DataLogger

Table of Contents

DataLogger

Help version 1.118

CONTENTS

Overview

DataLogger is an application that logs data from an OPC server to any ODBC-compliant database. DataLogger's tight integration with the OPC server provides substantial, unique benefits such as simple installation, high-efficiency performance, and easy tag browsing in the OPC browse space.

Feature Highlights

- Supports any ODBC-compliant database management system.
- Supports adding OPC data items through drag and drop.
- Has a user-friendly installation and configuration. If a Database Source is defined before DataLogger is launched, users can have an active logging configuration in less than ten steps.
- Has flexible triggering. Data logging can be enabled at the following times: always, at absolute times, or when an expression is true (such as when a tag's quality is bad). When enabled, logging can occur based on static / time interval, on log group item data change, on monitor item data change, and / or on start / stop condition transitions.
- Has improved, reliable information logging through the local store and forward file, which is used to bridge network and database outages or delays.
- Includes an OPC server Simulator Driver.
- Includes two hours for evaluation while in Time Limited mode.

Runtime Performance Features

- Runs as a System Service.
- Can be easily scaled through its support of multiple concurrent logging processes (threads).
- Logs data directly from the local item list without reliance on external OPC servers.
- Supports both automatic table creation and the ability to append data to an existing table.
- Supports error recovery and can automatically reconnect if a DSN connection is lost.
- Supports an optional automatic configuration backup (in which the most recent copy of the configuration file is saved).
- Supports _System Tags that allow optional Runtime control from OPC client applications (such as enabling / disabling logging and monitoring logging status).

Initial Setup Considerations

The following topics should be reviewed before the first DataLogger Configuration is created.

System Requirements
External Dependencies
Supported Data Types
SQL Authentication
Windows Authentication
CSV Import / Export

System Requirements

Software Requirements

The following Microsoft Windows operating systems are supported:

- Windows 8
- Windows 7 Professional, Enterprise, and Ultimate
- Windows Server 2012
- Windows Server 2008 R2
- Windows Server 2008
- Windows Vista Business, Enterprise, and Ultimate
- Windows Server 2003 (Service Pack 2)
- Windows XP Professional (Service Pack 2)

Note: When installed on a 64-bit operating system, the application runs in a subsystem of Windows called WOW64 (Windows-on-Windows 64-bit). WOW64 is included on all 64-bit versions of Windows and is designed to make differences between the operating systems transparent to the user.

Hardware Requirements

The minimum required hardware is as follows:

- 2.0 GHz processor
- 1 GB installed RAM
- 180 MB available disk space
- Ethernet card

See Also:

External Dependencies

External Dependencies

This application has external dependencies. It requires that the ODBC driver for the Database Management System being used be installed on the PC that is running the OPC server. DataLogger supports the following ODBC drivers:

- SQL Native Client (necessary for SQL Server 2005)
- SQL Server ODBC Driver (compatible with pre-SQL Server 2005)
- MyODBC Driver 3.51 (for MySQL)
- Microsoft Access 4.0 ODBC Driver
- Linked Excel table support provided through the Microsoft Access 4.0 ODBC Driver

Notes:

1. Although DataLogger supplies TimeStamp values with a resolution to one thousandth of a second, certain databases are not capable of displaying a Date Format to the resolution of below one second.
2. Some databases do not support millisecond resolution. For more information on a specific database, refer to the product's vendor.

Recordsets

Tip: MSSQL uses Dynamic-type recordsets; MS Access uses Dynaset-type recordsets; all others use Snapshot.

Dynaset-type Recordset — the result of a query that can have updatable records. A dynaset-type recordset object is a dynamic set of records that can add, change, or delete records from an underlying database table or tables. A dynaset-type recordset object can contain fields from one or more tables in a database. This type corresponds to an ODBC keyset cursor.

Snapshot-type Recordset — a static copy of a set of records that can find data or generate reports. A snapshot-type recordset object can contain fields from one or more tables in a database, but can't be updated. This type corresponds to an ODBC static cursor.

