Advanced Simulator Driver

Table of Contents

Welcome to the Advanced Simulator Driver Help Center

This is the user documentation for Kepware Advanced Simulator Driver. This documentation is updated regularly to reflect the latest functionality and information.

Overview
What is the Advanced Simulator Driver?

Setup
How do I configure the driver to communicate with ODBC data sources?

Data Types Descriptions
What data types are supported by this driver?

Address Descriptions
How do I address a data location in the ODBC data source?

Error Descriptions
What error messages are produced by the Advanced Simulator Driver?

Version 1.032

Overview

The Advanced Simulator Driver plugs into the industrial-based OPC server, which supplies reliable data access to any HMI, SCADA, Historian or Enterprise application supporting any platform interfaces. The Advanced Simulator Driver provides single-point access to multiple ODBC data sources via ODBC. It simulates live data by traversing through all the records in a selected table at a rate set by the user.

For information on general server topics, refer to the server's help documentation.

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

The Advanced Simulator Driver supports any ODBC-compliant data source, including the following:

Microsoft Access
Microsoft SQL
MySQL*
Oracle
Sybase

*If the project uses a MySQL data source, select **MySQL** from the **Model** drop-down list when creating a device. The driver will accommodate the MySQL-specific data handling requirements. Users should make sure that the MyODBC driver is installed for the PC that is running the OPC server.

Note: The Advanced Simulator Driver also supports the following:

- Multiple connection sessions to multiple ODBC data sources
- Automatic tag generation from the ODBC data source
- OPC DA
- Drag and drop editing
- CSV import and export for easy manipulation of many tags
- New tags that can be added to the application at any time (including Runtime)
- Running as a Windows Service

Communication Protocol

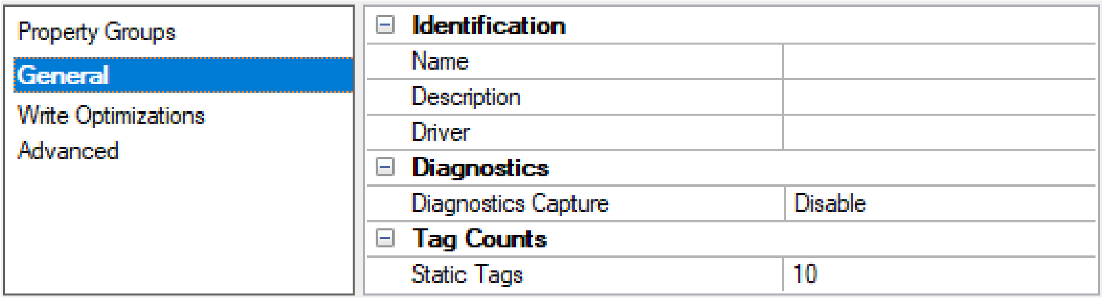
ODBC API

Channel and Device Limits

The maximum number of channels supported by this driver is 100. The maximum number of devices is 4096 per channel.

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
- General
- Write Optimizations
- Advanced

Identification	
Name	
Description	
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

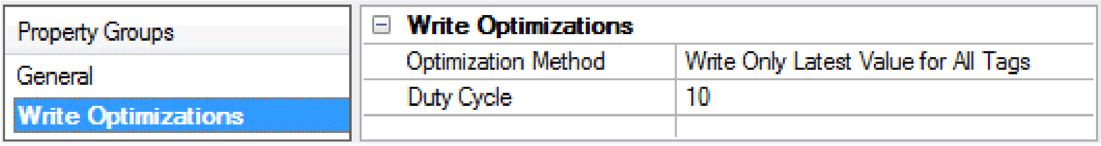
Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.
Note: This property is not available if the driver or operating system does not support diagnostics.
For more information, refer to Communication Diagnostics and Statistics Tags in 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.



