Integration Objects'
OPC DA & HDA Interface for Databases

OPC Driver for Databases
Version 1.2 Rev. 2

USER GUIDE

OPC Compatibility

OPC Data Access 2.05a
OPC Data Access 3.00
Historical Data Access 1.00
Historical Data Access 1.10
Historical Data Access 1.20

OPC Driver for Databases User's Guide Version 1.2Rev .2
Published October 2018

OPC Foundation Self-Tested for Compliance logo indicates that this product has been tested by the manufacturer to be compliant with the following OPC Specifications:
Historical Data Access 1.00
Historical Data Access 1.10
Historical Data Access 1.20

Data Access 2.05a
Data Access 3.0

Additional information about compliance testing, logo program and a summary of test results for **OPC Driver for Databases** can be found at http://www.opcfoundation.org.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

PREFACE

About this User Guide

This guide:

- Describes the main features offered by Integration Objects' OPC Driver for Databases.
- Lists the system requirements for installing and running Integration Objects' OPC Driver for Databases.
- Explains how to use and run this OPC server.
- Describes all required DCOM settings both on server and client sides.

Target Audience

This document is intended for users who are looking for applications that provide connectivity to ADO-Compliant databases including SQL Server, Oracle, MS Access, MySQL and Wonderware Historian. Knowledge of the basics of OPC DA (Data Access) and OPC HDA (Historical Data Access) is assumed. It is also expected that you have some prior knowledge of database configuration listed above and SQL queries.

Document Conventions

Convention	Description
Bold	Click/selection action required
	Information to be noted
Blue bold italics	Reference to other sections, or to other Integration Objects' User Guides

Customer Support Services

Phone	Email
Americas: +1 713 609 9208 **Europe-Africa-Middle East** +216 71 195 360	Support: customerservice@integrationobjects.com Sales: sales@integrationobjects.com Online: www.integrationobjects.com

INTRODUCTION

1. Overview

Integration Objects' OPC Driver for Databases is a plug and play software designed to provide full access to real-time and archived data in relational databases to any OPC DA and HDA client.

In this chapter, we will:

- Present a global architecture showing the interaction between this OPC server, OPC clients and the various supported databases in a client/server environment.
- Describe all OPC DA and HDA interfaces supported by this server.
- Enumerate software and hardware system requirements.

2. System Architecture

This OPC Server reads and updates data from/to the database via the ADO (Active Data Objects) technology. It can be accessed locally or remotely via DCOM by any OPC DA and HDA client.
The following figure illustrates the client/server architecture that demonstrates the interaction between the OPC DA/HDA clients, the OPC Driver for Databases and the databases.

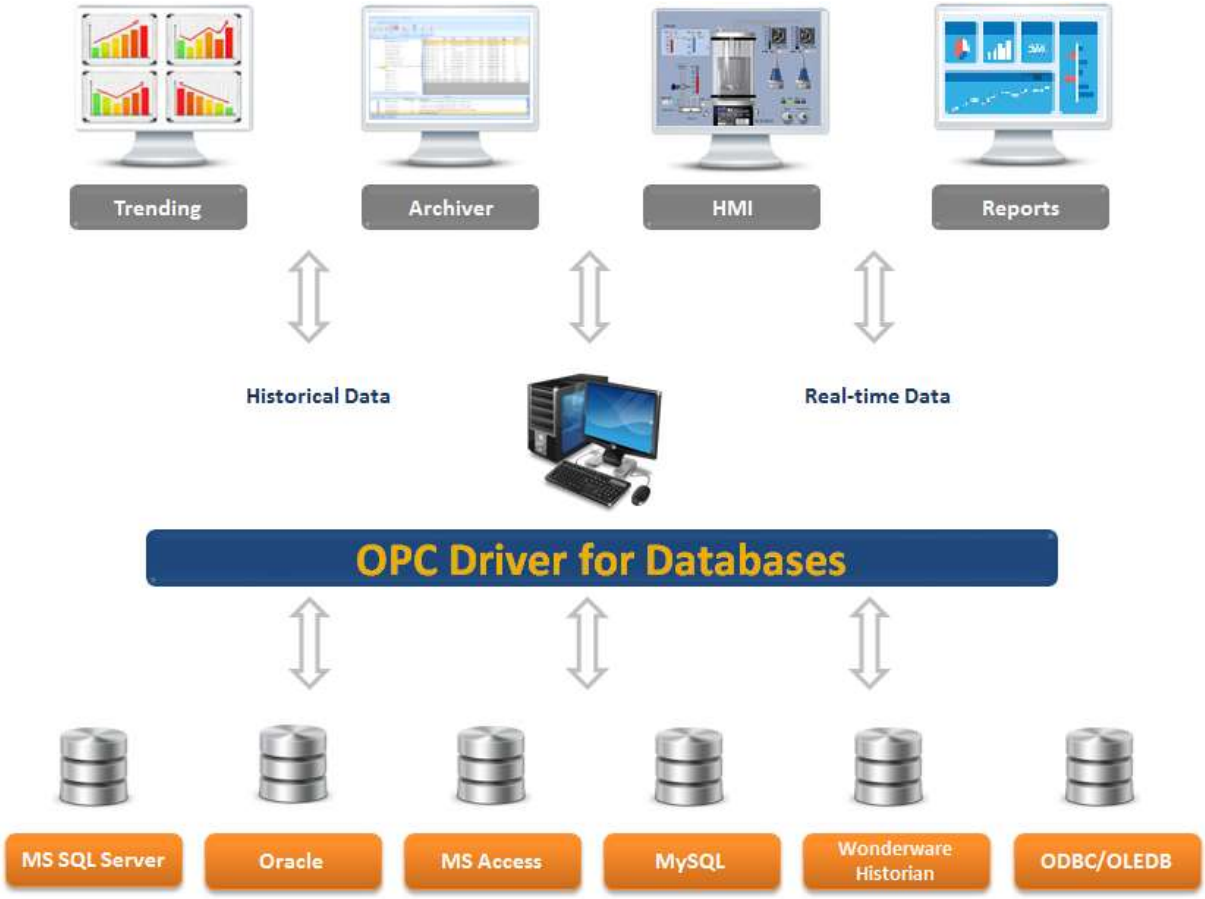

Figure 1: System Architecture

This OPC server can collect data from SQL Server, Oracle, MS Access, MySQL and InSQL databases via the ADO technology or any other database available in your network via OLEDB or ODBC providers.

3. OPC Compatibility

Integration Objects' OPC Driver for Databases implements:

- OPC Data Access specification version 2.05 and 3.0
- OPC Historical Data Access (HDA) specification version 1.1 and 1.2.

4. Features

This section details the features offered by this OPC server including OPC DA and HDA server/database configuration.

4.1. OPC Data Access Capabilities

This OPC server allows OPC DA Clients to retrieve in real-time the latest recorded data in the database. The following are the supported OPC DA interfaces:

Object	Interface	Supported
OPC DA Server	IUnknown	Yes
	IOPCCommon	Yes
	IOPCServer	Yes
	IConnectionPointContainer	Yes
	IOPCBrowseServerAddressSpace (Optional)	Yes
	IOPCItemProperties	Yes
OPC DA Group	IUnknown	Yes
	IOPCItemMgt	Yes
	IOPCGroupStateMgt	Yes
	IOPCPublicGroupStateMgt (Optional)	Yes
	IOPCSyncIO	Yes
	IOPCAsyncIO	Yes
	IOPCSyncIO2	Yes
	IOPCAsyncIO2	Yes
	IOPCAsyncIO3	Yes
	IOPCItemDeadbandMgt	Yes
	IConnectionPointContainer	Yes
	IOPCGroupStateMgt2	Yes

Table 1: Supported OPC DA Interfaces

4.2. OPC HISTORICAL DATA ACCESS CAPABILITIES

This OPC server allows OPC HDA clients to retrieve historical data from the connected databases. The retrieved data can be raw data using raw read and read at time requested or aggregated data, which is computed using analysis functions such as average, interpolation, etc. Historical data can also be updated. You can insert new data or replace existing raw(s). You can also delete raw(s) for a specified time or during a time range when required.

For more details, you can refer to the ***OPC HDA Fundamentals.pdf*** included with the installation of the OPC Driver for Databases.

The following are the current supported OPC HDA interfaces:

Object	Interface	Supported
OPCHDAServer	IOPCCommon	Yes
	IOPCHDA_Server	Yes
	IOPCHDA_SyncRead	Yes
	IOPCHDA_SyncUpdate (optional)	Yes
	IOPCHDA_SyncAnnotations (optional)	No
	IOPCHDA_AsyncRead (optional)	Yes
	IOPCHDA_AsyncUpdate (optional)	Yes
	IOPCHDA_AsyncAnnotations (optional)	No
	IOPCHDA_Playback (optional)	No
	IConnectionPointContainer	Yes
OPCHDABrowser	IOPCHDA_Browser	Yes

Table 2: Supported OPC HDA Interfaces

This server does not support the annotations and playback interfaces of the HDA standard, which both are optional interfaces.

- **Supported attributes**

These are the standard HDA attributes supported by the server:

- OPCHDA_DATA_TYPE: Specifies the data type for the item.
- OPCHDA_ITEMID: Specifies the item id.

- **Supported aggregates**

This server supports the following standard aggregates defined by the OPC HDA specification:

- OPCHDA_AVERAGE: The average data over the resample interval.
- OPCHDA_TOTAL: The total value (time integral) of the data over the resample interval.
- OPCHDA_COUNT: The number of raw values over the resample interval.
- OPCHDA_INTERPOLATIVE: Used for interpolated values.
- OPCHDA_START: The value at the beginning of the resample interval. The timestamp is the time of the beginning of the interval.

4.3. INTUITIVE USER INTERFACE FOR DATABASE CONFIGURATION

The OPC Driver for Databases offers an intuitive graphical user interface to configure the links to the databases. This configuration will be saved in an XML file and will be loaded at the next OPC server startup.

After startup, the server loads the configuration file to create tags and to build its address space. The server also retrieves all database information such as tables' names, mapping between tags and columns names to allow transactions from OPC server to database.

The OPC Driver for Databases can communicate with any compliant MS SQL Server database, Oracle, MySQL, Wonderware Historian, MS Access Database or any database using OLEDB or ODBC connection providers.

5. System Requirements

This driver was successfully installed and executed under the following operating systems:

- Windows XP SP2,
- Windows 2003 SP2,
- Windows Seven,
- Windows 8,
- Windows 10,
- Windows Server 2008,
- Windows Server 2012,
- and Windows Server 2016.

Moreover, you should take the following into consideration:

- Install the needed OLE DB providers
- Depending on your architecture, you may install:
  - Oracle,
  - Microsoft SQL Server,
  - MySQL,
  - Microsoft Access.
  - Required OPC DLLs (described in more details in the next chapter).
  - An OPC HDA client compliant with OPC HDA 1.1 and 1.2 standards.
  - An OPC DA client compliant with OPC DA2.05 and 3.0 standards. Click here if you need to download a client application.

6. Databases Compatibility

This driver is compatible with the following Database Management Systems:

- MS SQL Server 2005 or higher
- Oracle 8i or higher
- Microsoft Access 2003 or higher,
- MySQL 5.0 or higher,
- Wonderware Historian 9, 10, 2012 R2, 2014, 2014 R2, 2014 P01 and 2017.

GETTING STARTED

1. Pre-Installation Considerations

In order to properly run the OPC Driver for Databases, make sure to run the install program using an administrator account and have the following software components installed on the target system:

- The **OPC core components 3.00**, which consist of all shared OPC modules including the DCOM proxy/stub libraries, the OPC Server Enumerator, .NET wrappers, etc.
- .NET Framework version 4.0 or higher.

Make sure that there is no firewall or antivirus blocking the application.

The table below lists the prerequisites to communicate with databases per feature:

Feature	Database Connector Pre-requisite
OPC DA for MS SQL Server	Uses ADO .Net to communicate with the database. No pre-requisites need to be installed.
OPC HDA for MS SQL Server	Uses ADO .Net to communicate with the database. No pre-requisites need to be installed.
OPC DA for Oracle	Requires ODAC to communicate with the database.
OPC HDA for Oracle	Requires ODAC to communicate with the database.
OPC DA for MS Access	Requires Microsoft Office to be installed.
OPC HDA for MS Access	Requires Microsoft Office to be installed.
OPC DA for MySQL	Uses embedded MySQL connector to communicate with the database. No pre-requisites need to be installed.
OPC HDA for MySQL	Uses embedded MySQL connector to communicate with the database. No pre-requisites need to be installed.
OPC DA for ODBC	Requires the ODBC driver to be installed in order to communicate with the corresponding database source type.
OPC HDA for ODBC	Requires the ODBC driver to be installed in order to communicate with the corresponding database source type.
OPC DA for OLEDB	Requires the OLEDB driver to be installed in order to communicate with the corresponding database source type.

OPC HDA for OLEDB	Requires the OLEDB driver to be installed in order to communicate with the corresponding database source type.
OPC DA for Wonderware Historian	Uses ADO .Net to communicate with the database. No pre-requisites need to be installed.
OPC HDA for Wonderware Historian	Uses ADO .Net to communicate with the database. No pre-requisites need to be installed.

Table 3: Database Connector Pre-requisites

2. Installing OPC Driver for Databases

To install the OPC Driver for Databases:

1. Run the downloaded install program using an administrator account.
   The installation welcome dialog box will appear.

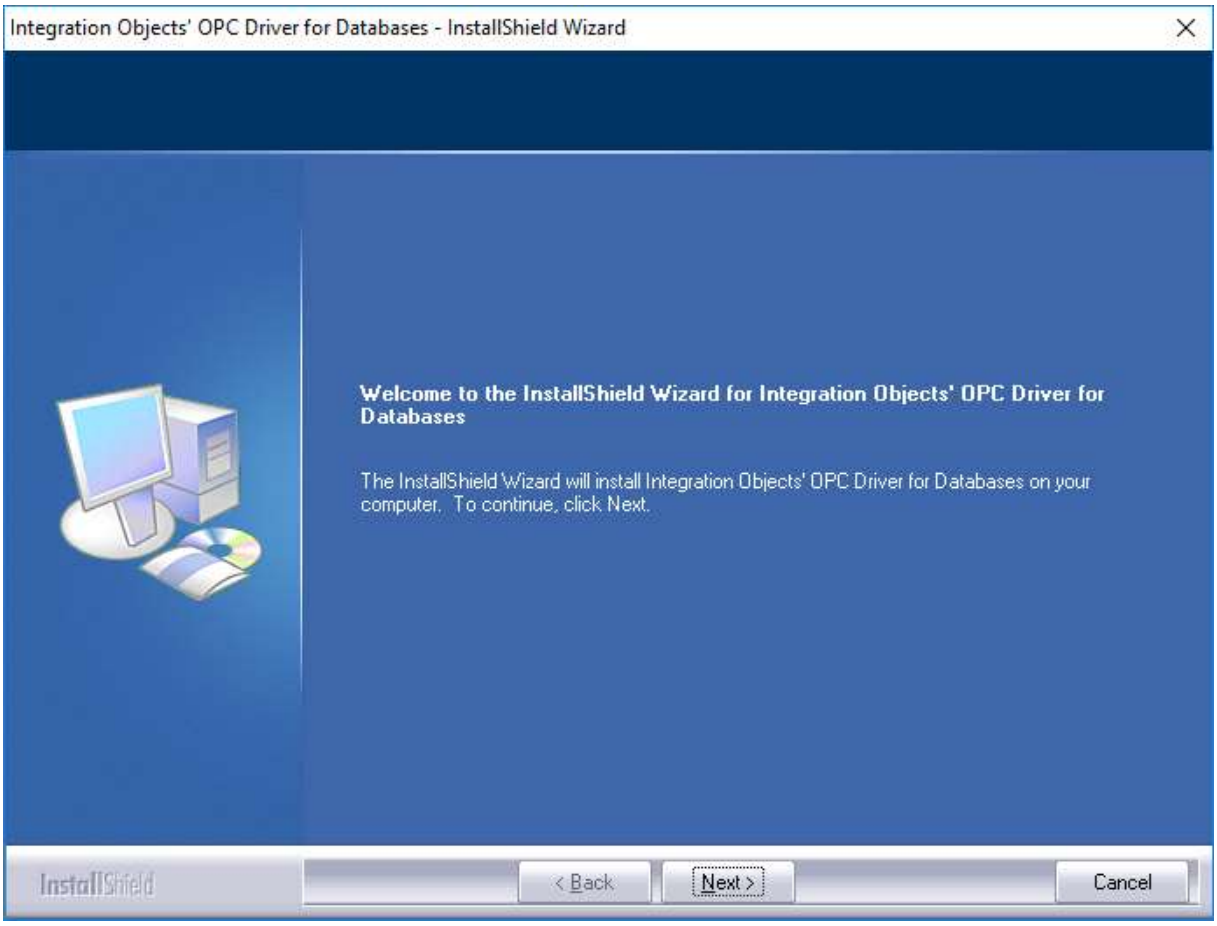

Figure 2: Installation Welcome Dialog Box

2. Click the **Next** button. The license agreement will be displayed

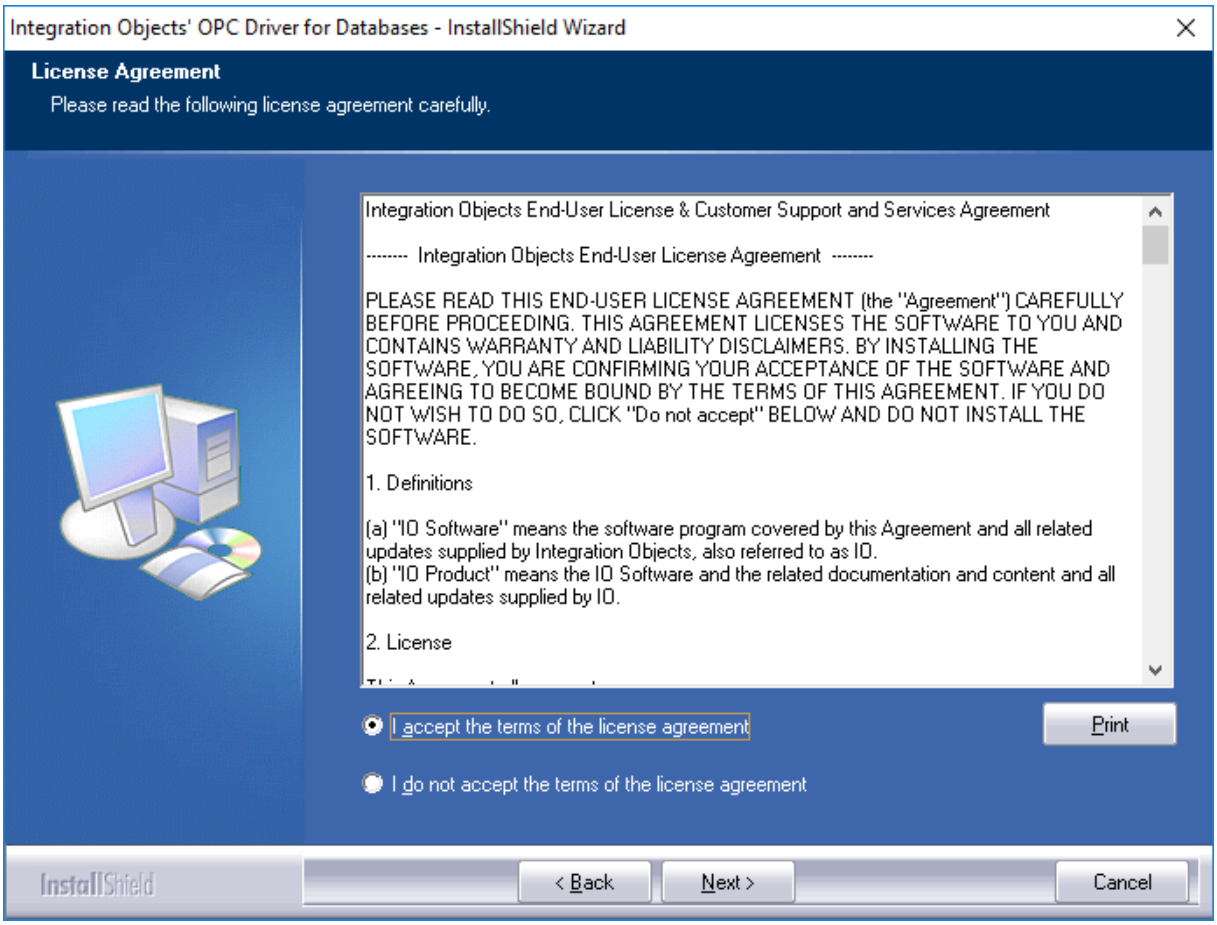

Figure 3: License Agreement Dialog Box

3. After reading the license agreement, select the first option and click the **Next** button. By proceeding, you are accepting all of the license agreement terms. Otherwise, you can cancel the installation. The customer information dialog box will then appear.

