ODBC Client Driver

Table of Contents

ODBC Client Driver

Help version 1.060

CONTENTS

Overview

The ODBC Client Driver provides a reliable way to connect ODBC data sources to OPC Client applications, including HMI, SCADA, Historian, MES, ERP, and countless custom applications. It provides single-point access to multiple ODBC data sources via ODBC. Through the power and speed of OPC, the server can offer ODBC data to any OPC/DDE enabled client.

ODBC Client Driver Features

- Multiple connection sessions to multiple ODBC data sources
- Read / Write to the SQL engine
- Automatic tag generation based on tables, queries, and stored procedures from the ODBC data source
- OPC DA
- Drag and drop editing
- CSV import and export for easy manipulation of tags
- Data Scaling
- System Tags for general error feedback to client applications
- Tags that can be added to the application at any time (even at Runtime)
- The ability to run as a System Service
- A built-in User Manager for complete control over individual user functionality and privileges

CAUTION: The ODBC Client Driver is a direct SQL client to the data source server, allowing the driver to perform all the common SQL CRUD operations (create, update, and delete). It is the database administrator's responsibility to limit access to the ODBC Client Driver to ensure data is properly secured from undesirable SQL operations through the least-privileges paradigm when an account for the driver to authenticate is being configured on the data source.

External Dependencies

This driver has external dependencies. It requires Microsoft Data Access Components (MDAC), which consist of several core components that provide various database technologies (including ODBC and its drivers).

Setup

Supported ODBC Data Sources

ODBC-compliant data sources are supported. Examples include:

- Microsoft Access
- Microsoft SQL
- MySQL*
- Oracle

*If the project uses a MySQL data source, users must select **MySQL** from the Model drop-down list when creating a device. This makes the driver accommodate MySQL's data handling requirements. The PC that is running the OPC server must have the MyODBC driver installed, as well.

⚙**CAUTION**: ODBC does not encrypt data for transmission. It is the network administrator's responsibility to provide security for the database traffic. This may be achieved through securing the network layer, such as use of a VPN or other secure tunneling protocol between the driver and the data source server.

Recordsets

The MySQL model uses Snapshot-type recordsets; all other ODBC data sources use Dynaset.

Communication Protocol

ODBC API

Channel and Device Limits

The maximum number of channels supported by this driver is 256. The maximum number of devices supported by this driver is 4096 per channel.

⚫**Note**: In this driver, the channel corresponds to the ODBC Driver and its particular ODBC data source.

Channel Properties — General

This server supports the use of multiple simultaneous communications drivers. Each protocol or driver used in a server project is called a channel. A server project may consist of many channels with the same communications driver or with unique communications drivers. A channel acts as the basic building block of an OPC link. This group is used to specify general channel properties, such as the identification attributes and operating mode.

Property Groups	Identification	
General	Name	
Write Optimizations	Description	
Advanced	Driver	
	Diagnostics	
	Diagnostics Capture	Disable
	Tag Counts	
	Static Tags	10

Identification

Name: Specify the user-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information. The property is required for creating a channel.

🔷 *For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.*

Description: Specify user-defined information about this channel.

🟢 Many of these properties, including Description, have an associated system tag.

Driver: Specify the protocol / driver for this channel. Specify the device driver that was selected during channel creation. It is a disabled setting in the channel properties. The property is required for creating a channel.

⚪ **Note**: With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. Changes to the properties should not be made once a large client application has been developed. Utilize proper user role and privilege management to prevent operators from changing properties or accessing server features.

Diagnostics

Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications allows the usage of statistics tags that provide feedback to client applications regarding the operation of the channel. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

⚪ **Note:** This property is not available if the driver does not support diagnostics.

🔷 *For more information, refer to "Communication Diagnostics" and "Statistics Tags" in the server help.*

Tag Counts

Static Tags: Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

Channel Properties — Write Optimizations

The server must ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties to meet specific needs or improve application responsiveness.

Property Groups	⊟ Write Optimizations	
General	Optimization Method	Write Only Latest Value for All Tags
Write Optimizations	Duty Cycle	10

Write Optimizations

Optimization Method: Controls how write data is passed to the underlying communications driver. The options are:

- **Write All Values for All Tags**: This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.

- **Write Only Latest Value for Non-Boolean Tags**: Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.

  ⬤ **Note**: This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.

- **Write Only Latest Value for All Tags**: This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

Duty Cycle: is used to control the ratio of write to read operations. The ratio is always based on one read for every one to ten writes. The duty cycle is set to ten by default, meaning that ten writes occur for each read operation. Although the application is performing a large number of continuous writes, it must be ensured that read data is still given time to process. A setting of one results in one read operation for every write operation. If there are no write operations to perform, reads are processed continuously. This allows optimization for applications with continuous writes versus a more balanced back and forth data flow.

⬤ **Note**: It is recommended that the application be characterized for compatibility with the write optimization enhancements before being used in a production environment.

Channel Properties — Advanced

This group is used to specify advanced channel properties. Not all drivers support all properties; so the Advanced group does not appear for those devices.

Property Groups	Non-Normalized Float Handling	
General	Floating-Point Values	Replace with Zero
Write Optimizations	Inter-Device Delay	
Advanced	Inter-Device Delay (ms)	0

Non-Normalized Float Handling: A non-normalized value is defined as Infinity, Not-a-Number (NaN), or as a Denormalized Number. The default is Replace with Zero. Drivers that have native float handling may default to Unmodified. Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. Descriptions of the options are as follows:

- **Replace with Zero**: This option allows a driver to replace non-normalized IEEE-754 floating point values with zero before being transferred to clients.
- **Unmodified**: This option allows a driver to transfer IEEE-754 denormalized, normalized, non-number, and infinity values to clients without any conversion or changes.

 Note: This property is disabled if the driver does not support floating-point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM data is not affected by this setting.

 For more information on the floating-point values, refer to "How To ... Work with Non-Normalized Floating-Point Values" in the server help.

Inter-Device Delay: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

 Note: This property is not available for all drivers, models, and dependent settings.

Channel Properties — Data Source Settings

The Data Source Settings group is used to select a data source and configure connection specific properties.

Property Groups	Data Source Settings	
General	Data Source	Excel Files
Write Optimizations	Username	
Advanced	Password	········
Data Source Settings	Login Timeout (s)	5

Data Source: This property specifies the Data Source Name (DSN). Click on the browse button to launch the Data Source Selection dialog. Select a DSN from the menu. The DSN must be created and configured for the data source before it will appear in the menu. Click on the **Configure DSN...**button to launch the **Microsoft® ODBC Data Source Administrator**.

Username: This property specifies the user name. It is only needed if required by the data source (such as in SQL server authentication).