Dynamic-type Recordset — a query result set from one or more base tables to add, change, or delete records from a row-returning query. Further, records other users add, delete, or edit in the base tables also appear in the recordset. This type corresponds to an ODBC dynamic cursor (ODBCDirect workspaces only).

Supported Data Types

The Data Logger Plug-In supports the following data types.

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8-bit value
Char	Signed 8-bit value
Word	Unsigned 16-bit integer
Short	Signed 16-bit integer
BCD	Two-byte packed binary coded decimal
LBCD	Four-byte packed binary coded decimal
DWord	Unsigned 32-bit integer
Long	Signed 32-bit integer
Float	32-bit floating point value
Double	64-bit floating point value
String	ASCII text string
Date	Floating-point OLE automation date

Note: 64-bit integer types, LLong and QWord, are not supported.

SQL Authentication

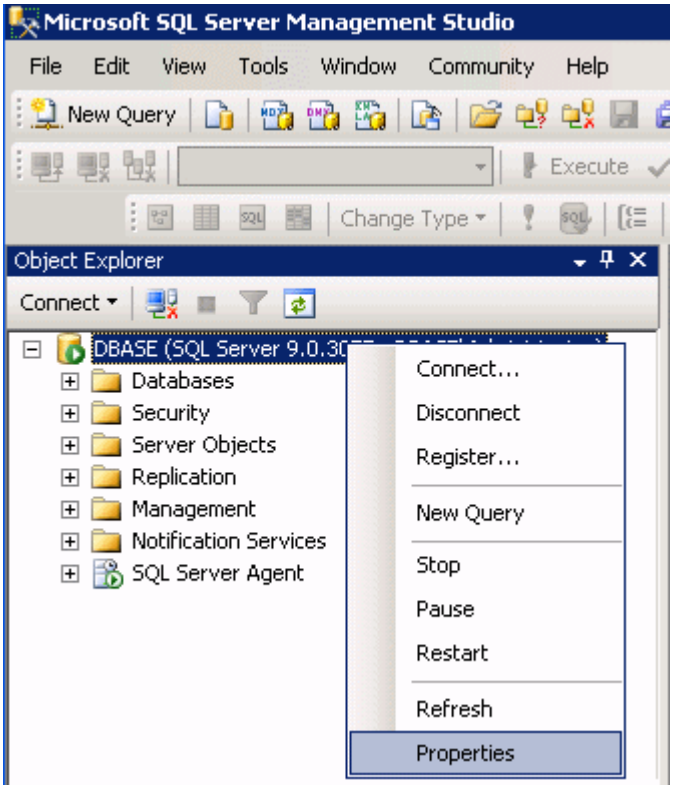
Select one of the links below to jump to that section of SQL Authentication setup.

Setting up SQL Authentication
Running as a System Service
Connecting Remotely as a System Service