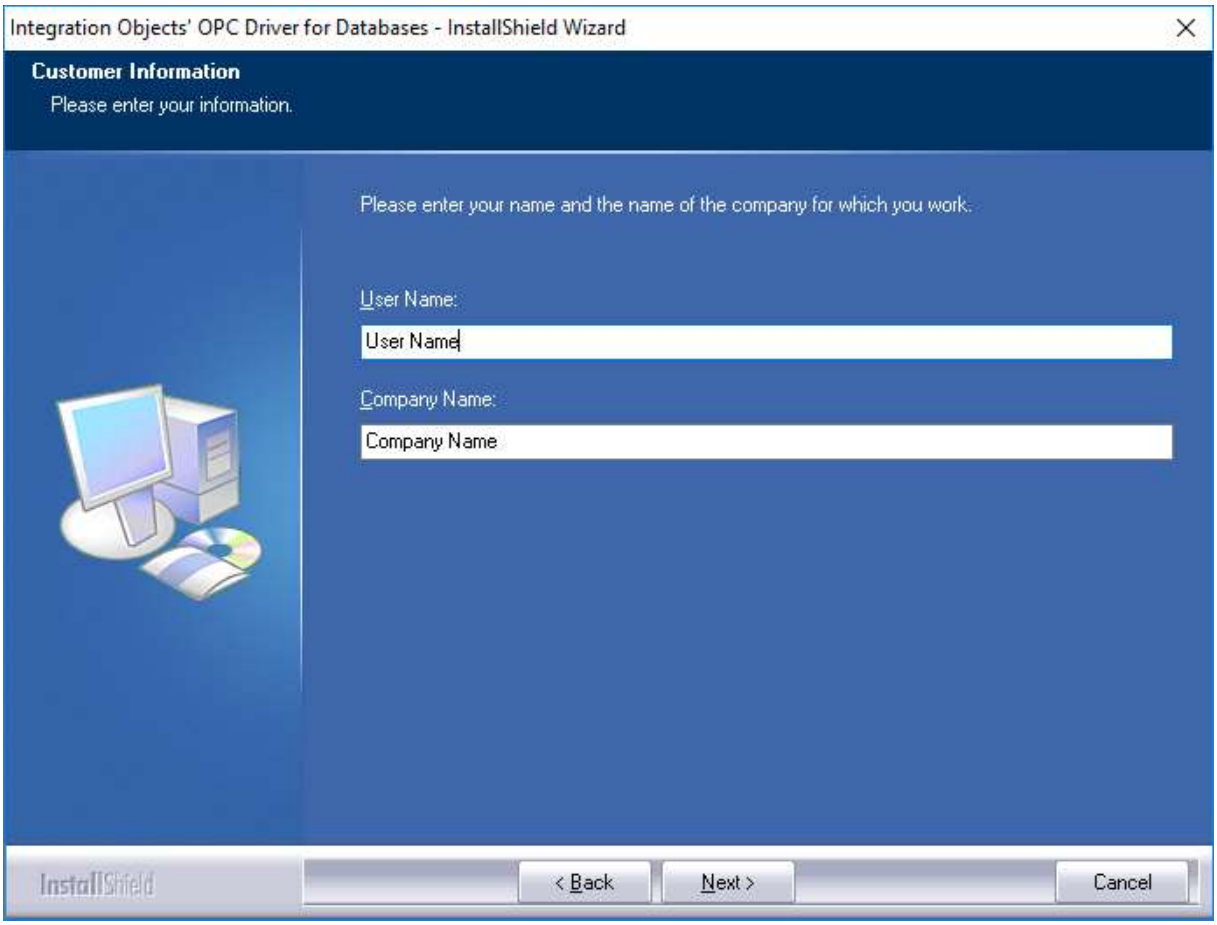

Figure 4: Customer Information Dialog Box

4. Enter the user name and the company name, and then click the **Next** button. The dialog box for selecting the setup type will be displayed.

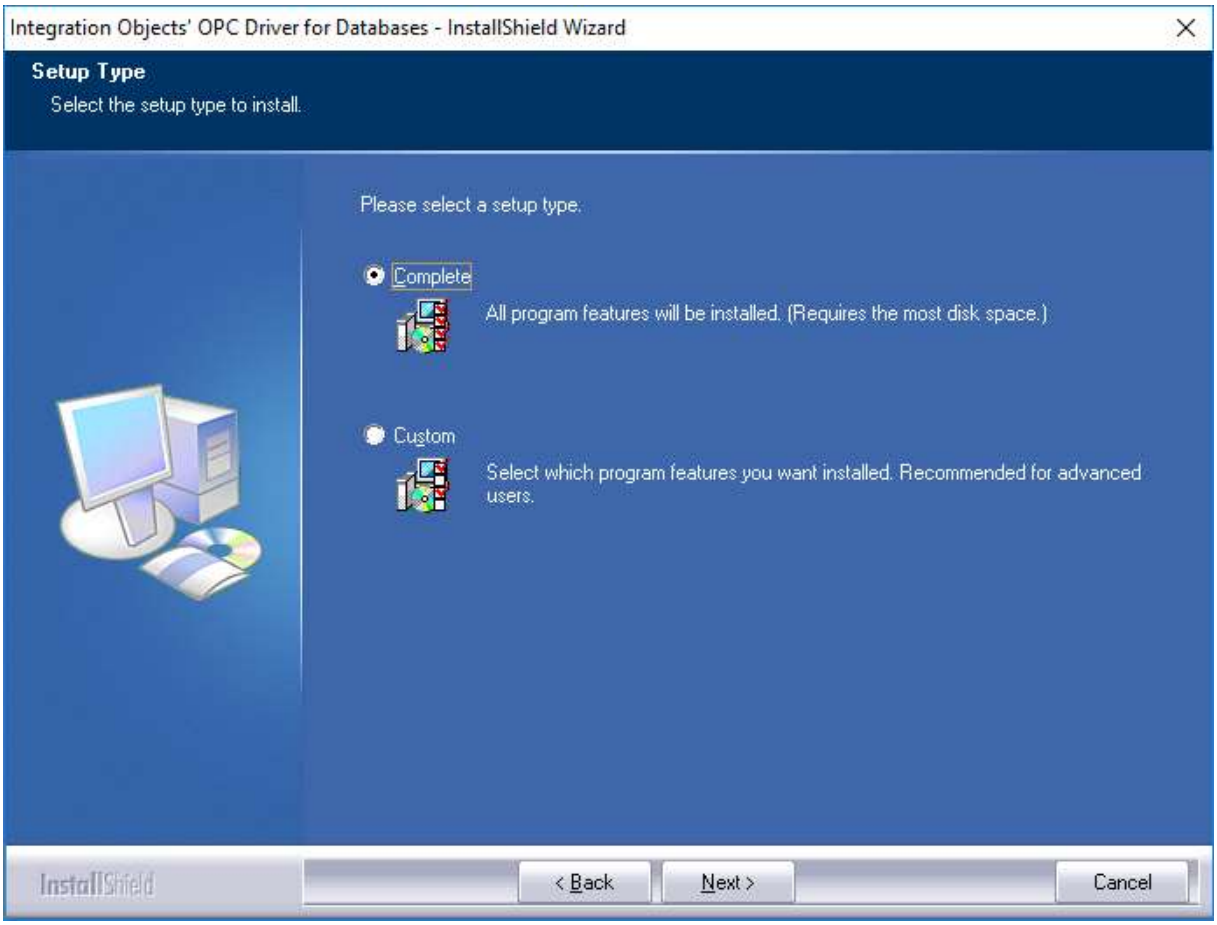

Figure 5: Setup Type Dialog Box

5. If you choose the "**Complete**" setup type, all features will be installed.
   If you choose "**Custom**" setup type, the following dialog box will be displayed and you will need to check the features you want to install:

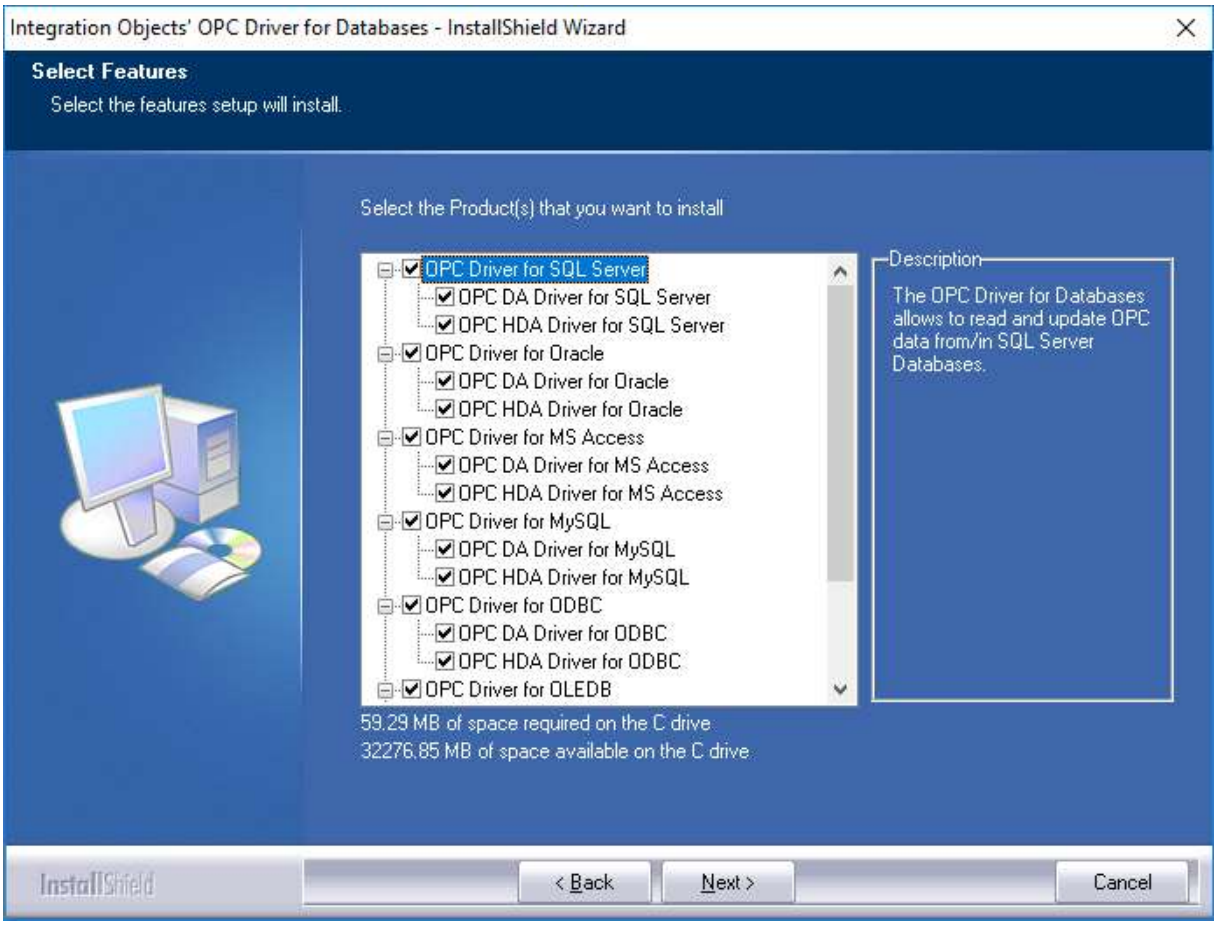

Figure 6: Features Dialog Box

- **OPC DA Driver for SQL Server** allows you to read and update OPC data from/in SQL Server databases through OPC Data Access specification.
- **OPC HDA Driver for SQL Server** allows you to read and update OPC data from/in SQL Server databases through the Historical Data Access specification.
- **OPC DA Driver for Oracle** allows you to read and update OPC data from/in Oracle databases through OPC Data Access specification.
- **OPC HDA Driver for Oracle** allows you to read and update OPC data from/in Oracle databases through OPC Historical Data Access specification.
- **OPC DA Driver for MS Access** allows you to read and update OPC data from/in Microsoft Access databases through OPC Data Access specification.
- **OPC HDA Driver for MS Access** allows you to read and update OPC data from/in Microsoft Access databases through OPC Historical Data Access specification.
- **OPC DA Driver for MySQL** allows you to read and update OPC data from/in MySQL databases through OPC Data Access specification.
- **OPC HDA Driver for MySQL** allows you to read and update OPC data from/in MySQL databases through OPC Historical Data Access specification.
- **OPC DA Driver for ODBC** allows you to read and update OPC data from/in ODBC databases through OPC Data Access specification.

- **OPC HDA Driver for ODBC** allows you to read and update OPC data from/in ODBC databases through OPC Historical Data Access specification.
- **OPC DA Driver for OLEDB** allows you to read and update OPC data from/in OLEDB databases through OPC Data Access specification.
- **OPC HDA Driver for OLEDB** allows you to read and update OPC data from/in OLEDB databases through the Historical Data Access specification.
- **OPC DA Driver for InSQL** allows you to read and update OPC data from/in Wonderware Historians through OPC Data Access specification.
- **OPC HDA Driver for InSQL** allows you to read and update OPC data from/in Wonderware Historians through OPC Historical Data Access specification.

6. After selecting the features to be installed, a dialog box, to choose the destination folder from, will be displayed.

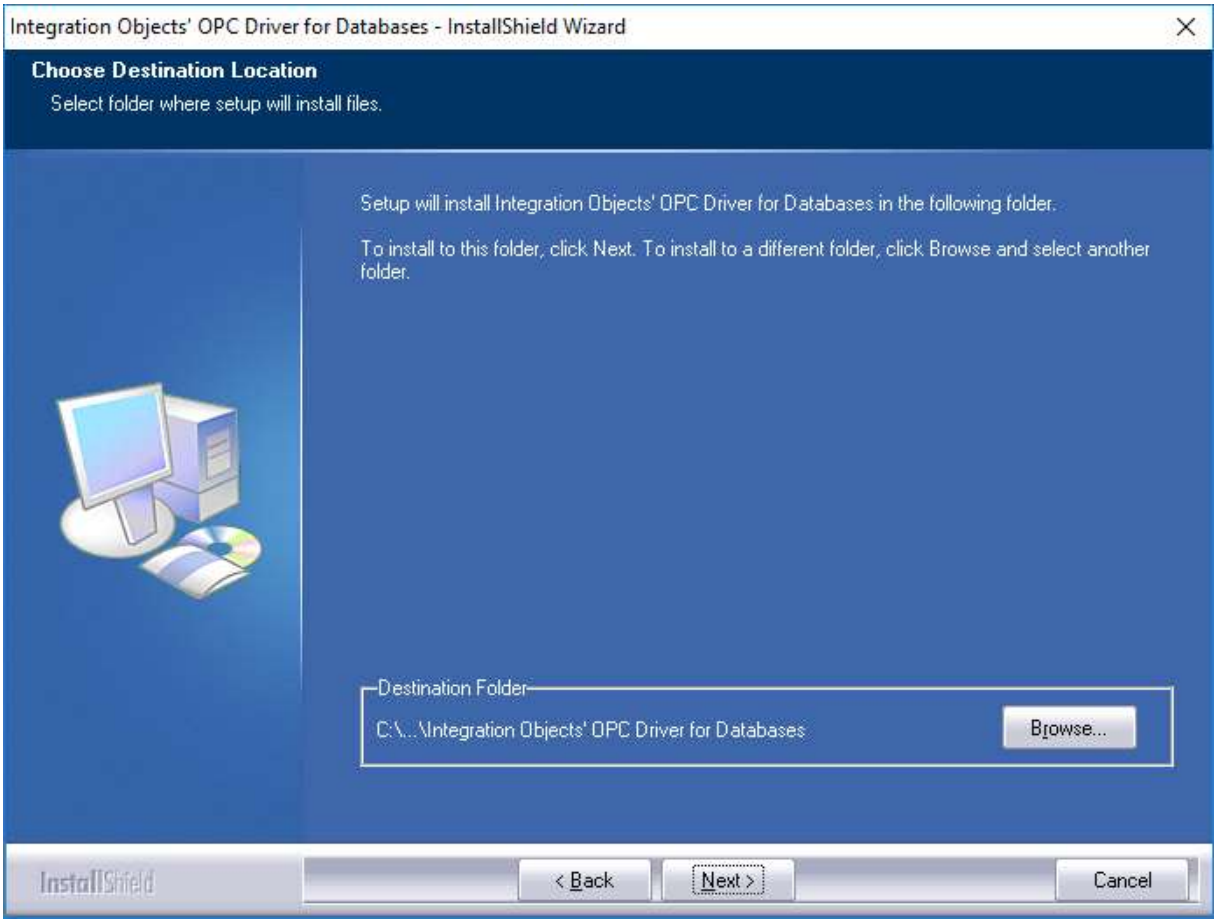

Figure 7: Choose Destination Folder Dialog Box

7. Click the **Next** button to continue the installation, or the **Browse** button to choose a different destination folder. The installation dialog box will then appear.

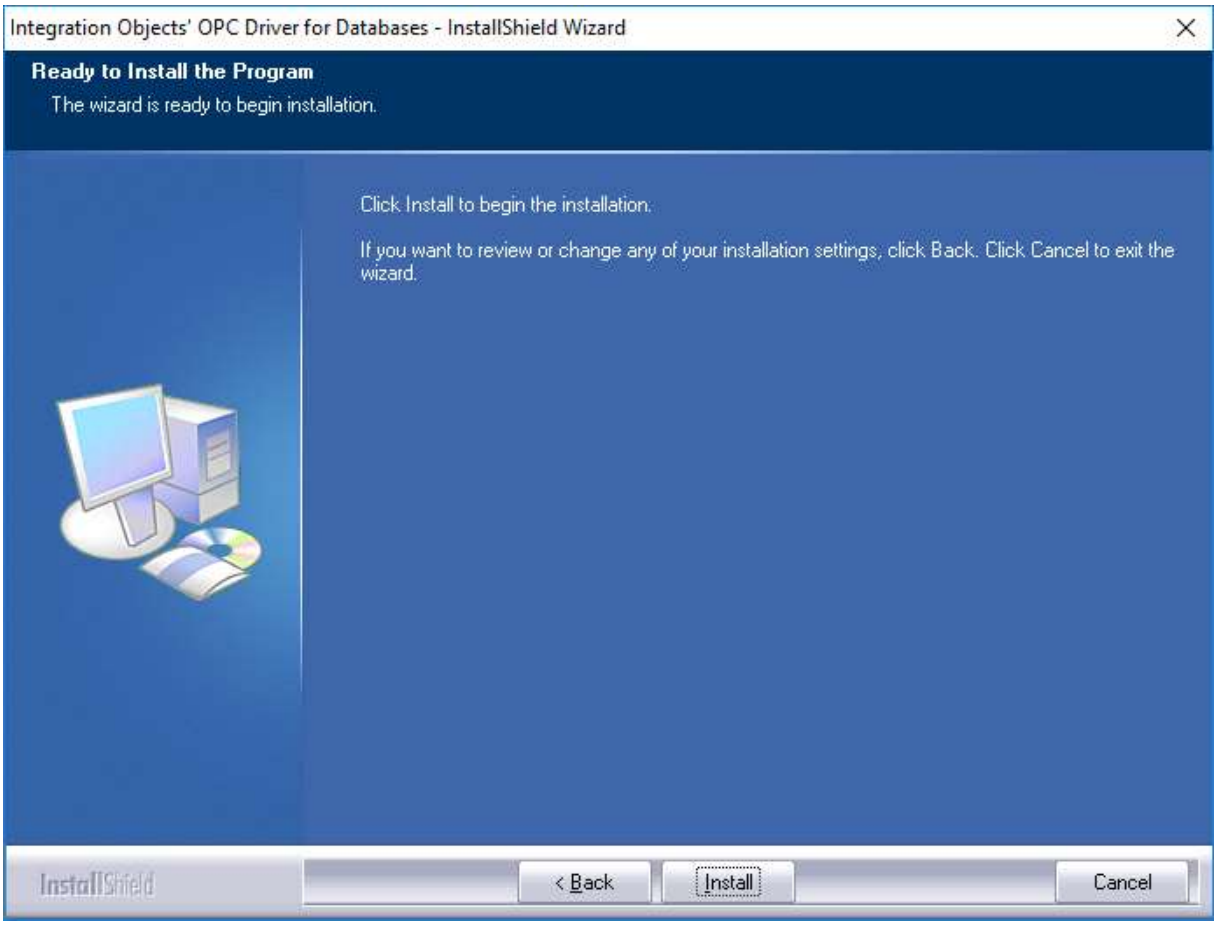

Figure 8: Installation Dialog Box

8. Click the **Install** button to start installation.

   The setup will then copy the necessary files to the selected target folder, create shortcut icons to launch the OPC Driver for Databases and authorization license program from the start menu and the desktop, and make an un-installation entry in the Add/Remove Programs in the Control Panel.

