Application Note: Bringing Data to Microsoft SQL Azure for Data Management and Analysis in the Cloud

Table of Contents

Overview

This document guides you through setting up data management and analysis in the cloud, using the Emerson Movicon.NExT™ SCADA platform the Microsoft Azure Cloud. Some goals of this application guide are to:

- Describe the procedures and actions to undertake to configure a Cloud platform based on the Microsoft Azure solution.
- Describe the procedures and actions to undertake along with useful tips to create a project based on the Movicon.NExT SCADA platform.
- Describe the procedures and actions to undertake to visualize data available in the Microsoft Azure Cloud by means of using the Microsoft Power BI solution.
- The Microsoft Azure and Microsoft Power BI products are not products managed and developed by Emerson. Therefore, we advise those using these services to contact Microsoft directly for any problems they may be experiencing and to obtain further information.

Cloud Computing Background

The term Cloud computing refers to the distribution of computer services and IT resources on the web such as data storage, processing or transmission. The Cloud is a container of data, information and services that can always be accessed from any device in any location connected in an internet infrastructure. The key point of this technology is the ability to use each of these individual things on demand over the internet without having to install programs or occupy memory on local devices.

There are many Cloud solutions on the market offered by web giants, such as Microsoft, Google and Amazon, and big specialized companies, such as QLik, Tableau, Pubnub and many others. Ultimately, there are many highly specialized solutions for IIoT, produced by specialists from the sector.

For example, the Databoom platform (www.databoom.com) is an example of how a Cloud platform can easily integrate with Movicon.NExT, thanks to a connector designed by Emerson for this purpose.

However, in this guide we have chosen to use Microsoft Azure as an example for Cloud storage and the PowerBI solution for Analytics. Microsoft Azure is one of the best known and most popular Cloud solutions available on the market for its openness, powerful performances and flexibility.

Naturally, any Movicon.NExT or Connext project can easily connect to the user's preferred Cloud solution among the many available on the market. Thanks to its openness and scalability, Movicon allows users to interface with any other Cloud solution available on the market.

What is Microsoft Azure?

Microsoft Azure (previously known as Windows Azure) is Microsoft's public cloud platform, that offers cloud computing services. The services offered by Microsoft Azure fall into three main categories, according to the delivery method adopted:

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)

- Software-as-a-Service (SaaS)

It also offers mBaaS services (mobile Backend as a Service). Specific services belonging to the categories include:

- Data processing resources
- Data storage and recording
- Data transmission and network interconnections
- Analysis
- Intelligence
- Automatic learning
- Security and identity management
- Data monitoring and management
- Services for developing applications

Although the creation of an account and its basic use is offered for free, each service charges a fee based on use and a payment method to determine the cost of a specific service. These services are delivered by a worldwide network of Microsoft datacenters.

What is Movicon?

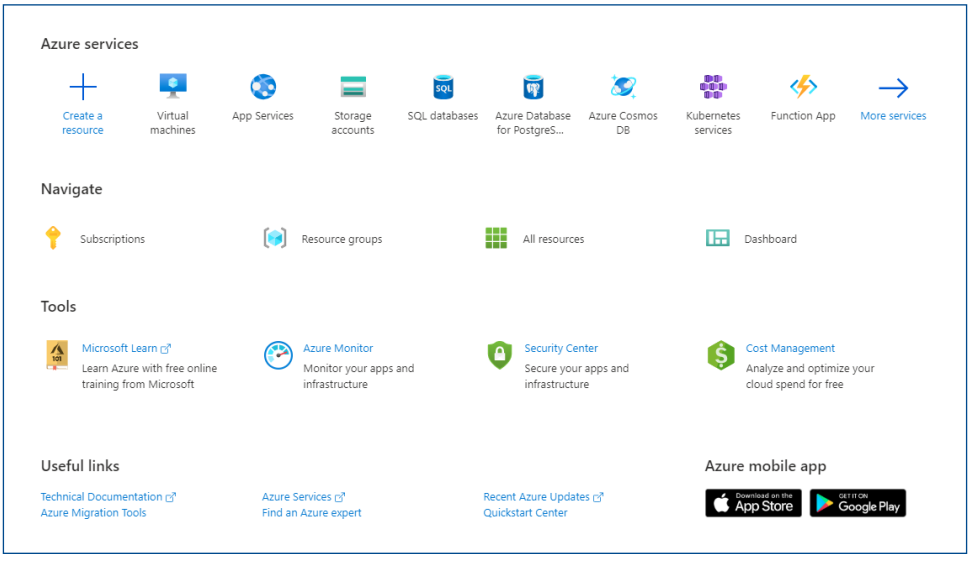
Emerson's Movicon.NExT and Connext products running on a range of hardware options are a flexible and scalable way of creating SCADA and IIoT solutions of all sizes. Users can deploy Movicon products where needed to connect with automation systems and other intelligent assets, and to route the data to cloud databases and other cloud-based resources.

Configuring Microsoft Azure

To use the Cloud Azure solution, you need to register on the dedicated portal at: https://azure.microsoft.com/.
It is best to access this portal with Microsoft Edge. This document does not describe the procedures used to register for an account on the Azure portal.

Creating a SQL Database Resource

After having created an account on the Azure portal, you can then create the various resources that are provided.



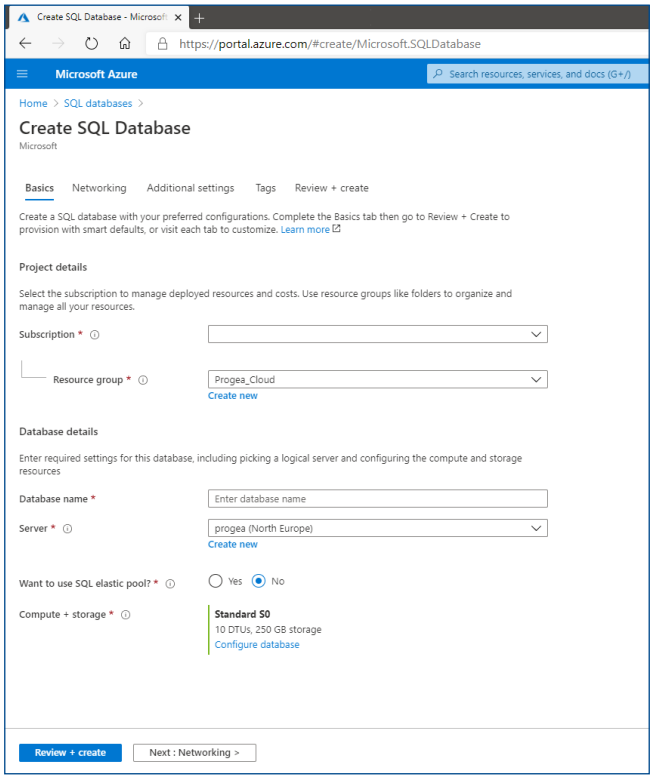
Azure SQL Databases

Clicking on the SQL Database icon will give you access to the SQL Cloud instance creation session.