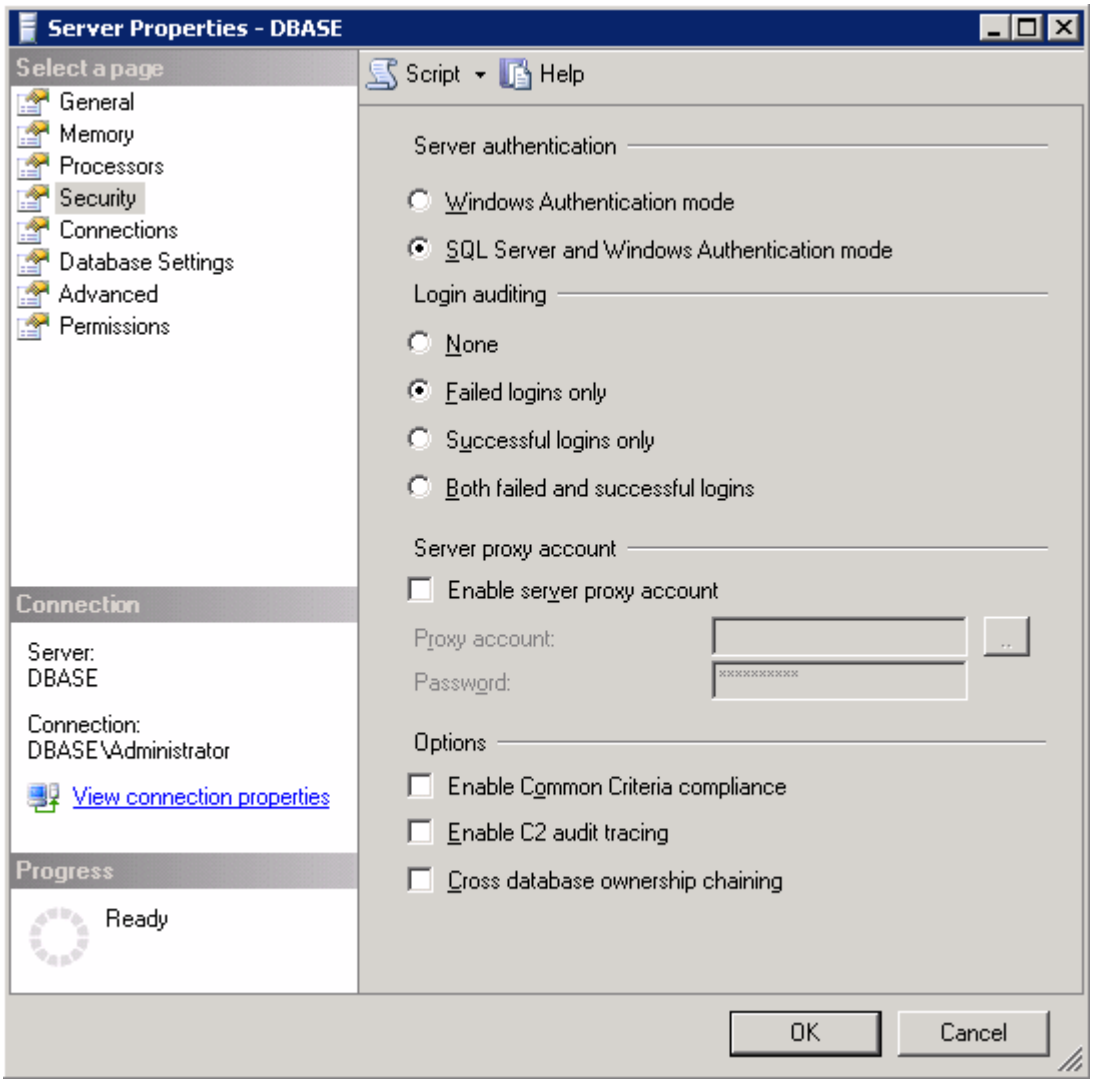
Setting up SQL Authentication

The following instructions contain information on setting up an SQL authentication. This process usually only has to be done when the application is running as a System Service and is attempting to connect remotely to SQL server.

