



Energy AnalytiX

Creating Energy Consumption Reports



APPLICATION NOTE

September 2013

Description: Guide to creating reports on Energy AnalytiX data.

OS Requirement: Vista x64/ Windows 7 x64/ Windows 8 x64/ Windows Server 2008 x64/ Windows Server 2008 R2 x64/ Windows Server 2012

General Requirement: Installation of AnalytiX v10.8 and ReportWorX v9.3. General Knowledge of Energy AnalytiX, ReportWorX32 and SQL.

Introduction

Energy AnalytiX runtime views are populated with data based on Common Language Runtime (CLR) stored procedures integrated within the Energy AnalytiX runtime database. The same CLR stored procedures are available to ReportWorX v9.3, BridgeWorX v9.3, GridWorX, Data Mining OPC Server, Microsoft SQL Server Reporting Service, etc.

Energy AnalytiX Runtime Views Support

The following CLR stored procedures are currently available for Energy AnalytiX runtime views.

- EA_GetAllAssetFinancialData
- EA_GetAllAssetNormalizedFinancialData
- EA_GetAllAssetSimpleFinancialData
- EA_GetAllAssetSimpleNormalizedFinancialData
- EA_GetAssetFinancialData
- EA_GetAssetNormalizedFinancialData
- EA_GetAssetNormalizedSimpleFinancialData
- EA_GetAssetSimpleFinancialData
- EA_GetCostCenterFinancialData
- EA_GetCostCenterSimpleFinancialData
- EA_GetDDCuSumData
- EA_GetDegreeDaysAnalysisData
- EA_GetDerivedCalculationValues
- EA_GetESEnergyUseReport
- EA_GetESITEnergyUseReport
- EA_GetESWaterUseReport
- EA_GetLoadConsumptionRatio
- EA_GetMeterValues
- EA_GetSimpleDerivedCalculationValue
- EA_GetSimpleLoadConsumptionRatio
- EA_GetSimpleMeterValue
- EA_GetSimpleStandardCalculationValue

- EA_GetSimpleWeatherData
- EA_GetStandardCalculationValues
- EA_GetWeatherData

Energy AnalytiX General Stored Procedure Support

The following CLR stored procedures are currently available to help you configure drill-down reports based on AnalytiX properties and Asset configuration data.

- EA_GetAssetConfigurationData
- EA_GetAllMetadataltems
- EA_GetAssetMetadataValues
- EA_GetChildrenStandardCalculationValues
- EA_GetChildrenDerivedCalculationValues
- EA_GetMultipleMeterValues
- EA_GetFilteredStandardCalculationValues
- EA_GetFilteredSimpleStandardCalculationValues
- EA_GetFilteredDerivedCalculationValues
- EA_GetFilteredSimpleDerivedCalculationValues
- EA_GetFilteredAssetFinancialData
- EA_GetFilteredAssetSimpleFinancialData
- EA_GetAssetConsumptionDifferentials

Energy AnalytiX General User Functions

The following CLR user functions are currently available to help you configure drill-down reports based on AnalytiX properties and Asset configuration data.

- SqlDateTime EA_ConvertToLocalTime(DateTime dateInUTC)
- SqlDateTime EA_ConvertToUTCtime(DateTime dateInLocal)

Each of these stored procedures is described in the GENESIS64 Silverlight Help, accessible from within Workbench-SL or PortalWorX-SL.

To access the Help, click on the blue question mark () on the top right of the Workbench-SL or PortalWorX-SL windows (within your Web browser) or on any 'Help' or question mark button within any open configuration properties. Then, navigate within the Table of Contents to Workbench SL → Energy AnalytiX → Energy AnalytiX Runtime Options → CLR Reporting.



Energy AnalytiX

Creating Energy Consumption Reports



APPLICATION NOTE

September 2013

A Complete description for the “EA_GetStandardCalculationValues” CLR stored procedure is included below as an example that will be used in this Application Note.Goal

We will configure and run a report in ICONICS ReportWorX using Energy AnalytiX Data and “EA_GetStandardCalculationValues” CLR stored procedure.

The following example assumes you have already configured Energy AnalytiX and have logged energy data. We also anticipate you have at least some familiarity with ICONICS ReportWorX from the BizViz suite.

EA_GetStandardCalculationValues

EA_GetStandardCalculationValues retrieves standard calculations data.