9. Before the completion of the installation, the following dialog will be displayed in order to configure the user account that will be used to run the OPC Driver for Databases services:

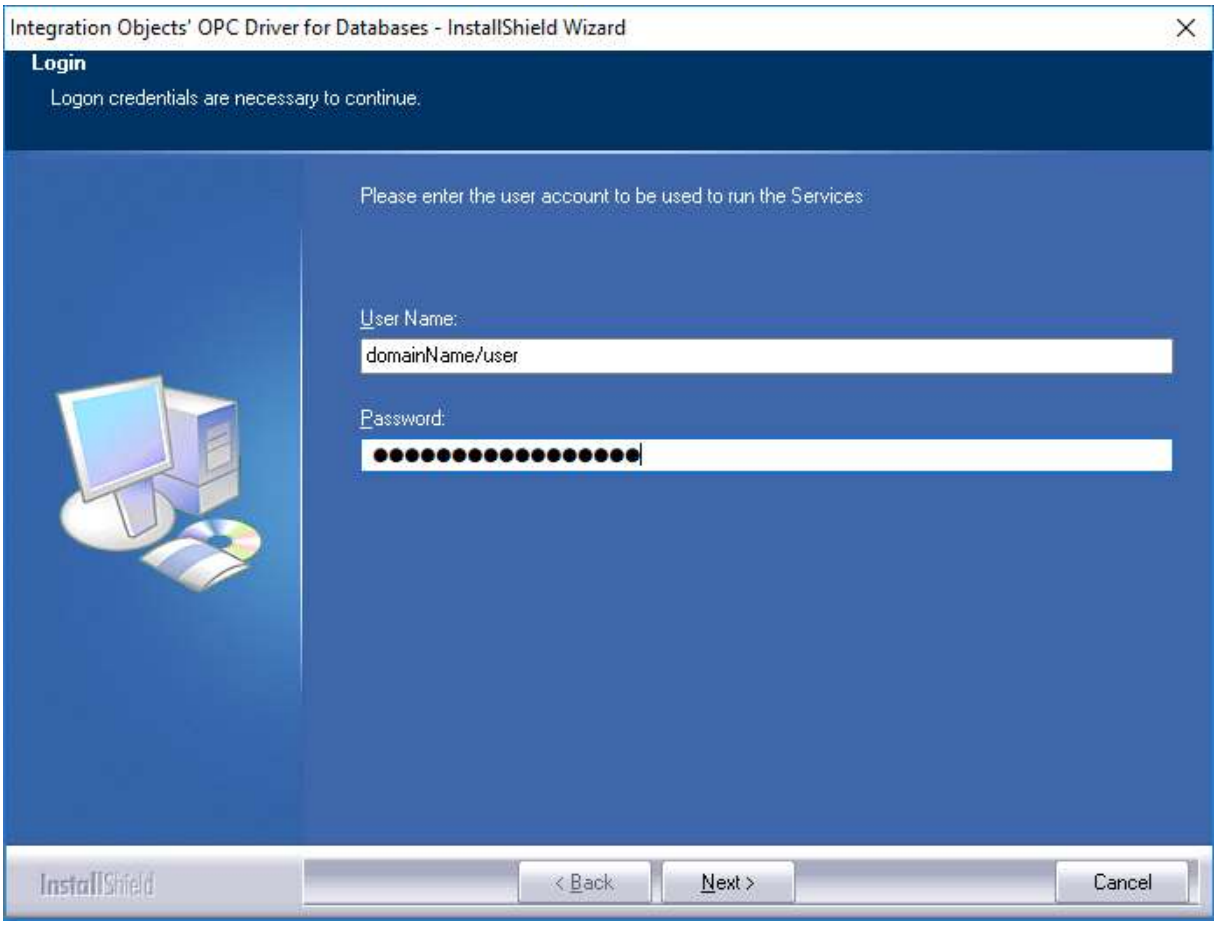

Figure 9: Configure Service Account

If you do not enter a valid account, the Local System account will be used to run the services. You can still modify this configuration after the installation using Windows services manager.

10. Click the **finish** button.

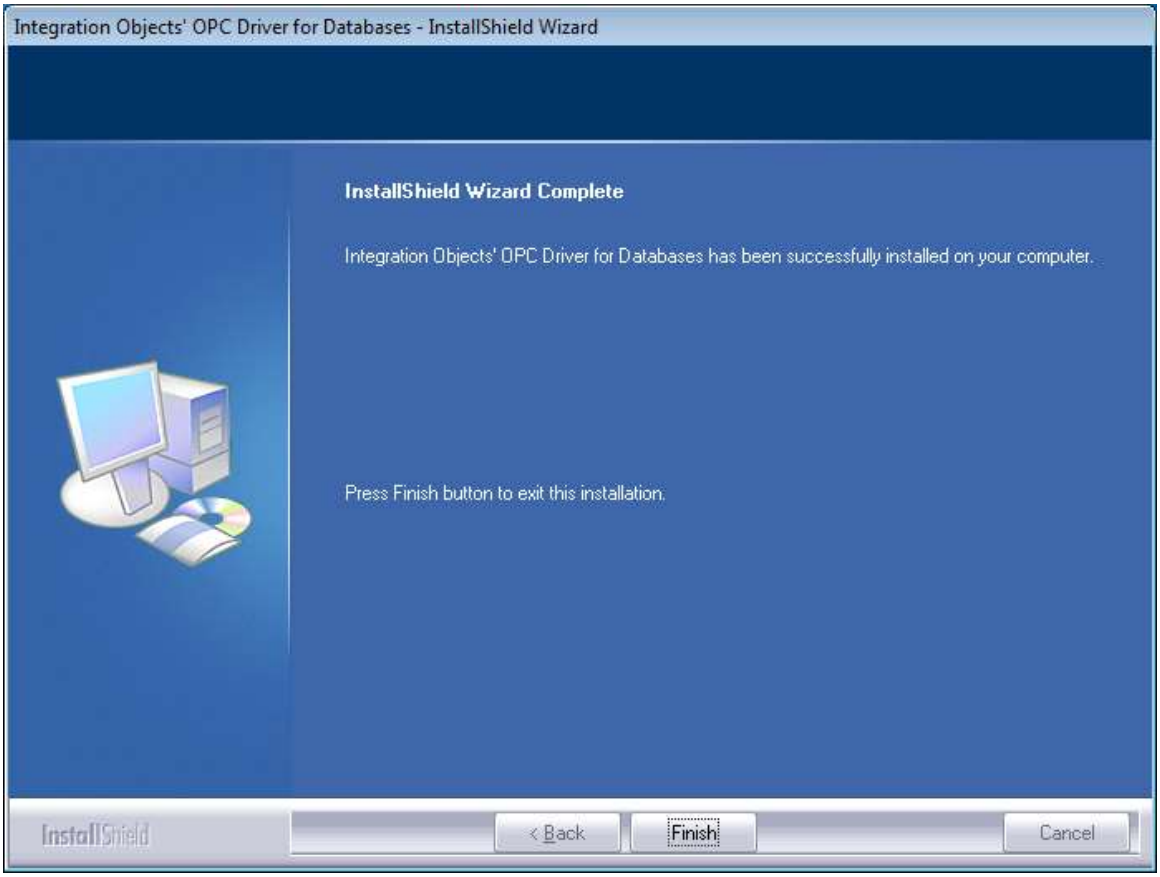

Figure 10: Installation Completed Dialog Box

3. Starting-up

The OPC Driver for Databases can be started manually from the start menu shortcut. To do so, click on Start ➔ Programs ➔ Integration Objects ➔ OPC Driver for Databases➔OPC Driver for Databases

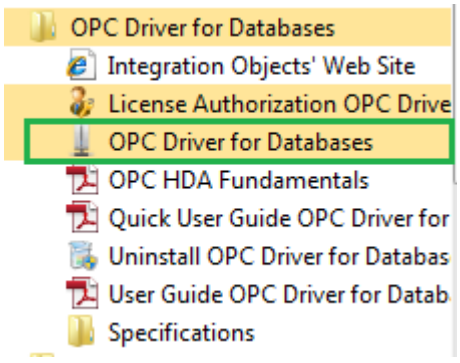

Figure 11: Starting the OPC Driver for Databases

The server can also be activated dynamically when an OPC DA or HDA client is connecting to it. If the server is not currently running, COM will launch it at the first OPC client connection.

4. Server Registration

In compliance with the OPC and COM specifications, the OPC Driver for Databases creates the following registry entries under HKEY_CLASSES_ROOT when installed on the target system. These entries will be removed when the server is uninstalled:

Registry Entry	Description
OPC DA Driver registry entries	
IntegrationObjects.OPCDriverForDatabases.1	Integration Objects' OPC Driver for Databases; http://www.integrationobjects.com
IntegrationObjects.OPCDriverForDatabases.1\CLSID	{**CLSID**} = {81ACD3D7-9F39-4776-845E-0AD85AB3306B}
CLSID\{**CLSID**}	Integration Objects' OPC Driver for Databases; http://www.integrationobjects.com
CLSID\{**CLSID**}\AppID	{**CLSID**}
CLSID\{**CLSID**}\ProgID	IntegrationObjects.OPCDriverForDatabases.1
OPC HDA Driver registry entries	
IntegrationObjects.OPCHDADriverForDatabases.1	Integration Objects' OPC HDA Driver for Databases; http://www.integrationobjects.com
IntegrationObjects.OPCHDADriverForDatabases.1\CLSID	{***CLSID***} = {BF6DEB69-A380-418F-BF86-E4DA44AE7962}
CLSID\{***CLSID***}	Integration Objects' OPC HDA Driver for Databases; http://www.integrationobjects.com
CLSID\{***CLSID***}\AppID	{***CLSID***}
CLSID\{***CLSID***}\ProgID	IntegrationObjects.OPCHDADriverForDatabases.1

Table 4: OPC Driver for Databases Registry Entries

5. Removing OPC Driver for Databases

You can remove the OPC driver from your machine by clicking on the "**Uninstall OPC Driver**" shortcut from the start menu.

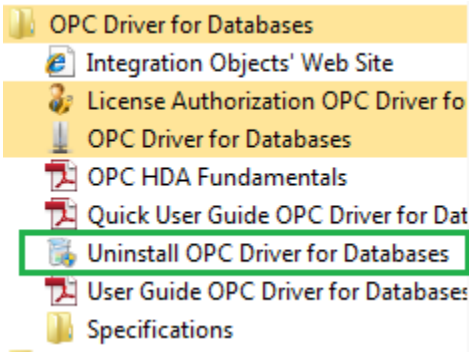

Figure 12: Start Menu – Uninstall Shortcut

If you are using Windows 10, Windows Server 2012 or windows Server 2016 operating systems, the uninstaller needs to be run from the start menu as illustrated below

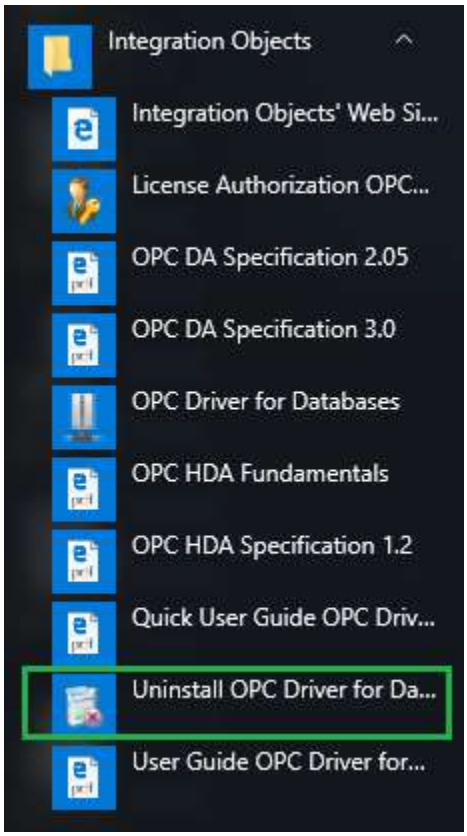

Figure 13: Windows 10 Startup Menu - Uninstall Shortcut

This OPC server can also be removed manually as follows:

1. Click **Start**.
2. Click **Settings**.
3. Click **Control Panel**.
4. Click **Add/Remove Programs**.
5. In Add/Remove Programs dialog screen select "**Integration Objects' OPC Driver for Databases**".
6. Click **Change/Remove** then **OK**.

CONFIGURING OPC DRIVER FOR DATABASES

1. Graphical User Interface Overview

Users can configure the OPC Driver for Databases with an intuitive GUI (Graphical User Interface).

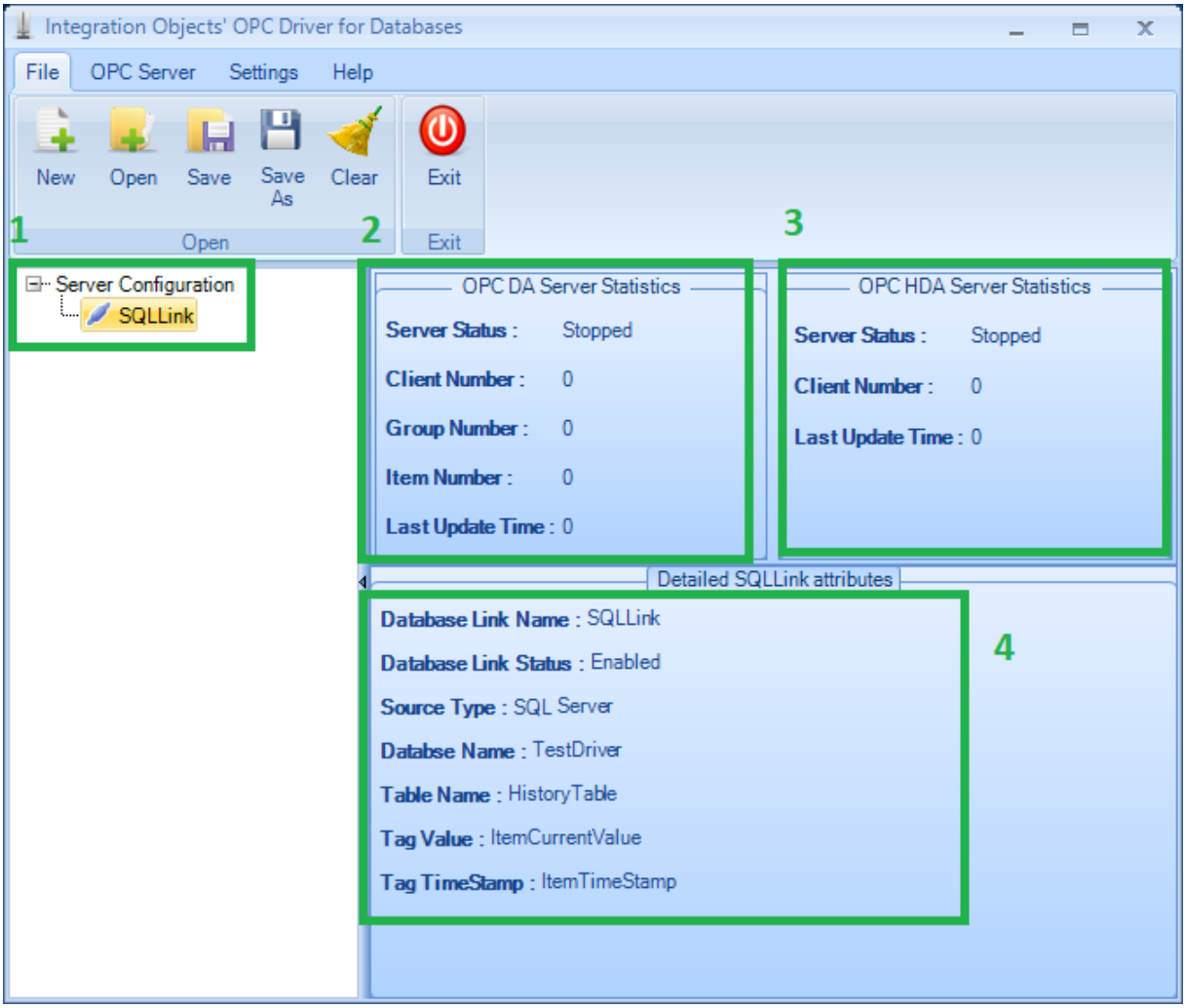

Figure 14: OPC Driver for Databases – Main View

We distinguish the following main sections:

- Server Configuration (1): describes how to configure links to any compliant database.

- DA Server statistics (2): displays the OPC DA server status (stopped or running or suspended), the connected clients number, the created groups number, the created items number and the last updated time.
- HDA Server statistics (3): displays the OPC HDA server status (stopped or running or suspended), the connected clients number and the last updated time.
- Detailed link's attributes (4): displays detailed attributes information related to the selected database link such as database connection name, the configured table name and the mapped tag values.

1.1. FILE MENU

Figure 15: File Menu

Using the File menu, you can**:**

- Create new configuration by clicking on New.
- Open an existing configuration by clicking on "Open" and selecting the appropriate ".xml" configuration file.
- Save your current configuration by clicking "Save "or "Save As".
- Clear the current configuration by clicking on "Clear".

1.2. OPC SERVER MENU

1.2.1. SERVER REGISTRATION

The OPC Server is registered automatically during the installation. The end user can also use the registration menu in the user interface to manually register/unregister the server.

Figure 16: OPC Server Menu

1.2.2. OPC DA Driver for Databases Service Management

Figure 17: OPC DA Driver for Databases Service

Under the same OPC Server menu, you can start or stop the OPC DA Driver for Databases service.

1.2.3. OPC HDA Driver for Databases Service Management

You can also manage the OPC HDA Server service by starting or stopping it as shown the figure below.

Figure 18: OPC HDA Driver for Databases Service

1.3. Settings Menu

Using the Settings menu, you can**:**

- Define the default configuration that will be loaded automatically when you restart the application.
- Remove the default configuration by clicking on the "Remove" button.
- Set up the configuration parameters through the displayed window when you click on the "Configure" button.
- Open the application log file by clicking on the “Log File” button.

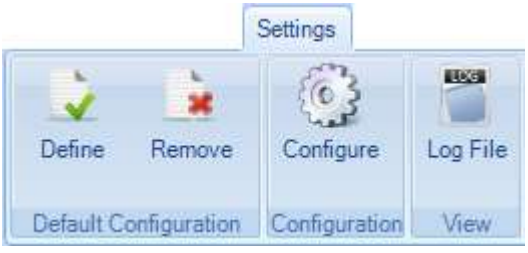

Figure 19: Settings Menu

When the user clicks the **Configure** button, the Settings window will be prompted.