Password: This property specifies the password. It is only needed if required by the data source (such as in SQL server authentication).

Login Timeout (s): This property specifies the number of seconds that the server will wait for a response before timing out when attempting to connect to the ODBC data source. The valid range is 1 to 99999 seconds. The default setting is 5 seconds.

 Note: Some data sources can use Windows Authentication to gain access to the data source. If the data source is configured to use Windows Authentication, it will ignore the User Name and Password and will use the Network Login ID instead.

Configuring a DSN

The example below uses System DSNs, which are local to the computer but not user-dedicated. This means that they can be accessed by any user with privileges. User DSNs, which are also local to the computer, are only accessible by the current user.

🔶 **Caution:** System DSNs must be used when the OPC server is configured to run as a System Service; otherwise, tags will neither be automatically generated nor be Good quality in projects in which they are already contained.

1. To start, click **Configure DSN** to invoke the **Microsoft® ODBC Data Source Administrator**.

2. Then, map an ODBC data source to a data source name.

⬤ **Note**: Once a DSN has been created successfully, it will appear in the Data Source drop-down menu.

Device Access Method	Properties Requiring Update When DSN Changes
Fixed Table	Table Selection Limit Import Columns Maximum Records to Import
Dynamic Table	Table Selection
Query	Query Trigger Column
Stored Procedure	Stored Procedure Selection

Data Source Authentication

Windows Authentication

Some data sources may be configured to use authentication for security. The SQL Server can authenticate with Windows by default using the Network Login ID. This makes the SQL server login transparent, and eliminates the need for a user name and password in the channel's Data Source Settings property group.

SQL Server Authentication

The SQL server also has its own authentication method. To connect the OPC server to the SQL server using SQL authentication, users must set the authentication method in the SQL server to allow SQL authentication. Users must also specify the Login ID and Password. After these properties are defined, the user name and password may then be entered in the OPC server in the channel's Data Source Settings tab. F

🔶 **See Also**: **Running as a System Service**

Device Properties — General

A device represents a single target on a communications channel. If the driver supports multiple controllers, users must enter a device ID for each controller.

Property Groups	Identification	
General	Name	
Scan Mode	Description	
	Channel Assignment	
	Driver	
	Model	
	ID Format	Decimal
	ID	2

Identification

Name: Specify the name of the device. It is a logical user-defined name that can be up to 256 characters long and may be used on multiple channels.

 Note: Although descriptive names are generally a good idea, some OPC client applications may have a limited display window when browsing the OPC server's tag space. The device name and channel name become part of the browse tree information as well. Within an OPC client, the combination of channel name and device name would appear as "ChannelName.DeviceName".

 For more information, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in server help.

Description: Specify the user-defined information about this device.

 Many of these properties, including Description, have an associated system tag.

Channel Assignment: Specify the user-defined name of the channel to which this device currently belongs.

Driver: Selected protocol driver for this device.

Model: Specify the type of device that is associated with this ID. The contents of the drop-down menu depend on the type of communications driver being used. Models that are not supported by a driver are disabled. If the communications driver supports multiple device models, the model selection can only be changed when there are no client applications connected to the device.

 Note: If the communication driver supports multiple models, users should try to match the model selection to the physical device. If the device is not represented in the drop-down menu, select a model that conforms closest to the target device. Some drivers support a model selection called "Open," which allows users to communicate without knowing the specific details of the target device. For more information, refer to the driver help documentation.

ID: Specify the device's driver-specific station or node. The type of ID entered depends on the communications driver being used. For many communication drivers, the ID is a numeric value. Drivers that support a Numeric ID provide users with the option to enter a numeric value whose format can be changed to suit the needs of the application or the characteristics of the selected communications driver. The format is set by the driver by default. Options include Decimal, Octal, and Hexadecimal.

 Note: If the driver is Ethernet-based or supports an unconventional station or node name, the device's TCP/IP address may be used as the device ID. TCP/IP addresses consist of four values that are separated by periods, with each value in the range of 0 to 255. Some device IDs are string based. There may be additional properties to configure within the ID field, depending on the driver.

Operating Mode

Data Collection: This property controls the device's active state. Although device communications are enabled by default, this property can be used to disable a physical device. Communications are not attempted when a device is disabled. From a client standpoint, the data is marked as invalid and write operations are not accepted. This property can be changed at any time through this property or the device system tags.

Simulated: Place the device into or out of Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

⬤ **Notes**:

1. This System tag (_Simulated) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.

2. When a device is simulated, updates may not appear faster than one (1) second client.

🔶 Simulation Mode is for test and simulation purposes only. It should never be used in a production environment.

Tag Counts

Static Tags: Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

Device Properties — Scan Mode

The Scan Mode specifies the subscribed-client requested scan rate for tags that require device communications. Synchronous and asynchronous device reads and writes are processed as soon as possible; unaffected by the Scan Mode properties.

Scan Mode: Specify how tags in the device are scanned for updates sent to subscribing clients. Descriptions of the options are:

- **Respect Client-Specified Scan Rate**: This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate**: This mode specifies the value set as the maximum scan rate. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
 Note: When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate**: This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
- **Do Not Scan, Demand Poll Only**: This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the OPC client's responsibility to poll for updates, either by writing to the _DemandPoll tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help.*
- **Respect Tag-Specified Scan Rate**: This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

Initial Updates from Cache: When enabled, this option allows the server to provide the first updates for newly activated tag references from stored (cached) data. Cache updates can only be provided when the new item reference shares the same address, scan rate, data type, client access, and scaling properties. A device read is used for the initial update for the first client reference only. The default is disabled; any time a client activates a tag reference the server attempts to read the initial value from the device.

Device Properties — Tag Generation

The automatic tag database generation features make setting up an application a plug-and-play operation. Select communications drivers can be configured to automatically build a list of tags that correspond to device-specific data. These automatically generated tags (which depend on the nature of the supporting driver) can be browsed from the clients.

 Not all devices and drivers support full automatic tag database generation and not all support the same data types. Consult the data types descriptions or the supported data type lists for each driver for specifics.

If the target device supports its own local tag database, the driver reads the device's tag information and uses the data to generate tags within the server. If the device does not natively support named tags, the driver creates a list of tags based on driver-specific information. An example of these two conditions is as follows:

1. If a data acquisition system supports its own local tag database, the communications driver uses the tag names found in the device to build the server's tags.

2. If an Ethernet I/O system supports detection of its own available I/O module types, the communications driver automatically generates tags in the server that are based on the types of I/O modules plugged into the Ethernet I/O rack.