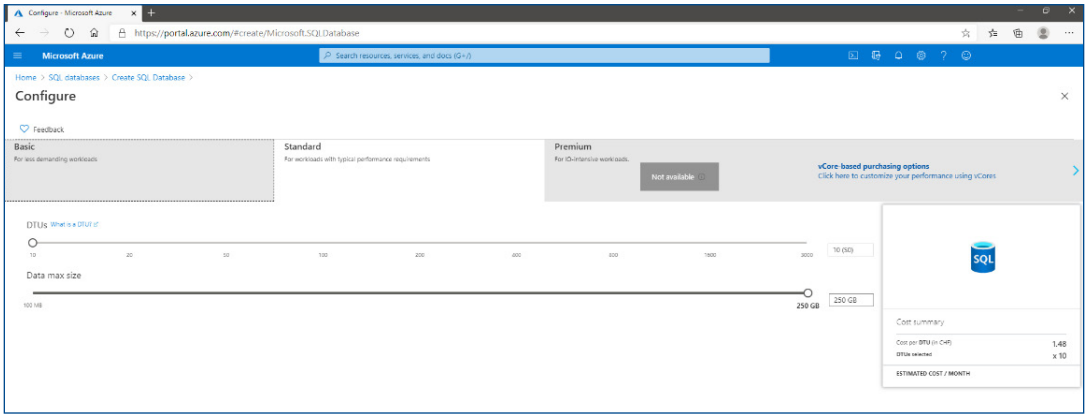
The following fields should be selected in this first step:

- Subscription
- Resource Group
- Identifying where process defects are originating, even if they are not detected until further in the production process

Afterwards define the name of the Database you wish to create.



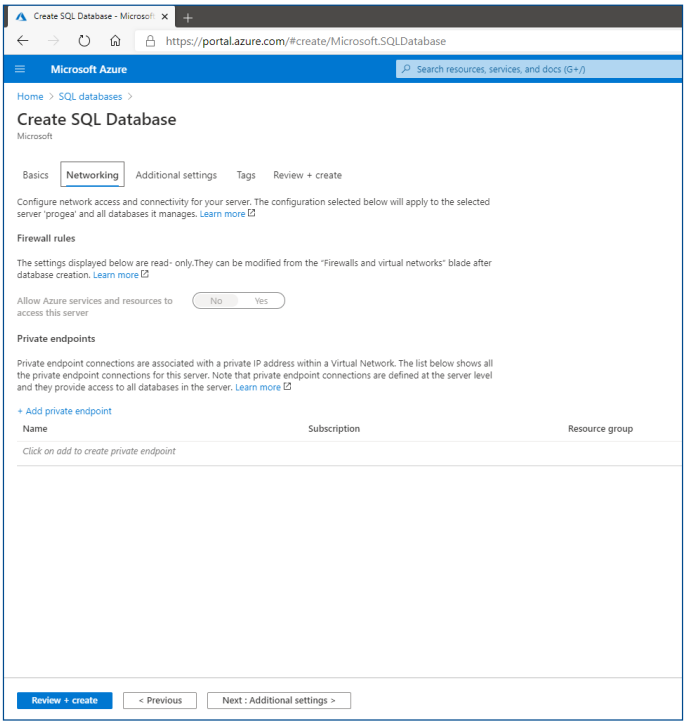
In this step you must also define where to place the Database resource graphically and its size in GBs.



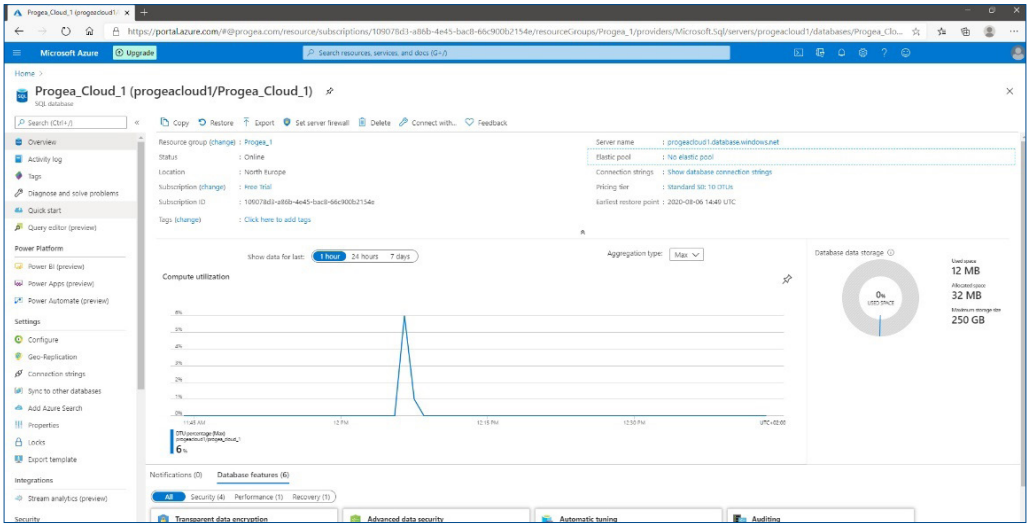
The next step allows you to configure Firewall rules and any endpoints defined within a network infrastructure.

NOTE: The public IP address used to access the Azure portal during creation will be added to the firewall rules by default. Should data recording or database access occur from different public IP addresses, you will also need to define their relative access rules as well.

After having confirmed the creation with the appropriate button, it will take a few minutes to deploy the resource before it



displays and is made available in the list of the resources within the selected subscription. Clicking on the resource will give you access to its summary page.



You can install Microsoft Visual Studio (available for download at the "Connect with…" link in the Azure portal), or Microsoft SQL Server Management Studio to visualize tables and data to be saved in the database.