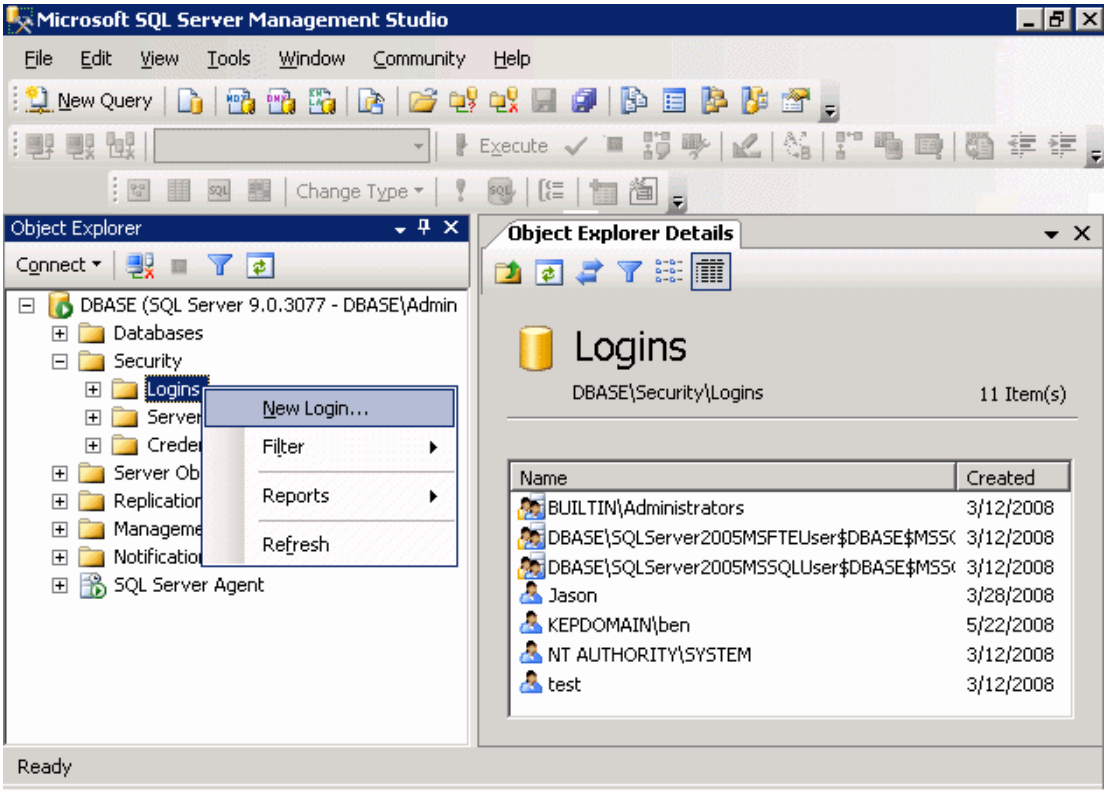
1. In the SQL manager, right-click on the SQL server icon and then open the SQL Server properties.



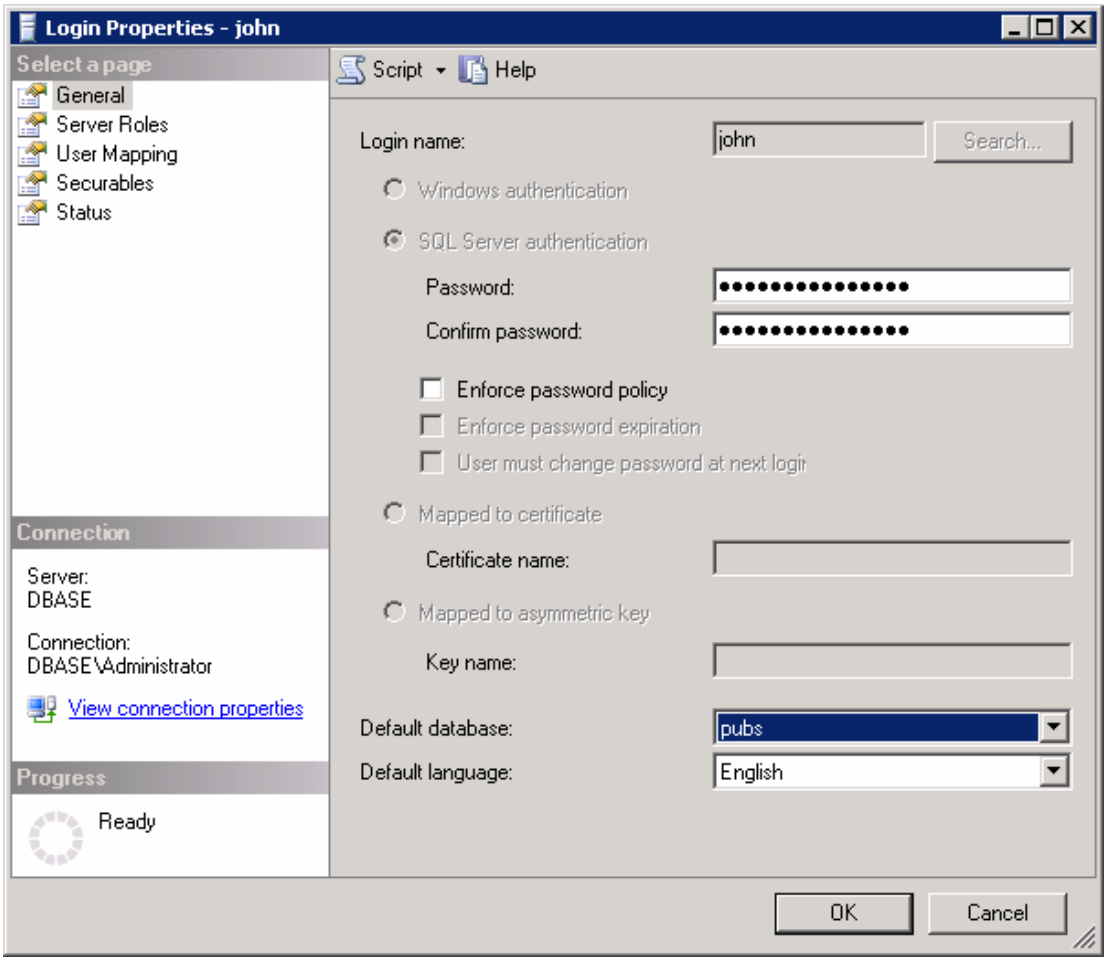
2. Select the **Security** page and choose the mixed authentication mode (**SQL Server and Windows Authentication mode** radio button).



