ADO Driver documentation

The core driver name is InterBaseSQL.Data.InterBaseClient.dll. This needs to be added to your project in the references section. If for some reason the client is not registered using GACUTIL (<https://docs.microsoft.com/en-us/dotnet/framework/app-domains/install-assembly-into-gac>) then you can manually locate it with Add Reference | Browse.

The ADO.NET driver does require that the InterBase client be installed on the client machine.

**Common Components**

**InterBaseSql.Data.Client.Native**

IBClientFactory – This gets an instance to the client library if you need direct access to the API.

GetGDSLibrary – This is a static public method to get access to an instance of either the server or embedded InterBase client. Platform/bitness is determined for you. It returns an interface of IIBClient defined an IIBClient.cs. The methods are documented in the InterBaseAPI pdf.

Use – var ibclient = IBClientFactory.GetGDSLibrary(IBServerType.Default);

**InterBaseSql.Data.InterBaseClient**

IBCommand – This is the DBCommand descendant. This represents an SQL statement or stored procedure to execute against a data source. <https://docs.microsoft.com/en-us/dotnet/api/system.data.common.dbcommand?view=net-5.0>

IBCommandBuilder – DbCommandBuilder descendant. This automatically generates single-table commands used to reconcile changes made to a [DataSet](https://docs.microsoft.com/en-us/dotnet/api/system.data.dataset?view=net-5.0) with the associated database. <https://docs.microsoft.com/en-us/dotnet/api/system.data.common.dbcommandbuilder?view=net-5.0>

IBConnection – DbConnection descendant. Defines the core behavior of database connections. Works from a connection string (see below). <https://docs.microsoft.com/en-us/dotnet/api/system.data.common.dbconnection?view=net-5.0>

IBConnectionStringBuilder – DbConnectionStringBuilder decendant. This can be used to create a connection string from the individual parts. <https://docs.microsoft.com/en-us/dotnet/api/system.data.common.dbconnectionstringbuilder?view=net-5.0>

IBDataAdapter – DbDataAdapter descendant. This implements a set of functions to provide strong typing, but inherit most of the functionality needed to fully implement a DataAdapter. <https://docs.microsoft.com/en-us/dotnet/api/system.data.common.dbdataadapter?view=net-5.0>

IBDatabaseInfo – Class that given a connection allows for retrieving information on that database.

Properties

IBConnection – The connection component to use to query against.

IscVersion

PageSize

AllocationPages

BaseLevel

DbId

Implementation

NoReserve

OdsVersion

OdsMinorVersion

MaxMemory

CurrentMemory

ForcedWrites

NumBuffers

SweepInterval

ReadOnly

Fetches

Marks

Reads

Writes

BackoutCount

DeleteCount

ExpungeCount

InsertCount

PurgeCount

ReadIdxCount

ReadSeqCount

UpdateCount

DatabaseSizeInPages

ActiveUsers

IBDataReader - DbDataReader descendant. This is a forward only stream of rows from a data source. <https://docs.microsoft.com/en-us/dotnet/api/system.data.common.dbdatareader?view=net-5.0>

IBEnlistmentNotification - IEnlistmentNotification implementation. <https://docs.microsoft.com/en-us/dotnet/api/system.transactions.ienlistmentnotification?view=net-5.0>

IBTransaction – DbTransaction descendant. Transaction component for transaction control. <https://docs.microsoft.com/en-us/dotnet/api/system.data.common.dbtransaction?view=net-5.0>

InterBaseClientFactory – DbProviderFactory descendant. This represents a set of methods for creating instances of a provider's implementation of the data source classes. <https://docs.microsoft.com/en-us/dotnet/api/system.data.common.dbproviderfactory?view=net-5.0>

**Connection Strings** –

The following describes the keys and synonyms for the Connection strings to attach to an InterBase database.

