SQL Databases Normalized Service

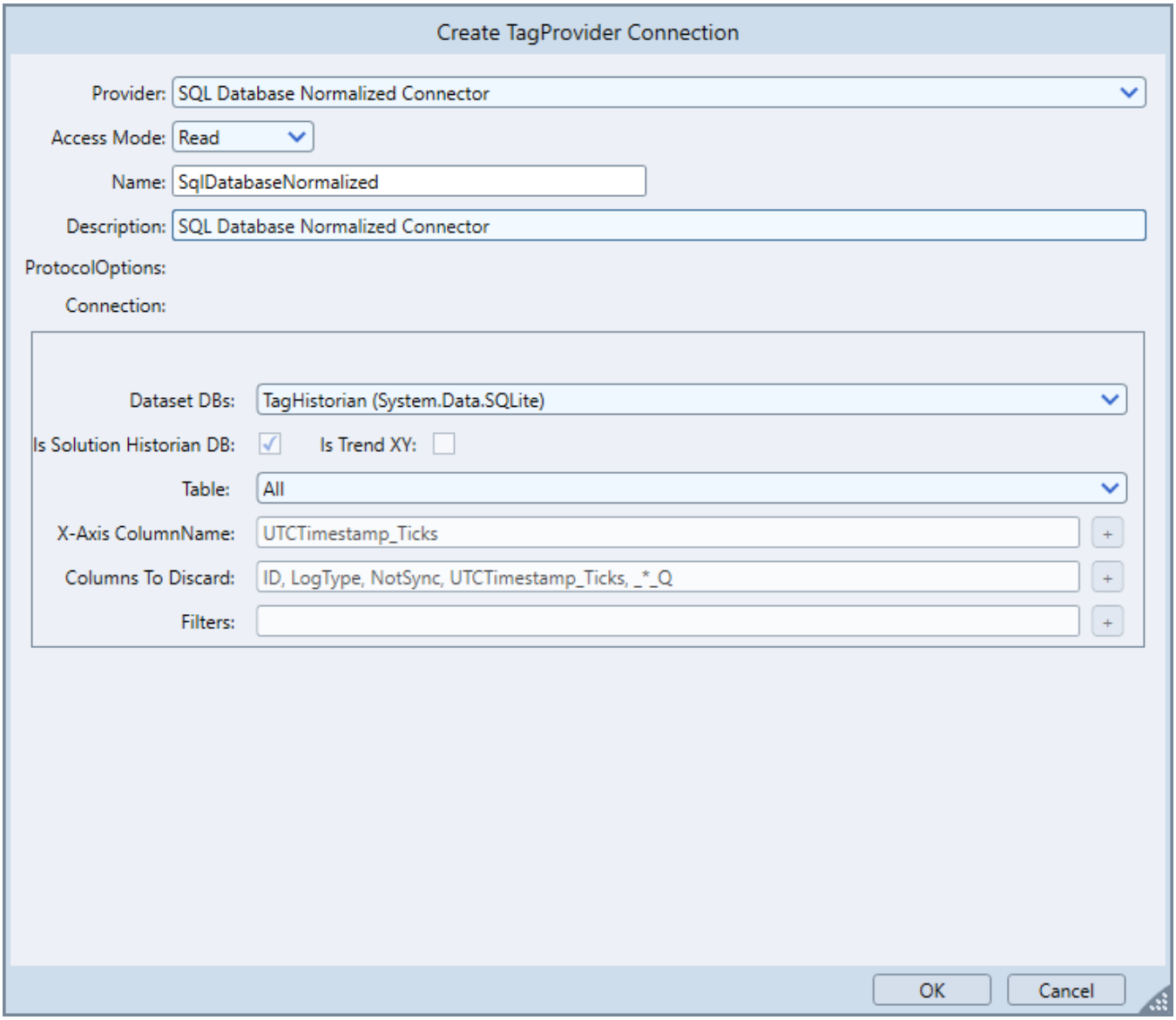
Information on how to set up a TagProvider connection with SQL Databases (Normalized).

The SQL Database Normalized Service connects to solution databases at runtime, enabling data querying without local Tags. Configure this TagProvider to display queried data on TrendCharts and other application areas, simplifying real-time access and utilization of normalized database information within the Unified Namespace.

Configuration
Connection Parameters
Connecting to a Solution Historian
Example

Configuration

To set up a new TagProvider connection navigate to **Unified Namespace/TagProvider Connections** and press the add button to open the New TagProvider Connection dialog. You can also create your TagProvider Connection directly from the AssetTree. To do this, navigate to **Unified Namespace/Assets Tree** and press  to open the New TagProvider Connection dialog. In the configuration dialog, select the **SQL Database Normalized Connector** on the Provider field and insert the required connection parameters. After that, press the Ok button to finish the provider configuration. Creating the connection adds no folder to the Asset Tree. To use its data with Asset(), open **Unified Namespace / Assets Tree**, select a folder (create one first if needed) and click **Insert data from TagProvider** on the toolbar: the folder linked to the TagProvider is created inside the selected folder. A linked folder cannot sit at the root of the Asset Tree.