Syntax:

EA_GetStandardCalculationValues(*int catalogAssetID*, *bool retrieveChildren*, *string calculationName*, *DateTime startDate*, *DateTime endDate*, *short timeAggregate*, *short dayAggregate*, *short dayOfWeek*, *int meterTypeID*)

Parameters:

- **catalogAssetID:** The a=Asset catalog based ID.
- **retrieveChildren:** A flag to indicate if child assets' data should be retrieved.
- **calculationName:** The calculation's internal name (i.e., "Consumption").
- **startDate:** The start date of data retrieval.
- **endDate:** The end date of data retrieval.
- **timeAggregate:** The time period aggregate for data retrieval. Values are: Base Interval = 0 (not currently available), Hour = 1, Day = 2, Week = 3, Month = 4, Year = 5, Entire Interval = 6.
- **dayAggregate:** The day aggregate for data retrieval. Values are: All Days = 0, Week days Only = 1, Weekends Only = 2, Specific Day = 3.
- **dayOfWeek:** The specific day for data retrieval. Values are: Sunday = 0, Monday = 1, Tuesday = 2, Wednesday = 3, Thursday = 4, Friday = 5, Saturday = 6. It is used only if the *dayAggregate* is set to 3.
- **meterTypeID:** The meter type ID for data retrieval.

Result set:

CatalogAssetEntryID, SqlDbType.Int
 EnergyAssetEntryID, SqlDbType.Int
 AssetName, SqlDbType.NVarChar, 255
 CostCenterID, SqlDbType.Int
 CostCenterName, SqlDbType.NVarChar, 256
 MeterTypeID, SqlDbType.Int
 MeterTypeName, SqlDbType.NVarChar, 128
 RateModelID, SqlDbType.Int
 CalculationID, SqlDbType.Int
 CalculationName, SqlDbType.NVarChar, 64
 CalculationDisplayName, SqlDbType.NVarChar, 256

IsRollup, SqlDbType.Bit
 IsDerived, SqlDbType.Bit
 StartDate, SqlDbType.DateTime
 EndDate, SqlDbType.DateTime
 Value, SqlDbType.Float
 UnitsID, SqlDbType.Int
 UnitDisplayName, SqlDbType.NVarChar, 64
 FullSetOfData, SqlDbType.Bit
 Status, SqlDbType.SmallInt
 ErrorCode, SqlDbType.SmallInt
 AverageValue, SqlDbType.Float
 MaxValue, SqlDbType.Float
 MinValue, SqlDbType.Float

Creating a new report template

1. Open ReportWorX configurator
2. Create a new Report Template and name it **tmp_EA_GetStdCalcVal**
3. Open report template, excel will launch with an empty worksheet

Creating a new connection to database

4. Right-click on cell and select **Connect to new Data Source** → **Open Database.**
5. In the Database Connection Configuration Wizard create a new connection to the **Energy AnalytiX** database.
6. Once you create connection string you will be able to create a **Database Query**
7. Click on **Edit** button and select **Store Procedure Call**

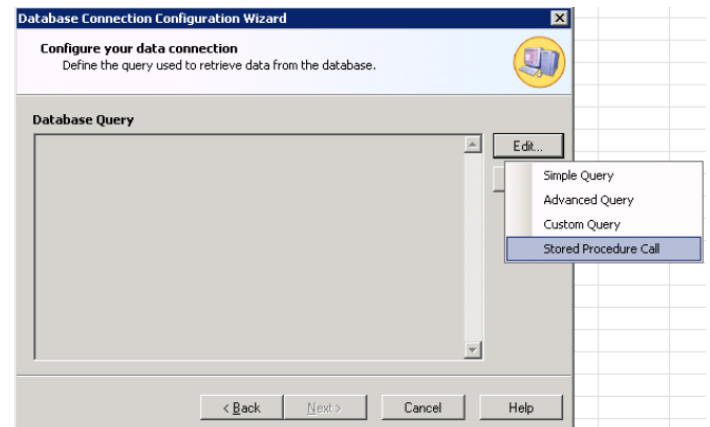


Figure 1 - The Database Connection Wizard

8. Select the **EA_GetStandardCalculationValues** CLR Stored Procedure.



Energy AnalytiX

Creating Energy Consumption Reports



APPLICATION NOTE

September 2013

Configuring Stored Procedure Parameter Values

We will define three different parameter types depending on the parameter usage.

- Manually enter a value
- Automatically evaluate an expression
- Select from a dynamic list of values

The **Procedure Parameters** are already populated in the **SQL Command Designer** dialog.