|  |  |  |
| --- | --- | --- |
| Primary name | Alternatives | Description |
| Datasource | Data Source, server, host | This is the IP address or the Host name for the location of the server. |
| Port | Port Number | The port the IB server is listening on. Default 3050. |
| Database | Initial Catalog | Path to the database or the database alias. |
| User | Userid, uid, user name, username, user id | The user to log in as. |
| Password | User password, userpassword | Password to use logging in. |
| Dialect |  | Dialect to force the connection to be in. Default 3. |
| Role |  | The role to log in as. |
| Charset | Character Set | The character set to establish the connection with. Default is none (ANSII) |
| Packetsize | Packet Size | Packet size to use when communicating with the server. Default 8192 |
| Pooling |  | When true the connection is grabbed from a pool or, if necessary, created and added to the appropriate pool. Default true. |
| Connectionlifetime | Connection Lifetime | When a connection is returned to the pool, its creation time is compared with the current time, and the connection is destroyed if that time span (in seconds) exceeds the value specified by connection lifetime. Default 0 |
| Timeout | Connectiontimeout, connection timeout | The time (in seconds) to wait for a connection to open. Default 15. |
| Min pool size |  | The minimum number of connections allowed in the pool. Default 0 |
| Max pool size |  | The maximum number of connections allowed in the pool. Default 100 |
| Festchsize | Fetch size | The maximum number of rows to be fetched in a single call to read into the internal row buffer. Default 200 |
| Servertype | Source | The type of server used. Default IBServerType.Default (Server). Other option is IBServerType.Embedded |
| Isolationlevel | Isolation level | The default Isolation Level for implicit transactions. Default ReadCommitted |
| Records Affected |  | Get the number of rows affected by a command when true. Default true. Note Select statements cannot give an accurate number until all records fetched. |
| Enlist |  | If true, enlists the connections in the current transaction. Default true |
| embedded | clientEmbedded | If true indicates to act as if the Server Type is Embedded. Default false |
| Cachepages | Pagebuffers, page buffers, cache pages | How many cache buffers to use for this session. Default 0 (server side determined) |
|  |  |  |
|  |  |  |

Simple example connecting to the employee database on the local server –

server=localhost;database=c:\embarcadero\InterBase64\Examples\Database\employee.gdb;user=sysdba;password=masterkey

Entity Framework

Some of the implementation of the EntityFramework requires the EntityUDF. The script for this UDF is included as well as the dlls for the 32/64 bit platforms supported. UDF is documented below.

To Do – Take one of the examples and basically go through how to setup a DBContext etc.

Entity UDF

This is a UDF designed to add some missing internal functions. Since not all were in the shipping IB\_UDF library it was descided to have a sefl contained library. These can be used just fine outside the EntityFramework too. Place the corrct bit (32 or 64) dll into the <InterBase>\UDF directory then run the entityUDF.sql script against your Database to use.

Math functions

EF\_Abs –

Input Double

Output Double

Returns the absolute value of the input.

EF\_Ceiling –

Input Double

Output Numeric(18, 0)

Rounds up towards positive infinity.

EF\_Floor –

Input Double

Output Numeric(18, 0)

Rounds down towards negative infinity.

EF\_Round –

Input num Double

scale integer

Output Double

Rounds a number to the nearest integer. If the fractional part is exactly 0.5, rounding is upward for positive numbers and downward for negative numbers. With the optional scale argument, the number can be rounded to powers-of-ten multiples (tens, hundreds, tenths, hundredths, etc.) instead of just integers.

Examples

* EF\_ROUND(123.654, 1) returns 123.7
* EF\_ROUND(8341.7, -3) returns 8000.0
* EF\_ROUND(45.1212, 0) returns 45.0

EF\_Power –

Input x Double

y Double

Ouput Double.

Returns x to the y'th power.

EF\_Truncate –

Input num Double

scale Double

Output Double.

Returns the integer part of a number. With the optional scale argument, the number can be truncated to powers-of-ten multiples (tens, hundreds, tenths, hundredths, etc.) instead of just integers. Examples

* TRUNC(789.2225, 2) returns 789.22
* TRUNC(345.4, -2) returns 300.0
* TRUNC(-163.41, 0) returns -163.0

EF\_BitAnd –

Input x Numeric(18,0)

y Numeric(18, 0)

Ouput Numeric(18, 0).

Returns the result of the bitwise AND operation on the arguments.

EF\_BitOr –

Input x Numeric(18,0)

y Numeric(18, 0)

Ouput Numeric(18, 0).

Returns the result of the bitwise OR operation on the arguments.

EF\_BitXor –

Input x Numeric(18,0)

y Numeric(18, 0)

Ouput Numeric(18, 0).

Returns the result of the bitwise Xor operation on the arguments.

EF\_BitNot –

Input x Numeric(18,0)

Ouput Numeric(18, 0).

Returns the result of the bitwise Not operation on the argument.

String Functions – All string functions have in their SQL definition sizes of 2048. If you need larger or smaller you can always change the definitions up to 32K before running the script.

Also note that because UDF string types are CString (basically a CHAR datatype) they will always return right padded with spaces, but it you cast the result to a VarChar(2048) you will get the unpadded results.

