyID [segments](#) of the Entity must be properly filled),
- **entityMask** – an optional [Mask](#) that describes which fields of the Entity must be filled,
- **entityResultRequired** – a request to obtain an Entity inside the [answer](#) that will be sent from the server.

New and Past Transactions

Once a new transaction is created its [TransactionID](#) may be retrieved (via the [transactionID](#) property) and saved (saving the 5 numbers representing the TransactionID as described in **TransactionID**).

From these saved values it's possible to recreate the original TransactionID and then re-create (using the [constructor](#) with a non-null TransactionID) a Transaction object that represents this past transaction.

In this manner this re-created Transaction object may be used to [query the status](#) of the past Transaction to the server (e.g. in the cases where there is a connection broken after the transaction was sent to the server and before the server-answer is received by the client application).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[Transaction Members](#) | [ftapi_dnet Namespace](#) | [TransactionID](#)






Transaction Members

[Transaction overview](#)











Public Instance Constructors

 Transaction	Overloaded. Initializes a new instance of the Transaction class.
---	--



Public Instance Properties

 connection (inherited from ConnectionActivity)	Gets the associated Connection.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .
 transactionID	Gets the associated TransactionID.

Public Instance Methods

 Close (inherited from Disposable)	Logically destroy the current object, making it again available.
 Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 onQuery	Method automatically invoked when the server answer to queryStatus arrives.
 onSend	Method automatically invoked when the server answer to send arrives.
 queryStatus	Query the server for the status of this, already sent, transaction.
 send	Overloaded. Sends this new transaction to the server.
 ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

Public Instance Events

 queryEvent	Event asynchronously raised when the server answer to queryStatus arrives.
 sendEvent	Event asynchronously raised when the server answer to send arrives.

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [TransactionID](#)

Transaction Constructor

Initializes a new instance of the Transaction class.

Overload List

Initializes a new instance for a new Transaction to be sent to the server.

[public Transaction\(Connection\);](#)

Initializes a new instance for a past already created Transaction or for a new Transaction to be sent to the server.

[public Transaction\(Connection,TransactionID\);](#)

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [TransactionID](#)

Transaction Constructor (Connection)

Initializes a new instance for a new Transaction to be sent to the server.

```
public Transaction(  
    Connection connection  
);
```

Parameters

connection
the parent Connection.

Remarks

Status

- on successfully Exit – [opening](#)

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [Transaction Constructor Overload List](#) | [TransactionID](#)

Transaction Constructor (Connection, TransactionID)

Initializes a new instance for a past already created Transaction or for a new Transaction to be sent to the server.

```
public Transaction(  
    Connection connection,  
    TransactionID transactionID  
);
```

Parameters

connection

the parent Connection.

transactionID

it may be

- **null** – to create a new Transaction to be sent to the server.
- **non-null** – to re-create a Transaction given its TransactionID.

Remarks

Status

- on successfully Exit
 - if *transactionID* is **null** – [opening](#)
 - if *transactionID* is **non-null** – [running](#)






See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [Transaction Constructor Overload List](#) | [TransactionID](#)

Transaction Properties

The properties of the **Transaction** class are listed below. For a complete list of **Transaction** class members, see the [Transaction Members](#) topic.

Public Instance Properties

 connection (inherited from ConnectionActivity)	Gets the associated Connection.
 name (inherited from Disposable)	Gets or sets the name used by ToString to represent the current object.
 status (inherited from Disposable)	Gets the current Status of this object.
 suppressFinalizeForbidden (inherited from Disposable)	Gets or set an indication about the automatic invocation of Finalize .
 transactionID	Gets the associated TransactionID.

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [TransactionID](#)

Transaction.transactionID Property

Gets the associated TransactionID.

```
public virtual TransactionID transactionID {get;}
```

Property Value

This value may be:

- **a new unique transactionID** – if the transaction was constructed to make a true new transaction to send to the server (i.e. without using any past transactionID in the [Transaction](#) constructor).
- **a past transactionID** – if the transaction was constructed to access a previously created transaction: in this case the value is the same past non-null transactionID given to the [Transaction](#) constructor.

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#)

Transaction Methods

The methods of the **Transaction** class are listed below. For a complete list of **Transaction** class members, see the [Transaction Members](#) topic.

Public Instance Methods

✦ Close (inherited from Disposable)	Logically destroy the current object, making it again available.
✦ Dispose (inherited from Disposable)	Logically destroy the current object, making it un-available.
✦ Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
✦ GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
✦ GetType (inherited from Object)	Gets the Type of the current instance.
✦ onQuery	Method automatically invoked when the server answer to queryStatus arrives.
✦ onSend	Method automatically invoked when the server answer to send arrives.
✦ queryStatus	Query the server for the status of this, already sent, transaction.
✦ send	Overloaded. Sends this new transaction to the server.
✦ ToString (inherited from Disposable)	Returns a single-line String that represents the current object.

See Also

[Transaction Class](#) | [ftapi dnet Namespace](#) | [TransactionID](#)

Transaction.onQuery Method

Method automatically invoked when the server answer to [queryStatus](#) arrives.

```
public virtual void onQuery(  
    QueryEventArgs qe  
);
```

Parameters

qe
Server answer to [queryStatus](#).

Remarks

The current [Status](#) remains [running](#).

If *qe.transactionStatus* is [pending](#) then the server can accept additional [queryStatus](#) invocation otherwise it is a good practice to [Dispose](#) this transaction.

This method must be properly extended in Transaction sub-classes since it does nothing in the Transaction class.

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [queryStatus](#) | [Transaction.QueryEventHandler](#)

Transaction.onSend Method

Method automatically invoked when the server answer to [send](#) arrives.

```
public virtual void onSend(  
    SendEventArgs se  
);
```

Parameters

se
Server answer to [send](#).

Remarks

The current [Status](#) remains [running](#).

If *se.transactionStatus* is [pending](#) then the server can accept a [queryStatus](#) invocation otherwise it is a good practice to [Dispose](#) this transaction.

This method must be properly extended in Transaction sub-classes since it does nothing in the Transaction class.

Example

```
public override void onSend(SendEventArgs se)  
{  
    using(this)  
    {  
        // base.onSend(se);  
        Lib.lib.trace(TraceLevel.full, "onSend result: "  
            + (se.entity == null ? "NULL" : se.entity.ToString(true, 0)));  
        string message = "Result of sending " + this + " is " + se.transactionStatus;  
        if(se.transactionStatus == TransactionStatus.conversationClosed)  
            Lib.lib.trace(TraceLevel.warning, "Transaction response lost. " + message);  
        else  
            Lib.lib.trace(TraceLevel.normal, message);  
        if(se.entity != null)  
        {  
            if(se.entity.entityClass.classId == FT_C_ERROR_INFO_ID)  
                Lib.lib.trace(TraceLevel.normal,  
                    "ReasonCode " + se.entity.getFieldLong("ReasonCode") +  
                    " " + se.entity.getFieldString("ErrorString"));  
            else  
                Lib.lib.trace(TraceLevel.warning,  
                    "Transaction Response Class " + se.entity.entityClass +  
                    " not managed");  
        }  
    }  
}
```

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [send](#) | [Transaction.SendEventHandler](#)



Transaction.queryStatus Method

Query the server for the status of this, already sent, transaction.

```
public virtual void queryStatus(
    bool entityResultRequired
);
```

Parameters

entityResultRequired

A request to obtain an Entity inside the [answer](#) that will be sent from the server.

When this value is

- **false** – the server will not send back an Entity to the client inside the [entity](#) field (it will be always equal to `null`).
- **true** – the server may choose to send an Entity in the answer; in this case the **entity** field may be not-`null`.

Remarks

If this method invocation completed successfully, then

- the query was sent to server,
- when the server-answer will be available the [queryEvent](#) will be raised.

otherwise

- the client has rejected the query,
- the query was *not* sent to server,
- no **queryEvent** will be raised,
- an appropriate [FastTrackException](#) will be thrown.

In both cases the current [Status](#) remains unchanged.

Status

- required on Entry – [running](#)
- on successfully Exit – **running**

Events

Event Type	Reason
queryEvent	Event asynchronously raised when the server answer to a successfully <code>queryStatus</code> arrives.