Movicon.NExT and Microsoft Azure

Running Movicon as an Application in the Cloud

Movicon can be run as a Cloud application. In this case, the project is run in the Cloud without physically needing any local hardware machine. This type of use could be envisaged, for example, when wishing to run a SCADA/HMI application in the cloud by carefully evaluating the connectivity of devices that can be reached on a network with the most appropriate communication protocol.

To use this type of solution, you will need to obtain an Azure account that allows you to configure and use a “virtual machine” on which Movicon.NExT is to be installed.

Movicon in the Field to Record Data in the Cloud

Movicon can be run as a local application (on-premises), as is usually done for SCADA/HMI applications connected to the field. The difference in this case, is that the project does not record data locally but in the Cloud using a data repository that is neither local nor physically present but accessible to all despite this.

This type of solution is ideal for those users who need to record data from different sites and installations, by centralizing them in a repository to allow global data analysis and access from anywhere.

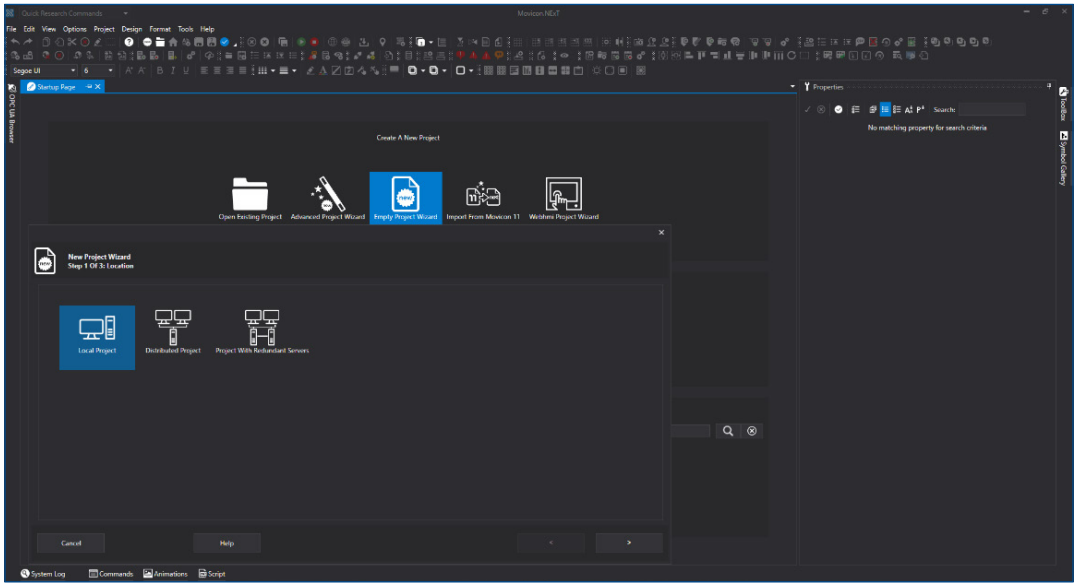
To use this type of solution, you will need an Azure account that allows you to use the SQL Azure functions for recording data in the Cloud. We will see how this is done in this document.

The next section will continue by giving an example of how to create a locally run Movicon.NExT project to record data in Microsoft Azure.

Creating a Movicon Project

The following instructions assume that the user is already familiar with creating Movicon.NExT projects as described in the examples and tutorials provided.

Create a new Movicon.NExT project using the normal procedures using the “Startup Page” as shown below, for example.



Continue using the normal procedures to create your desired project by adding Tags and associating them to the communication driver according to what and how the data is to be used for.

We will not go into detail on how to create a Movicon project in this document as this can be done by referring to the user guide documents, examples and tutorials dedicated to this.

You should create the project to perform the normal I/O Data Server configuration procedures for field connectivity, and with local screens to visualize data.

Data collected by the project can be recorded locally with the Historian or Data Logger recording engines, as set to do so for default.

The next section will continue by showing you how to set the Historian or Data Logger properties so that data are recorded in the Azure Cloud instead of locally.

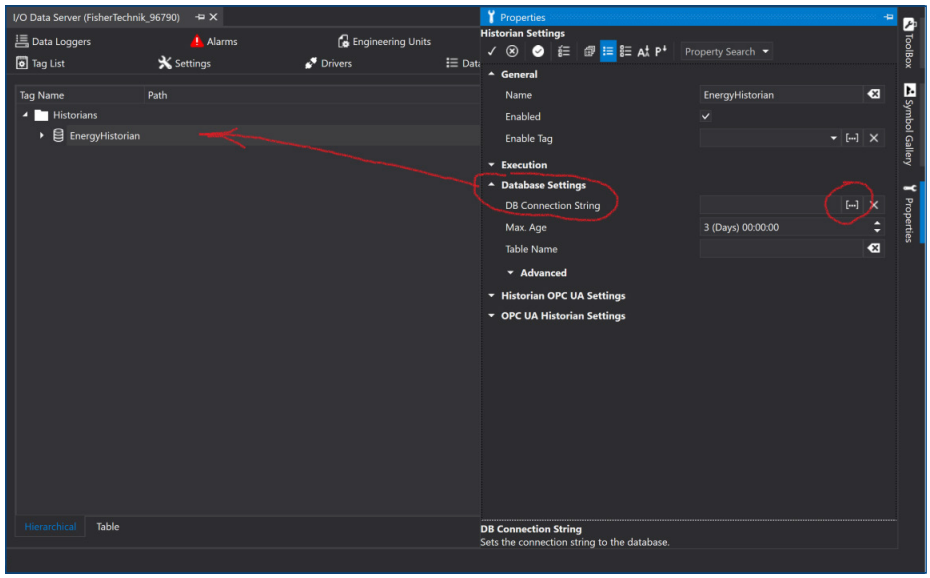
Configuring Historian or Data Logger Properties

For a Movicon project to record data in the Cloud, you will need to configure the recording properties of either the project's Historian or Data Logger objects in the I/O Data Server section.

The Default configuration has been set to record data locally in SQL Server tables. In our case, we shall configure the property relating to the Database connection by setting it according to your needs. Movicon uses three database connections:

- A default database connection for recording variable data (Historian default connection)
- A database connection for recording events (Event Log default connection)
- A database connection for Audit Trial information (Default Audit Trail connection).