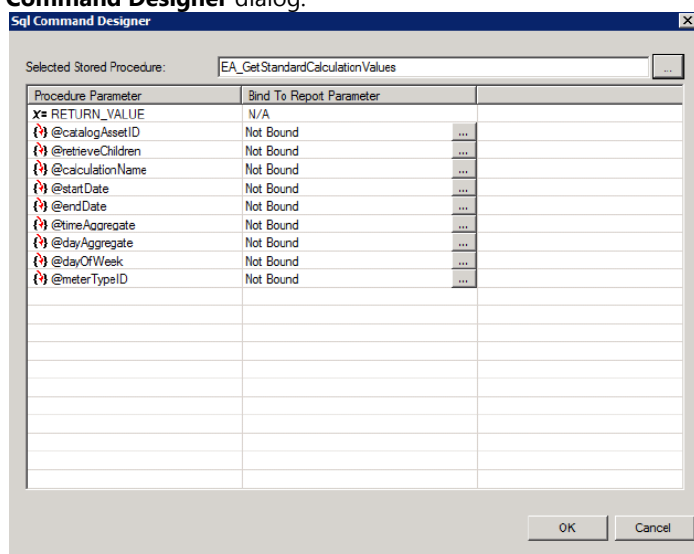


Figure 2 – The SQL Command Designer

Manually entering a value - Binding To Report Parameters

1. We will define a **Parameter Type** with name "**Manually enter a value**" for the following three procedure parameters in the bullet list below.

• Manual Parameter value bullet list:

- o @retrieveChildren = 0
- o @startDate = 8/28/2013 4:00:00 AM (**select the date**)
- o @endDate = 8/28/2025 4:00:00 AM (**select today's date**)

NOTE: All input date parameters are in local time format. The CLR stored procedures internally convert to UTC time, retrieve the data, and send the results to the client in local time format.

2. We also have to enter text into the **Web Prompt** field.

• Manual Web Prompt:

- o @retrieveChildren = **Include children assets (1=yes, 0=no):**
- o @startDate = **Enter Start Date:**
- o @endDate = **Enter End Date:**

3. Within the "Configure the parameter properties" dialog, enter the **Default value:** as defined in the previous bullet list.

Automatically evaluate an expression - Binding To Report Parameters

1. We will define a **Parameter Type** with name "**Automatically evaluate an expression**" for the following procedure parameters in the bullet list.

• Automatic Parameter value bullet list:

- o @calculationName = Consumption
- o @timeAggregate = 1
- o @dayAggregate = 0
- o @dayOfWeek = 0

Within the "Configure the parameter properties" dialog, enter the **Constant value:** as defined in the previous bullet list.

Select from a dynamic list of values - Binding To Report Parameters

1. We will define a **Parameter Type** with name "**Select from a dynamic list of values**" for the following two procedure parameters in the bullet list,

• Select Parameter value bullet list:

- o @meterTypeID = Selected from a list
- o @catalogAssetID = Selected from a list

2. We also have to enter text into the **Web Prompt** fields.

• Select Web Prompt:

- o @meterTypeID = **Select a Meter Type:**
- o @catalogAssetID = **Select Asset:**

3. Within the "Configure the parameter properties dialog", we would have to implement the following steps:

For @meterTypeID parameter:

- Use the **Connection String** as we created earlier
- In the **Simple Query** connect to **EA_CurrentMeterTypes** view
- From the list of **Column Names** select **MeterTypeID** and **MeterTypeName**
- As the **Default Value**, select the **Electric type** entry
- Click **OK** until you return to the **SQL Command Designer** dialog

For @catalogAssetID parameter:



Energy AnalytiX

Creating Energy Consumption Reports



APPLICATION NOTE

September 2013

- Use the **Connection String** as we created earlier
 - In the **Simple Query** connect to **ASC_EquipmentView** view
 - From the list of **Column Names** select **ID** and **Name**
 - As the **Default Value**, select the equipment of your choice
 - Click **OK** until you return to the **SQL Command Designer** dialog
4. In **Parameter Browser** dialog you can change the Order of appearance in the Web Interface for parameters.

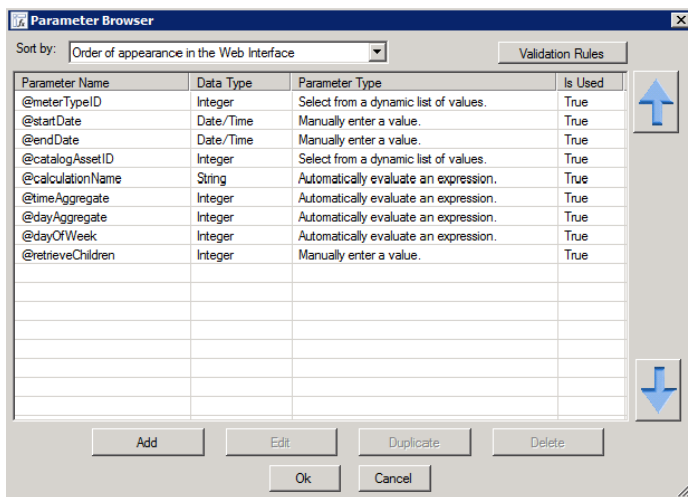


Figure 3 — Parameter Browser

Finishing the Database Connection Configuration Wizard

Your parameters should look similar to the following figure 4.