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [onQuery](#)

Transaction.send Method

Sends this new transaction to the server.

Overload List

Shorthand method to send this new transaction to the server using some defaults values.

[public virtual void send\(Entity,uint\);](#)

Full method to send this new transaction to the server.

[public virtual void send\(Entity,uint,EntityAction,bool,Mask\);](#)

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#)

Transaction.send Method (Entity, UInt32)

Shorthand method to send this new transaction to the server using some defaults values.

```
public virtual void send(
    Entity entity,
    uint keyID
);
```

Parameters

entity

[Entity](#) on which the action must be done.

keyID

Zero or a primary key identifier of the EntityClass of the *entity*'s EntityClass.

Remarks

The invocation

```
send(entity, keyID)
```

is equivalent to

```
send(entity, keyID, EntityAction.add, false, null)
```

so *please see* the [full send](#) method for a full detailed description of it.

Status & Events

Please see the **full send** method for a full detailed description of them.

Example

```
public TransactionOrd(Connection conn, Entity order, string newName) : base(conn)
{
    name = newName;
    Lib.lib.trace(TraceLevel.full, "Over " + this
        + " sending order: " + order.ToString(true,0));
    send(order, FT_C_ORDERKey);
}
```

See Also

[Transaction Class](#) | [ftapi dnet Namespace](#) | [Transaction.send Overload List](#)

Transaction.send Method (Entity, UInt32, EntityAction, Boolean, Mask)

Full method to send this new transaction to the server.

```
public virtual void send(  
    Entity entity,  
    uint keyID,  
    EntityAction entityAction,  
    bool entityResultRequired,  
    Mask entityMask  
);
```

Parameters

entity

[Entity](#) on which the action must be done.

Segments of this Entity that **must** be properly filled are:

- all segments described by *keyID* when it is different from zero.
- all segments described by *entityMask* when it is different from `null`.

All other fields **may** be properly filled (but see the proper FastTrack Market/Service specification to see all required well filled fields).

keyID

Zero or a primary key identifier of the EntityClass of the *entity*'s EntityClass.

All the KeyID [segments](#) of the Entity must be properly filled.

entityAction

[EntityAction](#) (add/modify/remove/kill) to be performed.

entityResultRequired

A request to obtain an Entity inside the [answer](#) that will be sent from the server.

When this value is

- **false** – the server will not send back an Entity to the client inside the [entity](#) field (it will be always equal to `null`).
- **true** – the server may choose to send an Entity in the answer; in this case the **entity** field may be not-`null`.

entityMask

An optional [Mask](#) that describes which fields of the Entity must be filled.

Remarks

If this method invocation completed successfully, then

- the transaction was sent to server,
- the current [Status](#) changes to [running](#),
- when the server–answer will be available the [sendEvent](#) will be raised.

otherwise

- the client rejected the send,
- the transaction was *not* sent to server,
- the current **Status** remains unchanged,
- no **sendEvent** will be raised,
- an appropriate [FastTrackException](#) will be thrown.

Status

- required on Entry – [opening](#)
- on successfully Exit – **running**

Events

Event Type	Reason
sendEvent	Event asynchronously raised when the server answer to a successfully send arrives.

Example

The following code–fragment shows a [shorthand send](#) invocation.

```
public TransactionOrd(Connection conn, Entity order, string newName) : base(conn)
{
    name = newName;
    Lib.lib.trace(TraceLevel.full, "Over " + this
        + " sending order: " + order.ToString(true,0));
    send(order, FT_C_ORDERKey);
}
```



See Also

[Transaction Class](#) | [ftapi dnet Namespace](#) | [Transaction.send Overload List](#) | [onSend](#)

Transaction Events

The events of the **Transaction** class are listed below. For a complete list of **Transaction** class members, see the [Transaction Members](#) topic.

Public Instance Events

 queryEvent	Event asynchronously raised when the server answer to queryStatus arrives.
 sendEvent	Event asynchronously raised when the server answer to send arrives.

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [TransactionID](#)

Transaction.queryEvent Event

Event asynchronously raised when the server answer to [queryStatus](#) arrives.

```
public event QueryEventHandler queryEvent;
```

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [onQuery](#)

Transaction.sendEvent Event

Event asynchronously raised when the server answer to [send](#) arrives.

```
public event SendEventHandler sendEvent;
```

See Also

[Transaction Class](#) | [ftapi_dnet Namespace](#) | [onSend](#)

Transaction.QueryEventArgs Class

Transaction server answer to [queryStatus](#).

For a list of all members of this type, see [Transaction.QueryEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

[ftapi_dnet.Transaction.TransactionEventArgs](#)

ftapi_dnet.Transaction.QueryEventArgs

public class Transaction.QueryEventArgs : [TransactionEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)








See Also

[Transaction.QueryEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onQuery](#)





Transaction.QueryEventArgs Members

[Transaction.QueryEventArgs overview](#)

Public Instance Fields

 classID (inherited from TransactionEventArgs)	ClassID on which the transaction was sent.
 entity (inherited from TransactionEventArgs)	Result entity.
 entityAction (inherited from TransactionEventArgs)	Requested EntityAction.
 keyID (inherited from TransactionEventArgs)	KeyID of the primary key on which the transaction was sent.
 marketReasonCode (inherited from TransactionEventArgs)	Specific market–related reason about the transaction abort.
 timeStamp (inherited from TransactionEventArgs)	Result timestamp.
 transactionStatus (inherited from TransactionEventArgs)	Status of a transaction sent to a server.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.

See Also

[Transaction.QueryEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onQuery](#)

Transaction.SendEventArgs Class

Transaction server answer to [send](#).

For a list of all members of this type, see [Transaction.SendEventArgs Members](#).

[System.Object](#)

[System.EventArgs](#)

[ftapi_dnet.FastTrackEventArgs](#)

[ftapi_dnet.Transaction.TransactionEventArgs](#)

ftapi_dnet.Transaction.SendEventArgs

public class Transaction.SendEventArgs : [TransactionEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)








See Also

[Transaction.SendEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onSend](#)





Transaction.SendEventArgs Members

[Transaction.SendEventArgs overview](#)

Public Instance Fields

 classID (inherited from TransactionEventArgs)	ClassID on which the transaction was sent.
 entity (inherited from TransactionEventArgs)	Result entity.
 entityAction (inherited from TransactionEventArgs)	Requested EntityAction.
 keyID (inherited from TransactionEventArgs)	KeyID of the primary key on which the transaction was sent.
 marketReasonCode (inherited from TransactionEventArgs)	Specific market–related reason about the transaction abort.
 timeStamp (inherited from TransactionEventArgs)	Result timestamp.
 transactionStatus (inherited from TransactionEventArgs)	Status of a transaction sent to a server.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.

See Also

[Transaction.SendEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onSend](#)

Transaction.TransactionEventArgs Class

Transaction server answer to [send/queryStatus](#).

For a list of all members of this type, see [Transaction.TransactionEventArgs Members](#).

System.Object

System.EventArgs

[ftapi_dnet.FastTrackEventArgs](#)

ftapi_dnet.Transaction.TransactionEventArgs

[ftapi_dnet.Transaction.QueryEventArgs](#)

[ftapi_dnet.Transaction.SendEventArgs](#)

public abstract class Transaction.TransactionEventArgs : [FastTrackEventArgs](#)

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[Transaction.TransactionEventArgs Members](#) | [ftapi_dnet Namespace](#) | [onSend](#) | [onQuery](#)

Transaction.TransactionEventArgs Members

[Transaction.TransactionEventArgs overview](#)

Public Instance Fields

classID	ClassID on which the transaction was sent.
entity	Result entity.
entityAction	Requested EntityAction.
keyID	KeyID of the primary key on which the transaction was sent.
marketReasonCode	Specific market–related reason about the transaction abort.
timeStamp	Result timestamp.
transactionStatus	Status of a transaction sent to a server.

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetType (inherited from Object)	Gets the Type of the current instance.
ToString (inherited from FastTrackEventArgs)	Overloaded. Returns a single/multiple line String that represents the current public fields values.