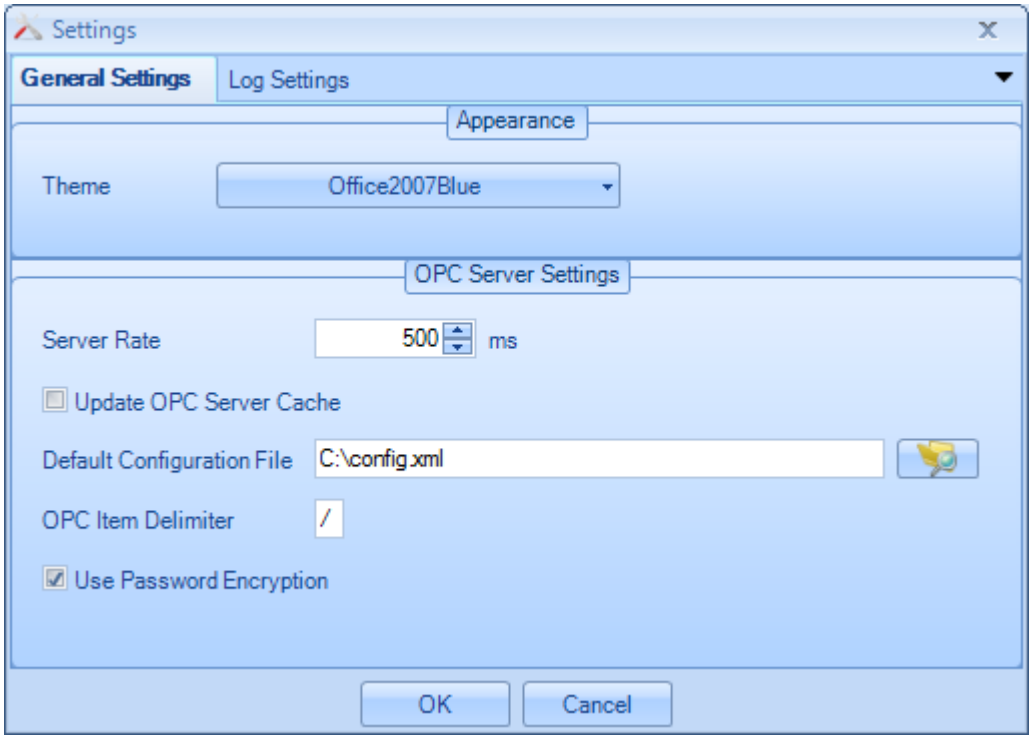

Figure 20: General Settings

Under the **General Settings** tab, you can update the following OPC Server parameters:

Parameter	Description	Default Value
Server Rate	This parameter is the maximum frequency at which the server handles the asynchronous reads/updates.	500 ms (milliseconds)
Update OPC Server Cache	**True:** Update OPC Server cache from database. **False:** Disable the update OPC Server cache procedure.	True
Default Configuration File	The path of the startup configuration file. This configuration will be loaded automatically at the application start-up.	Empty
OPC Item Delimiter	The character used to specify the boundary between OPC Tags names and the database link name in the address space	/
Use Password Encryption	**True**: Servers passwords will be encrypted in the configuration	True

	files. **False**: Servers passwords will not be encrypted in the configuration files	

Table 5: OPC Server Parameters

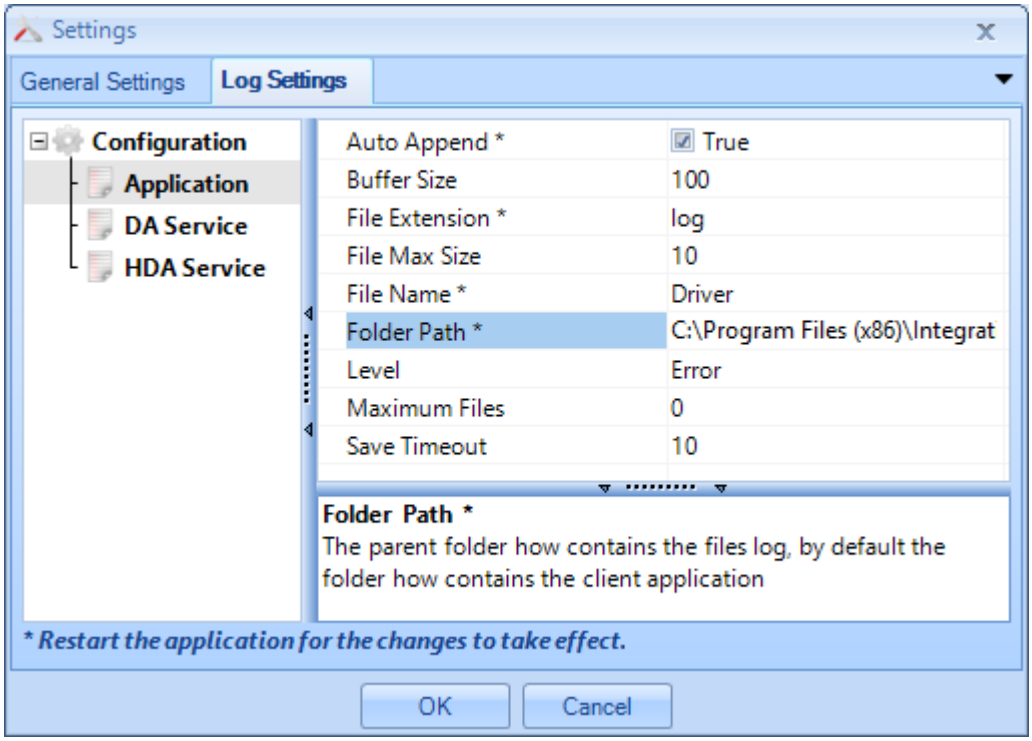

Figure 21: Log Settings

Under the **Log Settings** tab, you can set the following parameters for the GUI, the DA Service and the HDA Service:

Log Setting	Description	Default Value
Auto Append	Set to true to continue writing log messages in the existed log file or to false to create a new file.	True
Buffer Size	The maximum number of messages to be stored in the runtime memory before launching a write action in the hard disk. The specified value must be greater than 100.	100
File Max Size	This is the maximum log file size, in Mega-Bit. Once it is reached, the OPC Driver for Databases will automatically create a new log file and archive the last one.	10MB

Level	There are five log levels: 1. Control: Logs only control messages. This log level is the lowest level. 2. Error: Logs error and control messages. 3. Warning: Logs warning, error and control messages 4. Inform: Logs information, warning, error and control messages. 5. Debug: Logs all messages. This is the highest level. The higher the log level, the more information are recorded.	Error
Maximum Files	Set to 0 means that log files will be created in an unlimited way.	0
Save Timeout	Specifies the time period to wait before writing the log messages stored in the in-memory buffer to the hard disk. Note that the minimum value is 10 seconds.	60 s

Table 6: Log Settings

2. Databases Configuration

In this section, we will describe how to configure the connections to ADO-Compliant database (Oracle, SQL Server, etc.) or any other database via OLEDB or ODBC providers. The OPC Driver for databases main view as shown below lists at the left side the configured databases in Tree view format under the "Server Configuration" node.

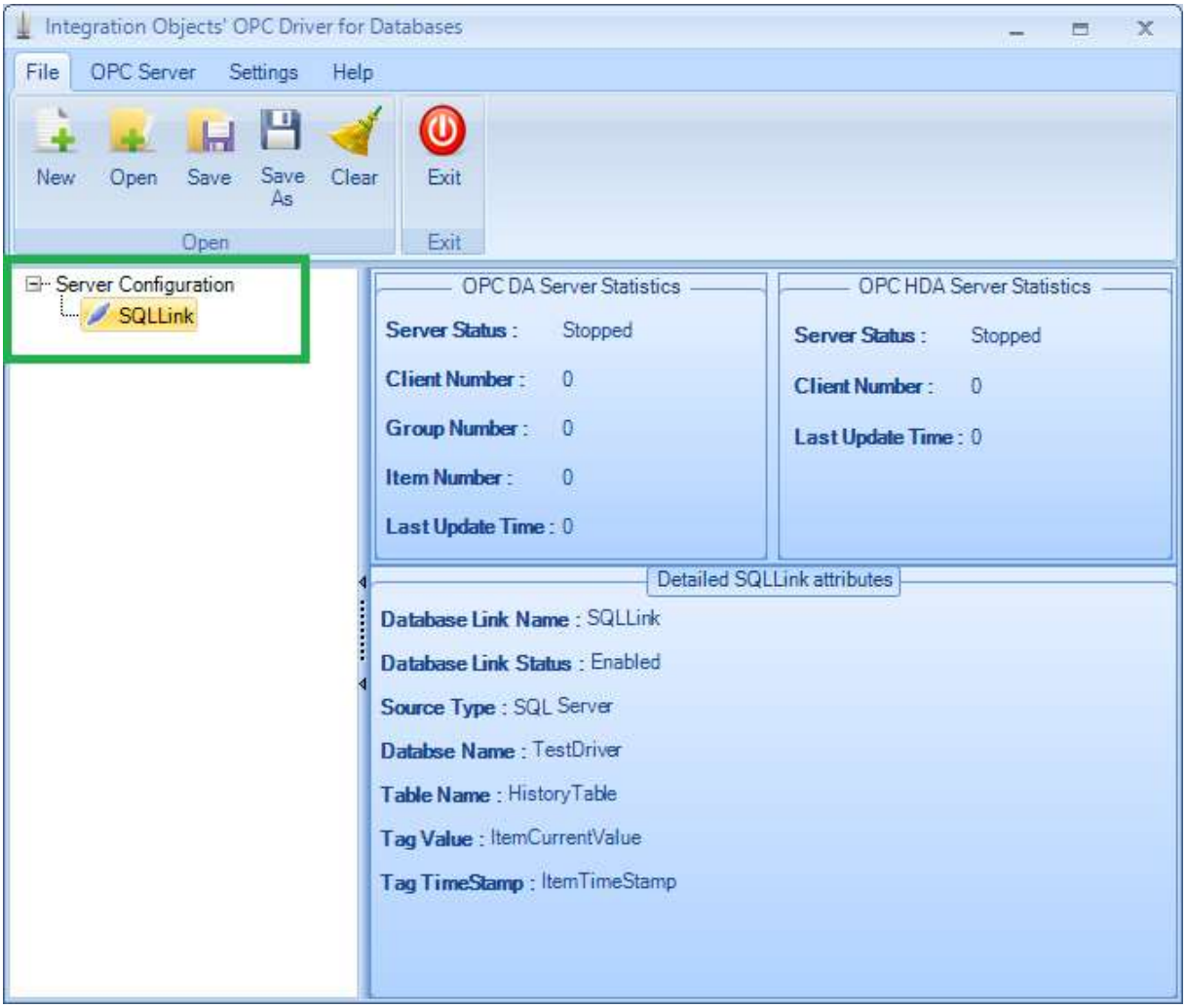

Figure 22: OPC Driver for Databases – Tree View

The below describes how to add, remove, view and edit the specified database configuration.

2.1. ADDING NEW DATABASE LINK

Right click on the "Server Configuration" node, and select **"New database link"** from the displayed server menu strip.

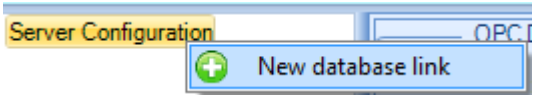

Figure 23: Add a New Database Link

Then, the new database link wizard will be displayed.

2.1.1. STEP 1

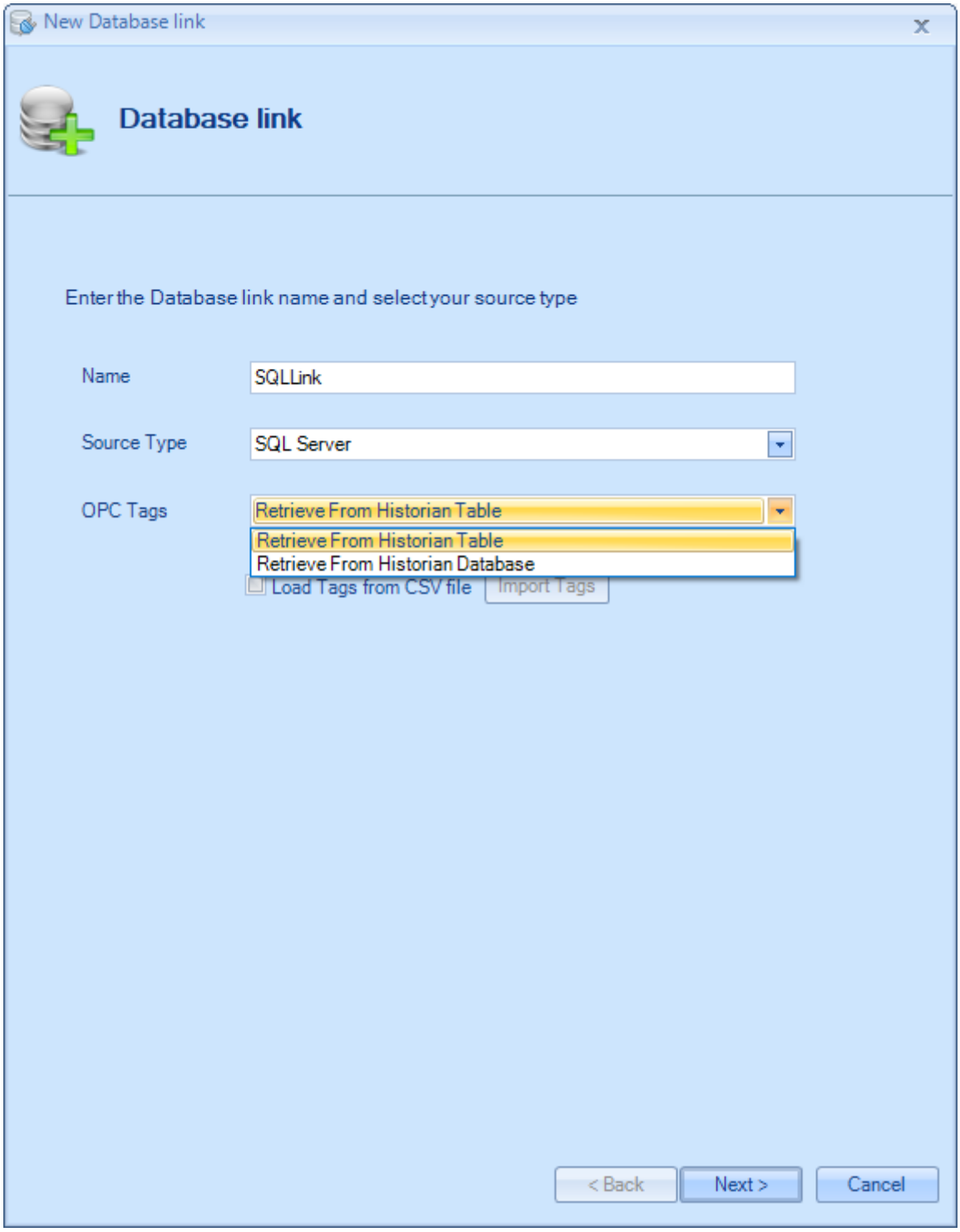

Figure 24: Add New Database Link

Parameter	Description
Name	The server link name.
Source Type	Specifies the provider. Currently, this version supports: • DB Provider for Oracle • DB Provider for SQL Server • DB Provider for MSAccess • DB Provider for MySQL • DB Provider for InSQL • ODBC • OLEDB
OPC Tags	Specifies how to retrieve OPC Tags list. • Retrieve From Historian Table • Retrieve From Historian Database
Load Tags from CSV file	Use this option to specify the tags to be included in the OPC Server address space. This option overrides the automatic discovery of the tags.

Table 7: Database Link Parameters

2.1.2. STEP 2

Click the next button to configure the connection string related to the selected database provider.

- **Microsoft SQL Server**

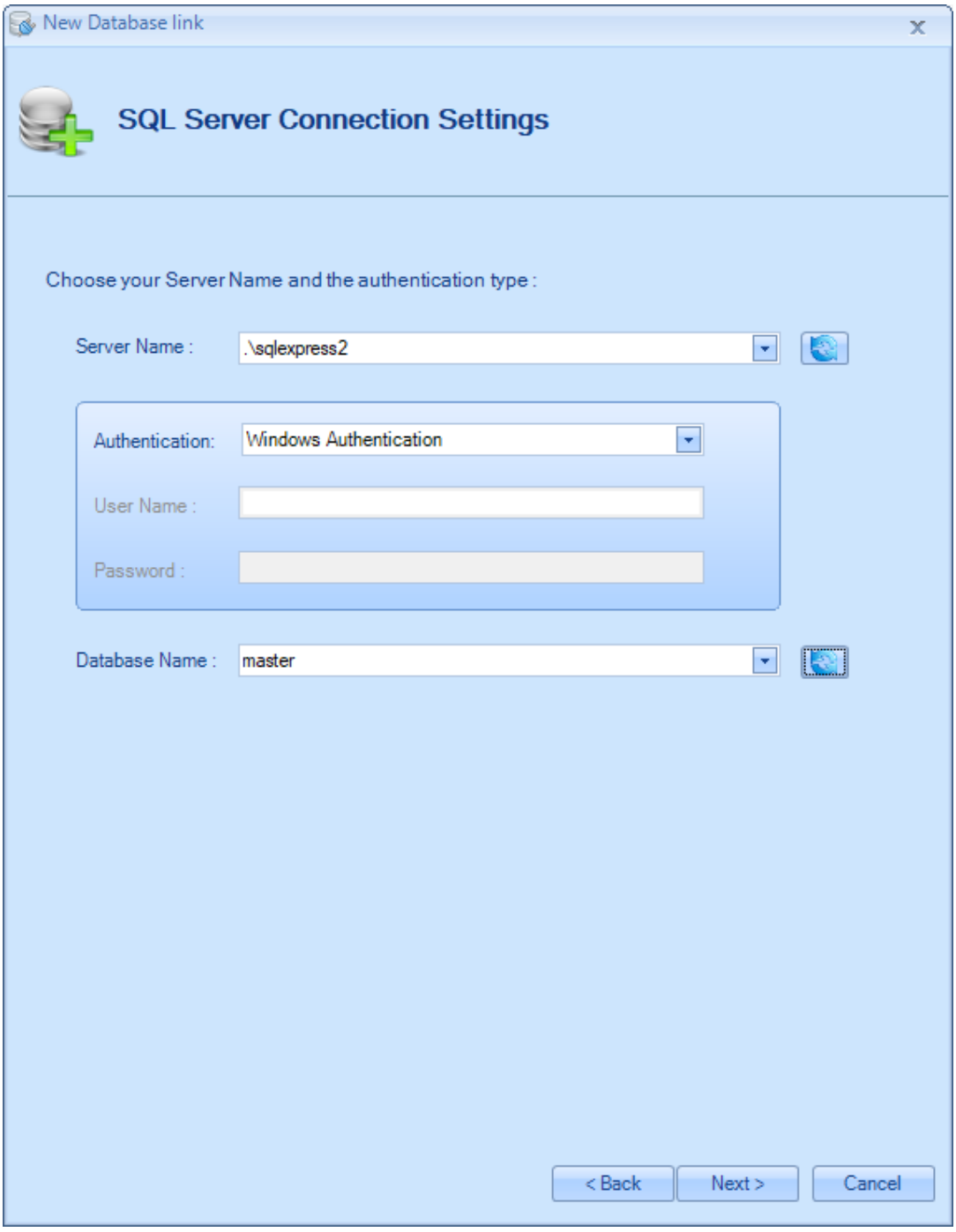

Figure 25: SQL Server Connection Settings

Parameter	Description
Server Name	SQL Server instance name
Authentication	Used to specify the SQL Server authentication mode: • Windows Authentication • SQL Server Authentication

User name	The SQL Server instance user name
Password	The SQL Server instance password
Database name	The SQL Server database name

Table 8: SQL Server Connection Parameters

- **Oracle**

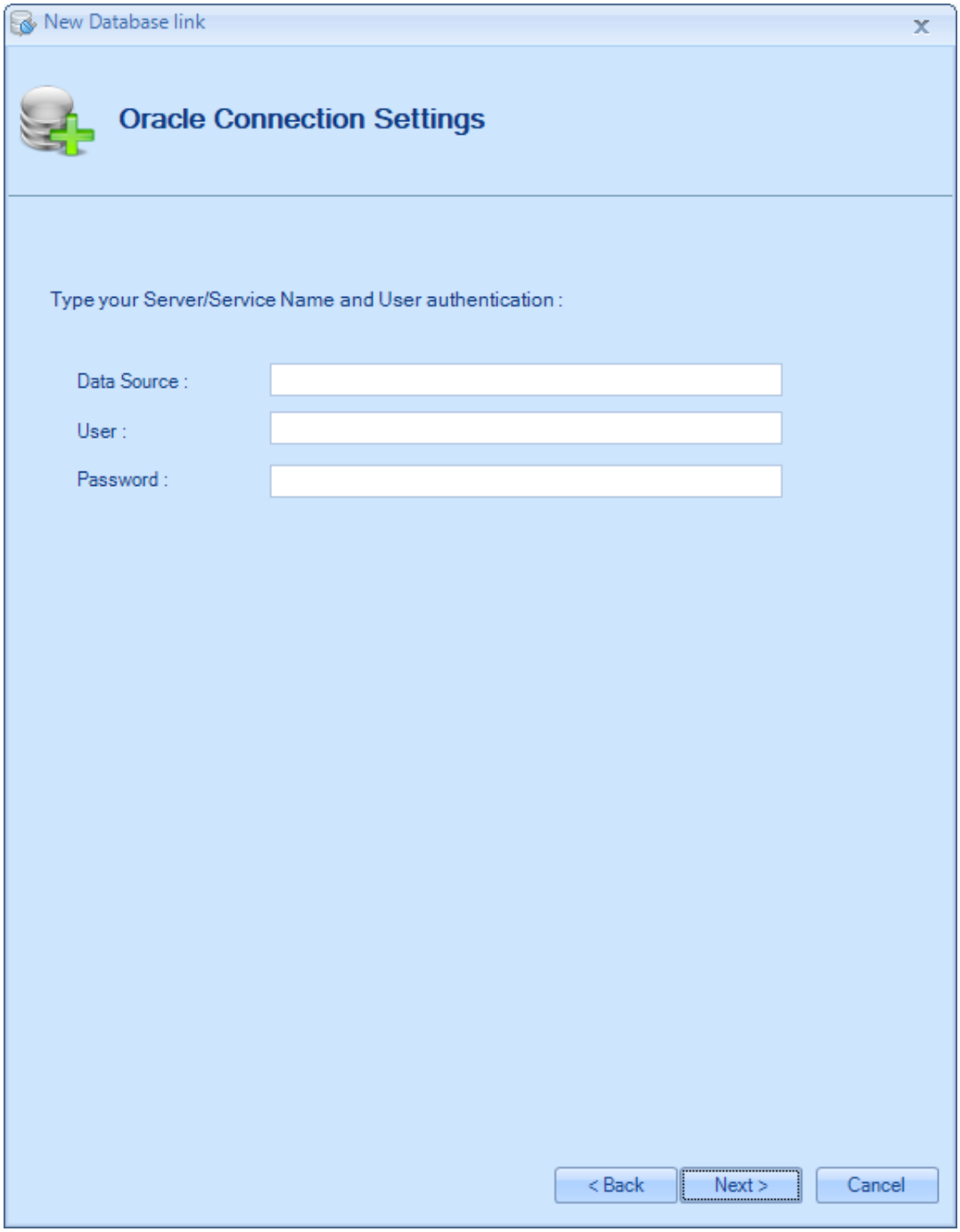

Figure 26: Oracle Connection Settings

Parameter	Description
Data Source	Oracle Server Instance name
User	The Oracle Server instance user name
Password	The Oracle Server instance password