 Note: Automatic tag database generation's mode of operation is completely configurable. *For more information, refer to the property descriptions below.*

Tag Generation	
On Device Startup	Do Not Generate on Startup
On Duplicate Tag	Delete on Create
Parent Group	
Allow Automatically Generated Subgroups	Enable
Create	Create tags

Property Groups
General
Scan Mode
Timing
Auto-Demotion
Tag Generation
Communications
Redundancy

On Property Change: If the device supports automatic tag generation when certain properties change, the **On Property Change** option is shown. It is set to **Yes** by default, but it can be set to **No** to control over when tag generation is performed. In this case, the **Create tags** action must be manually invoked to perform tag generation. To invoke via the Configuration API service, access /config/v1/project/channels/{name}/devices/{name}/services/TagGeneration.

On Device Startup: Specify when OPC tags are automatically generated. Descriptions of the options are as follows:

- **Do Not Generate on Startup**: This option prevents the driver from adding any OPC tags to the tag space of the server. This is the default setting.
- **Always Generate on Startup**: This option causes the driver to evaluate the device for tag information. It also adds tags to the tag space of the server every time the server is launched.
- **Generate on First Startup**: This option causes the driver to evaluate the target device for tag information the first time the project is run. It also adds any OPC tags to the server tag space as needed.

> **Note**: When the option to automatically generate OPC tags is selected, any tags that are added to the server's tag space must be saved with the project. Users can configure the project to automatically save from the **Tools | Options** menu.

On Duplicate Tag: When automatic tag database generation is enabled, the server needs to know what to do with the tags that it may have previously added or with tags that have been added or modified after the communications driver since their original creation. This setting controls how the server handles OPC tags that were automatically generated and currently exist in the project. It also prevents automatically generated tags from accumulating in the server.

For example, if a user changes the I/O modules in the rack with the server configured to **Always Generate on Startup**, new tags would be added to the server every time the communications driver detected a new I/O module. If the old tags were not removed, many unused tags could accumulate in the server's tag space. The options are:

- **Delete on Create**: This option deletes any tags that were previously added to the tag space before any new tags are added. This is the default setting.
- **Overwrite as Necessary**: This option instructs the server to only remove the tags that the communications driver is replacing with new tags. Any tags that are not being overwritten remain in the server's tag space.
- **Do not Overwrite**: This option prevents the server from removing any tags that were previously generated or already existed in the server. The communications driver can only add tags that are completely new.
- **Do not Overwrite, Log Error**: This option has the same effect as the prior option, and also posts an error message to the server's Event Log when a tag overwrite would have occurred.

Note: Removing OPC tags affects tags that have been automatically generated by the communications driver as well as any tags that have been added using names that match generated tags. Users should avoid adding tags to the server using names that may match tags that are automatically generated by the driver.

Parent Group: This property keeps automatically generated tags from mixing with tags that have been entered manually by specifying a group to be used for automatically generated tags. The name of the group can be up to 256 characters. This parent group provides a root branch to which all automatically generated tags are added.

Allow Automatically Generated Subgroups: This property controls whether the server automatically creates subgroups for the automatically generated tags. This is the default setting. If disabled, the server generates the device's tags in a flat list without any grouping. In the server project, the resulting tags are named with the address value. For example, the tag names are not retained during the generation process.
Note: If, as the server is generating tags, a tag is assigned the same name as an existing tag, the system automatically increments to the next highest number so that the tag name is not duplicated. For example, if the generation process creates a tag named "AI22" that already exists, it creates the tag as "AI23" instead.

Create: Initiates the creation of automatically generated OPC tags. If the device's configuration has been modified, **Create tags** forces the driver to reevaluate the device for possible tag changes. Its ability to be accessed from the System tags allows a client application to initiate tag database creation.
Note: **Create tags** is disabled if the Configuration edits a project offline.

Device Properties — Data Source Access Method

Access Method Selection
Once a channel with a defined ODBC data source is created, a device must be created for each access method desired. Options include **Fixed Table**, **Dynamic Table**, **Query**, and **Stored Procedure**. This can only be selected during device creation.
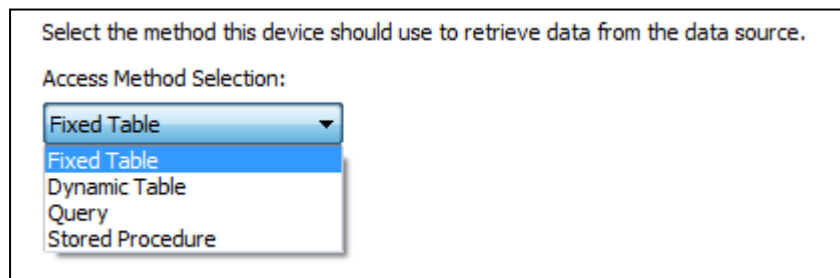
Fixed Table
Dynamic Table
Query
Stored Procedure

Notes:

1. The account configured in the DSN must have appropriate access permissions for these access methods to execute properly.

2. Users may receive a message that states the driver was unable to retrieve a data source component. It is important to verify in the ODBC administrator that the path to the data source is correct for the DSN. *For more information, refer to* Data Source Setup *and* **ODBC Client Driver Specific Messages**.

Fixed Table

Data can be retrieved from an ODBC data source table through the Fixed Table access method which allows access to each item in the ODBC data source table. After creating a device through the specified Fixed Table method, the driver will create tags in the server's tag database that correspond to each item in the ODBC data source table.
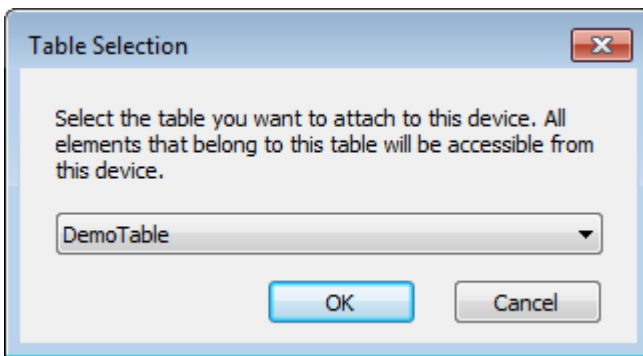
The basic flow of defining a fixed table is shown in the diagram below. Click on each step for instructions.

Table Selection

Once an access method has been selected, select a table from which to access data. All table names in the ODBC data source are available for selection. Once a table name is selected, the driver can be configured to automatically generate tags to represent the data in the table.

When changing the Table Selection of an existing Fixed Table Device, the following device properties need to be updated and tags regenerated: **Limit Import Columns** and **Maximum Records to Import**. In the configuration user interface, a change in Table Selection will automatically update the Limit Import Columns with all of the columns in the table and the Maximum Records to Import with the number of records in the table.