See Also

[Transaction.TransactionEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onSend](#) | [onQuery](#)

Transaction.TransactionEventArgs Fields

The fields of the **Transaction.TransactionEventArgs** class are listed below. For a complete list of **Transaction.TransactionEventArgs** class members, see the [Transaction.TransactionEventArgs Members](#) topic.

Public Instance Fields

 classID	ClassID on which the transaction was sent.
 entity	Result entity.
 entityAction	Requested EntityAction.
 keyID	KeyID of the primary key on which the transaction was sent.
 marketReasonCode	Specific market–related reason about the transaction abort.
 timeStamp	Result timestamp.
 transactionStatus	Status of a transaction sent to a server.

See Also

[Transaction.TransactionEventArgs Class](#) | [ftapi_dnet Namespace](#) | [onSend](#) | [onQuery](#)

Transaction.TransactionEventArgs.classID Field

ClassID on which the transaction was sent.

```
public readonly uint classID;
```

Remarks

This is different from `entity.entityClass.classId`

See Also

[Transaction.TransactionEventArgs Class](#) | [ftapi_dnet Namespace](#)

Transaction.TransactionEventArgs.entity Field

Result entity.

public readonly [Entity](#) entity;

Remarks

This field is meaning only when the transaction committed (i.e. [transactionStatus](#) is [OK](#)).

This field is null when the result entity was not required (see `entityResultRequired` parameter in [send](#) or [queryStatus](#)) or when the server choose to not send it.

See Also

[Transaction.TransactionEventArgs Class](#) | [ftapi_dnet Namespace](#)

Transaction.TransactionEventArgs.entityAction Field

Requested EntityAction.

```
public readonly EntityAction entityAction;
```

See Also

[Transaction.TransactionEventArgs Class](#) | [ftapi_dnet Namespace](#)

Transaction.TransactionEventArgs.keyID Field

KeyID of the primary key on which the transaction was sent.

```
public readonly uint keyID;
```

See Also

[Transaction.TransactionEventArgs Class](#) | [ftapi_dnet Namespace](#)

Transaction.TransactionEventArgs.marketReasonCode Field

Specific market–related reason about the transaction abort.

```
public readonly int marketReasonCode;
```

Remarks

This field is meaning only when the transaction aborted (i.e. [transactionStatus](#) is [NO](#)).

The exact meaning of this field depends on the specific market and it is documented in the correspondig market server manual, apart from the following generic values:

- **10000** – internal error
- **10001** – not logged
- **10002** – inadequate privileges
- **10003** – invalid request action
- **10004** – invalid TransactionID

See Also

[Transaction.TransactionEventArgs Class](#) | [ftapi_dnet Namespace](#)

Transaction.TransactionEventArgs.timeStamp Field

Result timestamp.

```
public readonly TimeStamp timeStamp;
```

Remarks

This field is meaning only when the transaction committed (i.e. [transactionStatus](#) is [OK](#)).

See Also

[Transaction.TransactionEventArgs Class](#) | [ftapi_dnet Namespace](#)

Transaction.TransactionEventArgs.transactionStatus Field

Status of a transaction sent to a server.

```
public readonly TransactionStatus transactionStatus;
```

See Also

[Transaction.TransactionEventArgs Class](#) | [ftapi_dnet Namespace](#)

TransactionID Class

An unique Transaction identifier.

For a list of all members of this type, see [TransactionID Members](#).

[System.Object](#)

ftapi_dnet.TransactionID

[\[Serializable\]](#)

public class TransactionID

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread–safe.

Remarks

Each TransactionID is identified by

- the triplet ([businessServiceID](#), [clientServiceID](#), [clientID](#)) i.e. the client from which the transaction was sent, normally automatically generated by [Transaction](#) constructor with the corresponding triplet ([businessServiceID](#), [clientServiceID](#), [clientID](#)) of the Connection on which the transaction was sent,
- the client [timeStamp](#) referring the transaction construction time.

This class is marked with [SerializableAttribute](#) in order to be able to save and then re–create TransactionID objects.

As an alternative an application may save the quadruplet ([businessServiceID](#), [clientServiceID](#), [clientID](#), [timeStamp](#)) and then re–create the same TransactionID object using the given constructor.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[TransactionID Members](#) | [ftapi_dnet Namespace](#)





TransactionID Members

[TransactionID overview](#)





Public Instance Constructors

 TransactionID Constructor	Initializes a new TransactionID instance given the quadruplet (businessServiceID, clientServiceID, clientID, timeStamp).
---	--

Public Instance Fields

 businessServiceID	Same businessServiceID value of the connection on which the transaction was made.
 clientID	Same clientID value of the connection on which the transaction was made.
 clientServiceID	Same clientServiceID value of the connection on which the transaction was made.
 timeStamp	Timestamp of transaction construction.

Public Instance Methods

 Equals	Determines whether another TransactionID is equal to this TransactionID.
 GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current TransactionID.

See Also

[TransactionID Class](#) | [ftapi_dnet Namespace](#)

TransactionID Constructor

Initializes a new TransactionID instance given the quadruplet (businessServiceID, clientServiceID, clientID, timeStamp).

```
public TransactionID(  
    uint bsID,  
    uint csID,  
    uint cID,  
    TimeStamp ts  
);
```

Parameters

<i>bsID</i>	businessServiceID value.
<i>csID</i>	clientServiceID value.
<i>cID</i>	clientID value.
<i>ts</i>	timeStamp reference.

Remarks

This convenience constructor may be used to re-create a TransactionID previously saved as a four-fields quadruplet ([businessServiceID](#)), [clientServiceID](#), [clientID](#), [timeStamp](#)).





See Also

[TransactionID Class](#) | [ftapi_dnet Namespace](#)

TransactionID Fields

The fields of the **TransactionID** class are listed below. For a complete list of **TransactionID** class members, see the [TransactionID Members](#) topic.

Public Instance Fields

 businessServiceID	Same businessServiceID value of the connection on which the transaction was made.
 clientID	Same clientID value of the connection on which the transaction was made.
 clientServiceID	Same clientServiceID value of the connection on which the transaction was made.
 timeStamp	Timestamp of transaction construction.

See Also

[TransactionID Class](#) | [ftapi_dnet Namespace](#)

TransactionID.businessServiceID Field

Same [businessServiceID](#) value of the connection on which the transaction was made.

```
public readonly uint businessServiceID;
```

See Also

[TransactionID Class](#) | [ftapi_dnet Namespace](#)

TransactionID.clientID Field

Same [clientID](#) value of the connection on which the transaction was made.

```
public readonly uint clientID;
```

See Also

[TransactionID Class](#) | [ftapi_dnet Namespace](#)

TransactionID.clientServiceID Field

Same [clientServiceID](#) value of the connection on which the transaction was made.

```
public readonly uint clientServiceID;
```

See Also

[TransactionID Class](#) | [ftapi_dnet Namespace](#)

TransactionID.timeStamp Field

Timestamp of transaction construction.

```
public readonly TimeStamp timeStamp;
```





See Also

[TransactionID Class](#) | [ftapi_dnet Namespace](#)

TransactionID Methods

The methods of the **TransactionID** class are listed below. For a complete list of **TransactionID** class members, see the [TransactionID Members](#) topic.

Public Instance Methods

 Equals	Determines whether another TransactionID is equal to this TransactionID.
 GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	Returns a single-line String that represents the current TransactionID.

See Also

[TransactionID Class](#) | [ftapi_dnet Namespace](#)

TransactionID.Equals Method

Determines whether another TransactionID is equal to this TransactionID.

```
public override bool Equals(  
    object anotherTransactionID  
);
```

Parameters

anotherTransactionID

The other TransactionID to compare with this TransactionID.

Return Value

true if the other TransactionID is equal to this TransactionID; otherwise, *false*.

See Also

[TransactionID Class](#) | [ftapi_dnet Namespace](#)

TransactionID.GetHashCode Method

Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.

```
public override int GetHashCode();
```

Return Value

A hash code for the current TransactionID.

See Also

[TransactionID Class](#) | [ftapi dnet Namespace](#)

TransactionID.ToString Method

Returns a single-line [String](#) that represents the current TransactionID.

```
public override string ToString();
```

Return Value

A single-line [String](#) that represents the current TransactionID.