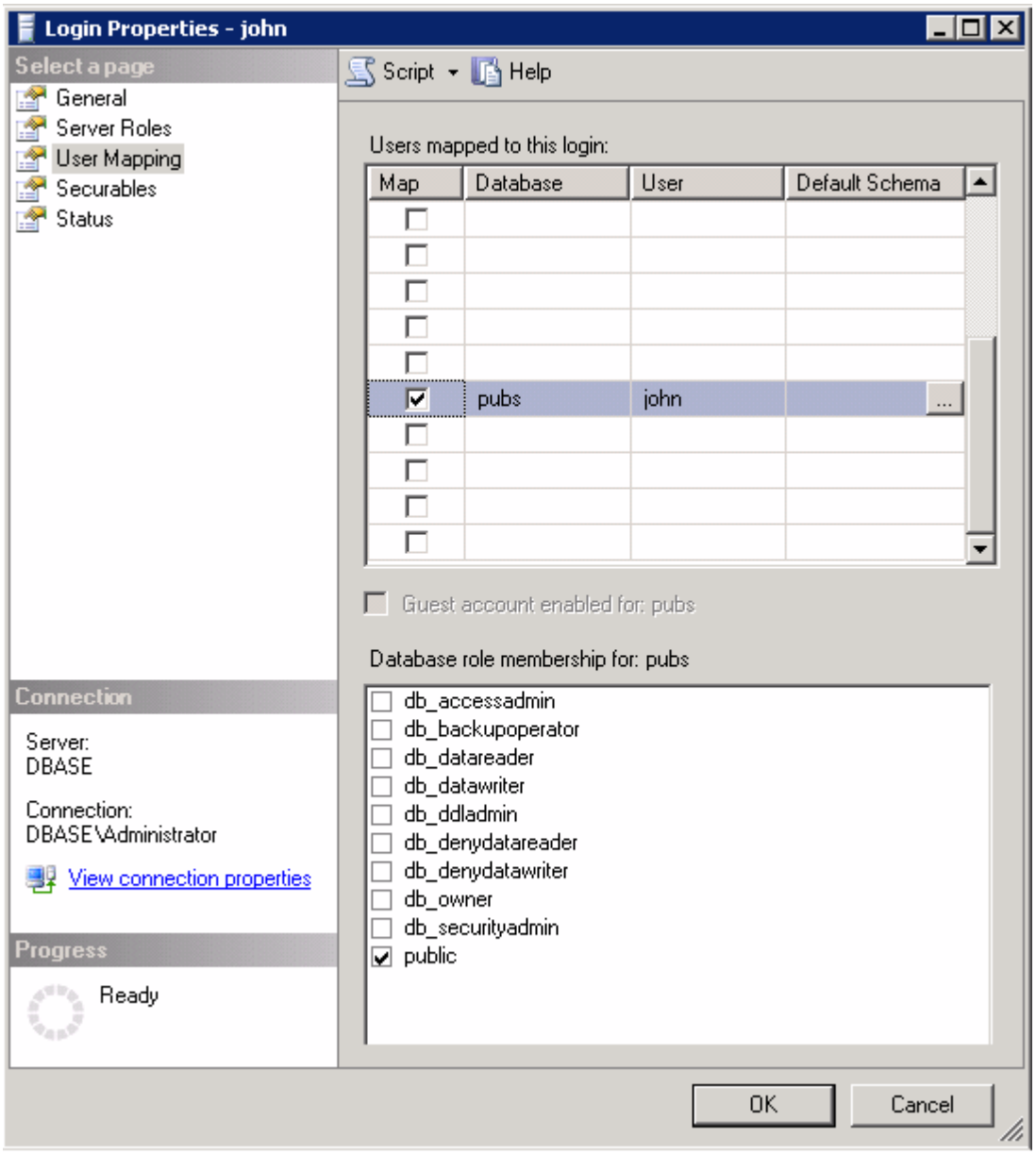
3. Within the tree menu, right-click on the security folder. Select **Logins** | **New user**.
4. Create and define a user's privileges.