Auto Tag Database Generation Options
The Automatic Tag Generation (ATG) Options can be used to control how tags are automatically generated in the server's tag database. Tags that are automatically generated will differ depending on the type of access method selected.
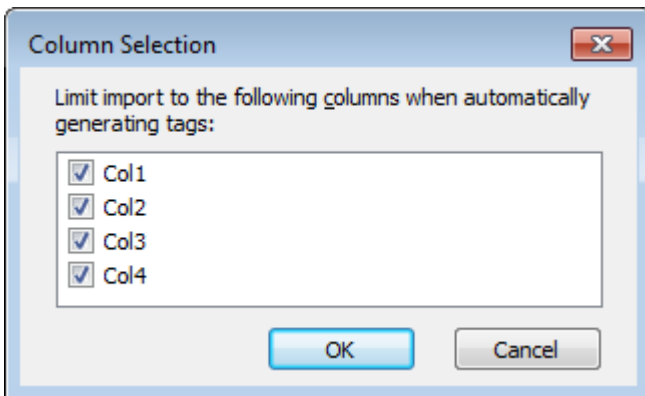
Write Access: This property specifies the default Write privileges when automatically generating tags. It only sets the tag's access privileges to Read/Write in the server: it does not affect the field's access privileges in the database.

⬤ **Notes**:

1. Some data sources (such as Excel) do not support Writes.
   - **Limit Import Columns:** This property specifies the data source's table columns that will be imported into the tag database. This property only applies to Fixed Table. The string lists the column names to use for tag import, separated by a semi-colon. Click on the browse button to launch the Column Selection dialog. After selecting the columns and clicking on OK, the string is updated with the newly configured column list.

2. If the column name includes a semi-colon, it is enclosed in quotation marks. If that same column name includes quotation marks, these are doubled up. For example, the string format is:Col1;"Col-with;1";"Colwith;and""1" for columns named:
   - Col1
   - Colwith;1
   - Colwith;and"1

3. If the string begins with a quotation mark, but does not end with one, then any quotation marks are doubled up. For example, the string format is: Col1;"""Col2;Col"3" for columns named:
   - Col1
   - "Col2
   - Col"3"

- **Maximum Records to Import:** This property specifies the maximum number of records to import. The default setting will automatically generate tags for all records in the table, but users can limit the number is desired. This property only applies to Fixed Table. The default setting is 0.

Timing Options — Fixed

For better performance, the ODBC Client Driver can query the data source for updates at an interval that is independent of tag update rates.

> Specify the minimum frequency at which the driver will query the data source for the latest data values.
>
> Query Interval (ms):
>
> 250

Query Interval (ms): This property specifies the frequency that the driver will query the data source for the latest data values. The valid range is 100 to 99999999 milliseconds. The default setting is 250 milliseconds.

⬤ **Notes**: It is recommended that the client update rate for the tags generated for the table be at least **twice as fast as this query interval**.

Dynamic Tables

Data can be retrieved from an ODBC data source table through the Dynamic Table access method, which allows access to a single record set in an ODBC data source table via record tags. The record tags correspond to their respective columns in the data source table, and are dynamically controlled via internal control tags (which are accessible from a client application). This allows users to work with a particular record set in the table. The table below displays a list of the internal control tags that can be used to access the data.

Dynamic Table Control Tags	Access	Description
Move	Read/Write	Move a specified number of records forward (positive value) or backward (negative value) in a table.
MovePrev	Read/Write	Move to the previous record (on write complete).
MoveNext	Read/Write	Move to the next record (on write complete).
MoveLast	Read/Write	Move to the last record in the table (on write complete).
MoveFirst	Read/Write	Move to the first record in the table (on write complete).
CurrentRecord	Read Only	The index of the record that is currently being accessed.
RecordCount	Read Only	Number of records are in the table.

The basic flow of defining a dynamic table is shown in the diagram below. Click on each step for instructions.

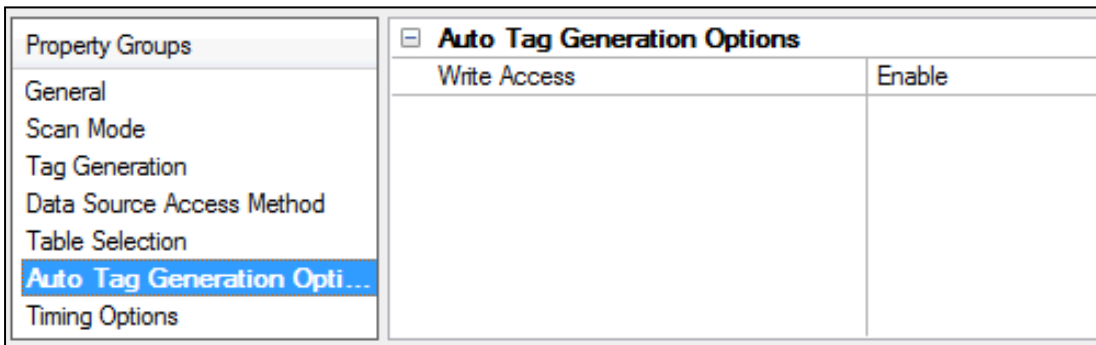Access Method → Table Selection → Tag Database → Timing Options

Table Selection

Once an access method has been selected, select a table from which to access data. All table names in the ODBC data source are available for selection. Once a table name is selected, the driver can be configured to automatically generate tags to represent the data in the table.
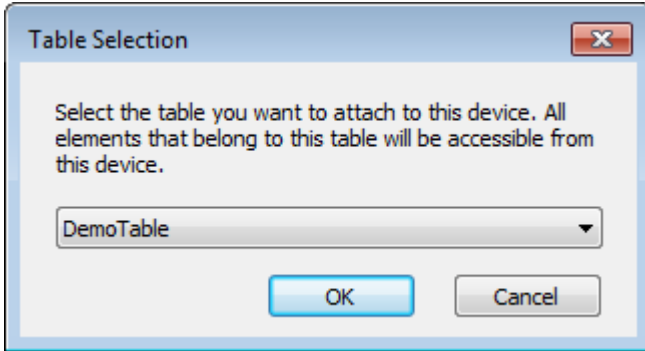
Auto Tag Database Generation Options

The Automatic Tag Generation (ATG) Options can be used to control how tags are automatically generated in the server's tag database.
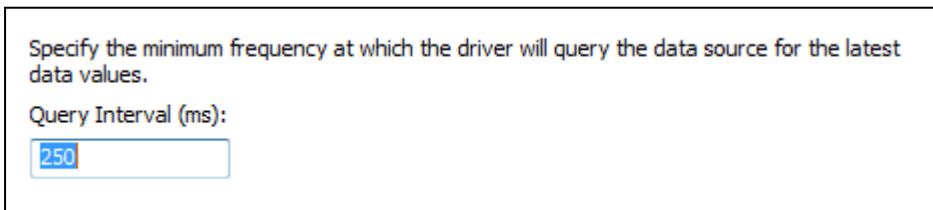

Write Access: This property specifies the default Write privileges when automatically generating tags. It only sets the tag's access privileges to Read/Write in the server: it does not affect the field's access privileges in the database.