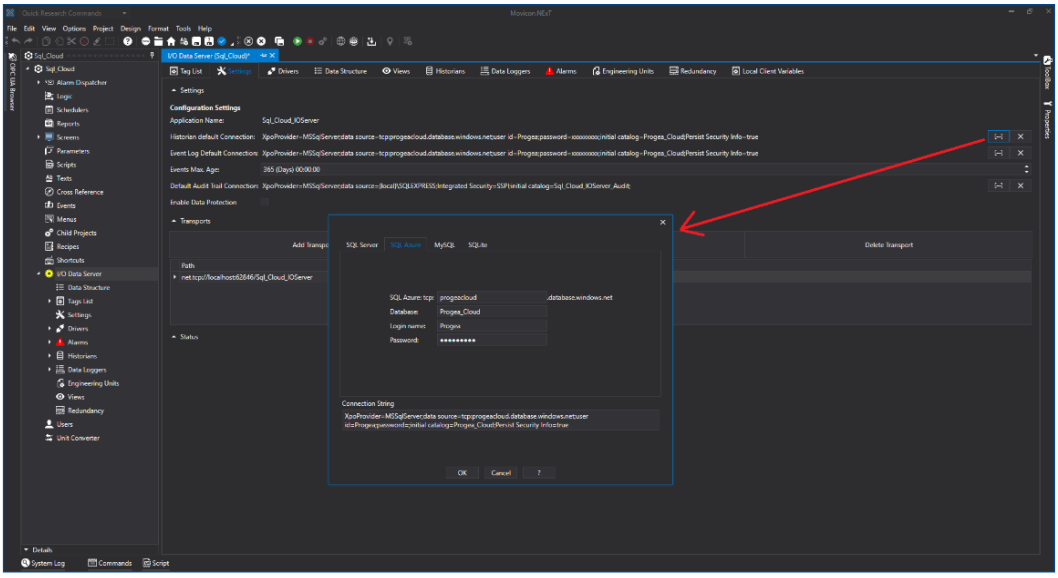
The database connections for recording variable data can also be set individually in the Historian or Datalogger objects. In this way, for example, it will be possible to record parts of the data in the PC locally and parts in the cloud, or both in the PC locally and in the cloud by duplicating the recording objects.



To configure the recording in the SQL Azure database, you will need to insert the SQL Azure server's name, database name, user, and password parameters, which were previously configured in the Azure portal, in the connection strings using the appropriate settings window.

Example:

- Server name: progeacloud.database.windows.net
- Database: Progea_Cloud
- Login name: Progea
- Password: xxxxxxxxx



When everything has been configured correctly, confirm with the "OK" button to save the configuration.

We have just seen how to set the default connections of the part concerning data collected by Movicon.NExT with a SQL Azure database. The concept to keep a local database connection for default and subsequently configure which data we want to have available in the Azure SQL database (see the part relating to database connections in the chapters dedicated to the Historian and Data Loggers) remains valid.

- Please note: For intrinsic security reasons, the database connection for the Audit Trail information (Default Audit Trail connection) cannot be set to use SQL Azure but must be configured towards a database which can be accessed by the user, who was created in Windows with the Movicon setup, using the integrated authentication process.

Configuring Tags

Recording data obviously requires tags to be configured and connected to field devices using Communication Drivers or OPC UA.

We will now go to the I/O Data Server resource to create and configure tags appropriately according to your project needs and the project development procedures defined in the programming manual.

Configuring Recording Engines

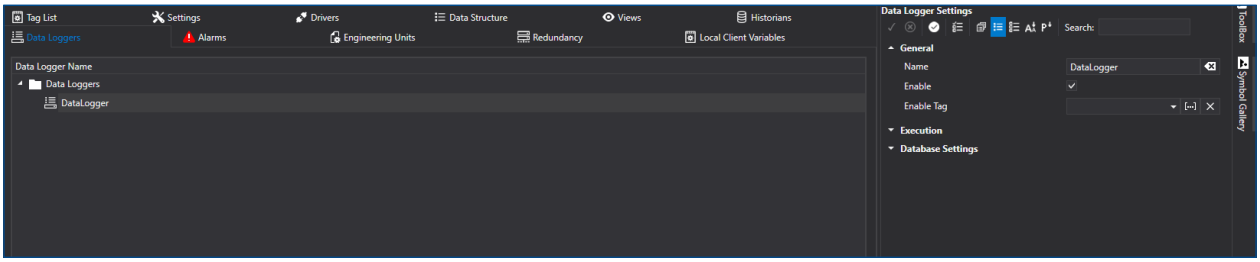
We will continue with configuring the project by creating and configuring data recording engines. The Movicon project can record data collected from field devices, which have been associated to tags, in a target database by using the Historian or Data Logger resources. As previously indicated, in our case, the target database in which data will be recorded is the database in the Cloud Microsoft SQL Azure.

The choice of using the Historian or Data Loggers depends on your project needs. Both these resources record data but each one has a different way of doing so and this largely depends on which choice the designer has made as described in the programming manual.

In our example, we will use both the Data Loggers and Historian.

Configuring Data Loggers

We will now access the I/O Data Server resource and select the “Data Loggers” tab. Once this tab has opened, right click on the “Data Loggers” folder to add a new Data Logger to the list.