This TagProvider is **Read-Only**.

Connection Parameters

Field	Description
Provider	Specifies the system the TagProvider will interface with. In this case, it interfaces with the SQL Database Connector.
Access Mode	Determines permissions for interacting with tags. **Read** allows data reading.
Name	User-defined name for the TagProvider instance. Identifies the specific configuration in the system.
Description	Explains the TagProvider's purpose or the systems it interfaces with. Provides context for its usage.
ProtocolOptions	Defines specific communication settings for the connection to the device. True or False.
Connection	Defines connection string for the SQL database connection settings, including database source, path, and additional parameters.

Field	Description
Dataset DBs	Specifies the database to be used for storing and retrieving tag data.
Is Solution Historian DB	Indicates if the database is used as a solution historian.
Is Trend XY	Specifies if the data is to be treated as XY trend data, where X is NOT time-based.
Table	Defines the table within the database to retrieve or store data from.
X-Axis ColumnName	Specifies the column name to be used for the X-axis in trend data.
Columns To Discard	List of columns to be discarded, separated by comma (',')
Filters	Specifies filters to be applied to the data retrieval process. It defines a hierarchical order for columns, separated by comma (','), when assembling the TreeViewStructure. You can use an **Asterisk** (*) as wildcard for column filtering. Some usage examples are listed below. *Temperature: Will filter out any column that **EndsWith** the **Temperature**. LineMonitor*: Will filter out any column that **StartsWith** the **LineMonitor**. _*_Q: Will filter out any column that **StartsWith** an underscore ' _ ', and **EndsWith** Q. Is useful for filtering out TagQuality Info.

Connecting to a Solution Historian

The SQL Database Normalized Connector can be used with any SQL database and schema that uses normalized datatables, but when used in conjunction with the SQL Historian tables, generated by the framework itself, some extended functional is available.

The flag for **Is Solution Historian DB** is **always true**, automatically filling some station parameters.

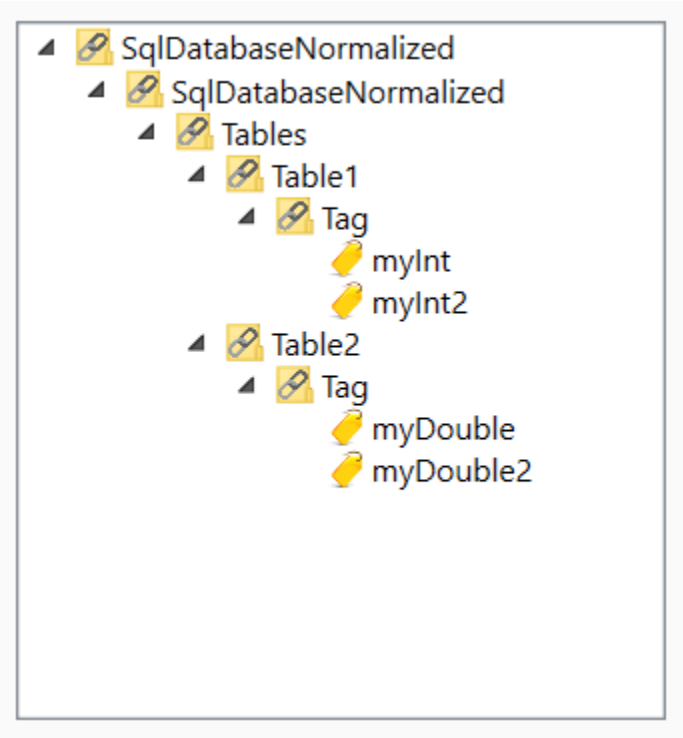
Is Trend XY: Input field is disabled. Default is false as it cannot be used for Normalized Datatables.

X-Axis Column Name: Input field is disabled. Default value is UTCTimestamp_Ticks.

Columns To Discard: Input field is disabled. Default value is ID, LogType, NotSync, UTCTimestamp_Ticks, _*_Q.