Table 9: Oracle Connection Parameters

- **Microsoft Access**

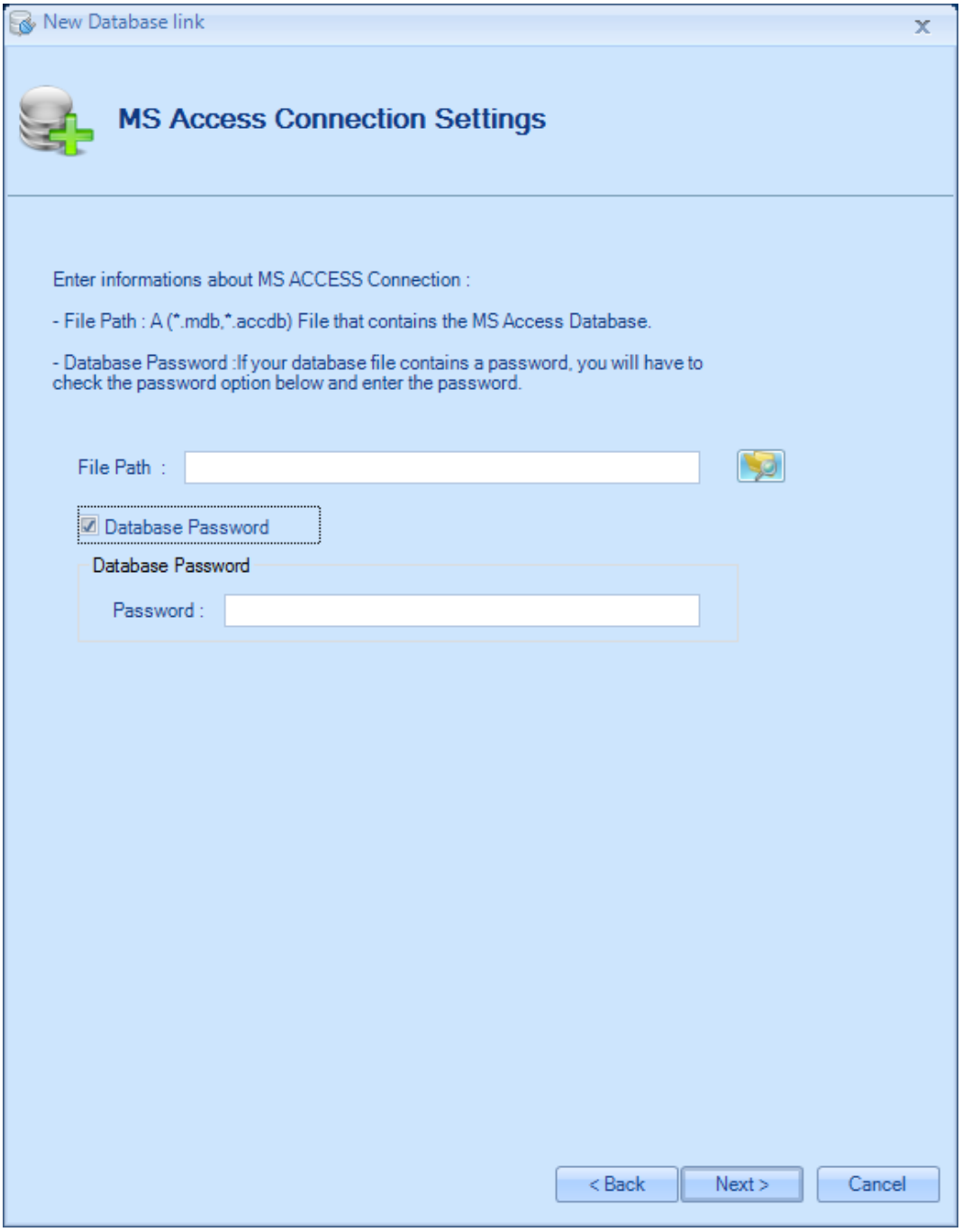

Figure 27: Microsoft Access Connection Settings

Parameter	Description
File Path	Microsoft Access file path
Password	Microsoft Access database password

Table 10: Microsoft Access Connection Parameters

- **MySQL**

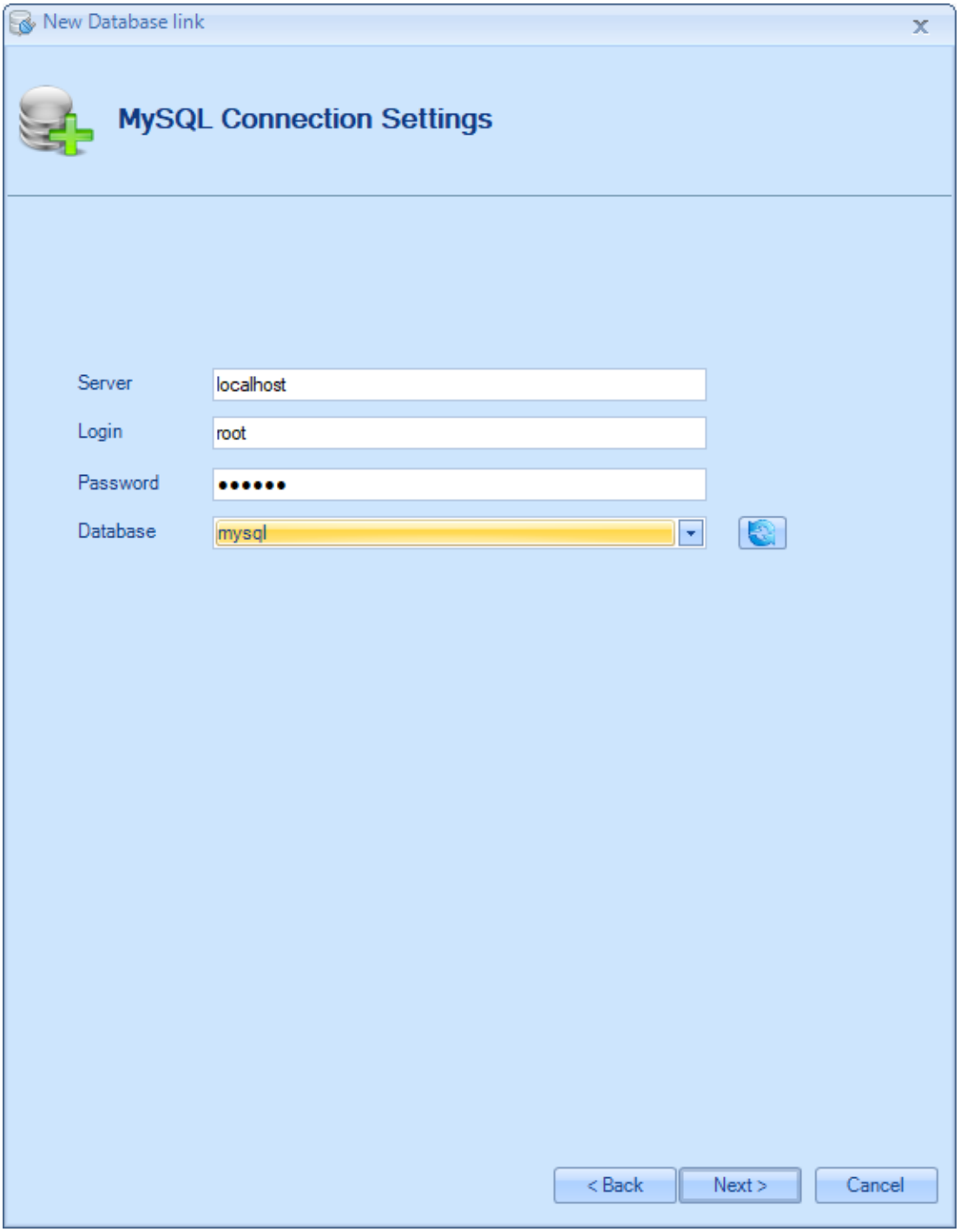

Figure 28: MySQL Connection Settings

Parameter	Description
Server	MySQL server instance name
Database	MySQL database name

Login	MySQL database user name
Password	MySQL database password

Table 11: MySQL Connection Parameters

- **ODBC**

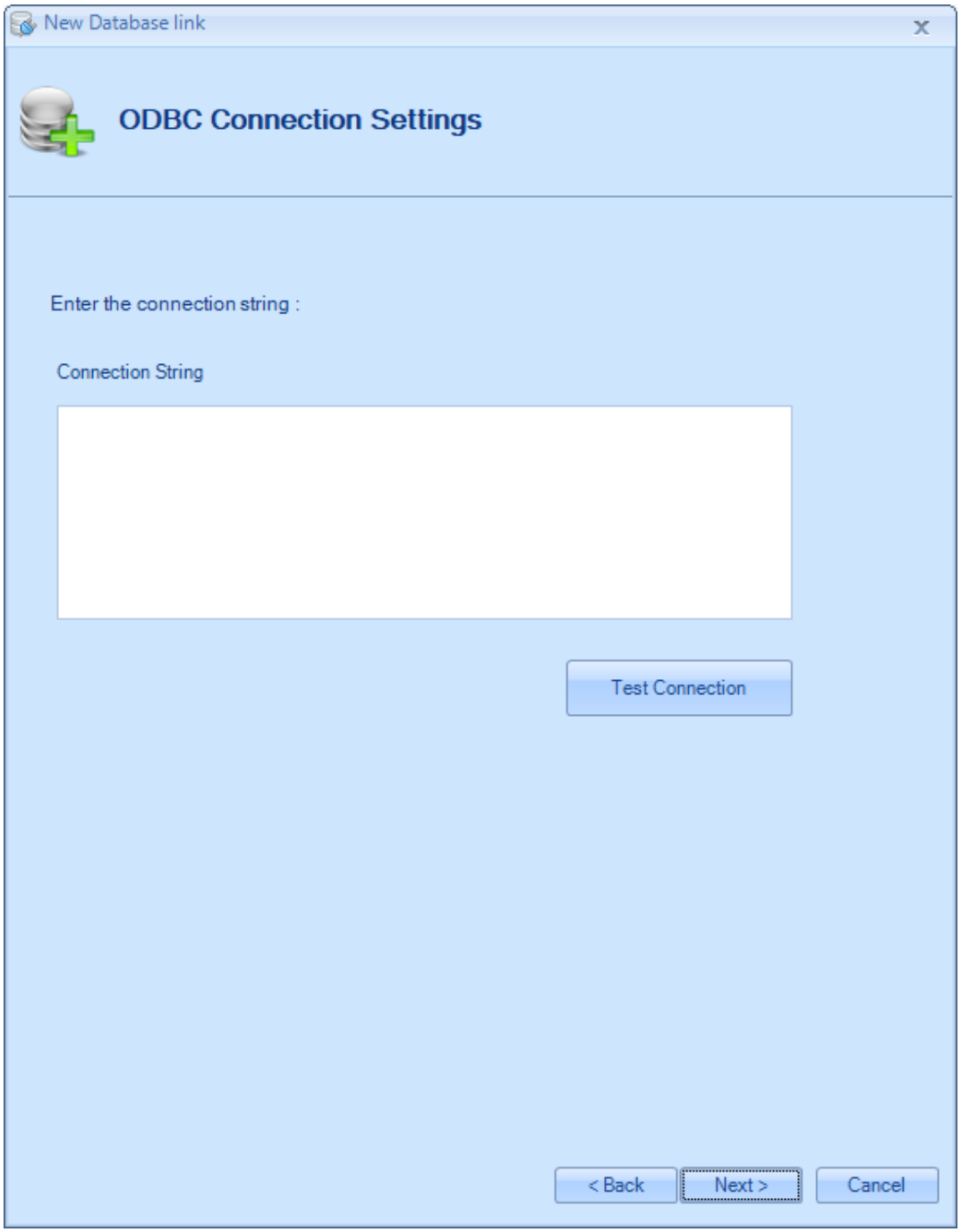

Figure 29: ODBC Connection Settings

Parameter	Description
Connection String	**ODBC SQL Server connection string example :** Driver={SQL Server};Server=.\sqlexpress; Database=master;Trusted_Connection=yes;

Table 12: ODBC Connection Parameters

- **OLEDB**

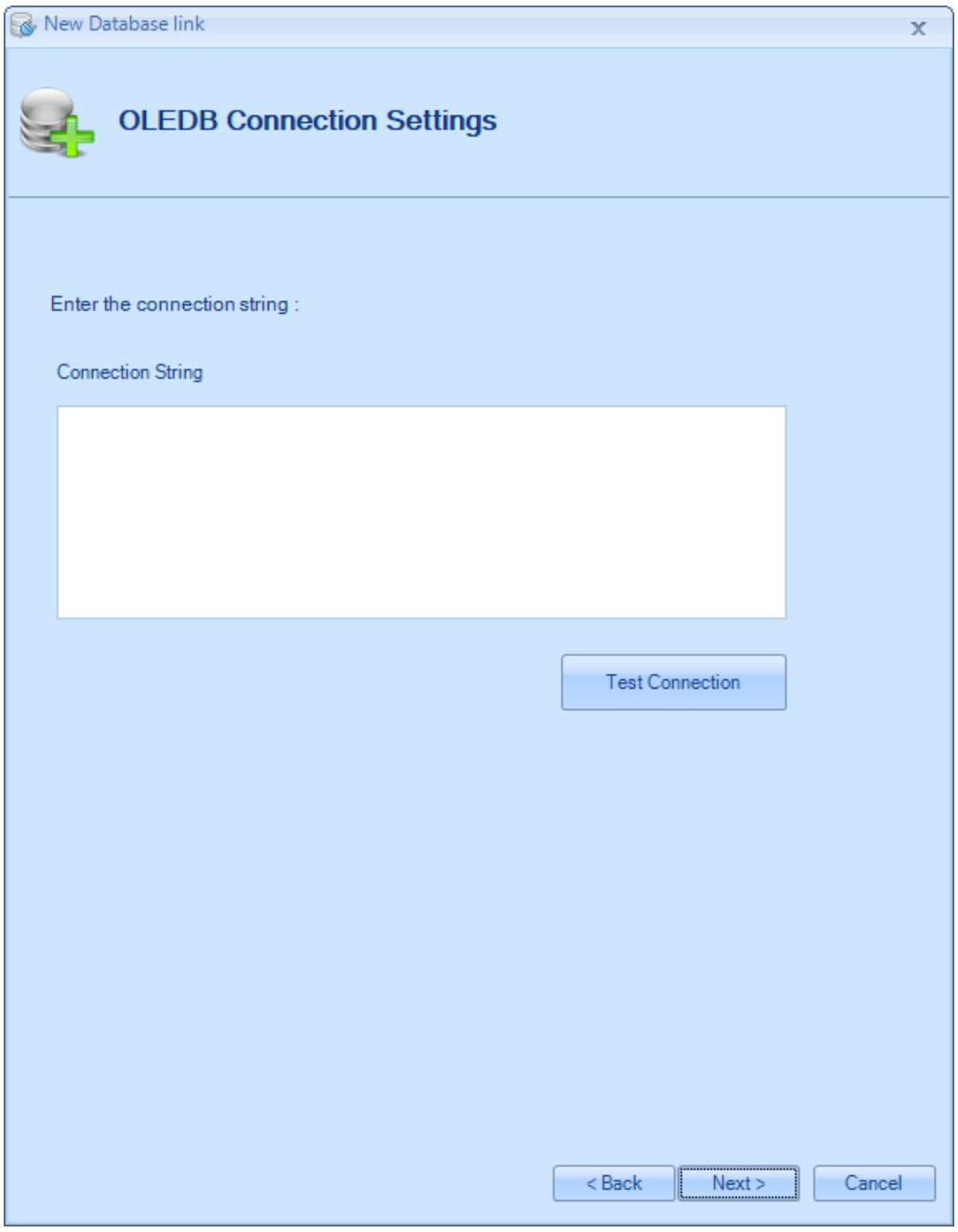

Figure 30: OLEDB Connection Settings

Parameter	Description
Connection String	**OLEDB SQL Server connection string example:** Provider=sqloledb;Data Source=.\sqlexpress;Initial Catalog=master;Trusted_Connection=yes;

Table 13: OLEDB Connection Parameters

- **Wonderware Historian**

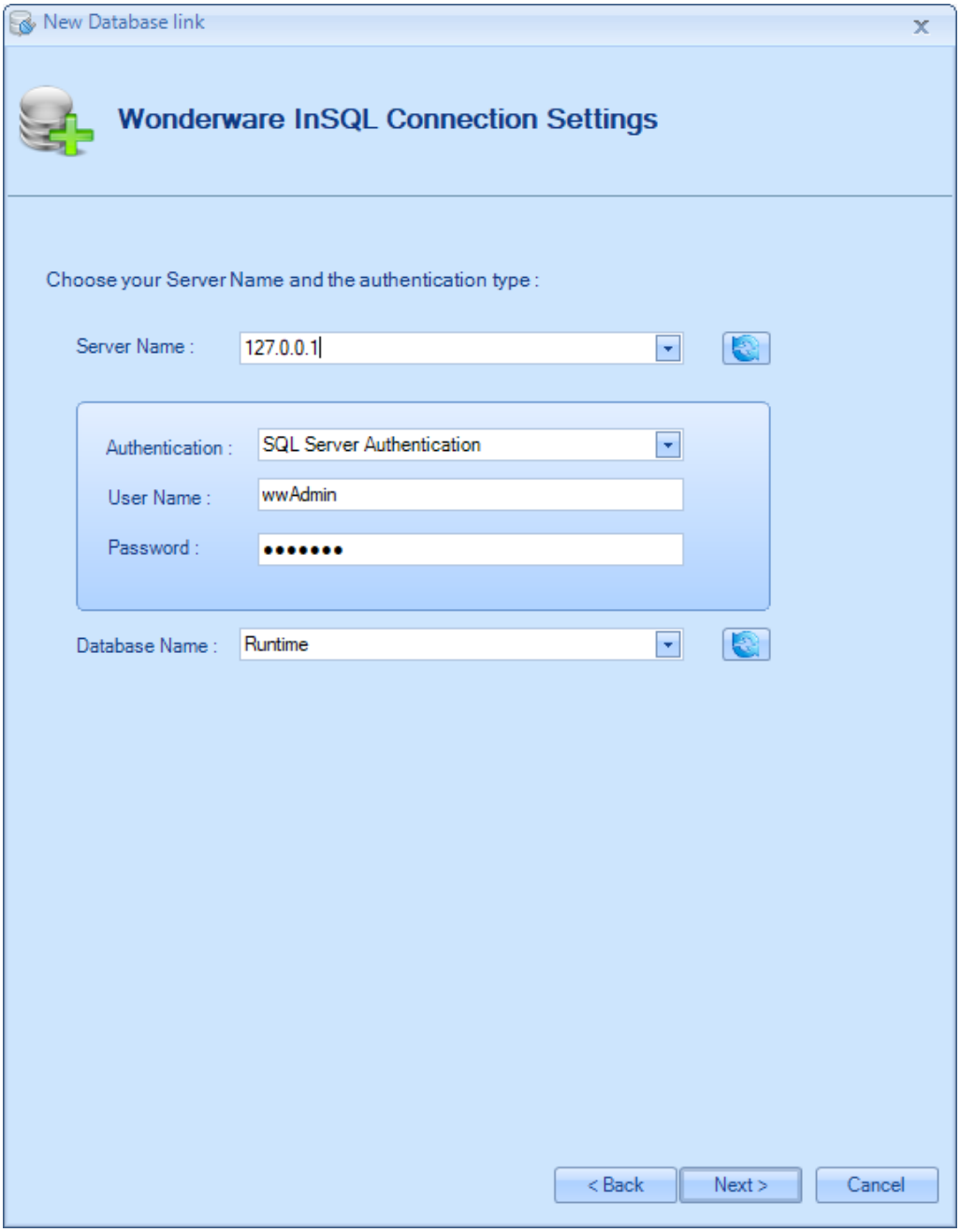

Figure 31: Wonderware InSQL Connection Settings

Parameter	Description
Server Name	Wonderware InSQL Instance name
Authentication	Used to specify the Wonderware InSQL connection type: • Windows Authentication • SQL Server Authentication

User name	The Wonderware InSQL instance user name
Password	The Wonderware InSQL instance password
Database name	The Wonderware InSQL Database name

Table 14: Wonderware InSQL Connection Parameters

2.1.3. STEP 3

Once you have selected your source type and configured the connection settings, press **Next** button to proceed.