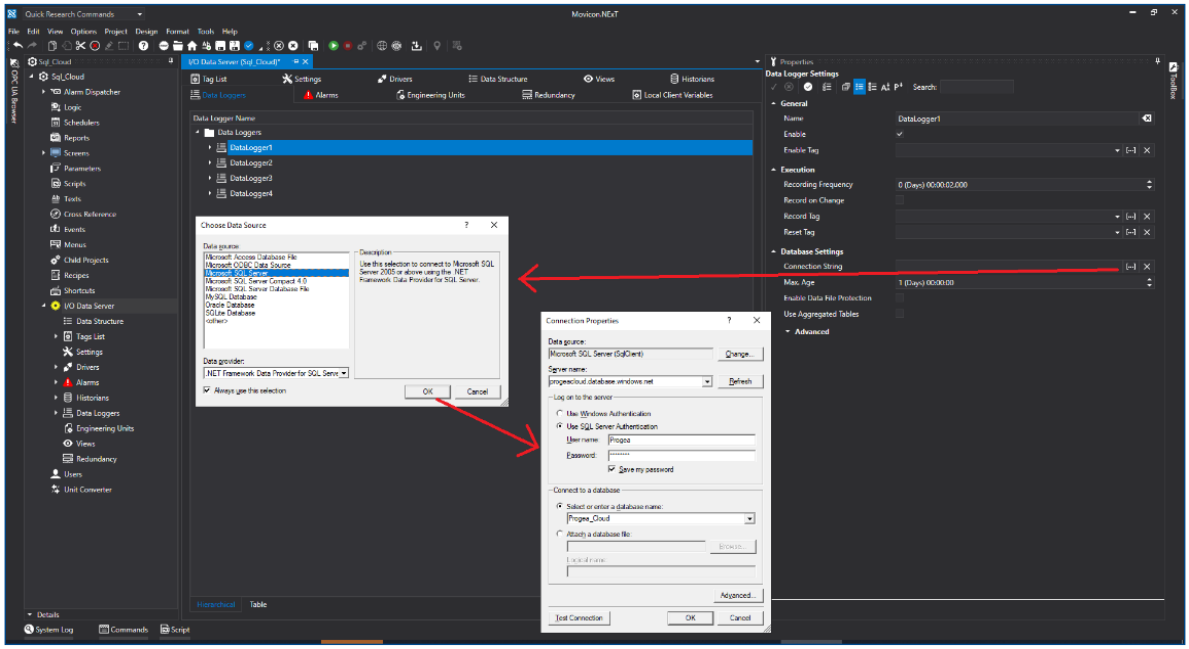


In the Property settings of each Data Logger object, we will define the recording times of each one of the four Data Loggers, which will be 2 seconds for Datalogger1, 5 seconds for Datalogger2, 1 minute for Datalogger3 and 5 minutes for Datalogger 4 respectively.

Even though we have linked only one tag to each Data Logger, you can link as many tags as needed according to the maximum number of tags contained in a table in the database structure you are using.

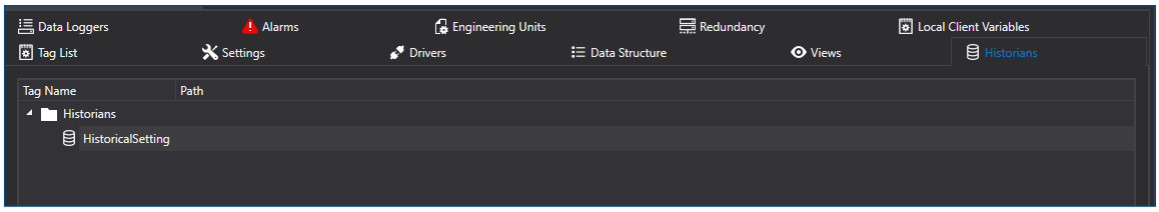
As indicated in the section relating to the default database configurations, unless specified otherwise, the Data Loggers use the default configuration for connections to the database. If desired, it is also possible to customize the connection string to the database for each of the configured Data Loggers. When this is the case, Movicon.NExT will not use the string defined for default in the I/O Data Server side but will use the one set in the “Connection String” field under the “Database Setting” item.

In this case, as in the default database configuration, to set the recording in SQL Azure, you will need to select the “Microsoft SQL Server” database type and insert the appropriate parameters in the configuration mask.

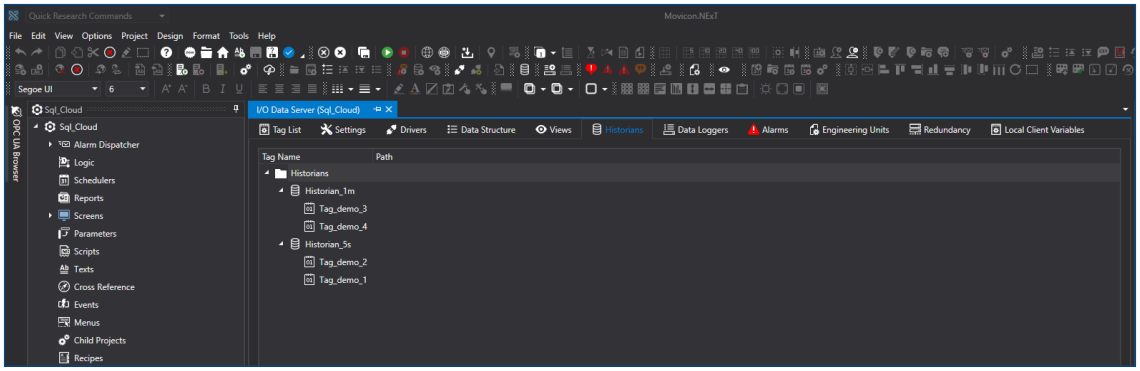


Configuring Historians

After having created the Data Logger resources, remain in the I/O Data Server side, and go to the “Historians” tab to add the Historian resources to allow data to be stored in the SQL Azure database. As previously seen for the Data Logger resources, right click on the “Historian” folder, and add a new Historian to the list.

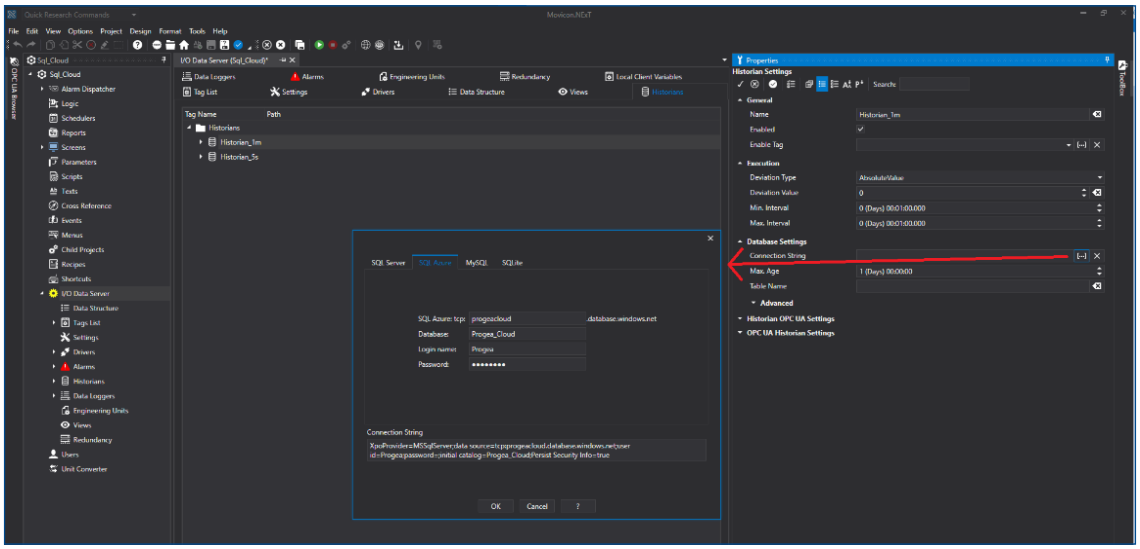


We will now create two different Historian resources. We shall call the first one “Historian_1m” which will be given a 1-minute data acquisition time and will contain data deriving from the following tags: Tag_demo_3 and Tag_demo_4. We shall call the second one “Historian_5m” which will be given a data acquisition time of five minutes and will contain data deriving from the following tags: Tag_demo_1 and Tag_demo_2.