Example:

Select EF\_Reverse(‘ abc ’) from RDB$DATABASE

Returns ‘ cba ‘ (padded out to 2048)

Select Cast(EF\_Reverse(‘ abc ’) as VarChar(2048)) from RDB$DATABASE

Returns ‘ cba ‘

EF\_Reverse –

Input CSTRING(2048)

Output CSTRING(2048).

Returns a string backwards.

Examples

1. EF\_Reverse ('spoonful') -- returns 'lufnoops'
2. EF\_Reverse ('Was it a cat I saw?') -- returns '?was I tac a ti saW'

EF\_Position –

Input substr CSTRING(2048)

str CSTRING(2048)

startIdx integer

Output Integer.

Returns the (1-based) position of the first occurrence of a substring in a host string. With the optional third argument, the search starts at a given offset, disregarding any matches that may occur earlier in the string. If no match is found, the result is 0.

Examples

1. EF\_position ('be', 'To be or not to be', 1) -- returns 4
2. EF\_position ('be', 'To be or not to be', 4) -- returns 4
3. EF\_position ('be', 'To be or not to be', 8) -- returns 17
4. EF\_position ('be', 'To be or not to be', 18) -- returns 0

EF\_Length –

Input CSTRING(2048)

output Integer.

Returns the length of the character string.

EF\_Lower -

Input cstring(2048)

returns cstring(2048).

Returns the ANSI lower case for the string. It should respect things like Ä and return ä. Note the internal Upper function does not do this.

EF\_Trim –

Input type CSTRING(8)

str CSTRING(2048)

Output CString(2048).

Input Types can be ‘BOTH’ or ‘LEADING’ or ‘TRAILING’ (case insensitive). Trims spaces from either before, after or both of the inputed string. Note that the after side will only work if you cast the result to a VarChar as it will get padded by the Server even though properly trimmed.

EF\_Left –

Input Str CSTRING(2048)

len Integer

Output CSTRING(2048).

Returns the leftmost part of the argument string. The number of characters is given in the second argument.

EF\_Right –

Input Str CSTRING(2048)

len Integer

Output CSTRING(2048).

Returns the rightmost part of the argument string. The number of characters is given in the second argument.

EF\_Replace –

Input str CSTRING(2048)

Find CSTRING(2048)

Repl CSTRING(2048)

Output CSTRING(2048)

Replaces all occurrences of a substring in a string.

Example

* EF\_replace ('Billy Wilder', 'il', 'oog') -- returns 'Boogly Woogder'
* EF\_replace ('Billy Wilder', 'il', '') -- returns 'Bly Wder'
* EF\_replace ('Billy Wilder', 'xyz', 'abc') -- returns 'Billy Wilder'
* EF\_replace ('Billy Wilder', '', 'abc') -- returns 'Billy Wilder'

EF\_SubStr –

Input str cstring(2048),

startIdx integer,

length integer

Ouput cstring(2048)

Returns the substring starting at the StartIdx and going to length # of characters.

NewGuid

Ouput CSTRING(16) character set OCTETS

Returns a 16 byte GUID Array. Should go into an OCTET variable

Time Functions

EF\_UTCCurrentTime

Ouput timestamp

Returns the current UTC time.

EF\_DateAdd

Input unit cstring(7)

amount numeric(18, 0)

toDate timestamp

output timestamp

Returns the date added by the type passed. Valid units are - 'YEAR' , 'MONTH' , 'WEEK', 'DAY', 'HOUR', 'MINUTE', 'SECOND'.

Examples

* EF\_dateadd (‘day’, 28, current\_date)
* EF\_dateadd (‘hour’, -6, current\_time)
* EF\_dateadd (‘month’, 9, DateOfConception)
* EF\_dateadd (‘minute’, 90, 'now')

EF\_DateDiff –

Input units cstring(12)

from timestamp

to timestamp

Output numeric(18,0)

Returns the number of years, months, days, hours, minutes, seconds elapsed between two date/time values. Valid units are - 'YEAR' , 'MONTH' , 'WEEK', 'DAY', 'HOUR', 'MINUTE', 'SECOND'

DDEX

Installation instructions for the DDEX driver.

This section will be cobbled together from these resources mainly/

<https://www.visualstudioextensibility.com/2016/11/23/some-implications-of-the-new-modular-setup-of-visual-studio-2017-for-vsx-developers/>

<https://visualstudioextensions.vlasovstudio.com/2017/06/29/changing-visual-studio-2017-private-registry-settings/>

<https://www.tabsoverspaces.com/233604-installing-ddex-provider-for-firebird-into-visual-studio-2017>