See Also

[TransactionID Class](#) | [ftapi_dnet Namespace](#)

IDump Interface

Defines a method to retrieve a multi-line string representation of an object.

For a list of all members of this type, see [IDump Members](#).

public interface IDump

Types that implement IDump

Type	Description
Connection.MonitorEventArgs	Nam transaction status data.
Connection.OpenEventArgs	Connection server answer to open .
Entity	Value of an EntityClass .
FastTrackEventArgs	FastTrackEventArgs is the FastTrack base class for classes containing event data.
Filter.CloseEventArgs	Filter server answer to Close .
Filter.OpenEventArgs	Filter server answer to open .
Filter.SetEventArgs	Filter server answer to set .
Lib.TraceEventArgs	Data to be traced by the application.
Query.CloseEventArgs	Query server answer to Close .
Query.NotifyEventArgs	Query data coming from the server as result of a open or queryRows .
Query.OpenEventArgs	Query server answer to open .
Query.RowsEventArgs	Query server answer to queryRows .
Subscription.CloseEventArgs	Subscription server answer to Close .
Subscription.NotifyEventArgs	Subscription data coming from the server as result of a open or refreshEntity .
Subscription.OpenEventArgs	Subscription server answer to open .
Transaction.QueryEventArgs	Transaction server answer to queryStatus .
Transaction.SendEventArgs	Transaction server answer to send .
Transaction.TransactionEventArgs	Transaction server answer to send/queryStatus .

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)


See Also

[IDump Members](#) | [ftapi_dnet Namespace](#)

IDump Members

[IDump overview](#)

Public Instance Methods

 ToString	Returns a multi-line String that represents the current IDump object public fields.
--	---


See Also

[IDump Interface](#) | [ftapi_dnet Namespace](#)

IDump Methods

The methods of the **IDump** interface are listed below. For a complete list of **IDump** interface members, see the [IDump Members](#) topic.

Public Instance Methods

 ToString	Returns a multi-line String that represents the current IDump object public fields.
--	---

See Also

[IDump Interface](#) | [ftapi_dnet Namespace](#)

IDump.ToString Method

Returns a multi-line [String](#) that represents the current [IDump](#) object public fields.

```
string ToString(  
    bool withHeader,  
    uint offset  
);
```

withHeader

First line header request.

offset

Indentation value.

Typically: number of spaces = offset*4.

Return Value

A multi-line [String](#) that represents the current [IDump](#) object public fields.

Remarks

Each IDump field is written on a line alone with a

```
fieldName = fieldValue
```

format, eventually preceded by an header line.

Example

A typically generated string will be:

```
ftapi_dnet.Subscription+NotifyEventArgs  
    entityAction = add  
    classID = 30007  
    keyID = 1  
    timeStamp = 753179494.16  
    isMasked = False  
    entity = entity 33 of class FT_C_MARKET(30007)
```

See Also

[IDump Interface](#) | [ftapi_dnet Namespace](#)

Connection.BrokenEventHandler Delegate

Connection callback method to be called when the connection with the server crashed in an unrecoverable manner.

```
public delegate void Connection.BrokenEventHandler(  
    object connection,  
    EventArgs e  
);
```

Parameters

connection

Involved [Connection](#).

e

[Empty](#) event.

Remarks

Please see the [onBroken](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited `onBroken` method there are only two differences

- the absence of the *connection* parameter (in the `onBroken` method the connection defaults to `this`),
- the absence of the [Empty](#) event *e* parameter.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onBroken](#)

Connection.CloseEventHandler Delegate

Connection callback method to be called when the server answer to [Close](#) arrives.

```
public delegate void Connection.CloseEventHandler(  
    object connection,  
    EventArgs e  
);
```

Parameters

connection

Involved [Connection](#).

e

[Empty](#) event.

Remarks

Please see the [onClose](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onClose method there are only two differences

- the absence of the *connection* parameter (in the onClose method the connection defaults to `this`).
- the absence of the [Empty](#) event *e* parameter.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onClose](#)

Connection.MonitorEventHandler Delegate

Connection callback method to be called when there is a status-change on a NAM-market transaction monitored by [monitorNamTransactions](#).

```
public delegate void Connection.MonitorEventHandler(  
    object connection,  
    MonitorEventArgs me  
);
```

Parameters

connection

Involved [Connection](#).

me

Transaction status data.

Remarks

Please see the [onMonitor](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onMonitor method there is only one difference:

- the absence of the *connection* parameter (in the onMonitor method the connection defaults to this).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onMonitor](#)

Connection.OpenEventHandler Delegate

Connection callback method to be called when the server answer to [open](#) arrives.

```
public delegate void Connection.OpenEventHandler(  
    object connection,  
    OpenEventArgs oe  
);
```

Parameters

connection

Involved [Connection](#).

oe

Server answer to [open](#).

Remarks

Please see the [onOpen](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onOpen method there is only one difference:

- the absence of the *connection* parameter (in the onOpen method the connection defaults to `this`).

Example

```
public void myOpenEventHandler(Object connection, Connection.OpenEventArgs oe)  
{  
    if(oe.result == Error.OK)  
    {  
        Thread thread = new Thread(new ThreadStart(threadStart));  
        thread.IsBackground = true;  
        thread.Start();  
    }  
}
```

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onOpen](#)

Filter.CloseEventHandler Delegate

Filter callback method to be called when the server answer to [Close](#) arrives.

```
public delegate void Filter.CloseEventHandler(  
    object filter,  
    CloseEventArgs ce  
);
```

Parameters

filter
Involved [Filter](#).

ce
Server answer to [Close](#).

Remarks

Please see the [onClose](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onClose method there is only one difference:

- the absence of the *filter* parameter (in the onClose method the filter defaults to `this`).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onClose](#)

Filter.OpenEventHandler Delegate

Filter callback method to be called when the server answer to [open](#) arrives.

```
public delegate void Filter.OpenEventHandler(  
    object filter,  
    OpenEventArgs oe  
);
```

Parameters

filter Involved [Filter](#).

oe Server answer to [open](#).

Remarks

Please see the [onOpen](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onOpen method there is only one difference:

- the absence of the *filter* parameter (in the onOpen method the filter defaults to `this`).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onOpen](#)

Filter.SetEventHandler Delegate

Filter callback method to be called when the server answer to [set](#) arrives.

```
public delegate void Filter.SetEventHandler(  
    object filter,  
    SetEventArgs se  
);
```

Parameters

filter
Involved [Filter](#).

se
Server answer to [set](#).

Remarks

Please see the [onSet](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onSet method there is only one difference:

- the absence of the *filter* parameter (in the onSet method the filter defaults to `this`).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onSet](#)

Lib.TraceEventHandler Delegate

Lib callback method to be called when a trace message must be programmatically traced as requested by [setTraceMode](#).

```
public delegate void Lib.TraceEventHandler(  
    object lib,  
    TraceEventArgs te  
);
```

Parameters

lib
Involved [Lib](#).

te
Trace data.

Example

```
private void myTraceEventHandler(Object lib, Lib.TraceEventArgs te)  
{  
    #if DEBUG  
        if(te.traceSource != TraceSource.ftapi)  
            Console.WriteLine("TRACE {0,-11} {1,-7} {2}",  
                te.traceSource, te.traceLevel, te.message);  
    #else  
        if(te.traceSource == TraceSource.application)  
            Console.WriteLine(te.message);  
    #endif  
}
```

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [setTraceMode](#) | [trace](#)

Query.CloseEventHandler Delegate

Query callback method to be called when the server answer to [Close](#) arrives.

```
public delegate void Query.CloseEventHandler(  
    object query,  
    CloseEventArgs ce  
);
```

Parameters

query

Involved [Query](#).

ce

Server answer to [Close](#).

Remarks

Please see the [onClose](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onClose method there is only one difference:

- the absence of the *query* parameter (in the onClose method the query defaults to this).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onClose](#)

Query.NotifyEventHandler Delegate

Query callback method to be called when a new data comes from the server as result of a [open](#) or [queryRows](#).

```
public delegate void Query.NotifyEventHandler(  
    object query,  
    NotifyEventArgs ne  
);
```

Parameters

query

Involved [Query](#).

ne

Data coming from the server as result of a [open](#) or [queryRows](#).