Write Optimizations	
Optimization Method	Write Only Latest Value for All Tags
Duty Cycle	10

Write Optimizations

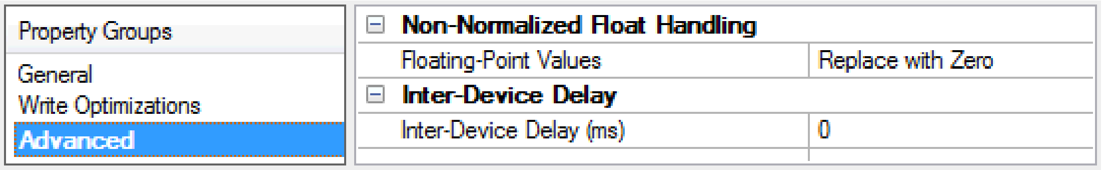
Optimization Method: Controls how write data is passed to the underlying communications driver. The options are:

- **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.



Non-Normalized Float Handling	
Floating-Point Values	Replace with Zero
Inter-Device Delay	
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.

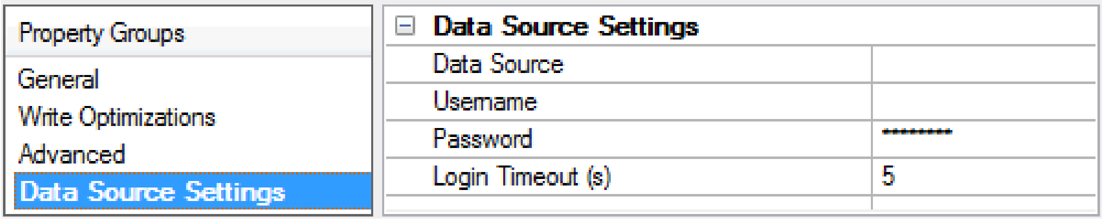
For more information on the floating-point values, refer to "How To ... Work with Non-Normalized Floating-Point Values" in the server help.

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 property group is used to configure various data source settings. It is located in Channel Properties.



Property Groups
- General
- Write Optimizations
- Advanced
- Data Source Settings

⊟ Data Source Settings	
Data Source	
Username	
Password	********
Login Timeout (s)	5

Data Source

This property is used to select a Data Source Name (DSN). A DSN must be created and configured for a data source before it will appear in this list and the ODBC database. For information on how to do so, follow the instructions below.

1. To start, Click the Data Source property to get the Data Source Selection pop up.
2. Click **Configure DSN** to open the **Microsoft® ODBC Data Source Administrator**.
3. Select either the **User DSN** or **System DSN** tab. Data source types are local to the computer: the User DSN is user-dedicated, whereas the System DSN can be accessed by any user with privileges.
4. Map an ODBC data source to a data source name.
5. Once a DSN has been configured successfully, it appears in the Data Source drop-down menu.

User Name and Password

The User Name and Password settings are only needed if required by the data source. For example, these fields are required when using SQL server authentication.

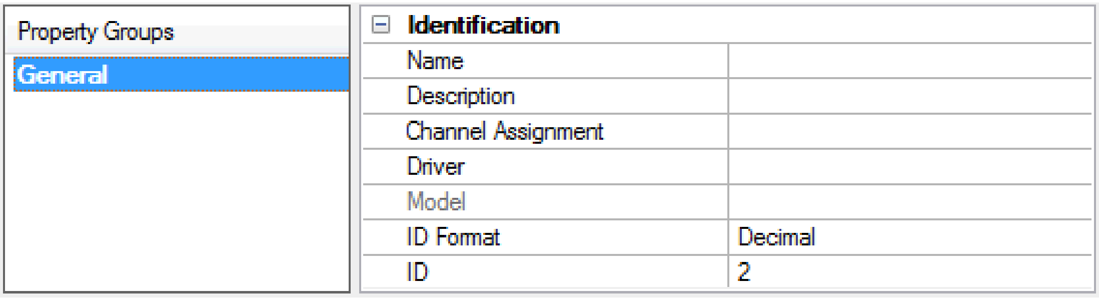
Note: Some data sources allow Windows Authentication be used to gain access to the data source. When used, the data source will ignore the User Name and Password specified in the Data Source Settings tab and will use the network Login ID instead. For more information, refer to Data Source Authentication.

