

Using XLReporter with DeltaV Event Journal

ISA-18.2 Alarm Management Compliance

Overview

XLReporter takes historical values directly from the DeltaV Event Journal database automatically to populate workbooks, periodically, via VBA or on event, without needing Excel. The award-winning **XLReporter** history engine can return either raw or calculated Industrial metrics directly to the reports.

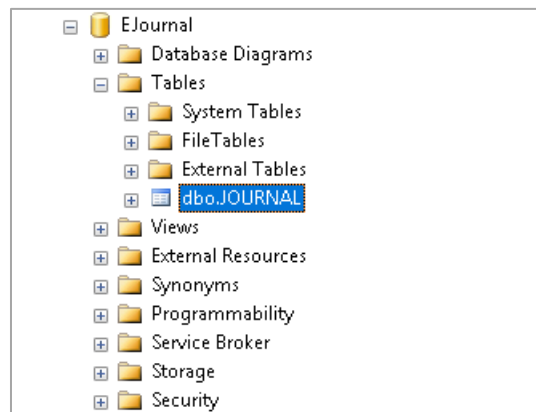
The DeltaV Event Journal is a Microsoft SQL Server database with event data logged by DeltaV.

Distribute the reports by email, FTP, file server and printers. Enable **XLReporter's** Web Portal and access reports from any device supporting a web browser such as a smart phone or tablet.

Prerequisites

Verify Database

To verify, open **Microsoft SQL Server Management Studio** and connect to the SQL Server instance where the Event Journal database is configured.



- Expand the **EJournal** database.
- Expand **Tables**.
- Select the **JOURNAL** table.

Right click the **JOURNAL** table and choose *SELECT TOP 1000 Rows*. This should display data from the table.

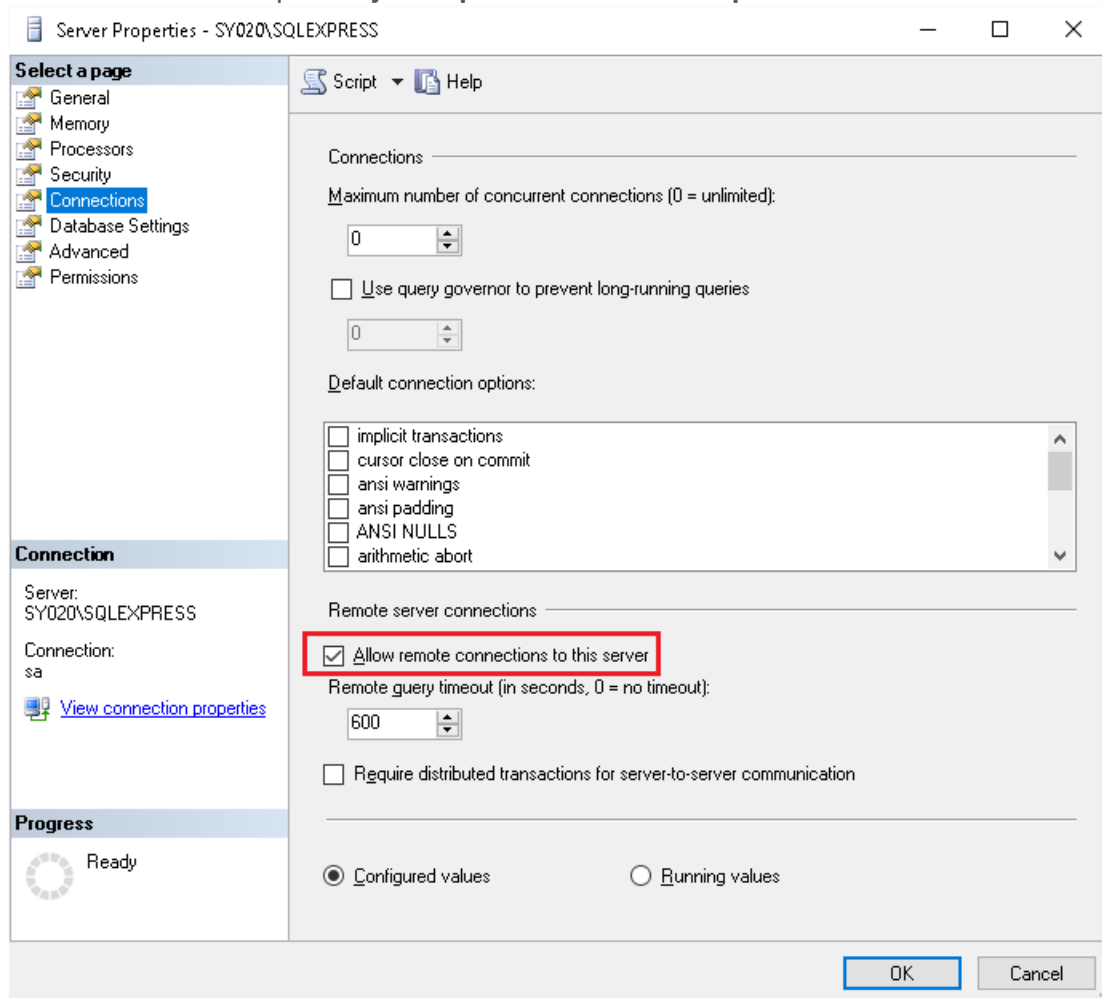
If the database or table does not exist, or no data is displayed, contact Emerson Automation Solutions technical support to troubleshoot and correct these issues.