Remarks

Please see the [onNotify](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited `onNotify` method there is only one difference:

- the absence of the *query* parameter (in the `onNotify` method the query defaults to `this`).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onNotify](#)

Query.OpenEventHandler Delegate

Query callback method to be called when the server answer to [open](#) arrives.

```
public delegate void Query.OpenEventHandler(  
    object query,  
    OpenEventArgs oe  
);
```

Parameters

query

Involved [Query](#).

oe

Server answer to [open](#).

Remarks

Please see the [onOpen](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onOpen method there is only one difference:

- the absence of the *query* parameter (in the onOpen method the query defaults to this).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onOpen](#)

Query.RowsEventHandler Delegate

Query callback method to be called when the server answer to [queryRows](#) arrives.

```
public delegate void Query.RowsEventHandler(  
    object query,  
    RowsEventArgs re  
);
```

Parameters

query

Involved [Query](#).

re

Server answer to [queryRows](#).

Remarks

Please see the [onRows](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited `onRows` method there is only one difference:

- the absence of the *query* parameter (in the `onRows` method the query defaults to `this`).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onRows](#)

Subscription.BrokenEventHandler Delegate

Subscription callback method to be called when the connection with the server crashed or when the server choose to terminate the connection.

```
public delegate void Subscription.BrokenEventHandler(  
    object subscription,  
    EventArgs e  
);
```

Parameters

subscription

Involved [Subscription](#).

e

[Empty](#) event.

Remarks

Please see the [onBroken](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onBroken method there are only two differences

- the absence of the *subscription* parameter (in the onBroken method the subscription defaults to this),
- the absence of the [Empty](#) event *e* parameter.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onBroken](#)

Subscription.CloseEventHandler Delegate

Subscription callback method to be called when the server answer to [Close](#) arrives.

```
public delegate void Subscription.CloseEventHandler(  
    object subscription,  
    CloseEventArgs ce  
);
```

Parameters

subscription

Involved [Subscription](#).

ce

Server answer to [Close](#).

Remarks

Please see the [onClose](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onClose method there is only one difference:

- the absence of the *subscription* parameter (in the onClose method the subscription defaults to `this`).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onClose](#)

Subscription.IdleEventHandler Delegate

Subscription callback method to be called when the flow of historical data is finished and the flow of actual data is starting.

```
public delegate void Subscription.IdleEventHandler(  
    object subscription,  
    EventArgs e  
);
```

Parameters

subscription

Involved [Subscription](#).

e

[Empty](#) event.

Remarks

Please see the [onIdle](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onIdle method there are only two differences

- the absence of the *subscription* parameter (in the onIdle method the subscription defaults to `this`),
- the absence of the [Empty](#) event *e* parameter.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onIdle](#)

Subscription.NotifyEventHandler Delegate

Subscription callback method to be called when a new data comes from the server as result of a [open](#) or [refreshEntity](#).

```
public delegate void Subscription.NotifyEventHandler(  
    object subscription,  
    NotifyEventArgs ne  
);
```

Parameters

subscription

Involved [Subscription](#).

ne

Data coming from the server as result of a [open](#) or [refreshEntity](#).

Remarks

Please see the [onNotify](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited `onNotify` method there is only one difference:

- the absence of the *subscription* parameter (in the `onNotify` method the subscription defaults to `this`).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onNotify](#)

Subscription.OpenEventHandler Delegate

Subscription callback method to be called when the server answer to [open](#) arrives.

```
public delegate void Subscription.OpenEventHandler(  
    object subscription,  
    OpenEventArgs oe  
);
```

Parameters

subscription

Involved [Subscription](#).

oe

Server answer to [open](#).

Remarks

Please see the [onOpen](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onOpen method there is only one difference:

- the absence of the *subscription* parameter (in the onOpen method the subscription defaults to `this`).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onOpen](#)

Transaction.QueryEventHandler Delegate

Transaction callback method to be called when the server answer to [queryStatus](#) arrives.

```
public delegate void Transaction.QueryEventHandler(  
    object transaction,  
    QueryEventArgs qe  
);
```

Parameters

transaction

Involved [Transaction](#).

qe

Server answer to [queryStatus](#).

Remarks

Please see the [onQuery](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onQuery method there is only one difference:

- the absence of the *transaction* parameter (in the onQuery method the transaction defaults to this).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onQuery](#)

Transaction.SendEventHandler Delegate

Transaction callback method to be called when the server answer to [send](#) arrives.

```
public delegate void Transaction.SendEventHandler(  
    object transaction,  
    SendEventArgs se  
);
```

Parameters

transaction

Involved [Transaction](#).

se

Server answer to [send](#).

Remarks

Please see the [onSend](#) related documentation for a full behaviour/parameters/events/examples description.

In the cited onSend method there is only one difference:

- the absence of the *transaction* parameter (in the onSend method the transaction defaults to `this`).

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onSend](#)

ContextAttribute Enumeration

Additional attributes to associate to a new Connection.

public enum ContextAttribute

Members

Member Name	Description
compressed	Compression type to use in the new connection. E.g. "ZIP" or "" (default for no compression).
applAuthKey	Authorization key for the new connection. An authorization key, normally given by List S.p.a. in an authorization file (see applAuthFile , is a key that allow the client to open and use successfully a connection with a given server. The server may subsequently returns a invalidConfigurationKey failure-code if it does not comply with this value.
applAuthFile	Pathname of file containg an authorization key for the new connection. An authorization file, given by List S.p.a., contains a key that allow the client to open and use successfully a connection with a given server. The server may subsequently returns a invalidConfigurationKey failure-code if it does not comply with the authorization key.
certificateX509	X509 certificate.
connectionTimeout	Timeout (in seconds) during which the Connection open will remain blocked waiting for a connection to the server. During this period the entire application will be blocked and no more responsive/sensitive to events arriving from the network. Minimum timeout value is 1 (one second).
alternativeHost1	First optional alternative requested server-host of a new connection. The optional pair given by this value together with alternativePort1 describe another server to which the client must talk in the case the first attempt to the principal server (see IPAddress and IPPort parameters of Connection open) failed.
alternativePort1	First optional alternative requested server-port of a new connection.

	The optional pair given by alternativeHost1 together with this value describe another server to which the client must talk in the case the first attempt to the principal server (see IPAddress and IPPort parameters of Connection.open) failed.
alternativeHost2	Second optional alternative requested server–host of a new connection.
alternativePort2	Second optional alternative requested server–port of a new connection.
alternativeHost3	Third optional alternative requested server–host of a new connection.
alternativePort3	Third optional alternative requested server–port of a new connection.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [Connection.open](#)

EntityAction Enumeration

Request/Notification about an [Entity](#) operation.

public enum EntityAction

Remarks

Requests	Subscription. open
	Transaction. send
Notifications	Subscription.NotifyEventArgs
	Transaction.TransactionEventArgs
	Connection.MonitorEventArgs

Members

Member Name	Description
add	<p>Entity added (or to be added) on the server.</p> <p>Historical data (i.e. data coming before onIdle invocation) notified in Subscription.NotifyEventArgs are always tagged with this value (never with rwt).</p>
del	<p>Entity logically removed (or to be removed) from the server</p> <p>Entity available in notifications (Subscription.NotifyEventArgs or Transaction.TransactionEventArgs or Connection.MonitorEventArgs) are generally undefined on any field apart from those associated with a given keyID.</p> <p>Historical data (i.e. data coming before onIdle invocation) are never tagged with this value.</p>
rwt	<p>Entity rewritten (or to be rewritten) on the server.</p> <p>Historical data (i.e. data coming before onIdle invocation) are never tagged with this value.</p>
kil	<p>Entity physically removed (or to be removed) from the server.</p> <p>Every time this value is notified the server EntityClass version changed: the new version is available in Subscription.NotifyEventArgs.timeStamp.seconds field and it's different from both</p>

- the initially required EntityClass version (classVersion parameter of Subscription [open](#))
- the initially subscribed EntityClass version ([classVersion](#) field of Subscription.OpenEventArgs)

With notifications ([Subscription.NotifyEventArgs](#) or [Transaction.TransactionEventArgs](#) or [Connection.MonitorEventArgs](#)) if the associated KeyID is

- **positive** – then entities are generally undefined on any field apart from those associated with the given positive keyID,
- **zero or negative** – entity are always null and the meaning is that all entities are physically removed from the server.