Synonymous with the Data Loggers, the Historians use the default Historian connection, unless the database connection string has been customized in each resource.

To customize the connection string, use the button in “Connection String” property to access its settings page and insert the appropriate SQL Azure access parameters.



Summary:
To recap, we have just finished creating the back end of the project by means of using the I/O Data Server.
We did this in three steps by:

1. Customizing the default database connection configuration with the one we wish to use. We chose to use SQL Azure.
2. Creating Tags and their connections to field devices using Communication Drivers.
3. Creating Data Logger and Historian recording engines to record the data available in Tags in the desired target database.

Now we will see how to display these tags locally using HMI Screen that make up the front end (Client) of the project.

Local Visualization Using Screens

After having configured the Server side correctly, we can now use the project's Screen resources to create screens to graphically display both real-time data in Tags and data stored in the database used, which in our case is the SQL Azure.

Please refer to the programming manual for all further information on graphic editing techniques for a project's HMI Client side.

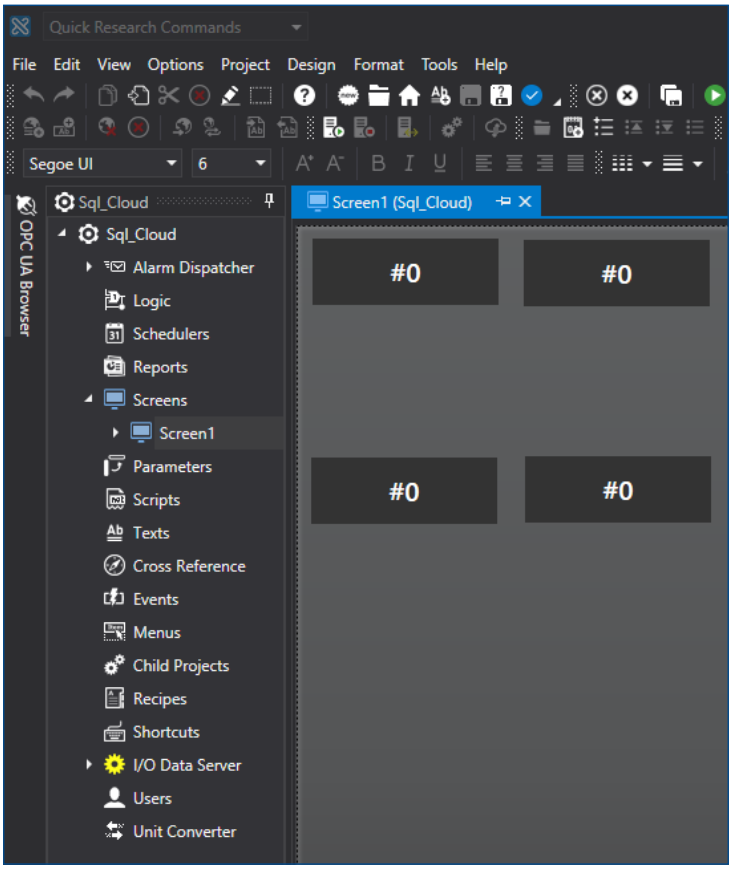
Creating Graphical Objects

After having created one or more Screen resources in the project, we will now begin inserting graphical objects from the Toolbox on the screens to create the project's graphical data representations.

The Display object is the most common one to use for displaying real-time Tag values and it can be withdrawn from the toolbox and inserted on screens.

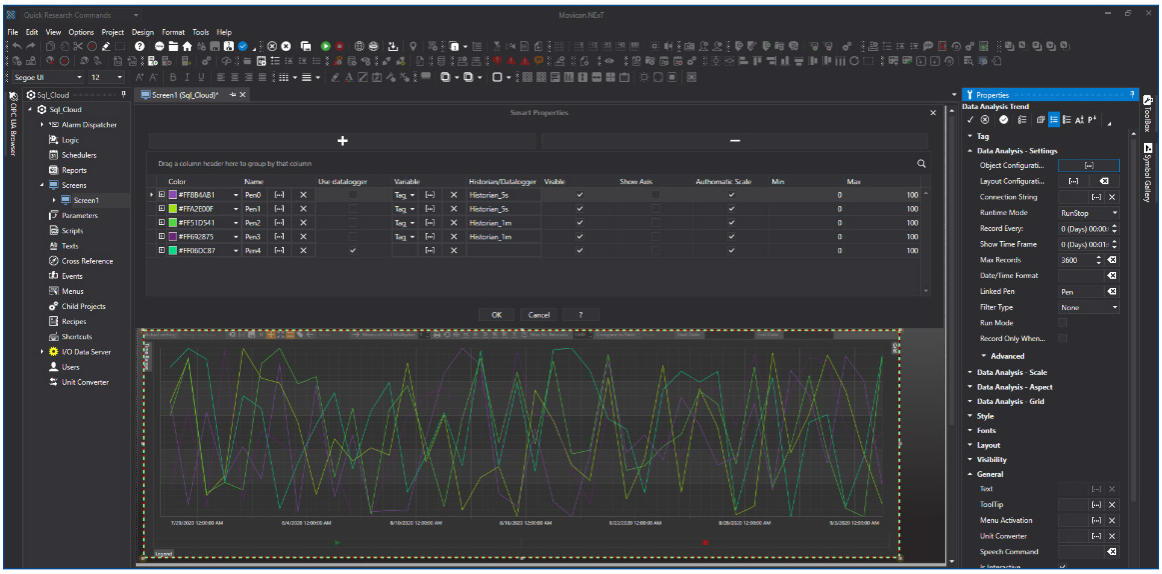
In our example, we will insert as many Display objects as there are tags and associate the tags desired in each object's properties as well as change their styles and colors as pleased.

At Runtime the Display objects will display the associated tag's value connected to the field device.



It is also possible to use purpose-built objects for displaying historical data contained in the database connected to the project. The Toolbox contains many objects for this purpose, such as the Trends, Charts, Tables and Reports.

We could, in this case, insert one or more Trend Data analysis objects to display graphical representations of recorded values. This can be done by simply associating a Data Logger or Historian object in the property settings of these objects.