2.1.3.1. RETRIEVE OPC TAGS FROM THE HISTORIAN DATABASE

In case you have selected **Retrieve OPC Tags from the Historian Database** when adding new Database Link in the first step, the figure below will be displayed.


When choosing this option, the OPC Driver for databases will consider that tags ID are the tables names existing in the chosen database.

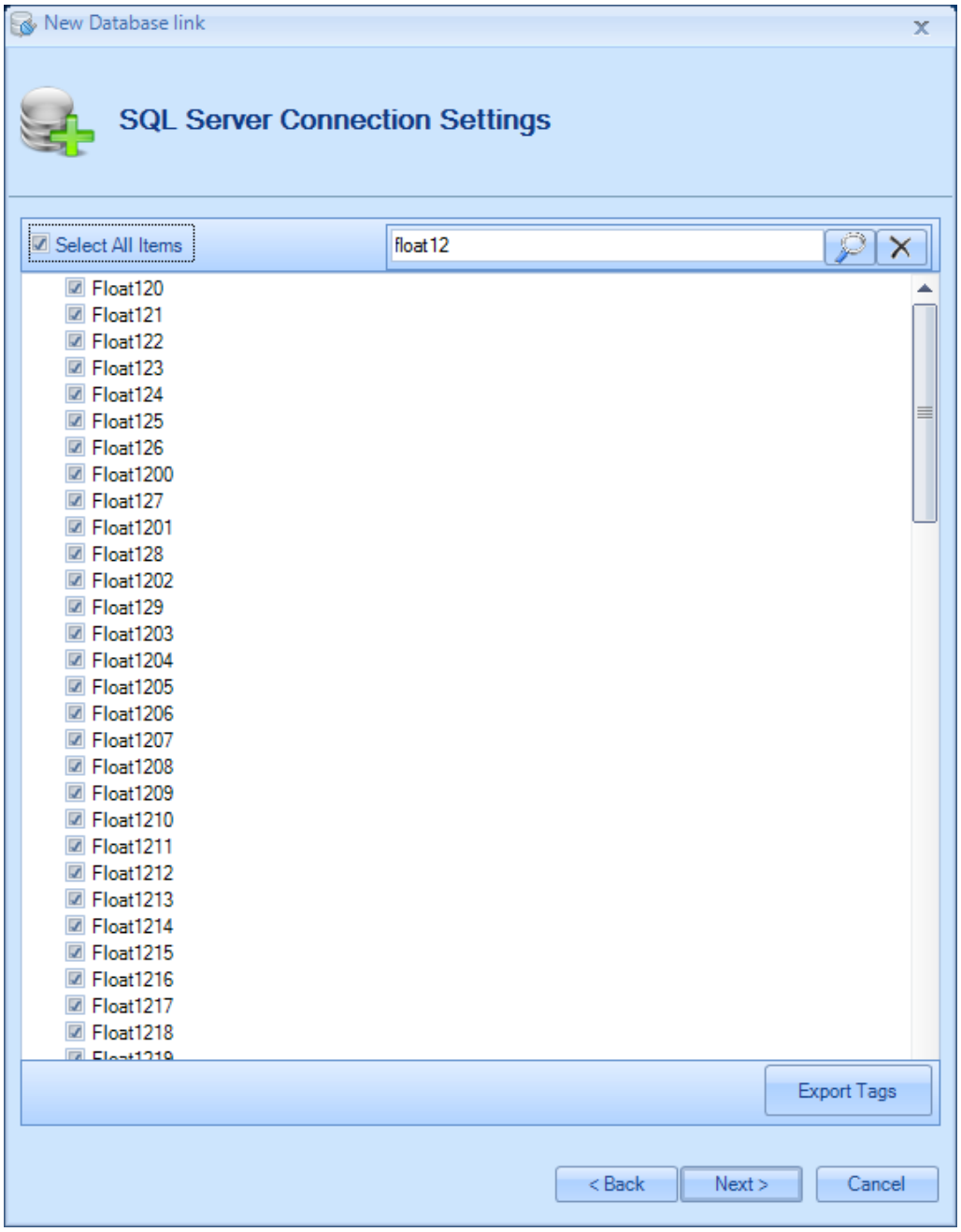

Figure 32: OPC Tags List

The tags list included in this window was retrieved from the selected database. Select the OPC Tags list and click **Next** button to map table fields with the OPC fields.

The **Export Tags** button allows you to export the selected tags into a csv file.

The search bar allows you to filter the tables names that you want to select.

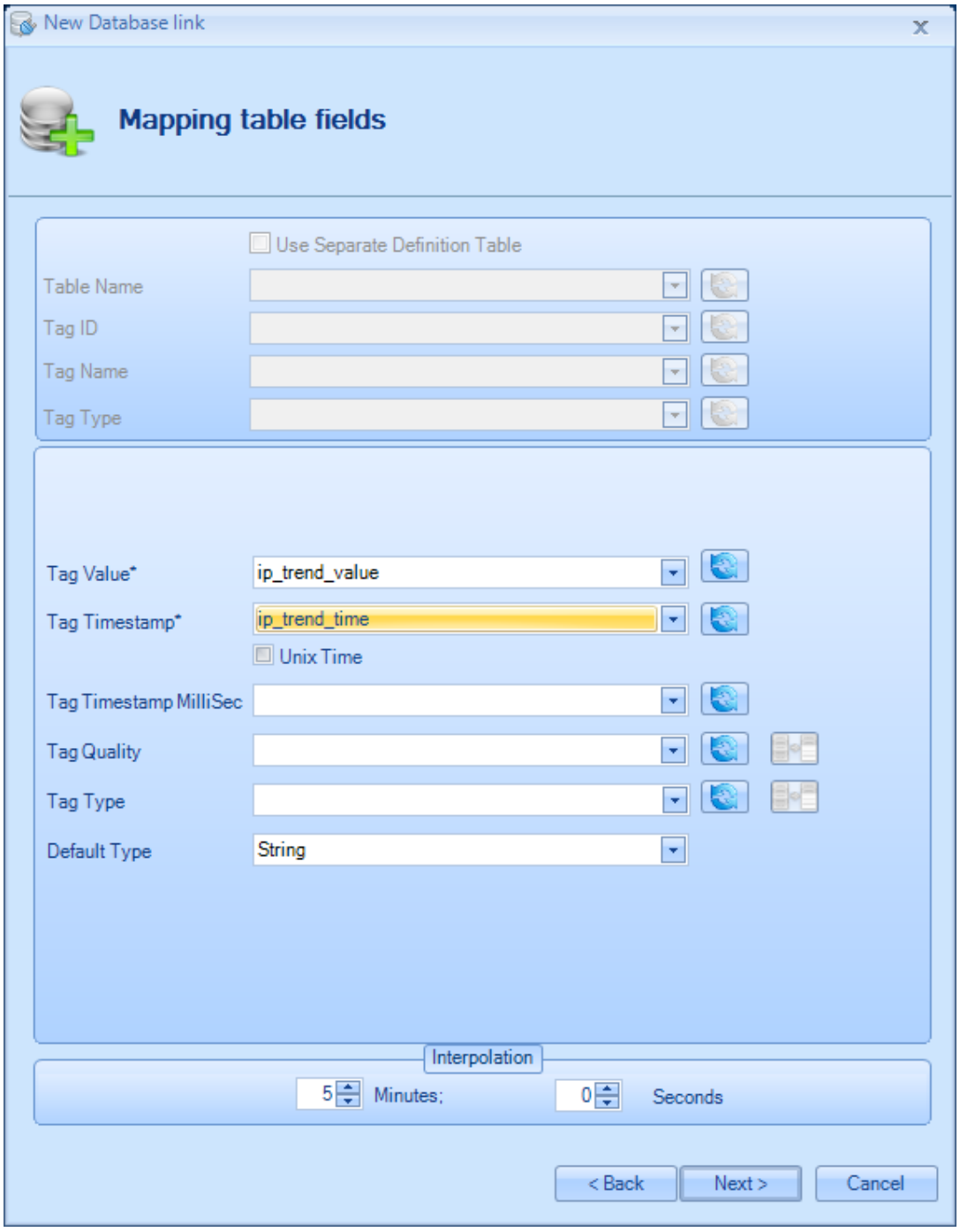

Figure 33: Mapping OPC Tag Attributes

This dialog box allows you to match OPC tag attributes including 'Tag Value', 'Tag Timestamp', 'Tag Quality' and 'Tag Type' with column names of the selected table.
Note that filling in the 'Tag Value' and 'Tag Timestamp' fields is mandatory. The selected columns should also be distinct. Otherwise, the database link configuration will be rejected.

Parameter	Description
Tag Value	Select a column from the selected table from which the tag

	value will be collected.
Tag Timestamp	Select a column from the selected table from which the tag timestamp will be collected.
Tag Timestamp MilliSec	Select a column from the selected table from which the tag timestamp millisecond will be collected.
Tag Quality	Select a column from the selected table from which the tag quality will be collected.
Tag Type	Select a column from the selected table from which the tag type will be collected.
Default Type	Used to specify the default supported type whenever the tag type is not mapped. It could be a string, numeric or date type.

Table 15: Table Fields

- **Interpolation:**

This is optional. You can customize the time range for ReadAtTime requests concerning interpolated values. The time range is [ftReadAtTime – Interpol, ftReadAtTime + Interpol] with ftReadAtTime is the filetime argument passed in the ReadAtTime request and Interpol is the configured time range.

Interpol = x_1**MN** + x_2**S**.
MN: minute.
S: second.
Interpolation parameters:

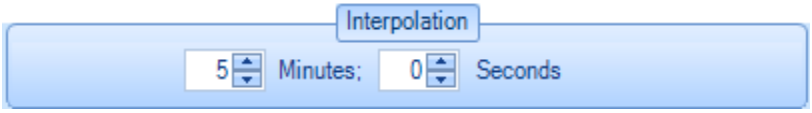

Figure 34: Interpolation Parameters

Example:
Interpol = 5MN + 0S

2.1.3.2. RETRIEVE OPC TAGS FROM THE HISTORIAN TABLE

In case you have selected Retrieve **OPC Tags from the Historian Table** when adding new Database Link in the first step, you can choose tags that will be added to the OPC Driver for Databases address space.

- **Use Separate Definition Table**

Once the **Mapping table fields** wizard page is displayed. You can use in this case a separate definition table to retrieve all OPC Tags information such as OPC Tag Name and OPC Tag Type then map it with the defined historian table. To do so, you need to:

- Check **Use Separate Definition Table** option,
- Match OPC tag attributes including 'Tag ID', 'Tag Name' and 'Tag Type' with column names of the selected definition table.

- Then, click **Map Separate Definition Table** button highlighted in green to specify the link between the two definition and historian tables.

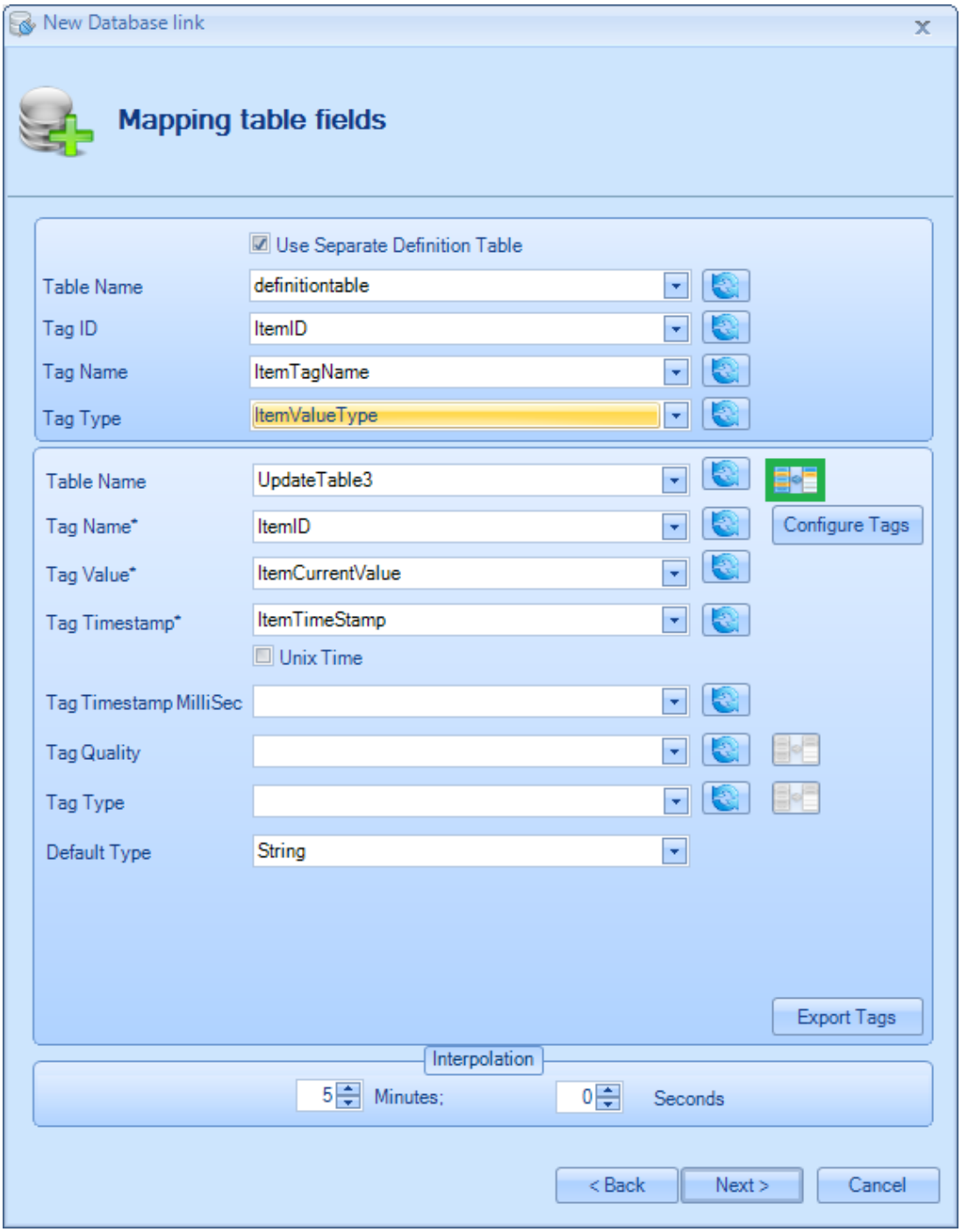

Figure 35: Mapping OPC Tag Attributes (Use Separate Definition Table)

Parameter	Description
Tag ID	Select a column from the selected table from which the tag ID will be collected.
Tag Name	Select a column from the selected table from which the tag

	Name will be collected.
Tag Type	Select a column from the selected table from which the tag type will be collected.

Table 16: Definition Table Fields

When clicking the **Map Separate Definition Table** button, the following window will be prompted.

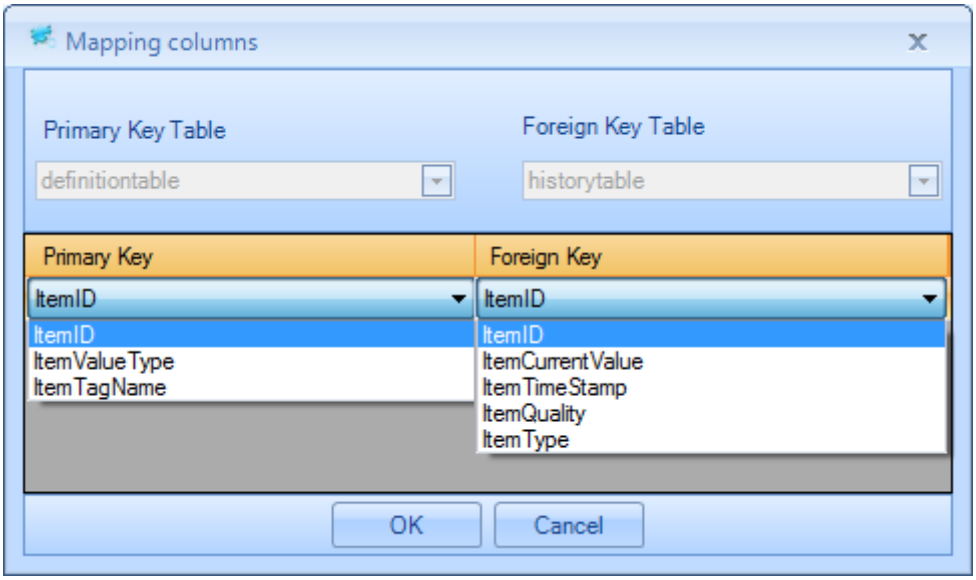

Figure 36: Mapping Columns

Parameter	Description
Primary Key	Used to choose the column that will be mapped with the historian table one, every value of that column will have a correspond value in the definition table column.
Foregin Key	Used to choose the column that will be mapped with the definition table one.

Table 17: Mapping Columns Fields

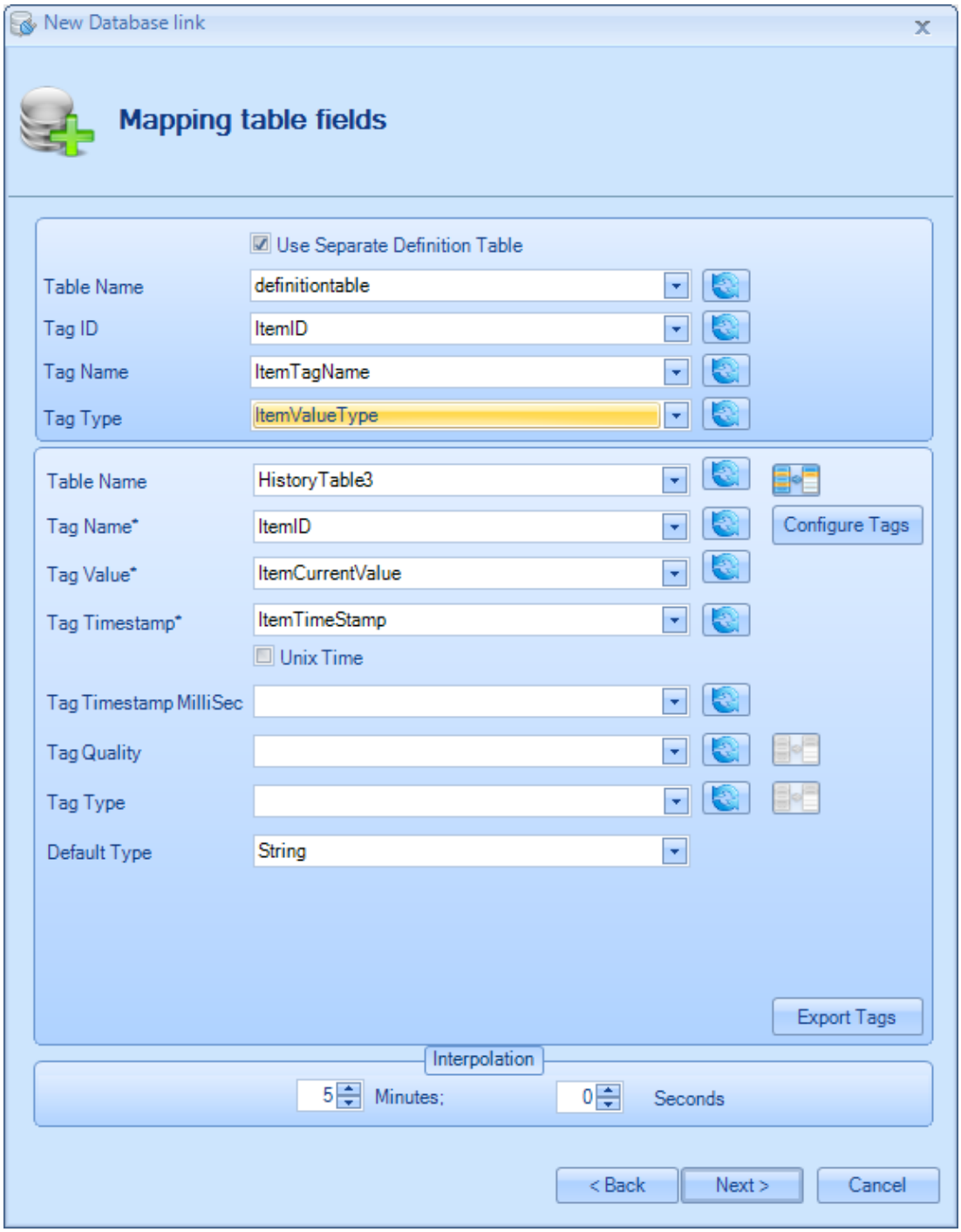

Figure 37: Configure Tags

Once the columns fields are configured, click the **Configure Tags** button to select the tags to be exposed by the OPC Driver for Databases. The following window will then be prompted:

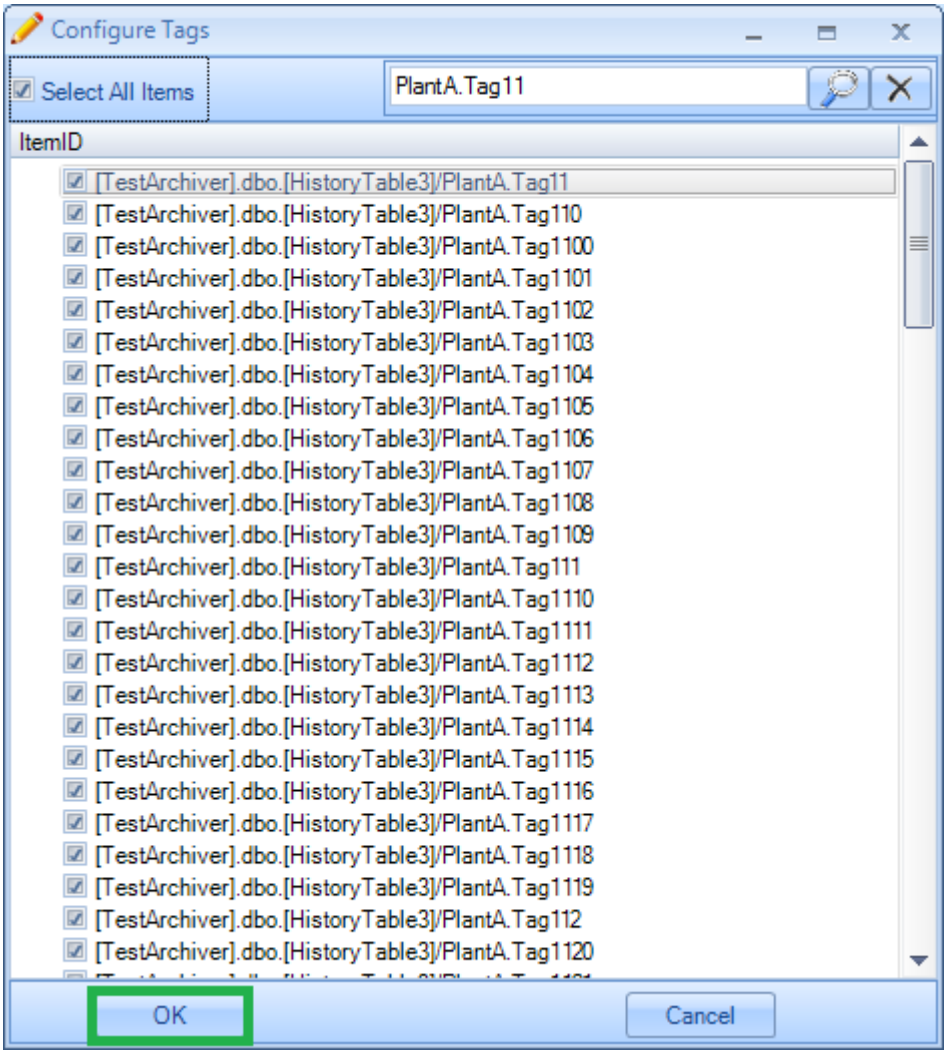

Figure 38: Choose Tags

Select your tags and then click **OK** to add the selected tags to the OPC Driver for databases address space.

- **Use Standard Historian Table**

You can use a standard historian table by unchecking the **Use Separate Definition Table** check button. Then, match OPC tag attributes including 'Tag ID', 'Tag Value', 'Tag Timestamp', 'Tag Quality' and 'Tag Type' with column names of the selected historian table.

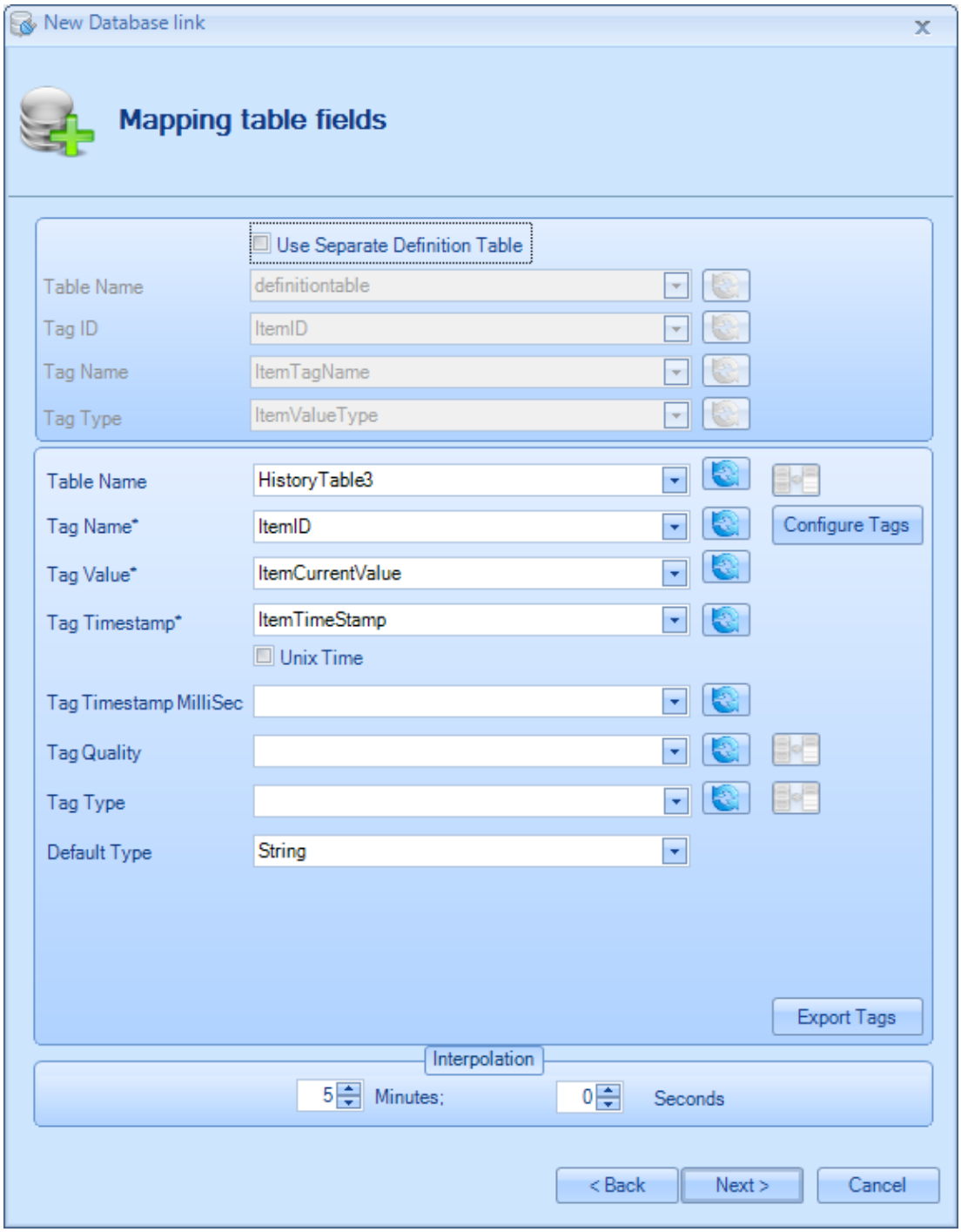

Figure 39: Mapping OPC Tag Attributes (Use Standard Historian Table)

Parameter	Description
Tag Name	Select a column from the selected table from which the tag Name will be collected.
Tag Value	Select a column from the selected table from which the tag value will be collected.
Tag Timestamp	Select a column from the selected table from which the tag

	timestamp will be collected.
Tag Timestamp MilliSec	Select a column from the selected table from which the tag timestamp millisecond will be collected.
Tag Quality	Select a column from the selected table from which the tag quality will be collected.
Tag Type	Select a column from the selected table from which the tag type will be collected.
Default Type	Used to specify the default supported type whenever the tag type is not mapped. It could be a string, numeric or date type.

Table 18: Standard Historian Table Fields

Configure the columns fields and then click the **Configure Tags** button to select the tags to be exposed by the OPC Driver for Databases.

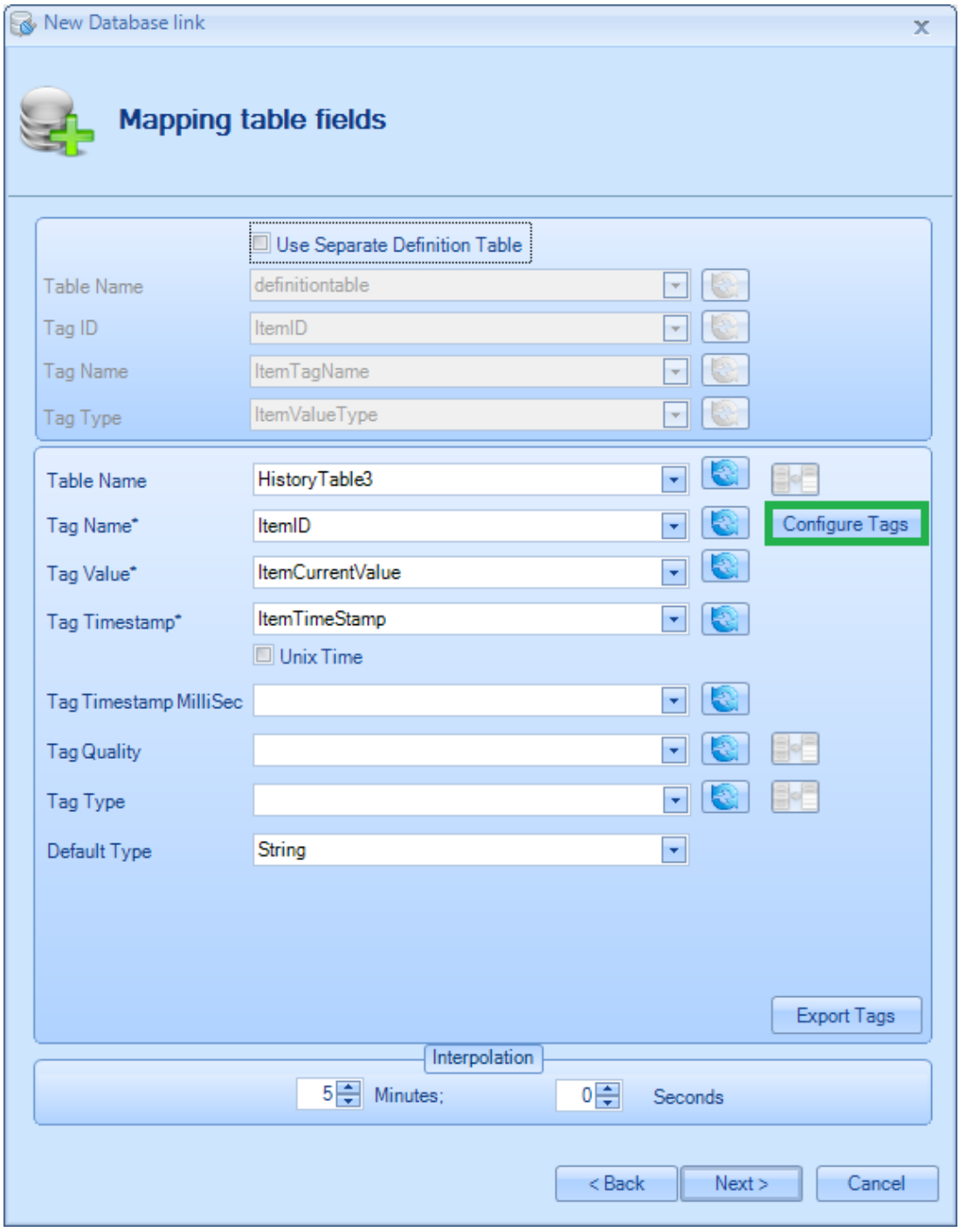

Figure 40: Configure Tags

The following window will then be prompted:

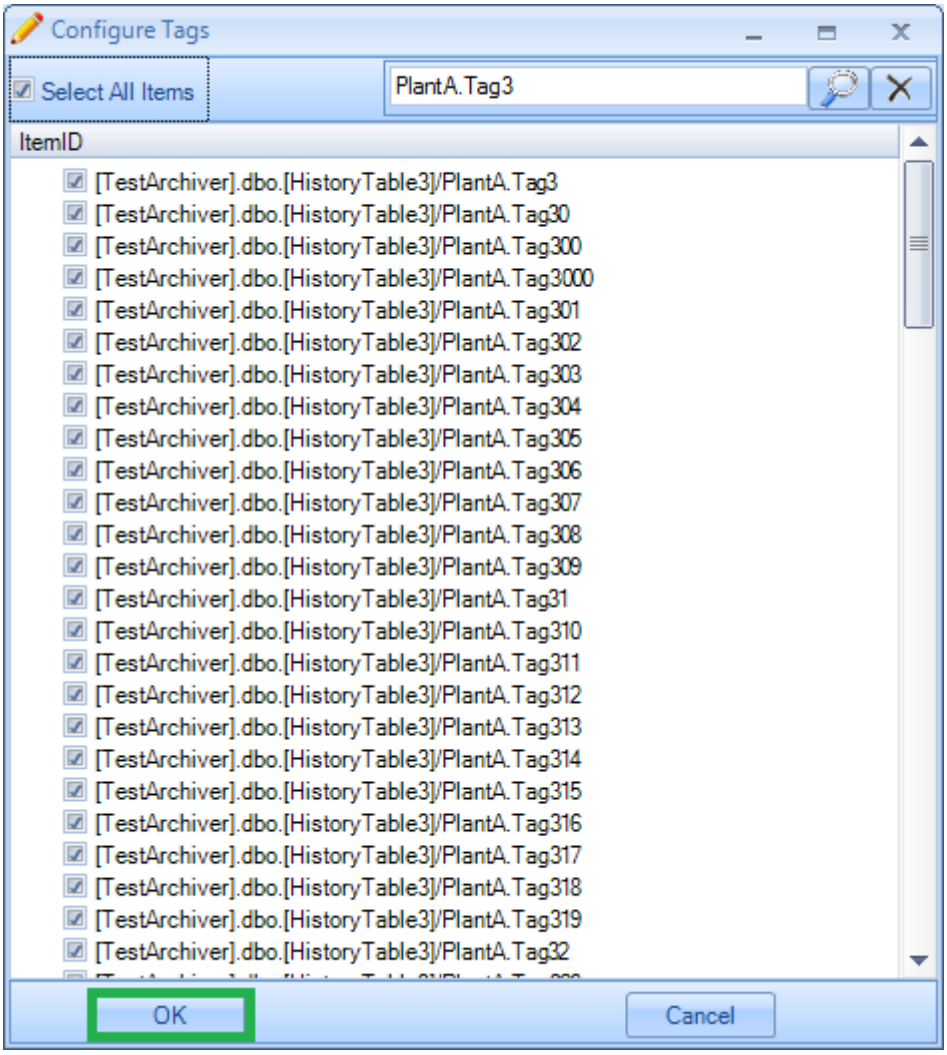

Figure 41: Choose Tags

Select your tags and then click **OK** to add the selected tags to the OPC Driver for Databases address space.

- **Interpolation:**

This is optional. You can customize the time range for ReadAtTime requests concerning interpolated values. The time range is [ftReadAtTime – Interpol, ftReadAtTime + Interpol] with ftReadAtTime is the filetime argument passed in the ReadAtTime request and Interpol is the configured time range.

Interpol = x_1**MN** + x_2**S**.
MN: minute.
S: second.

Interpolation parameters:

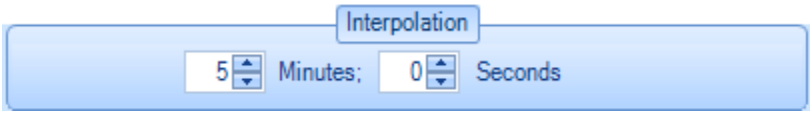

Figure 42: Interpolation Parameters

Example:
Interpol = 5MN + 0S.

- **Mapping Custom Qualities**

You should click on **Map Custom Qualities** button to configure your own quality values. You will get the following screen:

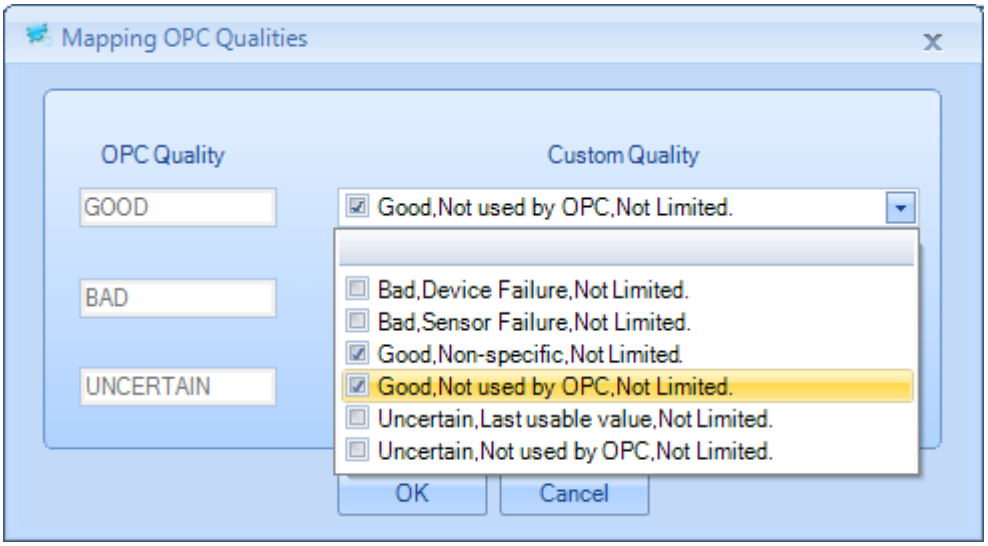

Figure 43: Mapping Custom Qualities

- **Mapping Custom Types**

OPC Driver for Databases uses standard variant types. If your database does not use those types you can map your custom types to the standard variant types. Click on **Map Custom Types** button, you will get the following window:

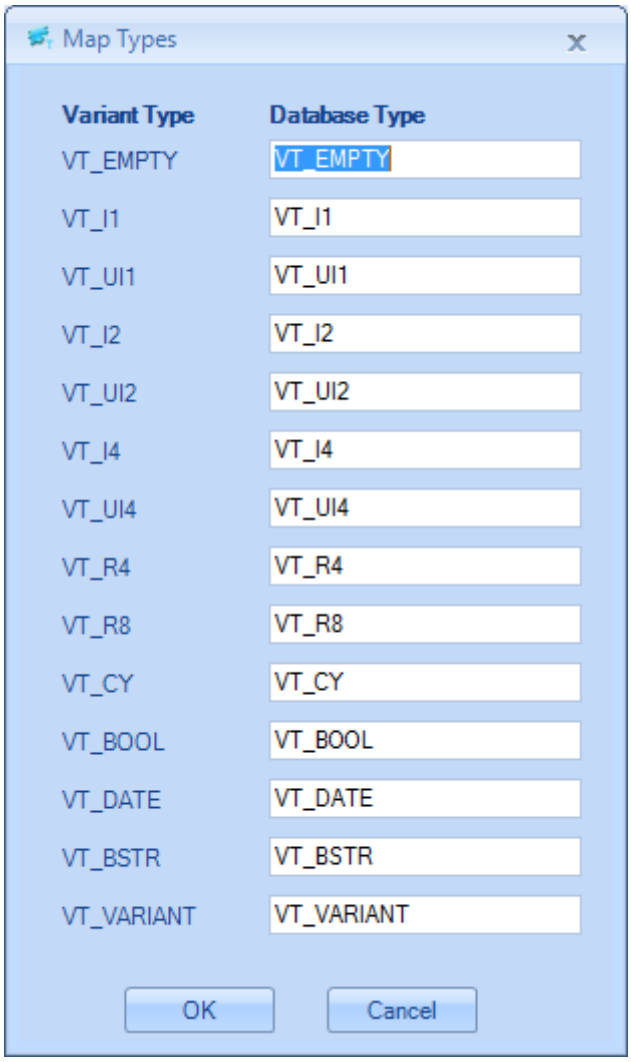

Figure 44: Mapping Custom Types

Replace the Variant type by your corresponding custom type that your database uses to get the type of your items.

Once the configuration is completed, you need to:

1. **Save the configuration into an xml file that could be loaded with the next start up**
2. **Define the saved XML file as a default configuration**
3. **Start the appropriate service from the service menu**

2.2. VIEW DATABASE LINK PROPERTIES

You can display the selected database link properties by right clicking on its node and selecting **View the database connection** option from the displayed menu.