Login Timeout

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 default setting is 5 seconds. The range is 1-99999 seconds.

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

General

⊟ Identification	
Name	
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.

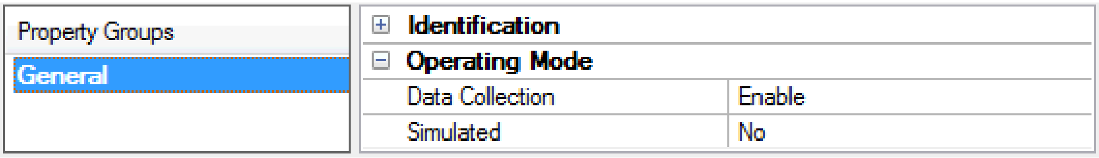
Channel Assignment: Specify the user-defined name of the channel to which this device currently belongs.

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 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



Property Groups

General

⊞ Identification	
⊟ Operating Mode	
Data Collection	Enable
Simulated	No

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.

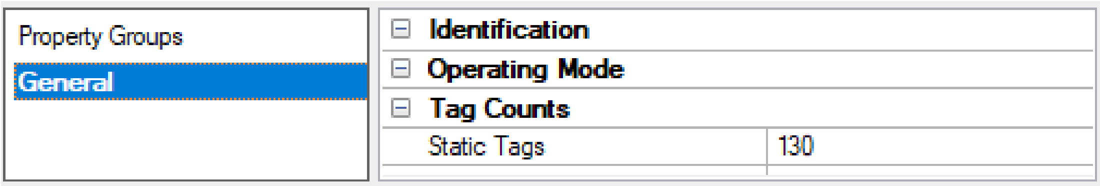
Simulated: Place the device into or out of Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client 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. Updates are not applied until clients disconnect and reconnect.
2. The 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.
3. In Simulation mode, the item's memory map is based on client update rate(s) (Group Update Rate for OPC clients or Scan Rate for native and DDE interfaces). This means that two clients that reference the same item with different update rates return different data.
4. When a device is simulated, updates may not appear faster than one (1) second in the client.

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

Tag Counts

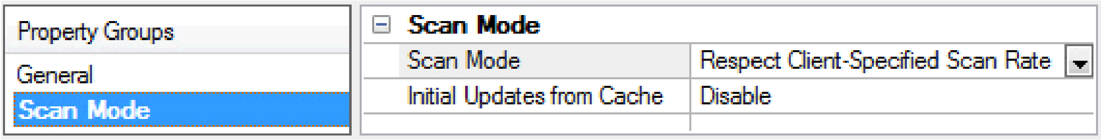


Static Tags 130

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 Respect Client-Specified Scan Rate

Initial Updates from Cache Disable

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.

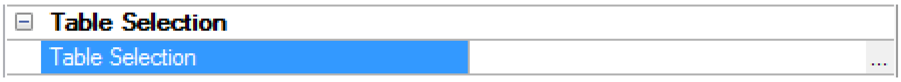
- **Do Not Scan, Demand Poll Only**: This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the OPC client's responsibility to poll for updates, either by writing to the _DemandPoll tag or by issuing explicit 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 – Table Selection

Selecting Tables

This property is used to specify a table from which data will be accessed. All table names available from the ODBC data source are available for selection.



⊟ **Table Selection**

Table Selection	...

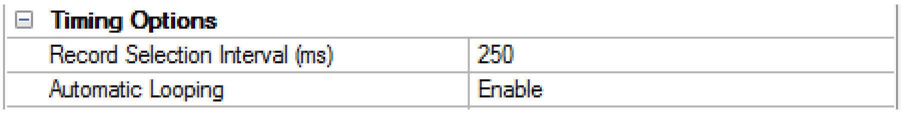
A number of tags are automatically generated in the server's tag database depending on the number of fields in the selected table. An internal control tag is also generated to access data.