Historical data (i.e. data coming before [onIdle](#) invocation) are never tagged with this value.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#)

EntityClassType Enumeration

EntityClass type.

```
public enum EntityClassType
```

Members

Member Name	Description
NONE	Not a EntityClass. This type is never a legitimate value for the type property
STRUCT	A typical EntityClass. All market EntityClass have this type. Some EntityField may have this type (if their type is NONE).
ENUM	An enumerated list. Some EntityField may have this type (if their type is NONE).
UNION	Planned, but currently unused, type.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [EntityClass.type](#)

EntityFieldType Enumeration

EntityField type.

public enum EntityFieldType

Remarks

Numbers may have:

- **integer signed two's-complement binary representation** – values in range $[-2^{8*\text{dataSize}-1}..2^{8*\text{dataSize}-1}-1]$.
- **integer unsigned binary representation** – values in range $[0..2^{8*\text{dataSize}}-1]$.
- **32-bit or 64-bit IEEE 754 floating-point number representation** – values specified in *IEEE Standard for Binary Floating-Point Arithmetic*, ANSI/IEEE Standard 754-1985 (IEEE, New York).

EntityFieldType Compendium

Size (i.e. dataSize)	Representation	FastTrack type	Corresponding .NET type
exactly 1 byte	unsigned	BYTE	Byte
	signed	CHAR	SByte
	unsigned	UCHAR	Byte
	unsigned	BOOLEAN	Byte
typically 2 bytes	signed	SHORT	Int16
	unsigned	USHORT	UInt16
typically 4 bytes	signed	INT	Int32
	unsigned	UINT	UInt32
	unsigned	DATE	UInt32
	unsigned	TIME	UInt32
	unsigned	TDATE	UInt32
	unsigned	LTIME	UInt32
	unsigned	LDATE	UInt32
	unsigned	MTIME	UInt32
	unsigned	DTIME	UInt32
typically 4 or 8 bytes	signed	LONG	Int32 or Int64
	unsigned	ULONG	UInt32 or UInt64

	signed	POINTER	Int32 or Int64
exactly 4 bytes	single precision floating number	FLOAT	Single
exactly 8 bytes	double precision floating number	DOUBLE	Double
numElements +1 bytes	encoded with encoding and right-filled with a single 0x00 byte	STRING	String

Members

Member Name	Description
NONE	An EntityClass. See the entityClass property.
INT	Signed integer.
UINT	Unsigned integer.
SHORT	Signed integer.
USHORT	Unsigned integer.
LONG	Signed integer.
ULONG	Unsigned integer.
FLOAT	Single-precision floating number.
DOUBLE	Double-precision floating number.
BYTE	Unsigned integer. Values in range [0..255] (superset of ASCII values) are <i>never</i> interpreted with encoding .
CHAR	Signed integer. Values in range [-128..127] (superset of ASCII values) are <i>never</i> interpreted with encoding .
UCHAR	Unsigned integer. Values in range [0..255] (superset of ASCII values) are <i>never</i> interpreted with encoding .
STRING	String. They are <i>always</i> interpreted with encoding . Please note that numElements and dataSize properties (in this case the maximum length and the allocated space in bytes) refer to the encoded string and not to the string itself. E.g. the "A?B" string (i.e. "A\u2260B") is a 3 characters string that requires a 5 bytes sequence {0x41, 0xE2, 0x89, 0xA0, 0x42} to be represented with UTF-8 encoding (so in this case dataSize must be at least 6).

	<p>The same "A?B" string requires only 3 bytes {0x41, 0x3F, 0x42} to be represented with ASCII encoding, but in this case it will be represented as "A?B" because there is not a representation for ? (i.e. 0x2260) in ASCII.</p> <p>Furthermore please consider that decoding a byte sequence with a different encoding will result in a different string: e.g. decoding the previous 5 bytes UTF-8 encoded sequence {0x41, 0xE2, 0x89, 0xA0, 0x42} using an ASCII encoding will result in the "A???B" string, and using the windows-1252 encoding will result in the "Aâ% B" (i.e. "A\u00E2\u2030\u00A0B") string.</p>
DATE	<p>Unsigned integer.</p> <p>A date represented with the number of seconds elapsed since 1/1/1970.</p>
TIME	<p>Unsigned integer.</p> <p>Currently unused.</p>
TDATE	<p>Unsigned integer.</p> <p>Currently unused.</p>
LTIME	<p>Unsigned integer.</p> <p>A timestamp with a HHMMSScc decimal format representation.</p>
LDATE	<p>Unsigned integer.</p> <p>A date with a YYYYMMDD decimal format representation.</p>
BOOLEAN	<p>Unsigned integer.</p> <p>Values in range [0..255] (0 ? false, others ? true).</p>
MTIME	<p>Unsigned integer.</p> <p>A timestamp with a HHMMSSmmm decimal format representation.</p>
DTIME	<p>Unsigned integer.</p> <p>A timestamp with a HHMMSSdddd decimal format representation.</p>
POINTER	<p>Signed integer.</p> <p>Currently unused.</p>

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [EntityField.type](#)

Error Enumeration

Failure result–code.

public enum Error

Remarks

These values are available inside [FastTrackException](#) and in some notifications arriving from the server: [Connection.OpenEventArgs](#), [Subscription.OpenEventArgs](#) and [Subscription.CloseEventArgs](#).

Members

Member Name	Description
OK	Positive answer notified by the server when the operation completed successfully.
doubleInit	Failure code with a self–describing meaning.
internalError	Failure code with a self–describing meaning.
init	Failure code with a self–describing meaning.
close	Failure code with a self–describing meaning.
invalidIPAddress	Failure code with a self–describing meaning.
invalidIPPort	Failure code with a self–describing meaning.
noServer	Failure code with a self–describing meaning.
transport	Failure code with a self–describing meaning.
invalidUserType	Failure code with a self–describing meaning.
invalidUserName	Failure code with a self–describing meaning.
invalidPassword	Failure code with a self–describing meaning.
invalidRevision	Failure code with a self–describing meaning.
invalidCID	Failure code with a self–describing meaning.
alreadyLog	Failure code with a self–describing meaning.
invalidConvId	Failure code with a self–describing meaning.
generic	Failure code with a self–describing meaning.
invalidSubscribeType	Failure code with a self–describing meaning.
invalidClassID	Failure code with a self–describing meaning.
doubleSubscription	Failure code with a self–describing meaning.
noSubscribedClass	Failure code with a self–describing meaning.
invalidConfigurationKey	Failure code with a self–describing meaning.
memoryFull	Failure code with a self–describing meaning.
invalidServerStatus	Failure code with a self–describing meaning.
invalidArgument	Failure code with a self–describing meaning.
unresolvedService	Failure code with a self–describing meaning.
invalidMask	Failure code with a self–describing meaning.
exceedSession	Failure code with a self–describing meaning.

invalidProfile	Failure code with a self-describing meaning.
invalidFilter	Failure code with a self-describing meaning.
noSuchEntityClass	Failure code with a self-describing meaning.
noSuchField	Failure code with a self-describing meaning.
invalidEntity	Failure code with a self-describing meaning.
invalidFieldType	Failure code with a self-describing meaning.
invalidStatus	Failure code with a self-describing meaning.
multipleDefinedField	Failure code with a self-describing meaning.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [FastTrackException](#)

FilterClose Enumeration

Server filter–destruction failure–code.

`public enum FilterClose`

Remarks

These values are carried out by the server on the [result](#) field of [Filter.CloseEventArgs](#) as an answer to a [Filter.Close](#) destruction request.

Members

Member Name	Description
OK	Positive answer: filter successfully destroyed on the server.
invalidFilterID	Negative answer: an internal error occurred.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onClose](#)

FilterOpen Enumeration

Server filter—creation failure—code.

`public enum FilterOpen`

Remarks

These values are carried out by the server on the [result](#) field of [Filter.OpenEventArgs](#) as an answer to a `Filter.open` creation request.