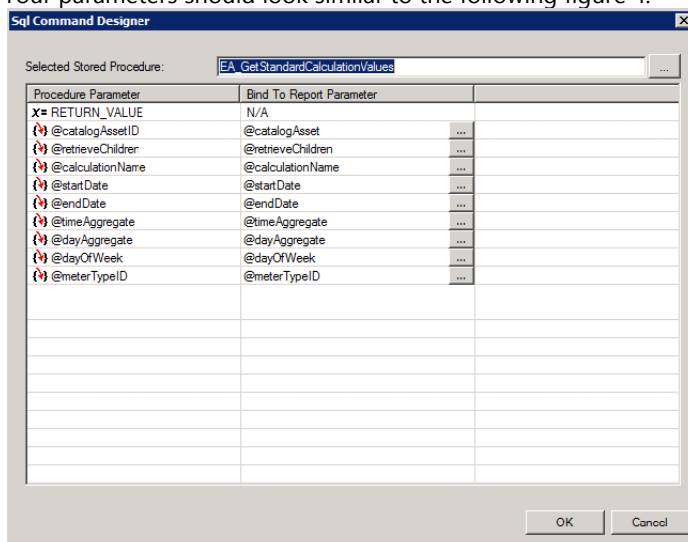


Figure 4 - The final configured parameters.

1. Click **OK**. You'll see the following message.

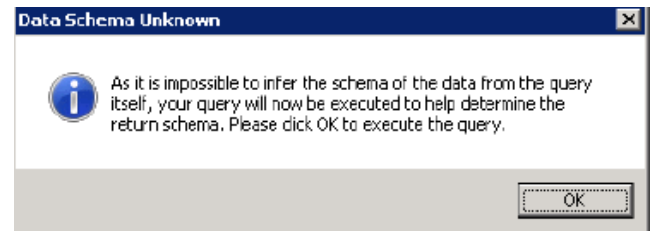


Figure 5

2. Click **OK** to proceed.
3. This opens the **Parameter Values** window.

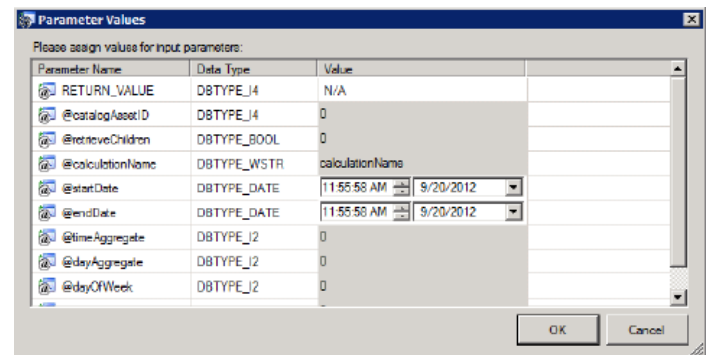


Figure 6 - The parameter values window.

4. Click **OK**. This returns you to the **Database Connection Configuration Wizard** dialog.
5. Click **Next** to continue.

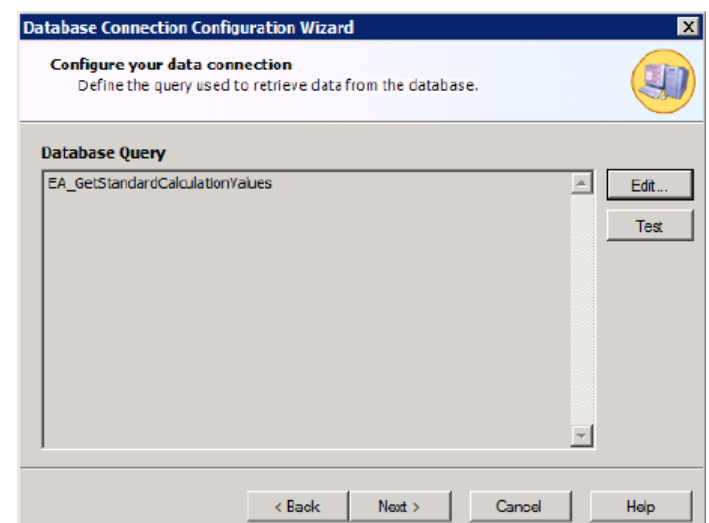


Figure 7 – The Database Connection Wizard dialog.