● **Note**: Some data sources (such as Excel) do not support Writes.

Timing Options — Dynamic

For better performance, the ODBC Client Driver can query the data source for updates at an interval that is independent of tag update rates.

Query Interval (ms): This proeprty specifies the frequency that the driver will query the data source for the latest data values. The valid range is 100 to 99999999 milliseconds. The default setting is 250 milliseconds.

💧**Note:** It is recommended that the client update rate for the tags generated for the table be at least **twice as fast as this query interval**.

Query

The ODBC Client Driver supports SELECT and EXECUTE queries in which a tag database can be created and defined according to specific needs. Users can retrieve tags from whole tables and specific records. Although the query access method can be used to execute stored procedures that return many result sets, the generated tags are from the first result set only. *For more information, see* **Choosing Query or Stored Procedure Access Method**.

When changing the Query string of an existing Query Device, if the Trigger Column is configured, it may need to be updated and tags regenerated.

The basic flow of defining a query is shown in the diagram below. Click on each step for instructions.

💧**See Also:**
Using the Query Access Method
Data Source Query Format
Modifying a Query

Building a Valid Query Statement

Descriptions of the properties are as follows:

Query: This property is used to enter a valid query statement. SELECT and EXECUTE statements are supported.

💧**Note**: Supported EXECUTE statements include those with a valid stored procedure and an optional list of unnamed or named input properties. The database and/or schema should not be used to qualify the stored procedure name. To reference a stored procedure in another database, create another ODBC data source to that database. Then configure a channel for that ODBC data source and a device to directly reference the stored procedure.

Validate...: Clicking on this hyperlink/button allows the driver to determine if the query statement is valid or invalid by executing the query. A pop-up message will indicate the query is valid or provide details for an invalid query

Notes:

1. Tags will be generated if all properties are valid after the wizard **Finish** button has been pressed.

2. The **Validate** button triggers validation of the query string. The validation process includes executing the query statement.

See Also:
Using the Query Access Method
Data Source Query Format
Modifying a Query

Miscellaneous Query Trigger Definition

The Miscellaneous property dialog allows setting a column that should act as the trigger to the client that the query execution completed and the tags have all been updated.

Set the column that should act as the trigger for this query.

Trigger Column:

Trigger Column: This property specifies the field or column in the database that should be updated last. It is typically used as a trigger to indicate to the connected client that all other fields have been updated and can be processed.

For example, when the Query statement contains "SELECT job_id, job_name, job_desc WHERE job_name = 'manager'" and the Trigger column property is set to "job_id", the tags in the job_id tag group will be updated last.

See Also:
Using the Query Access Method
Data Source Query Format
Modifying a Query

Timing Options — Query

For better performance, the ODBC Client driver can query the data source for updates at an interval that is independent of tag update rates.

Query Interval: When enabled, this property enables the query to run each time the query interval elapses. When disabled, the query will not run unless a write to the Execute internal control tag occurs. The default setting is enabled.

Query Interval (ms): This property is available to configure if the Query Interval property is enabled. If available, this property specifies the minimum frequency at which the driver will query the data source for the latest data values. It is recommended that the client update rate for tags generated for the results of the query be at least twice as fast as this query interval. The query interval may need to be adjusted to ensure all the tags are requested by the client within that interval. The valid range is 100 to 99999999 milliseconds. The default setting is 250 milliseconds.

● **Note**: When configured with a disabled or long query interval, synchronous requests from a client should be avoided as they will not complete until the query interval elapses or a write to the internal Execute tag occurs.

Auto Tag Database Generation Options
The Automatic Tag Generation (ATG) Options can be used to control how tags are automatically generated in the server's tag database. Tags that are automatically generated will differ depending on the type of access method selected.

Write Access: This property specifies the default Write privileges when automatically generating tags. It only sets the tag's access privileges to Read/Write in the server: it does not affect the field's access privileges in the database.

● **Notes**:

- For queries, this property only applies to SELECT statements. Tags generated from EXECUTE statements are always read only.

· Some data sources (such as Excel) do not support Writes.

Using the Query Access Method

For information on retrieving tags based on a query, refer to the instructions below.

1. In **Query**, type a SELECT or an EXECUTE query. Supported EXECUTE statements are limited to those with a valid stored procedure and an optional list of unnamed or named input properties. The stored procedure name should not be qualified by a database and/or schema. To reference a stored procedure in another database, create another ODBC data source to that database. Then configure a channel for that ODBC data source and a device to directly reference the stored procedure.

   Note: Some ODBC data sources do not allow queries. *For more information, refer to the data source's help documentation.*

2. Pressing the **Validate**, **Update Tags**, **Apply**, or **OK** button executes the query statement to validate the statement. If no error message is received, the query is valid. A pop-up error message displays if the query is invalid. If the query is invalid, make changes to the query statement or the data source connection and then re-attempt validation.

3. Pressing the **Update Tags**, **Apply**, or **OK** button executes the query, generates an internal Execute tag, and a tag database based on the results of the query. If a message is displayed indicating that the query is invalid, make changes to the query statement or the data source connection and then re-attempt validation and tag generation.

Important: For tags to generate, the query must result in at least one returned record. A null return will neither generate tags nor post error messages (since it is not considered to be an error).

Note: If a SELECT statement returns at least one record, an internal Execute tag is also generated to allow the client to trigger query execution. The internal Execute tag is always available for an EXECUTE statement even when no other tags result from the query.

Query Tags	Access	Description
Execute	Read/Write	Execute query statement (on write complete).

Data Source Query Format

For information on creating a query to retrieve specific data rows from a table, refer to the instructions below.

1. In **Query**, add a search condition to the query using the following format:

   SELECT [*column1, column2, ...*] FROM [*table*] WHERE [*search condition*];

2. The general format of a search condition is as follows:

   WHERE [*expression*] [*comparison operator*] [*expression*]

   Note: When using specific strings in the expression, single quotes may be needed. For example, "Select job_desc WHERE jobname = 'manager'".

A tag group is created for each column in the result set. A tag for each record in that column is created in the tag group with an address of the column name and record index.

Modifying a Query Statement

Once a device has been defined with a query statement, changes to the statement can be made by opening the Device Properties and selecting the Build Query Settings property group.

● **Note**: On the Build Query Device property group, the **Validate**, **Update Tags**, **Apply**, and **OK** buttons all cause validation of the query string by executing the query statement.