Thanks to the Movicon screens, we can easily create a local graphical interface to view the project's real-time data or historical data, regardless of which database has been configured in the project and the historical data's physical location.

Data Visualization Using Cloud

Displaying data recorded by Movicon.NExT and saved in the Azure SQL database can be done in various ways. For example, they can be displayed by creating a website that hosts a web-based solution that interfaces with data residing in the Azure SQL database. Even though this solution requires a mixture of skills to develop and support it, it does offer multiple advanced customization solutions. In this document, Emerson has chosen to use the Microsoft Power BI visualization solution.

Microsoft Power BI Configuration

To use the Power BI solution, you will need to sign up for an account for business or educational purposes at this link **https://powerbi.microsoft.com/en-us/landing/signin/**. We recommend that you access the portal with Microsoft Edge. The procedure to create a business or school organization account is not described in this document.

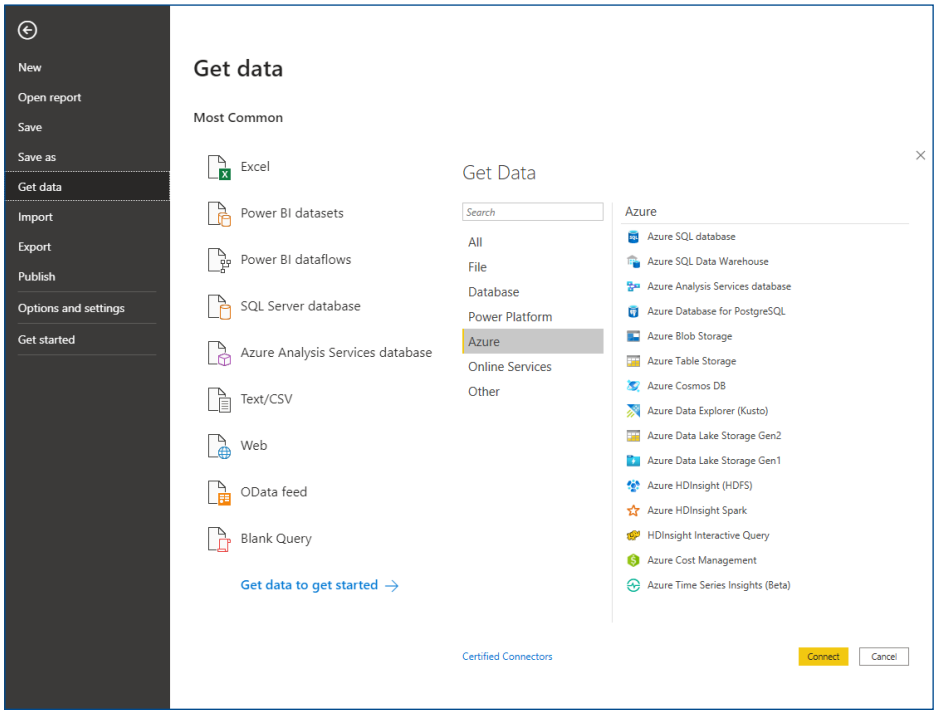
In addition to web access, the Power BI solution also requires the desktop application to be installed which we will use to create the contents to then be published to the web side.

The Power BI desktop installation can be obtained at: **https://powerbi.microsoft.com/en-us/desktop/** .

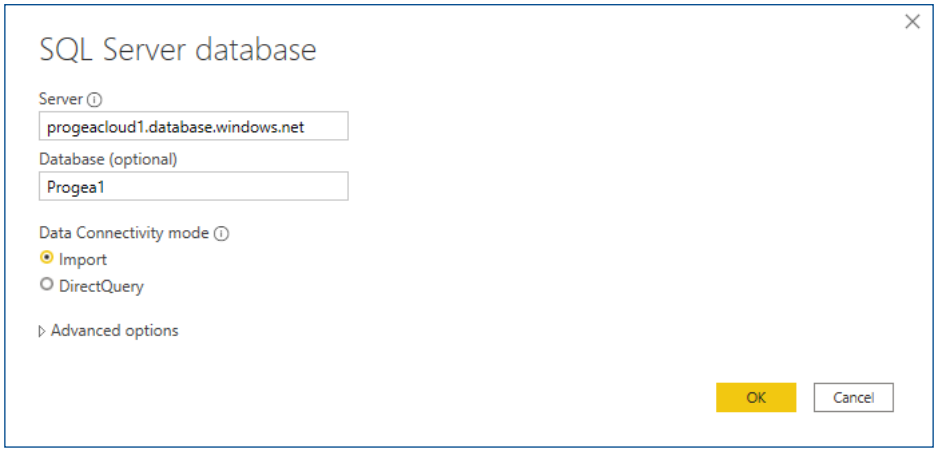
Microsoft Power BI Data Source Configuration

The first thing to do after creating a new project, is to connect the data source that we will use in our Power BI project.

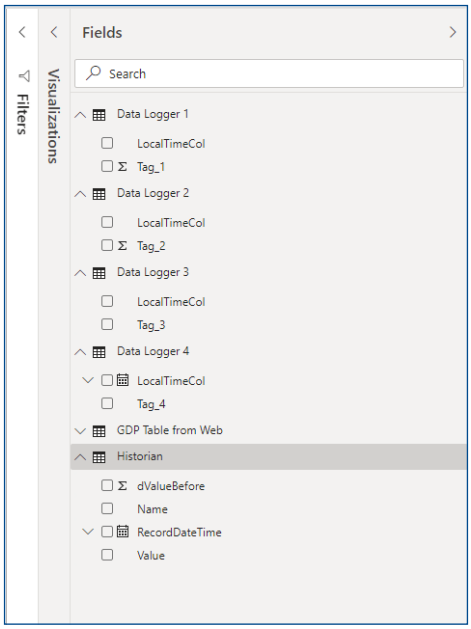
By accessing the Power BI “File” menu you will be able to select which data source to use. In our case, we will first use the data source from the Database instance configured in Azure (see the related document chapter).