6. Click **Finish**.

Defining the Report Layout in Excel



Energy AnalytiX

Creating Energy Consumption Reports



APPLICATION NOTE

September 2013

Although you could work with the window at the current size, we suggest resizing the **Layout Manager** window wider so you can view and work with more cells easily.

The columns in row 10 were automatically defined by ReportWorX based on the columns in the SQL Server database table(s).

As shown in the following figure we will select the column names from the drop down list according to the definitions below.

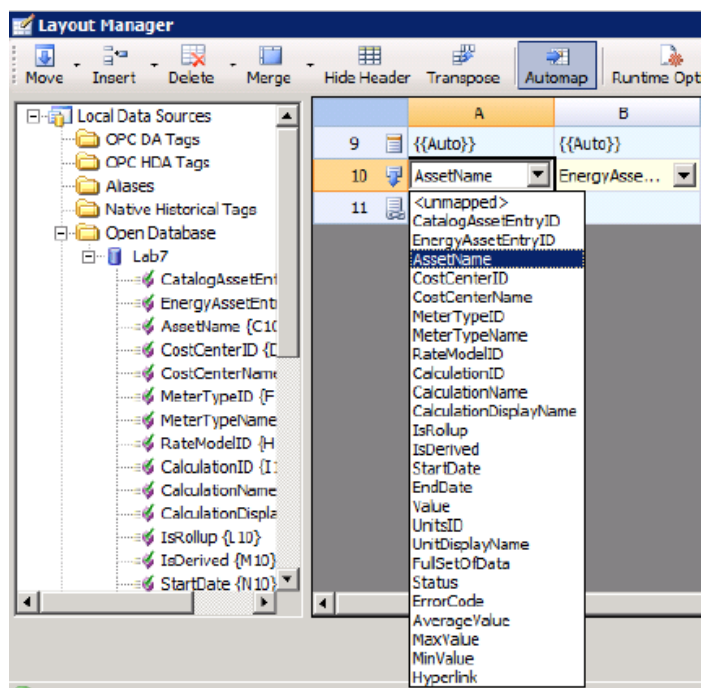


Figure 8 – The layout manager.

1. Set the column definitions.

• Report column definitions:

- A = AssetName
- B = CostCenterName
- C = MeterTypeName
- D = RateModelID
- E = StartDate
- F = EndDate
- G = Value
- H = AverageValue
- I = MaxValue
- J = MinValue

2. Close Layout manager

3. Save Report Template and return to ReportWorX Configurator

Creating a Report

1. In ReportWorX configurator create new report named **"rpt_EA_GetStdCalcVal"**
2. Assign the recently created template **"tmp_EA_GetStdCalcVal"** to this new report

Run Report

Now your report is ready to be executed. For execution you can use Unified Web Interface from the BizViz suite or PortalWorX-SL ReportWorX Executor in AnalytiX suite as it is necessary to manually type and select parameters for the report.

NOTE: Before executing the report through the ReportWorX Executor in PortalWorX-SL, it might be necessary to add your SQL server instance in the 'SQL Server Setup' application, available in ICONICS Tools.

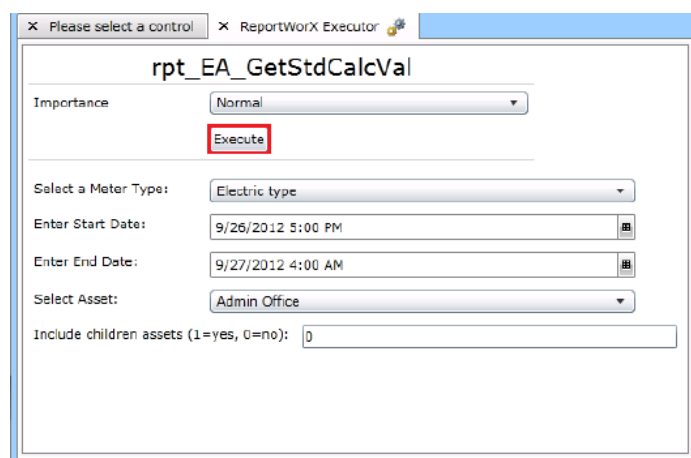


Figure 9 – Executing the report on the Unified Web Interface.