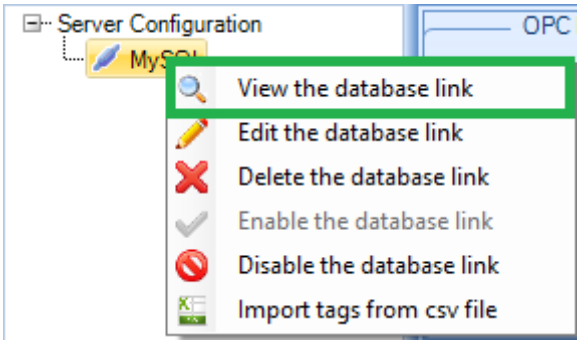

Figure 45: View the Database Link Properties

The main database link properties will be displayed in the specified window as shown below:

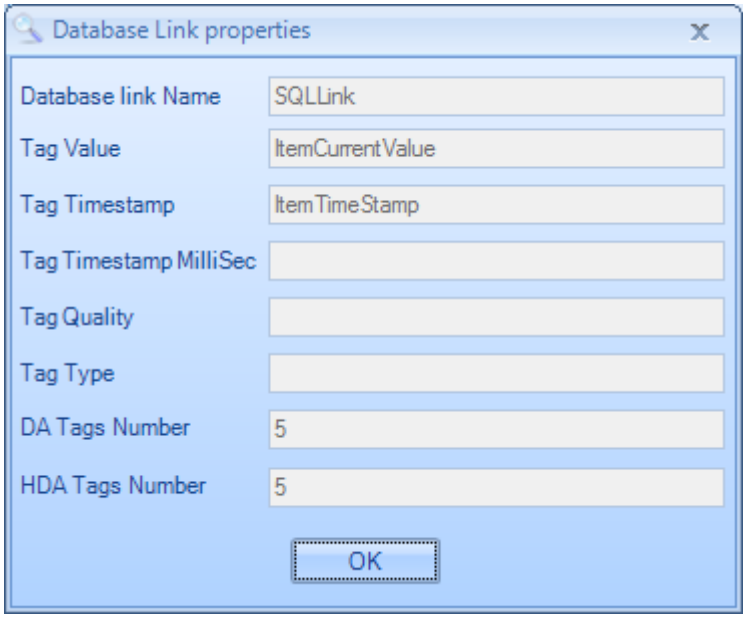

Figure 46: Database Link Properties

2.3. EDIT THE DATABASE LINK

You may update the configuration for a Database Link by right clicking on its node and selecting the **Edit the database connection** option from the displayed menu. Then, the a wizard for Database link edit will be displayed.

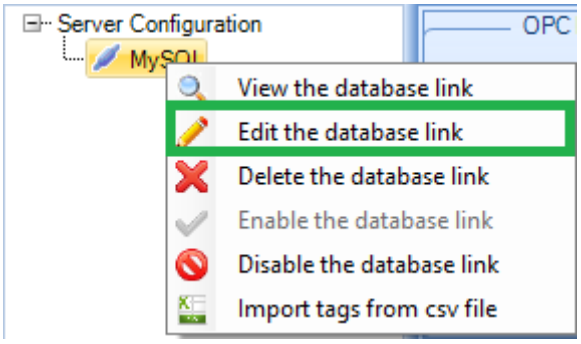

Figure 47: Edit the Database Link

2.4. REMOVE THE DATABASE LINK

You can also remove the selected database link from the server configuration by right clicking on its node and selecting on the **Delete the database link** option from the displayed menu.

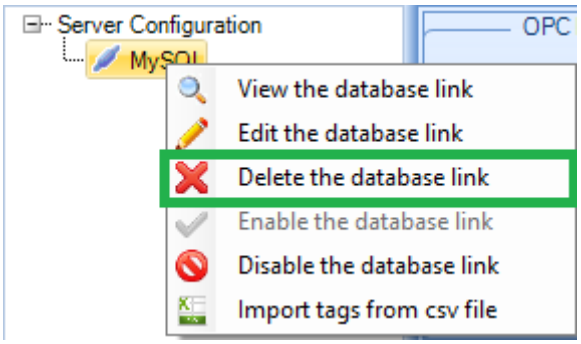

Figure 48: Remove the Database Link

You can enable your database connection by clicking on the contextual menu "Enable the database link" of the selected database connection node.
To disable it, you should click the Disable the database link menu item.

2.5. IMPORT TAGS FROM CSV FILE

You can also import tags from a csv file to the selected Database link from the server configuration by right clicking on its node and selecting **Import tags from csv file** option from the displayed menu.

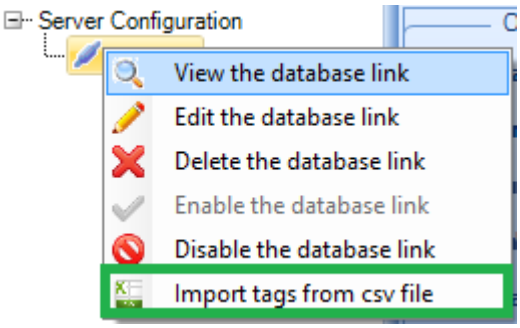

Figure 49: Import Tags from a csv File

TROUBLESHOOTING

1. Logging

The OPC server creates three log files named "Driver.LOG", "DriverService.LOG" and "HDADriverService.LOG" that record errors and debugging information for the server configuration and runtime execution.
This server also generates a log file dedicated to the details of operations of the OPC interfaces: "LogEvent.log" log file to easily diagnose the occurred problems and can be extremely valuable for troubleshooting. Under normal operations, the server logs very little information. These log files are generated at start-up under the installation folder where the executable file is located.

The OPC Driver for Databases is based on two configuration files:

- "SrvToolkit_CfgFile.ini"
- "OPCDriverConfig.ini"

These files include several logging parameters. You can update these parameters through the OPC Driver for Database user interface or by updating the "OPCDriverConfig.ini" and "SrvToolkit_CfgFile.ini" files in a text editor.

The following table describes the logging parameters saved on "SrvToolkit_CfgFile.ini":

Log Setting	Description	Default Value
LogFileMaxSize	The maximum log file size, in bytes. Once this size is reached during runtime, the log file will be overwritten.	2097152 bytes
LogLevel	The log level. Possible Values are: **Control (-1):** It is the lowest level. This log file contains at least a description of succeeded methods. **Fatal (0):** Only fatal error messages are logged. **Critical (1):** All critical error messages are logged. **Error (2):** All errors are logged. **Warning (3):** All warnings are logged. **Info (4):** All information is logged. **Debug (5):** For Debug information. The higher the log level, the more information are	Control

	recorded. We recommend using level 0 for a better performance of the server.	
ArchiveLastLog	**TRUE:** Old file is copied to an intermediate file with incremental extension, before being overwritten. **FALSE:** Any pre-existing log file is erased and overwritten at start-up.	False
LogFileName	Used to set the log file name	LogEvent

Table 19: SrvToolkit_CfgFile.ini

The following table describes the logging parameters saved on "OPCDriverConfig.ini":

Log Setting	Description	Default Value
WindowsLogConfiguration		
CreateNew	True to create a new event log or to append the old log.	False
Level	There are five log levels: 1. Control: Logs only control messages generated by OPC Driver for Databases. 2. Error: Logs error and control messages generated by the OPC Driver for Databases. 3. Warning: Logs warning, error and control messages generated by OPC Driver for Databases. 4. Inform: Logs information, warning, error and control messages generated by the OPC Driver for Databases. 5. Debug: Logs all messages generated by the OPC Driver for Databases.	Error
LogName	The OPC Driver for Databases log file name	Driver
Source	The Window event log source name	OPCDriverForDatabases
GUILogSettings		
AutoAppend	Set to true to continue writing log messages in the existing log file or to false to create a new file.	True
FileName	The OPC Driver for Databases log file name	Driver

MaximumFiles	Set to 0 means that log files will be created in an unlimited way.	1
FolderPath	Used to save the OPC Driver for Databases full installation directory path	
DAServiceLogSettings		
AutoAppend	Set to true to continue writing log messages in the existing log file, or to false to create a new file.	True
FileName	The OPC DA Driver for Databases log file name	DriverService
MaximumFiles	Set to 0 means that log files will be created in an unlimited way.	1
FolderPath	Used to save the OPC Driver for Databases full installation directory path	
HDAServiceLogSettings		
AutoAppend	Set to true to continue writing log messages in the existing log file, or to false to create a new file.	True
FileName	The OPC Driver for Databases log file name	HDADriverService
MaximumFiles	Set to 0 means that log files will be created in an unlimited way.	1
FolderPath	Used to save the OPC Driver for Databases full installation directory path	
PortConfiguration		
PortNumber	OPC DA Driver for Databases service port number	2501
HDAPortNumber	OPC HDA Driver for Databases service port number	2502
ConfigSetting		
ConfigFilePath	Used to save the default configuration file path	
ServerRate	This parameter is the frequency at which the server handles the asynchronous reads/updates.	500
UpdateOPCServerCache	**True:** Update OPC Server cache from database. **False:** Disable update OPC Server	True

	cache procedure.	
Delimiter	OPC Item Delimiter	/
DbCycle	This parameter is the frequency at which the server checks for the database connection state.	30 s (seconds)
ExecutionTimeout	Maximum duration to wait before the query execution expires	300 s (seconds)
RequireTagValidation	Used to validate the loaded OPC Tags from the XML configuration file with the retrieved OPC Tags from the historian table	False
UsePasswordEncryption	Used to enable or disable the password encryption	True

Table 20: OPCDriverConfig.ini

SrvToolkit_CfgFile.ini Configuration File

```
[LogSetting]
LogFileMaxSize=2097152
LogLevel=0
ArchiveLastLog=FALSE
LogFileName =LogEvent
```

Figure 50: SrvToolkit_CfgFile.ini File

OPCDriverConfig.ini Configuration File

```
[WindowsLogConfiguration]
CreateNew=False
Level=Error
LogName=OPC Driver for Databases
Source=OPC Driver for Databases
[GUILogSettings]
AutoAppend=True
BufferSize=100
FileName=Driver
MaximumFiles=0
Level=Error
FolderPath=C:\Program Files (x86)\Integration Objects\Integration Objects'
OPC Driver for Databases\LogFiles\
[ServiceLogSettings]
AutoAppend=True
BufferSize=100
FileName=DriverService
MaximumFiles=0
Level=Error
FolderPath=C:\Program Files (x86)\Integration Objects\Integration Objects'
OPC Driver for Databases\
AutoSaveTimeOut=60
[HDAServiceLogSettings]
AutoAppend=True
BufferSize=100
FileName=HDADriverService
MaximumFiles=0
Level=Error
FolderPath=C:\Program Files (x86)\Integration Objects\Integration Objects'
OPC Driver for Databases\
AutoSaveTimeOut=60
[PortConfiguration]
PortNumber=2501
HDAPortNumber=2502
[ConfigSetting]
ConfigFilePath=
ServerRate=500
Style=Office2007Blue
UpdateOPCServerCache=True
Delimiter=/
DbCycle =30
ExecutionTimeout=300
RequireTagValidation=false
UsePasswordEncryption=True
```

Figure 51: OPCDriverConfig.ini file

2. Most Common Issues & Solutions

Problem1: Cannot launch the OPC Driver for Databases.

You should check the license validity by launching the "LicenseAuthorization.exe" existing under the OPC Driver for Databases installation folder, or start it directly from the startup menu:

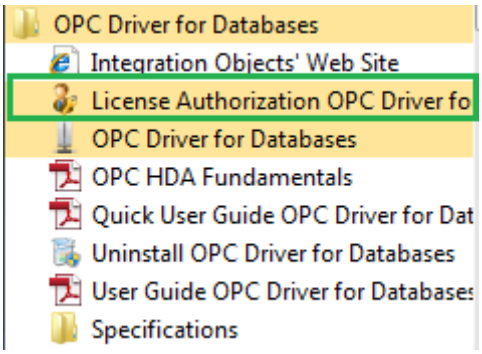

Figure 52: OPC Driver for Databases Startup Menu

If the License Authorization tool shows that the demo has expired and you want to activate it using your full activation license, you should follow the enumerated steps in the Frequently Asked Question section.

Problem2: Cannot browse the OPC Driver for Databases tags list.

You need to check the following items:

- Check if the configuration was successfully saved in the xml configuration file.
- Check if the OPC Driver for Databases service is running.
- Check the database connection status in the "Driver" log file.

Problem3: Cannot load the XML file configuration.

You need to set this file as a default configuration by following these steps:

1. Stop the OPC Driver for Databases service if it is running
2. Click on Define button existing under the Settings section
3. Select the xml configuration file from the displayed window
4. Apply your changes
5. Restart the OPC Driver for Databases service.

3. DCOM Encountered Issues

This section addresses some DCOM related problems while using OPC servers:

Problem 1: You have an "Access denied" error on the client machine. The client and server are running on standalone machines (meaning not on the same domain).

Let's assume that the OPC client is running on machine A and the OPC server on machine B.

When the OPC client and server are on different computers, you have to give each computer access to the other by giving access permissions. The permission issue is crucial to proper DCOM configurations.

Here the server is running on a standalone machine. So the ONLY user accounts, it will trust, are those it finds in its own "local" security database. Here is how this can get you into trouble on setting up an OPC client to server connection.

To allow a remote client to access the DCOM server, the DCOM utility uses the Windows Security database. For this reason, you cannot give access to a user account that does not figure in this database.

Here is the resolution:

1. You can add Machine B onto the same domain as Machine A (or in a trusted domain), which is the safest way to correctly set up the communication between the OPC client and the OPC server.
2. You need to create the EXACT SAME user account name AND password on BOTH machines (for example User1 (login), PWD1 (password)). Once you have that set up, when Machine A comes calling on Machine B with an OPC request and identifies himself as User1 with PWD1 password, Machine B will look in its database, see the same account name, the same password, and same "come on in request from Machine A". When Machine B goes to return its data from the OPC server to the OPC client on machine A, the OPC server will call Machine A as User1 with a password -- Machine A will look in its database, see that it has that account, and accept the call. This workaround should resolve the communication problem between the OPC client and server.

You can refer to DCOM_Config_Guideline_WinSeven_Workgroup.pdf DCOM guideline document to successfully configure DCOM for OPC Driver for Databases.

Problem 2: You have been running your OPC client on a Windows XP machine. When upgrading the machine to XP Service Pack 2, the OPC client becomes unable to connect to the OPC server.

This is a common problem when using OPC via DCOM with Microsoft Windows XP Service Pack 2.

In fact, when Service Pack 2 is installed with its default configuration settings, OPC communication via DCOM will cease to work.

To resolve this issue, you have to reconfigure your settings for:

1- The windows XP firewall.
2- And DCOM.

You can refer to Using OPC via DCOM with XP SP2.pdf OPC Foundation document that describes all steps to apply new settings.

FREQUENTLY ASKED QUESTIONS

Question 1: How can I activate the OPC Driver for Databases license?

You should check the license validity by launching the "LicenseAuthorization.exe" existing under the OPC Driver for Databases installation folder.

You need to launch the "LicenseAuthorization.exe" existing under the OPC Driver for Databases installation folder or start it directly from the startup menu.
Once the LicenseAuthorization.exe was launched, you should:

- Select the feature(s) that you want to activate
- Specify the Tags number
- Click Generate button to generate the user ID
- Copy and send the User ID to the sales team so they can generate the dedicated activation code.

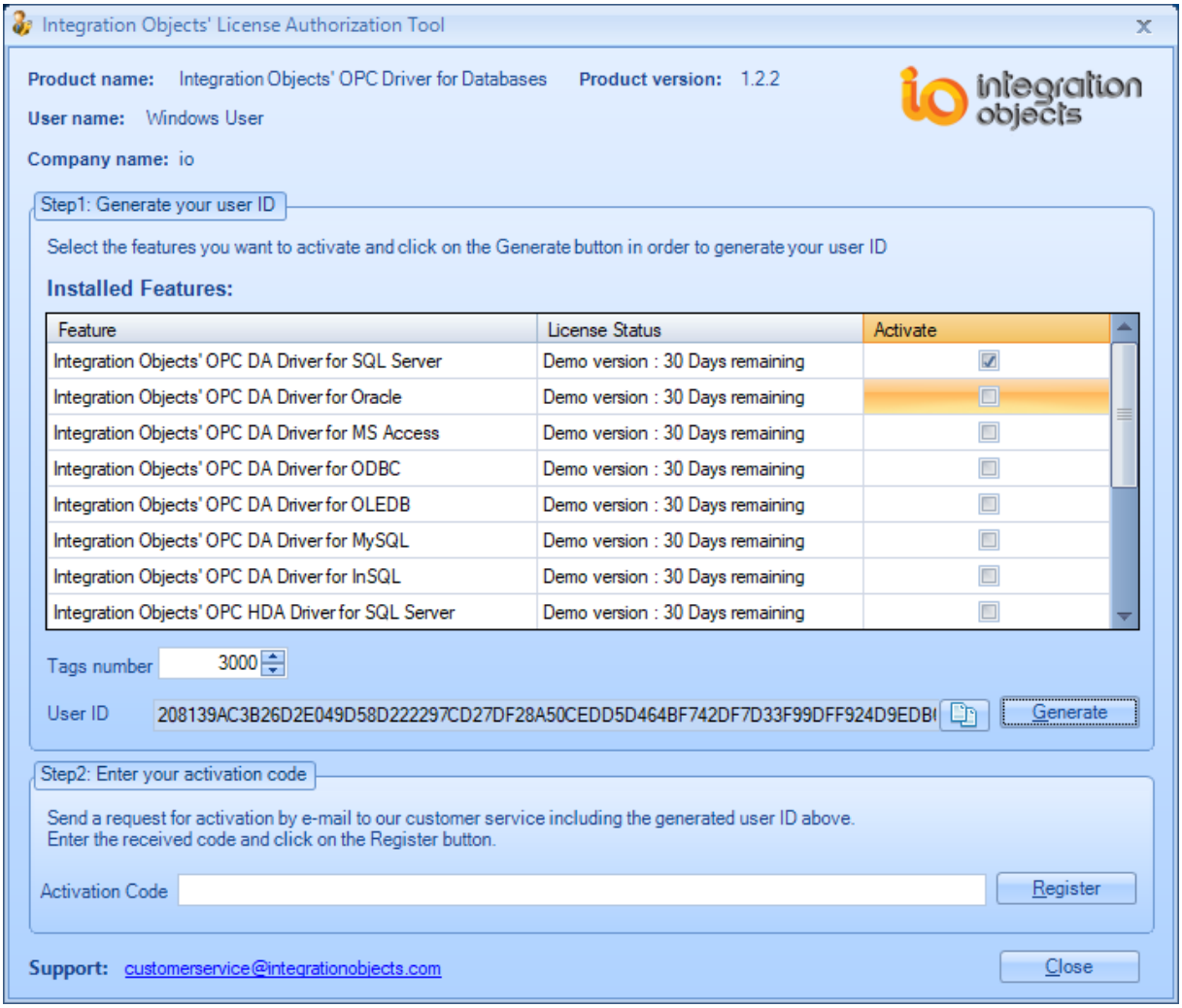

Figure 53: License Registration

Note that the license of your OPC Driver for Databases depends on the selected features during the generation of the user ID.

- Copy and paste the received activation code and click on the Register button.

Question 2: How can I restart OPC Driver for Databases with the default configuration?

You need to follow these steps:

- Stop the OPC Driver for Databases, if it is running
- Go to settings section, and select Define button

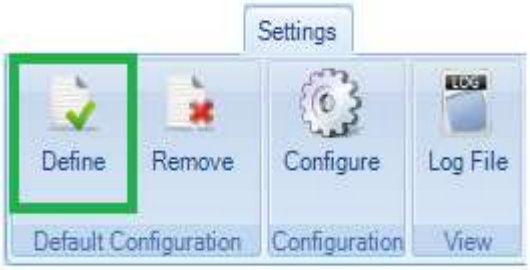

Figure 54: Settings Menu

- Select the xml configuration file from the displayed window
- Save your changes
- Restart the OPC Driver for Databases service

Question 3: Where should I install the OPC Driver for Databases?

You have 3 installation options:

1. You can install the OPC Driver for Databases in the same machine as your OPC client. In this case, you do not have to configure DCOM but you need to make sure that you can connect to your database(s) remotely.
2. You can install the OPC Driver for Databases in the same machine as your database. In this case, you will need to configure DCOM to establish communication with your OPC Client.
3. You can install the OPC Driver for Databases in a dedicated machine. In this case, you will need to configure DCOM and make sure that you can connect to your database(s) remotely.

The correct choice depends mainly on your network architecture and existing hardware.

Question 4: What are the Oracle versions that OPC Driver for Databases supports?

OPC Driver for Databases is compatible with Oracle 9i, Oracle 10g and Oracle 11g

For additional information on this guide, questions or problems to report, please contact:

Offices
- Americas: +1 713 609 9208
- Europe-Africa-Middle East: +216 71 195 360

Email
- Support Services: customerservice@integrationobjects.com
- Sales: sales@integrationobjects.com

To find out how you can benefit from other Integration Objects products and custom-designed solutions, please visit our website www.integrationobjects.com.