5. Under the **General** page, a user name and password must be defined.

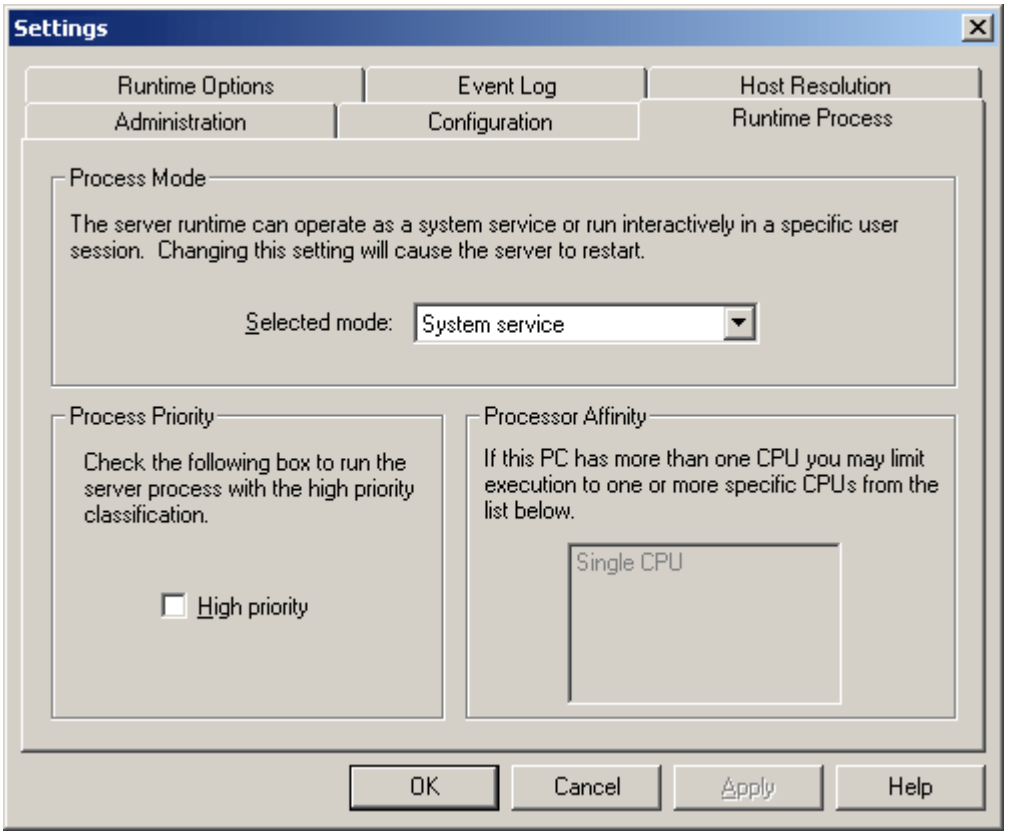


6. Next, select the **User Mapping** tab, then the database to connect. Select a role for the selected database. In this example, **Public** is used.