Query: This property is used to enter a valid query statement. SELECT and EXECUTE statements are supported.

> ● **Note**: Supported EXECUTE statements are limited to those with a valid stored procedure and an optional list of unnamed or named input properties. The stored procedure name should not be qualified by a database and/or schema. To reference a stored procedure in another database, create another ODBC data source to that database. Then configure a channel for that ODBC data source and a device to directly reference the stored procedure.

Validate...: Clicking on this hyperlink/button, executes the query to determine whether the query statement is valid or invalid.

Update Tags: Clicking on this hyperlink, applies all changes to the device properties, executes the query , and refreshes the tag database. A tag group is created for each column in the result set. Within each group a tag is created for each row. Additionally, the internal Execute tag is created at the device level to allow the client to execute the query.

Write Access:This property specifies the default Write privileges when automatically generating tags. It only sets the tag's access privileges to Read/Write in the server: it does not affect the field's access privileges in the database. This property only applies to SELECT statements. Tags generated from EXECUTE statements are always Read-Only

See Also:
Using the Query Access Method
Building a Query Statement
Data Source Query Format

Stored Procedure

The ODBC Client Driver supports access to stored procedures. *For more information, see **Choosing Query or Stored Procedure Access Method**.*

Guidelines for accessing stored procedures on the ODBC data source are as follows:

- The data source must support stored procedures.
- At least one custom stored procedure must be contained and tested by the data source before users can attempt to create and connect to the tag database via the ODBC Client Driver.
- The current account must have permission to access the stored procedure.
- The driver must have access to the data source via ODBC so that it can access and execute the stored procedure properly.

Note: The ODBC Client Driver uses MyODBC to communicate with MySQL. At the time of release, MyODBC did not support stored procedures; as such, the ODBC Client Driver's support of MySQL does not include stored procedures.

Stored Procedure Selection

Select the stored procedure you want to attach to this device.

Stored Procedure Selection:

PROC52

When Enable is selected, the OPC server clears input values after the procedure is executed returning them to default values for the data types.

Clear Inputs After Each Execution:

Enable

Specify Enable so that the OPC server will reset the input values after the last active client is disconnected from the server.

Reinitialize Stored Procedure:

Enable

Descriptions of the properties are as follows:

- **Stored Procedure Selection:** This property specifies the stored procedure that executes when a user writes to the internal Execute tag. Click on the browse button to launch the Stored Procedure Selection dialog.

Select the stored procedure you want to attach to this device.

Stored Procedure Selection:

STORED_PROC

When Enable is selected, the OPC server clears input values after the procedure is executed returning them to default values for the data types.

Clear Inputs After Each Execution:

Disable

Specify Enable so that the OPC server will reset the input values after the last active client is disconnected from the server.

Reinitialize Stored Procedure:

Enable

- **Clear inputs after each execution:** When enabled, the OPC server clears the input values after the stored procedure is executed, returning them to the default values for the data types. The default setting is disabled.
- **Reinitialize Stored Procedure:** When enabled, the OPC server will reset the input values after the last active client is disconnected from the server. When disabled, the input properties will retain their last known values when the client is reconnected. The default setting is enabled.

See Also:

Choosing Query or Stored Procedure Access Method

Creating the Tag Database

Selecting a Different Stored Procedure

Creating the Tag Database

Once the stored procedure has been selected, users can continue through and finish the wizard. The tag database of 'input', 'output', and 'return value' tags are created based on the selected stored procedure. An internal Execute tag is created in the tag database to allow the client to control when to execute the procedure. After writing appropriate values to the input tags (if applicable), users can write to the Execute tag to execute the stored procedure.

Stored Procedure Tags	Access	Description
Execute	Read/Write	Execute stored procedure (on write complete).

The example below displays a tag database that was created from a successful stored procedure.
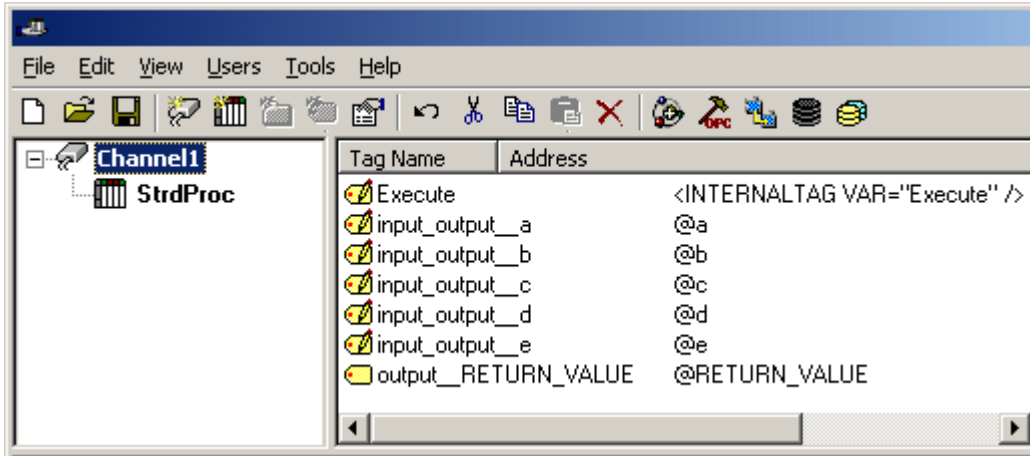
Selecting a Different Stored Procedure

Once a device has been defined with a stored procedure, a new stored procedure may be selected by opening the **Device Properties** editor and selecting the **Stored Procedure** property group. The list of available stored procedures is generated based on the configured ODBC data source.

● **Note**: Pressing the **Update Tags**, **Apply**, or **OK** button generates an internal Execute tag and input, output, and return value tags based on the definition of the stored procedure.

Descriptions of the properties are as follows:

Stored Procedure Selection: This property specifies the stored procedure that will execute when a user writes to the Execute internal control tag.

Update Tags...: Clicking on this hyperlink refreshes the tag database with an internal Execute tag and input, output, and return value tags based on the definition of the stored procedure.

Clear Inputs After Each Execution: When enabled, the OPC server will clear the input tag values after the stored procedure is executed, thus returning them to the appropriate default value for its data type. The default setting is unchecked.

Reinitialize Stored Procedure: When enabled, the OPC server will reset the input values after the last active client is disconnected from the server. When disabled, the input properties will retain their last known values when the client is reconnected. The default setting is enabled.

🔷 **See Also**: **Choosing Query or Stored Procedure Access Method**

Choosing Query or Stored Procedure Access for Stored Procedure Execution

Only the Query Access Method allows directly executing a SELECT statement. However, both the Query Access Method and the Stored Procedure Access Method allow executing stored procedures.