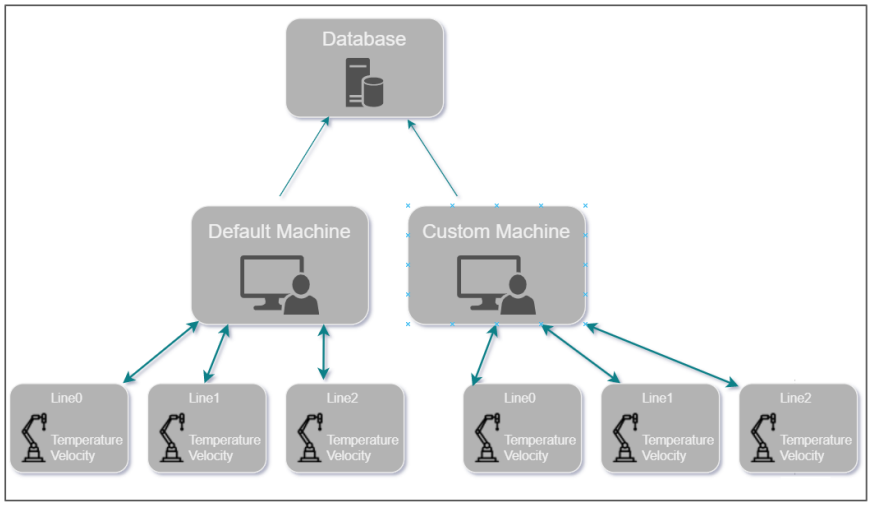
Filters: Input field is disabled. Default value is empty.

In Runtime, the DataModel Structure will behave as follows:



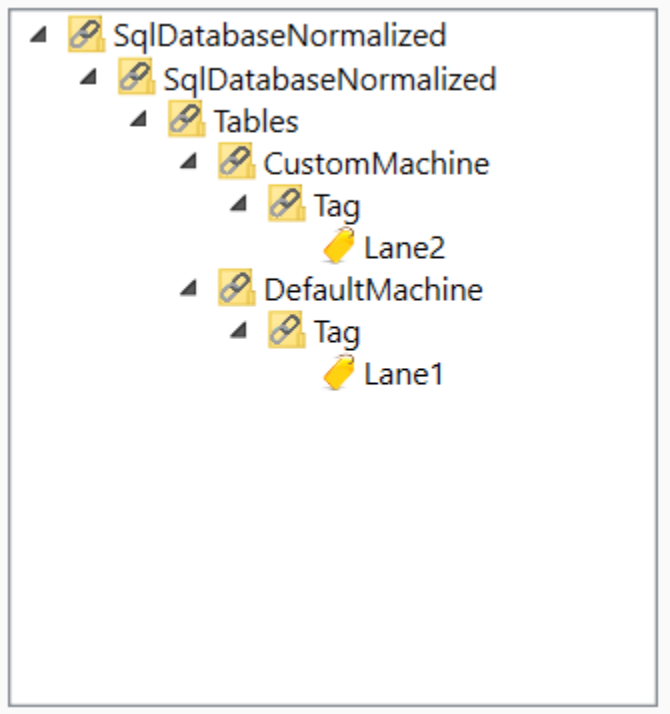
Example

Our Example will be a Multi-Line system, where different machines (PLCs) are acquiring data and sending them to a Main DataServer. All data is being stored in the same Table.



The result Table would look like this:

	ID	UTCTimestamp_Tic	UTCTimestamp_Tic	ObjIndex	ObjValue	ObjQuality	NotSync
▶	0	6388274416543...	6388274416543...	6	57,75264967152...	192	☑
	1	6388274416543...	6388274416543...	5	9,724745982167...	192	☑
	2	6388274416691...	6388274416691...	6	22,19022342994...	192	☑
	3	6388274416691...	6388274416691...	5	85,75502183347...	192	☑
	4	6388274416793...	6388274416793...	6	23,07630028827...	192	☑
	5	6388274416793...	6388274416793...	5	65,31241959988...	192	☑
	6	6388274416943...	6388274416943...	6	20,65126074189...	192	☑
	7	6388274416943...	6388274416943...	5	11,50521855517...	192	☑
	8	6388274417091...	6388274417091...	6	79,74868001279...	192	☑
	9	6388274417091...	6388274417091...	5	34,17961826528...	192	☑



In order to retrieve data from the Lines, based on which Machine stored them, the **TagProvider Configuration** parameters can be kept as default, and you can obtain them using the "Asset" method.
The paths below assume a folder named Plant in the Asset Tree, with a folder named SqlNormalized inside it that is linked to the root of the SqlDatabaseNormalized TagProvider. After the linked folder, the path follows the TagProvider's own tree (Tables, then the table, then each level of the tag name), with the TagProvider's separator, a dot.
For obtaining Lane 1 members we can use the following paths:
Lane1.Temperature: /Plant/SqlNormalized/Tables.DefaultMachine.Tag.Lane1.Temperature
Lane1.Velocity: /Plant/SqlNormalized/Tables.DefaultMachine.Tag.Lane1.Velocity
For obtaining Lane 2 members we use the following ones:
Lane2.Temperature: /Plant/SqlNormalized/Tables.CustomMachine.Tag.Lane2.Temperature
Lane2.Velocity: /Plant/SqlNormalized/Tables.CustomMachine.Tag.Lane2.Velocity

As an example, you can retrieve those values in a TextBox using the following LinkedValue:

```
Asset("/Plant/SqlNormalized/Tables.DefaultMachine.Tag.Lane1.Temperature")
```

In this section...