SQL Server Considerations

Remote Connection

In order for XLReporter to communicate to SQL Server remotely, remote server connections must be enabled in SQL Server. Open the SQL Server Management Studio and connect.

- Right click the server at the top of **Object Explorer** and select **Properties**.



- Under **Select a Page**, select **Connections**.
- In the **Remote server connections** section, check **Allow remote connections to this server**.

Authentication

Emerson Automation Solutions recommends that the SQL Server instance allows Windows Authentication only. This means that to connect remotely the Windows user on the remote machine must be valid on the machine with SQL Server.

Server Name and Port

Emerson Automation Solutions recommends that the SQL Server Browser Service not be running for the SQL Server instance containing the **EJournal** database. Therefore, when connecting remotely, the SQL Server Instance name must be manually entered. It will not appear in any list.

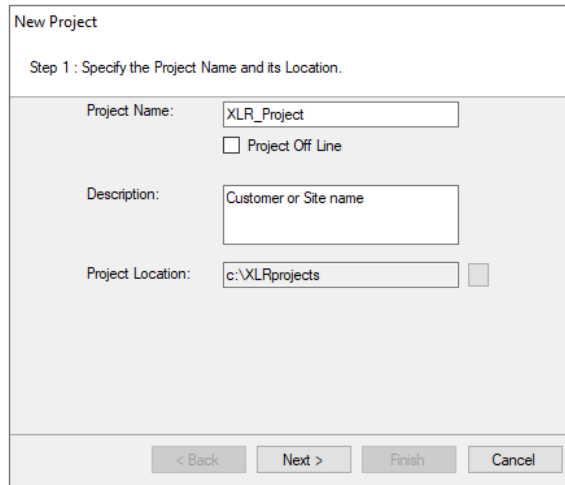
In DeltaV version 14.3 and above, the Port Number for the SQL Server instance containing the **EJournal** database is **55114**, which is not the default port of SQL Server.

Create a Project

From the **XLReporter Project Explorer** select **File, New** to start the **Project Wizard**. This will give step-by-step instructions on creating a project.

Step 1

- Enter a **Project Name** and **Description** (optional).



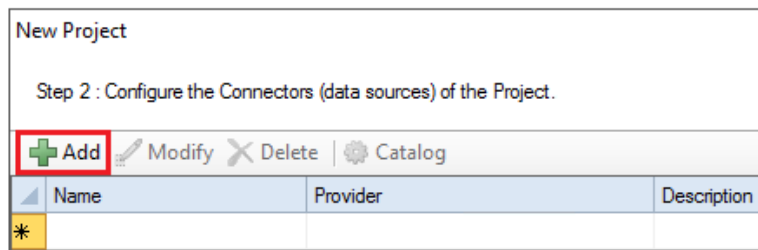
The screenshot shows the 'New Project' dialog box with the following fields and options:

- Project Name:** XLR_Project
- Project Off Line
- Description:** Customer or Site name
- Project Location:** c:\XLRprojects

Buttons at the bottom: < Back, Next >, Finish, Cancel.

Step 2

- Configure the data connector, click **Add**



The screenshot shows the 'New Project' dialog box with the following options and table:

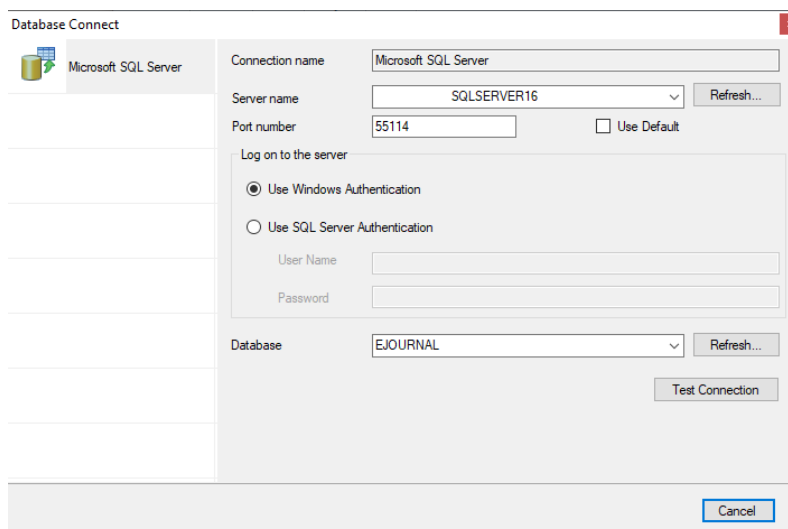
- Buttons:** + Add (highlighted), Modify, X Delete, Catalog

Name	Provider	Description
*		

Select **Advanced Modules, Alarm Management (ISA-18.2)**

- Set the **Provider** to *Emerson Automation Solutions DeltaV Event Journal*.