Members

Member Name	Description
OK	Positive answer: filter successfully created by the server.
syntaxError	Negative answer: syntax error in definition parameter.
invalidFilterLen	Negative answer: definition parameter too long.
invalidClass	Negative answer: wrong <code>classID</code> parameter.
invalidFilterType	Negative answer: wrong <code>type</code> parameter.
notImplemented	Negative answer: filter are not implemented by the server.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onOpen](#)

FilterSet Enumeration

Server filter–extension failure–code.

```
public enum FilterSet
```

Remarks

These values are carried out by the server on the [result](#) field of [Filter.SetEventArgs](#) as an answer to a `Filter.set` extension request.

Members

Member Name	Description
OK	Positive answer: filter successfully extended on the server.
syntaxError	Negative answer: syntax error in definition parameter.
invalidFilterID	Negative answer: an internal error occurred.
invalidFilterLen	Negative answer: definition parameter too long.
alreadySet	Negative answer: filter was already extended.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onSet](#)

QueryClose Enumeration

Server query–destruction failure–code.

`public enum QueryClose`

Remarks

These values are carried out by the server on the [result](#) field of [Query.CloseEventArgs](#) as an answer to a [Query.Close](#) destruction request.

Members

Member Name	Description
OK	Positive answer: query successfully destroyed on the server.
wrongQueryID	Negative answer: an internal error occurred.
error	Negative answer: other (unspecified) error.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onClose](#)

QueryOpen Enumeration

Server query—creation failure—code.

public enum QueryOpen

Remarks

These values are carried out by the server on the [result](#) field of [Query.OpenEventArgs](#) as an answer to a [Query.open](#) creation request.

Members

Member Name	Description
OK	Positive answer: query successfully created by the server.
badParameters	Negative answer: some (unspecified) bad parameter.
wrongQueryID	Negative answer: wrong queryID parameter.
genericError	Negative answer: other (unspecified) error.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onOpen](#)

QueryRows Enumeration

Server query—rows failure—code.

`public enum QueryRows`

Remarks

These values are carried out by the server on the [result](#) field of [Query.RowsEventArgs](#) as an answer to a [Query.queryRows](#) request.

Members

Member Name	Description
OK	Positive answer: query successfully made on the server.
wrongFirstRow	Negative answer: bad <code>firstRow</code> parameter.
wrongRowNumber	Negative answer: bad <code>rowNumber</code> parameter.
wrongQueryID	Negative answer: an internal error occurred.
genericError	Negative answer: other (unspecified) error.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onRows](#)

QueryType Enumeration

Subscription query—selection criteria.

public enum QueryType

Remarks

See [Partial Subscriptions](#) to understand how to use this value with partial subscriptions.

Members

Member Name	Description
all	<p>Request notifications for all entities.</p> <ul style="list-style-type: none"> • At first: many past entities values notifyEvent, • and then one single idleEvent, • and then many current entities values notifyEvent.
set	<p>Request notifications for entities that match a partial EntityKey.</p> <ul style="list-style-type: none"> • At first: many past entities values notifyEvent, • and then one single idleEvent, • and then many current entities values notifyEvent.
past	<p>Request notifications for only past entities values.</p> <ul style="list-style-type: none"> • At first: many past entities values notifyEvent, • and then one single idleEvent. <p>Please note that with this value the Subscription will be automatically disposed when the idle event will be received. This will always happen, whichever value the autoDisposeOnIdle property may have.</p>
onTime	<p>Request notifications for only current entities values.</p> <ul style="list-style-type: none"> • At first: many current entities values notifyEvent.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [Subscription.open](#) | [Partial Subscriptions](#)

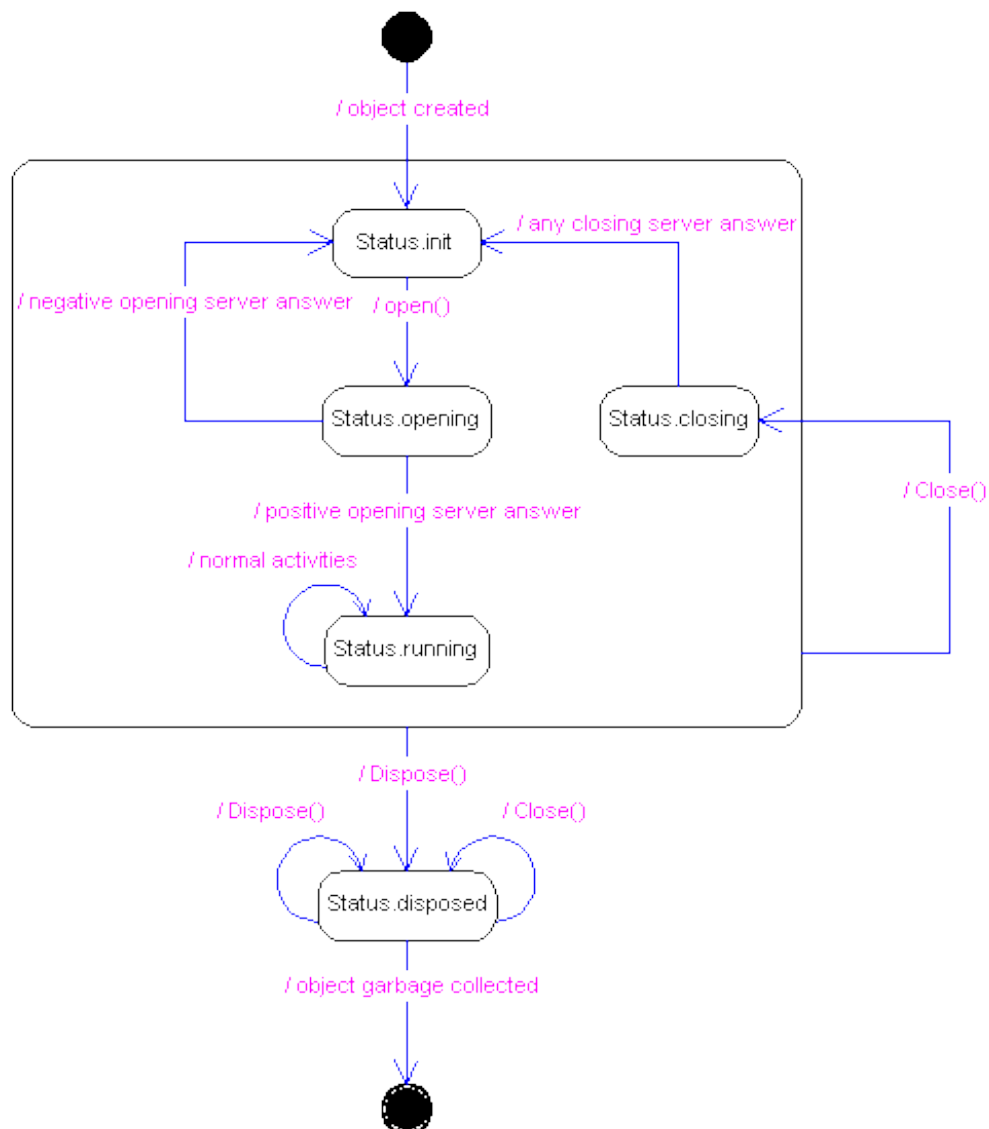
Status Enumeration

Status of every [Disposable](#) object.

```
public enum Status
```

Remarks

The lifecycle of every Disposable object is dipinct by the following state diagram.



