Technical Note

Database Authentication

Several Kepware products utilize a database for storage of values or archived data. The instructions below focus on the settings relevant for interaction with Kepware products; the exact configuration should be based on Windows best practices and security.

1. Types of Data Source Authentication

1.1 Windows Authentication

The SQL server is configured to use Windows authentication, by default using the network Login ID for security. This makes SQL server logon transparent and eliminates the need for a user name and password.

1.2 SQL Server Authentication

The SQL server has its own authentication process. To connect the OPC server to the SQL server using SQL authentication, first set the authentication method in the SQL server to allow SQL authentication. Define a Login ID and Password. Once completed, enter the user name and password within the OPC server under the channel properties Data Source Setting property group.

2. Data Source Setup

Important: Users can configure a DSN using the Microsoft® ODBC Data Source Administrator launched from Windows. Because the server is a 32-bit application, it is necessary that the 32-bit version of the administrator be used when configuring a System DSN. A 64-bit operating system launches the 64-bit version of the administrator by default; it may be necessary to browse to the SysWOW64 directory to access the 32-bit version.

Note: If using MySQL, the MyODBC driver must be installed on the PC running the OPC server.

1. Open the OPC server and create a new channel.
2. Right-click on the channel name and select **Properties** | **Data Source Settings**.
3. Click the **Browse...** button in the Data Source property.
4. In the Data Source selection dialog, click the **Configure DSN...** button.

Tip: This example uses System DSNs, which are local to the computer and can be accessed by any user with privileges. User DSNs, which are local to the computer but only accessible by the current user, can also be chosen.

5. Click **Add**.
6. Select the desired data source. In this example, a SQL Server data source is used.
7. Name the ODBC data source and select the SQL database to which it will connect.
8. Click **Next**.
9. In **Microsoft SQL Server DSN Configuration**, the **Windows NT Authentication** option should be selected by default. If it is not, select it and click **Next**.

If running the OPC server in System Service mode, refer to Running as a System Service.

10. Check **Change the default database to** and select a database from the drop-down menu. (Select the default database if the desired database is unavailable.)
11. Continue through the wizard, keeping the remaining settings at the default values.
12. Upon completion, test the data source connection by clicking **Test Data Source**.
    If the connection is good, the message "Test Successful" appears.

3. 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 applications where the server must provide data to OPC clients via DCOM. For these applications, the loss of a DCOM connection cannot be tolerated.

3.1 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.

3.1.1 Setting Up SQL Authentication

These steps are only necessary when attempting to connect remotely to SQL server while running as an NT service.

1. In SQL Manager, right-click on the SQL server icon in the tree and select **Properties**.
2. Open the **Security** tab and select the mixed authentication mode **SQL Server and Windows**.
3. In the tree menu, right-click on the **Security** folder.
4. Click **Logins** and select **New User**.
5. In the New Login dialog, select the **General** tab.
6. Define a user name and password.
7. Open the **Database Access** tab and select the database to which to connect.
8. Select a role for the database. In this example, **Public** is used.
9. When finished, click **Properties**.
10. Under **Database Role Properties**, click on **Permissions…**.
11. Check the boxes of the objects that the user will be able to access. In this example, the user is allowed to perform "Select" queries on the "author" table.
12. At this point, the Data Source Name (DSN) should already be set up. *If not, refer to Data Source Setup before continuing.*
13. In the OPC server, select **Tools** | **Options**.
14. On the **Service** tab, enable **Automatically Start as a Windows Service**.
15. Return to the OPC Server application.
16. Right-click on the channel and select **Properties**.
17. In the **Data Source Settings** property group, click **Configure DSN**.
18. Select the System DSN that was created and click **Configure**.
19. Confirm both the data source name and the server name and select **Next**.
20. Enable **With SQL Server Authentication using a login ID and password entered by the user**.
21. Check **Connect to SQL Server to obtain default settings for the additional configuration options**.
22. Enter the user name and password for the user defined in the SQL server.
23. Continue through the Wizard, keeping the default settings.

24. Test the data source connection at the end by selecting **Test Data Source**. If the connection is good, the message “Test Successful” appears.

4. SQL Authentication

4.1 Setting up a Microsoft SQL Server for SQL Authentication

The following instructions contain information on setting up an MS SQL Server for SQL authentication. This process is only required when the OPC server 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.
2. Open the SQL Server properties.
3. Select the **Security** page and choose the mixed authentication mode (**SQL Server and Windows Authentication mode** radio button).
4. Within the tree menu, right-click on the security folder. Select **Logins** | **New user**.
5. Create and define a user’s privileges.
6. Under the **General** page, a user name and password must be defined.
7. Select the **User Mapping** tab, then the database to connect.
8. Select a role for the selected database. In this example, **Public** is used.
9. Right-click the KEPServerEX **Administration** menu located in the System Tray.
10. Select **Settings...** | **Runtime Process**.
11. In **Selected Mode**, select **System Service** and click **OK**.
12. 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...**.
13. Enter the user's Login ID and password (defined in the SQL Server).

5. 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. 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.
3. Select **This account** and enter the domain name and password used for Windows Authentication.
4. Restart the service.

Note: The procedure described above may restrict the application's permissions if the domain account does not have administrative privileges on the system. If the account cannot be given administrative privileges, SQL Authentication should be used instead.