Use the Stored Procedure Access Method to execute a stored procedure to:

- Retrieve a return value and/or output properties
- Allow the client to vary the input properties
- Only execute the stored procedure when the client writes to the Execute tag (the Query Access Method executes on validation, Auto Tag Generation, and when the client writes to the Execute tag)

Use the Query Access Method to execute a stored procedure to:

- Retrieve data from the first result set
- Execute with the default property values

🔷 **See Also**:
Query Access Method
Stored Procedure Access Method

Running as a System Service

Unlike OPC servers that only support Stand Alone program operation, this server can continue to supply OPC data across user login or logout sessions by running as a System Service. This is crucial for applications in which the server must provide data to OPC clients via DCOM. For these applications, the loss of a DCOM connection cannot be tolerated.

The service must be able to access the configured Data Source with valid permissions to successfully connect and access the data source objects. Refer to Data Source Setup and ODBC Client Driver Specific Messages.

Connecting Remotely in System Service Mode

This ODBC communications application supports running as a service under supported Microsoft Windows operating systems. For operating system requirements, refer to the OPC server's help documentation.

Data Types Description

When creating tags, this driver converts the Transact_SQL data type to the following tag data types:

Transact-SQL Data Type	Tag Data Type	Tag Data Type Description
Bit	Boolean	Single bit
Tinyint	Byte	Unsigned - bit value bit 0 is the low bit bit 7 is the high bit
Int Smallint	Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
Real	Float	32-bit floating point value
Float	Double	64-bit floating point value
Bigint Char Date Datetime2 Datetimeoffset Decimal Money Nchar Numeric Nvarchar Smallmoney Time Varchar	String	Null-terminated character array. This driver limits string lengths to 256 characters.
Datetime Smalldatetime	Date	64-bit floating-point value representing the number of milliseconds since December 31, 1899. Tag value string literal format: YYYY-MM-DDThh:mm:ss.000

🔹 **Note:** The default data type is String.

Address Descriptions

Tag validation will only occur when a tag is requested by a client application. If a link to the ODBC data source cannot be made by the driver, the tag's data quality will be set to Bad. This will also occur if the address that the tag references cannot be resolved by the driver; such as, if the address cannot be mapped to the data set in the ODBC data source. When this occurs, an appropriate error message will be placed in the OPC server's Event Log.

See Also: **Error Descriptions**

Error Descriptions

The following categories of messages may be generated. Click on a link for a list of related messages.

Address Validation

Device Status Messages

ODBC Client Driver Messages

Address Validation

The following messages may be generated. Click on the link for a description of the message.

Missing address.

Device address <address> contains a syntax error.

Data type <type> is not valid for device address <address>.

Device address <address> is read only.

Missing address.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has no length.

Solution:
Re-enter the address in the client application.

Device address <address> contains a syntax error.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically contains one or more invalid characters.

Solution:

1. If the address is dynamic, re-enter it in the client application.

2. If the address is static, re-enter it in the server application.

Data Type <type> is not valid for device address <address>.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has been assigned an invalid data type.

Solution:

Modify the requested data type in the client or server application.

Device address <address> is read only.

Error Type:
Warning

Possible Cause:
A client has requested an access mode that is not compatible with what the data source supports for that tag address.

Solution:
Change the access mode in the client application, under device properties or under the actual tag in the server's tag data base.

Note:
Although some data sources do not support write access, others may support write for certain items.

Device Status Messages

The following message may be generated. Click on the link for a description of the message.

Write request rejected on read only item reference <channel name><device name><address>.

Write request rejected on read only item reference <channel name><device name><address>.

Error Type:
Serious

Possible Cause:

1. The item is read only; it cannot be written.

2. The Auto Tag Generation property to **set write access for imported database tags** is unchecked and the tag was created as read only.

Solution:

1. Determine if the item should be able to be written and change the tag access rights.

2. Enable the **Set write access for imported database tags** property and regenerate the tags.

ODBC Client Driver Specific Messages

The following messages may be generated. Click on a link for a description of the message.

Messages presented in a pop-up dialog:
Query validation failed due to the following: (Reason: ODBC Specific Error).
Unable to retrieve table names (Reason: ODBC Specific Error).
Unable to retrieve table column names (Reason: ODBC Specific Error).
Unable to retrieve table row count (Reason: ODBC Specific Error).

Unable to retrieve stored procedure names (Reason: ODBC Specific Error).
Unable to retrieve supported access methods (Reason: ODBC Specific Error).

Messages presented in the Event Log:
Call to execute stored procedure <stored procedure name> on device <device name> failed (Reason: ODBC Specific Error).
Data source <data source name> is configured as read only (all writes will be rejected). Please refer to the database configuration.
Unable to connect to data source on channel <channel name> (Reason: ODBC Specific Error).
Unable to open recordset on device <device name> (Reason: ODBC Specific Error).
Unable to query recordset on device <device name> (Reason: ODBC Specific Error).
Unable to retrieve database value for tag <tag address> on device <device name> (Reason: ODBC Specific Error).
Unable to write database value <value> for tag <tag address> on device <device name> (Reason: ODBC Specific Error).
Unable to read internal tag <tag address> on device <device name> (Reason: ODBC Specific Error).
Unable to write to internal tag <tag address> on device <device name> (Reason: ODBC Specific Error).
Unable to read parameter tag <tag address>. (Reason: ODBC Specific Error).
Unable to write to parameter tag <tag address> (Reason: ODBC Specific Error).
Unable to execute stored procedure <stored procedure name> on device <device name>. (Reason: ODBC Specific Error).
Unable to write to item <tag address> (Reason: ODBC Specific Error).

Call to execute stored procedure <stored procedure name> on device <device name> failed. (Reason: ODBC Specific Error).

Error Type:
Serious

Possible Cause:

1. The ODBC data source is not available.

2. The ODBC data source is not accessible based on user name and password.

3. The input properties are not valid.

Solution:

1. Verify that the ODBC data source is configured properly and that it is accessible by the user or from a remote machine.

2. Verify that input properties are set properly before executing the stored procedure.

 Note:

The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Data source <data source name> is configured as Read Only (all writes will be rejected).

Error Type:
Serious

Possible Cause:
The ODBC data source is configured such that writes are not allowed.

Solution:
If writes are required, ensure that the data source is configured to allow write access.

Query validation failed due to the following: (Reason: ODBC Specific Error).

Error Type:
Serious

Possible Cause:

1. The ODBC data source is not available.

2. The ODBC data source cannot satisfy the requirements of the query statement.

Solution:

1. Verify that the ODBC data source is configured properly and that it is accessible by the user or from a remote machine.

2. Verify that the query statement is valid for the ODBC data source.

Note:
The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to retrieve table names. (Reason: ODBC Specific Error).