Internal Control Tag	Access	Description
MoveTo	Read/Write	This internal tag moves forward or backward to the specified record number. It will continue updating the tags from its new position at the specified record selection interval.*

**For more information, refer to Timing.*

Device Properties – Timing Options

The driver simulates live data by traversing through all the records in the selected table at a rate set by the user. The tags are initialized to the data contained in the first record. The driver automatically selects the next record in the table at the set interval. The tags contain the data from the selected record. When the last record in the table is reached, users have the option to automatically start with the first record.



⊟ **Timing Options**

Record Selection Interval (ms)	250
Automatic Looping	Enable

Record Selection Interval

This property specifies the time interval for the driver to read the next record from the selected table. The driver updates the tags at these intervals. The valid range is 100 milliseconds to 99999999 milliseconds.

Automatic Looping

When enabled, this property causes the driver to start reading from the first record when the last record is reached. When unchecked, the driver will stop at the last record.

Data Types Description

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8-bit value bit 0 is the low bit bit 7 is the high bit
Char	Signed 8-bit value bit 0 is the low bit bit 6 is the high bit bit 7 is the sign bit
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
Float	32-bit Floating point value. The driver interprets two consecutive 16 bit registers as a Floating point value by making the second register the high word and the first register the low word.
Double	64-bit Floating point value
String	Null-terminated character array. This driver limits string lengths to 256 characters.

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 or the address that the tag references cannot be resolved by the driver (meaning that the address cannot be mapped to the data set in the ODBC data source) then the tag's data quality will be set to bad. The appropriate error message will be placed in the OPC server's event log.

For more information, refer to Error Descriptions.

Error Descriptions

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

Address Validation

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.
Array support is not available for the specified address: <address>.

Device Status Messages

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

Advanced Simulator Driver Specific Messages

Configuration
Unable to retrieve table names.
Unable to retrieve table column names.
Unable to retrieve table row count.

Runtime
Unable to connect to data source on channel <channel name>.
Unable to open recordset on device <device name>.
Unable to query recordset on device <device name>.
Unable to retrieve database value for tag <tag address> on device <device name>.
Unable to read internal tag <tag address> on device <device name>.
Unable to write to internal tag <tag address> on device <device name>.
Data source <data source name> is configured as Read Only (all writes will be rejected).

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. Re-enter the address in the client application if the address is dynamic.
2. Re-enter the address in the server application if the address is static.

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 tag address accessed by the client has requested an access mode that is not compatible with what the data source supports for that address.

Solution:

Change the access mode in the client application, Device Properties or under the actual tag in the server's tag database.

Note:

Some data sources do not support Write access, whereas others support it for specific items only.

Array support is not available for the specified address: <address>.

Error Type:

Warning

Possible Cause:

A tag address that has been specified statically contains an array reference for an address type that doesn't support arrays.

Solution:

1. Re-enter the address in the client application to remove the array reference.
2. Correct the address type.

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

Error Type:

Serious

Possible Cause:

A tag that has been specified as read only cannot be written to.

Solution:

N/A

Unable to retrieve table names.

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.

Error Type:

Serious

Possible Cause:

1. The ODBC data source is not available.
2. 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 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.

Error Type:

Serious

Possible Cause:

1. The ODBC data source is not available.
2. 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 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 connect to data source on channel <channel name>.

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.

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).

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

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:

This should not create any trouble as all the tags are Read Only.

Unable to open recordset on device <device name>.

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>.

Error Type:

Serious

Possible Cause:

The ODBC data source is no longer available.

Solution:

Verify that the ODBC data source 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 retrieve database 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.

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.

Unable to read internal tag <tag address> on device <device name>.

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>

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.

Index