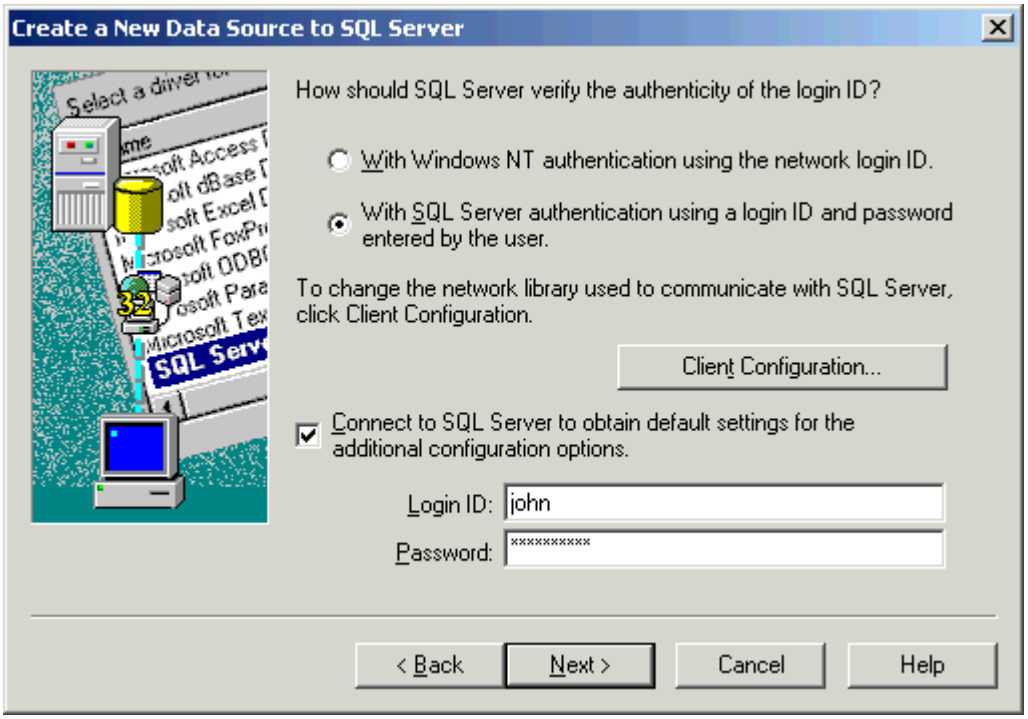


7. Next, right-click the KEPServerEX **Administration** menu located in the System Tray. Then, select **Settings...** | **Runtime Process**.

8. In **Selected Mode**, select **System Service**, and click **OK**.



9. When the DSN is configured, a series of DSN setup dialogs appear. In **Create a New Data Source to SQL Server**, enable **With SQL Server authentication using...** and **Connect to SQL Server to obtain...**. Then enter the user's Login ID and password (that were defined in the SQL Server).



Running as a System Service

Normally, an OPC server that only supports stand alone program operation is forced to shut down when its host machine experiences a user login or logout. However, this server can continue to supply OPC data across user login sessions by running as a System Service. The ability to run as a System Service is crucial for many applications where the server must provide data to OPC clients via DCOM. For these applications, the loss of a DCOM connection cannot be tolerated.

Note: For more information on running as a System Service, refer to the server's help documentation.

Connecting Remotely as a System Service

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

Windows Authentication

Windows Authentication allows the application to authenticate with the SQL server using Windows credentials. It requires that both the application and the SQL server be located on the same domain.

When the application is running in Interactive Mode, the Windows credentials of the user that launched the application are used during authentication. In most cases, this is the current logged-in user. As long as the user is part of the domain, and the SQL server is configured for Windows Authentication, it passes authentication.

When the application is running in System Service Mode, the NT AUTHORITY\SYSTEM account is used during authentication. This is a local account that fails Windows authentication. Users that require Windows Authentication in System Service Mode should refer to the instructions below.

1. To start, open the **Windows Service Configuration Manager**, and locate the Runtime service.
2. Right-click on the service and select **Properties**. Access the **Log On** tab.