Members

Member Name	Description
init	Initial status.
	Entry undefined ? object creation ? init

		opening ? negative opening server answer ? init ? onOpen/openEvent fired
		closing ? any closing server answer ? init ? onClose/closeEvent fired
	Activities	access to object properties
	Exit	init ? open methods ? opening any status ? Close ? closing any status ? Dispose ? disposed
opening	Waiting a server answer after an opening request.	
	Entry	init ? open methods ? opening
	Activities	access to object properties
	Exit	opening ? positive opening server answer ? running ? onOpen/openEvent fired opening ? negative opening server answer ? init ? onOpen/openEvent fired any status ? Close ? closing any status ? Dispose ? disposed
running	Normal running status.	
	Entry	opening ? positive opening server answer ? running ? onOpen/openEvent fired
	Activities	all object normal activities
	Exit	any status ? Close ? closing any status ? Dispose ? disposed
closing	<i>Please note</i> that all normal (i.e. not onClose/closeEvent nor some onOpen/openEvent) event handlers are invoked in this state.	
	Waiting a server answer after a closing request.	
	Entry	any status ? Close ? closing
	Activities	access to object properties
disposed	Exit	closing ? any closing server answer ? init ? onClose/closeEvent fired any status ? Dispose ? disposed
	Final status.	
	Entry	any status ? Dispose ? disposed

	Activities	access to object properties
	Exit	disposed ? object garbage collected ? undefined
	<i>Please note</i> that no event handlers of any sort are invoked in this state.	

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [Disposable](#)

SubscribeFlow Enumeration

Subscription data–transmission policy.

public enum SubscribeFlow

Members

Member Name	Description
last	Only the most recent snapshot of each entity sent by the server.
all	All variations sent by the server.
lastZeroMasked	<p>Only the most recent snapshot of each entity sent by the server.</p> <p>The transmission is optimized in a transparent way in a manner that only non–zero values are sent. <i>Please note</i> that this is only an optimization hint and so it does not change the behaviour of isMasked.</p>
allZeroMasked	<p>All variations sent by the server.</p> <p>The transmission is optimized in a transparent way in a manner that only non–zero values are sent. <i>Please note</i> that this is only an optimization hint and so it does not change the behaviour of isMasked.</p>

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [Subscription.open](#)

ThreadingMode Enumeration

Library threading behaviour.

```
public enum ThreadingMode
```

Remarks

These values control the behaviour of `Lib.start`.

For a full treatment of this subject please see [Implementation Threads](#)

Members

Member Name	Description
singleThread	Library execute in the same caller's thread. Library access by other threads is not safe and <i>must not</i> be made. With this value the <code>Lib.Select</code> and <code>Lib.step</code> methods must be periodically invoked in order to guarantee a proper efficacy.
singleThreadSafe	Library execute in the same caller's thread. Library access by other threads is safe and <i>may</i> be made. With this value the <code>Lib.Select</code> and <code>Lib.step</code> methods must be periodically invoked in order to guarantee a proper efficacy.
multipleThreads	Library execute in an own thread (created by the <code>Lib.start</code> method invocation). Library access by other threads are safe and <i>may</i> be made. This is the normal library behaviour where there is no need to invoke <code>Lib.step</code> or <code>Lib.Select</code> .

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [Lib.open](#) | [Implementation Threads](#)

TraceLevel Enumeration

Trace message level.

public enum TraceLevel

Remarks

This enum may be used in

- [traceLevel](#) field of [Lib.TraceEventArgs](#) to see the level of message being traced,
- [setTraceMode](#) method to define the maximum level to trace,
- [trace](#) method to define the level of a specific application message.

Members

Member Name	Description
undefined	Currently unused.
full	Verbose level.
normal	Normal level.
warning	Warning level.
error	Potentially discovered errors or malfunctions are traced.
	Error level.
	Discovered errors or malfunctions are traced.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [Lib.TraceEventHandler](#)

TraceOptions Enumeration

Trace message options.

This enumeration has a [FlagsAttribute](#) attribute that allows a bitwise combination of its member values.

```
[Flags]  
public enum TraceOptions
```

Remarks

This enum may be used in the [setTraceMode](#) method to define the mask (logical OR) of the requested trace options.

Members

Member Name	Description	Value
none	No option.	0
ftapiInOut	Every call and/or result will be traced.	1
flowControl	The application Flow Control will be traced.	2
dataTransfer	The application communication will be traced.	4
dataIn	The incoming records (subscription data) will be dumped.	8
dataOut	The sent record (transaction data) will be dumped.	16
application	Messages given to trace will be traced.	32
ftapi_dnet	.NET FastTrack API info will be traced.	65536
ftapi_dnetInOut	All .NET FastTrack API methods invocations and related parameters will be traced.	131072
ftapi_dnetEventArgs	All .NET FastTrack API event-handlers and related parameters will be traced.	262144

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [setTraceMode](#)

TraceSource Enumeration

Trace message category.

public enum TraceSource

Remarks

This enum may be inspected in the [traceSource](#) field of [Lib.TraceEventArgs](#) to see the category of message being traced.

Members

Member Name	Description
ftapi_dnet	.NET FastTrack Api.
ftProtocol	FastTrack protocol.
ftapi	Low level FastTrack Api.
flowControl	Flow control.
dataTransf	Data transfers. <ul style="list-style-type: none">• in – subscription, query• out – transaction
application	Application. Messages generated by trace method invocation.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [Lib.TraceEventHandler](#)

TransactionNamStatus Enumeration

Status of a transaction sent to a NAM market.

`public enum TransactionNamStatus`

Remarks

This enum may be inspected in the [status](#) field of [Connection.MonitorEventArgs](#) as an answer to a [monitorNamTransactions](#) request to see the status of a transaction sent to a NAM market.

Members

Member Name	Description
flying	Transaction in progress.
committed	Transaction successfully completed.
aborted	Transaction aborted locally. In this case the localError field of Connection.MonitorEventArgs may be inspected in order to see the reason.
Haborted	Transaction aborted on the server. In this case the namError and marketError fields of Connection.MonitorEventArgs may be inspected in order to see the reason.
notManaged	Transaction cannot be monitored. This value may happen only for particular market classes.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onMonitor](#)

TransactionStatus Enumeration

Status of a transaction sent to a server.

public enum TransactionStatus

Remarks

This enum may be inspected in the [transactionStatus](#) field of [Transaction.TransactionEventArgs](#) to see the status of a transaction sent to a server.

Please note that for transactions sent to a NAM market these values (in particular the [OK](#) value) does **not represent** the final NAM market outcomes but only the intermediate server result: in this case the final outcomes results are only available through the [TransactionNamStatus](#) values during a [monitorNamTransactions](#).

Members

Member Name	Description
OK	<p>Transaction accepted.</p> <ul style="list-style-type: none"> • for NAM market – The transaction is still flying and the application have to make an explicit monitoring (via monitorNamTransactions) to see if the transaction is still flying or it's aborted or committed. • for FastTrack market/service – The transaction is successfully completed: in this case the application do not need to make any explicit queryStatus invocation.
NO	<p>Transaction aborted.</p> <p>In this case the marketReasonCode and entity fields of Transaction.TransactionEventArgs may be used to understand why the market aborted the transaction.</p>
invalidSessionID	<p>Transaction aborted.</p> <p>An internal error occurred.</p>
invalidTID	<p>Transaction aborted.</p> <p>An invalid TransactionID was used.</p>
pending	<p>Transaction flying.</p> <ul style="list-style-type: none"> • for NAM market – This value cannot never occur. • for FastTrack market/service – This value may occur only for very few FastTrack market/services immediatly answering this flying status when they receive a send request.

	<p>Almost every FastTrack market/service attempts to complete the transaction request before emitting an OK or NO answer.</p> <p>When this value is received, an application may subsequently use the queryStatus method to re-query the final transaction status.</p>
conversationClosed	Transaction answer lost due to a connection close.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [onSend](#) | [onQuery](#)

UserType Enumeration

Activities that can be made on a [Connection](#).

`public enum UserType`

Remarks

This enum may be used in the [open](#) method of a [Connection](#) to specify the activities that can be made on that connection. The server may subsequently returns a [invalidUserType](#) failure-code if it does not comply with this value.

Members

Member Name	Description
trader	To create a connection on which send data variations (transaction). view and trader are the most commonly used values.
autoTrader	To create a connection on which send data variations (transaction).
monitor	To create a connection on which request data (subscriptions and/or queries).
view	To create a connection on which request data (subscriptions and queries). view and trader are the most commonly used values.
controller	To create a specialized (no more here described) connection.
masterSlave	To create a specialized (no more here described) connection.

Requirements

Namespace: [ftapi_dnet](#)

Assembly: ftapi_dnet (in ftapi_dnet.dll)

See Also

[ftapi_dnet Namespace](#) | [Connection.open](#)