Error Type:
Serious

Possible Cause:

1. The ODBC data source is not available.

2. The ODBC data source does not contain any tables.

Solution:

1. Verify that the ODBC data source is configured properly and that it is accessible by the user or from a remote machine.

2. Verify that the ODBC data source contains at least one table.

Note:

The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to retrieve table column names. (Reason: ODBC Specific Error).

Error Type:

Serious

Possible Cause:

1. The ODBC data source is not available.

2. The ODBC data source table is not available.

3. The ODBC data source table does not contain any columns.

Solution:

1. Verify that the ODBC data source is configured properly and that it is accessible by the user or from a remote machine.

2. Verify that the ODBC data source table exists in the ODBC data source.

3. Verify that the ODBC data source table contains at least one column.

Note:

The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to retrieve table row count. (Reason: ODBC Specific Error).

Error Type:

Serious

Possible Cause:

1. The ODBC data source is not available.

2. The ODBC data source table is not available.

3. The ODBC data source table does not contain any records.

Solution:

1. Verify that the ODBC data source is configured properly and that it is accessible by the user or from a remote machine.

2. Verify that the ODBC data source table exists in the ODBC data source.

3. Verify that the ODBC data source table contains at least one record.

Note:

The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to retrieve stored procedure names. (Reason: ODBC Specific Error).

Error Type:
Serious

Possible Cause:

1. The ODBC data source is not available.

2. The ODBC data source does not contain any stored procedures.

Solution:

1. Verify that the ODBC data source is configured properly and that it is accessible by the user or from a remote machine.

2. Verify that the ODBC data source contains at least one stored procedure.

Note:
The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to retrieve supported access methods. (Reason: ODBC Specific Error).

Error Type:
Warning

Possible Cause:
The ODBC data source is not available.

Solution:
Verify that the ODBC data source is configured properly and that it is accessible by the user or from a remote machine.

Note:
The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to connect to data source on channel <channel name>. (Reason: ODBC Specific Error).

Error Type:
Serious

Possible Cause:

1. The ODBC data source is not available.

2. The ODBC data source is not accessible based on user name and password.

3. The ODBC data source is a User DSN and the server is running as a service without access to the User DSN.

Solution:

1. Verify that the ODBC data source is configured properly and that it is accessible by the user or from a remote machine.

2. Verify that the authentication settings are properly configured in the data source (if applicable).

3. When the server is running as a service, configure the channel with a System DSN. If using a User DSN, then the service must have access to that DSN.

Note:

The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

See Also:

Data Source Authentication

Unable to open recordset on device <device name>. (Reason: ODBC Specific Error).

Error Type:

Serious

Possible Cause:

1. The ODBC data source is not available.

2. The ODBC data source is not accessible based on user name and password.

3. The table or query specified by the device no longer exists.

Solution:

1. Verify that the ODBC data source is configured properly and that it is accessible by the user or from a remote machine.

2. Verify that the table or query is supported by the data source.

Note:

The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to query recordset on device <device name>. (Reason: ODBC Specific Error).

Error Type:

Serious

Possible Cause:

1. The ODBC data source is no longer available.

2. The table is created but it has had no rows inserted.

Solution:

1. Verify that the ODBC data source is accessible by the user or from a remote machine.

2. Insert at least one row into the table.

Note:

The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to retrieve database value for tag <tag address> on device <device name>. (Reason: ODBC Specific Error).

Error Type:

Serious

Possible Cause:

1. The ODBC data source is no longer available.

2. The tag represents an address that cannot be resolved into a value available from the data source.

3. The table is created but it has had no rows inserted.

Solution:

1. Verify that the ODBC data source is accessible by the user or from a remote machine.

2. Verify that the tag represents an address that is supported by the data source and/or access method.

3. Insert at least one row into the table.

Unable to write database value <value> for tag <tag address> on device <device name>.

Error Type:

Serious

Possible Cause:

1. The ODBC data source is no longer available.

2. The tag represents an address that cannot be resolved into a value available from the data source.

3. The ODBC data source does not allow write access to this value.

Solution:

1. Verify the ODBC data source is accessible by the user or from a remote machine.

2. Verify that the tag represents an address that is supported by the data source and/or access method.

3. Verify that the value is write-able in the ODBC data source configuration.

Unable to read internal tag <tag address> on device <device name> (Reason: ODBC Specific Error).

Error Type:
Serious

Possible Cause:
The internal tag depends on a successful response from the ODBC data source which is no longer available.

Solution:
Verify that the ODBC data source is still accessible by the user or from a remote machine.

⬤ Note:
The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to write to internal tag <tag address> on device <device name> (Reason: ODBC Specific Error).

Error Type:
Serious

Possible Cause:
The internal tag depends on a successful response from the ODBC data source which is no longer available.

Solution:
Verify the ODBC data source is still accessible by the user or from a remote machine.

⬤ Note:
The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to read parameter tag <tag address>. (Reason: ODBC Specific Error).

Error Type:
Serious

Possible Cause:
The parameter tag for a stored procedure is no longer available or requires a data type not supported by the driver.

Solution:
Verify that the parameter tag for the stored procedure is valid and the data type is supported by the driver.

⬤ Note:
The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to write to parameter tag <tag address> (Reason: ODBC Specific Error).

Error Type:

Serious

Possible Cause:

The parameter tag for a stored procedure is no longer available or requires a data type not supported by the driver.

Solution:

Verify that the parameter tag for the stored procedure is valid and the data type is supported by the driver.

Note:

The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to execute stored procedure <stored procedure name> on device <device name>. (Reason: ODBC Specific Error).

Error Type:

Serious

Possible Cause:

An operation that is required to be performed prior to calling the stored procedure has failed. Examples may include:

- Failed to bind parameter <parameter>.
- Procedure requires unsupported parameter type(s) <parameter type>.
- Failed to bind column <column>.
- Fetch result data failed.
- A connection to the data source failed.

Solution:

Verify that the ODBC data source is configured properly for this stored procedure. Respond to the specific reasons provided.

Note:

The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

Unable to write to item <tag address> (Reason: ODBC Specific Error).

Error Type:

Serious

Possible Cause:

1. The tag is no longer available or requires a data type not supported by the driver.

2. The OPC Server is unable to communicate with the data source.

3. The database is read only.

4. If the data source is on another PC (other than the OPC server), communications to that PC may be lost.

Solution:

1. Verify that the tag is valid and the data type is supported by the driver.

2. Verify that the data source configuration is correct.

3. Verify that the data source has write permissions.

4. Check physical connections between the OPC server PC and the data source PC.

Note:

The ODBC Specific Error returned by the ODBC data source driver contains specific information on the cause of the problem.

See Also:
Data Source Setup

Index

E

F

G