Under **Primary Server**, click the browse pushbutton (...) to connect to the database.



The screenshot shows the 'Database Connect' dialog box with the following settings:

- Connection name:** Microsoft SQL Server
- Server name:** SQLSERVER16 (Refresh...)
- Port number:** 55114 (Use Default)
- Log on to the server:**
 - Use Windows Authentication
 - Use SQL Server Authentication
 - User Name:** (empty)
 - Password:** (empty)
- Database:** EJOURNAL (Refresh...)
- Buttons:** Test Connection, Cancel

If XLReporter is installed on the same machine as the SQL Server containing the Event Journal database, accept all default settings. Otherwise, with **Microsoft SQL Server** selected,

- Set **Server name** to the SQL Server instance with the DeltaV Event Journal. This will have to be manually entered.
- If DeltaV is version 14.3 or above, for **Port number** uncheck *Use Default* and specify 55114. For older versions leave *Use Default* checked.
- Set the **Log on to the server** to *Use Windows Authentication*.
- For **Database**, select *EJOURNAL*.

Click **Test** to verify the settings.

Metrics Table

When the data connector is saved, one of the templates automatically added to the project is called *AlarmReport* which contains key alarm metrics of the alarm system and the underlying alarms that influence the KPI. These KPIs can be exported periodically, such as every day, to an external database. With the KPIs stored in the **Metrics Table**, long term trends can be produced.

When **Create Metrics Tables** is checked, the **Connector** list is enabled and provides a list of every other connector in the project that is connected to a database.

If this option is enabled, when the connector is saved, the **Update Database** window appears showing that new tables have been added to the database to receive the KPI/alarm information.

```

Done DROP TABLE TableKPI
Done DROP TABLE TableTop10
' Create KPI table
Done CREATE TABLE TableKPI (Facility VARC
Done CREATE INDEX TableKPIIndex ON Table
'Create Top 10 Table
Done CREATE TABLE TableTop10 (Facility VA

```

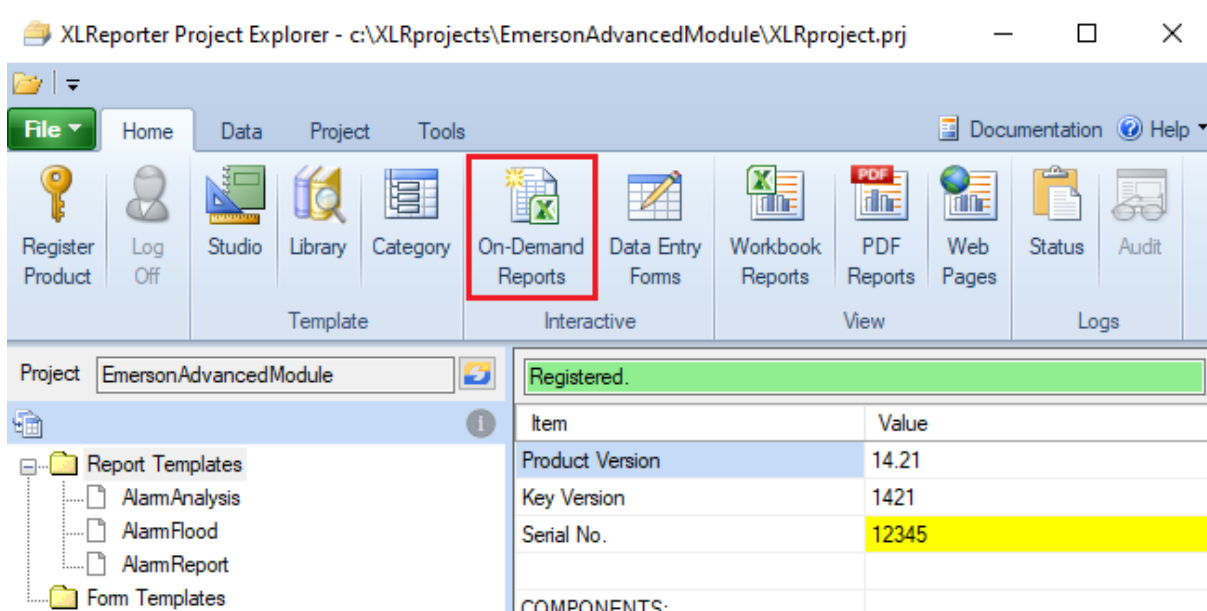
Under **Table/ View** set **Name** to *Journal*, click **OK**.

Three reports are automatically added to the project called *AlarmAnalysis*, *AlarmFlood*, and *AlarmReport* for ISA18.2.

Verify Data Communication

On-Demand Report

From the **XLReporter Project Explorer**, select **On-Demand Reports** to open the On-Demand application.



The templates *AlarmAnalysis*, *AlarmFlood*, and *AlarmReport* will be listed in the left pane.

Item	Value
Product Version	14.21
Key Version	1421
Serial No.	12345

Setting	Value
Position	All
Module	All
Attribute	All
Cycle	All

- Select *Alarm Report*.
- Enter a valid *start* and *end* date.
- Click **Refresh**.