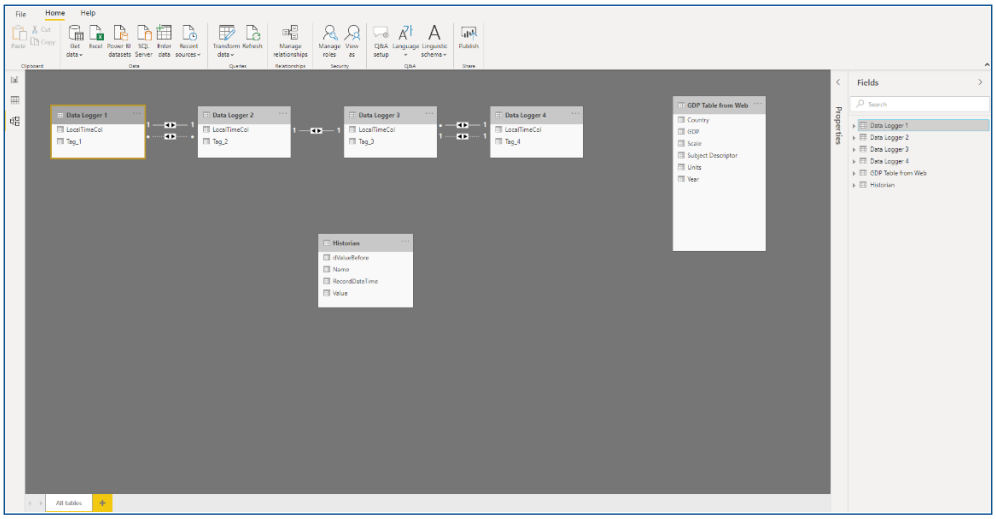
Once the connection has been selected by means of using the “Azure SQL database” object, a wizard will open to create and set the connection between Power BI and the database.



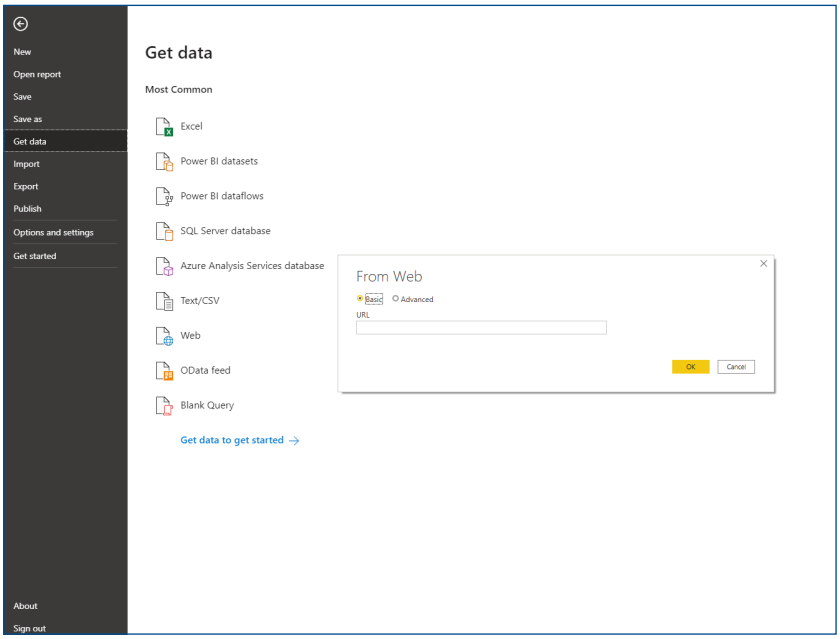
When pressing the “Ok” button, Power BI will load the database’s contents allowing us to scale them using the tools available in Power BI. In addition to scaling the available table fields, it is also possible to create a “relationship” between the different fields inserted in the tables. This will enable calculations to be performed automatically between data from different tables in graphical format.



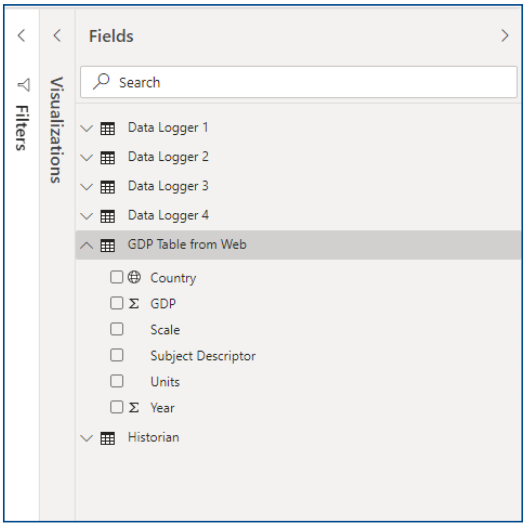
The screen shot shows an example of “relationships” applied to the different Data Logger tables which are contained within the database used in the Movicon.NExT project example.



The second data source, which we are going to connect to our Power BI, comes from a web link. In our example we will load data from a database which is available from the world economic forum website at **https://www.imf.org/external/pubs/ft/weo/2020/01/weodata/index.aspx.** Once accessed, we will then select the data we wish to display. In our example, we will select all the world nations and we will use “Gross domestic product, current prices” as our data reference. By accessing the Power BI “File” menu, we will be able to select which data source to use. In our specific case, we will select “Web” as our data source.



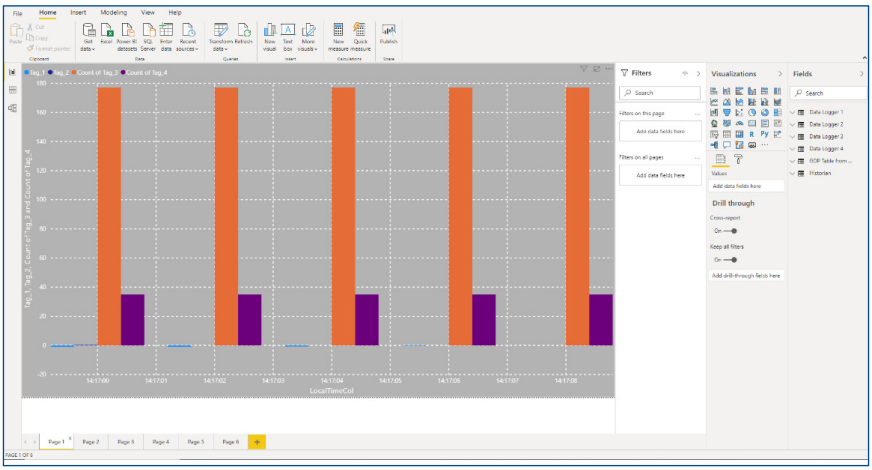
Once the reference URL has been entered, press the OK button to make the new data source available in BI.
As done previously, we can also scale the data using the tools provided by Power BI.
In our specific case, we will rename the data source as “GDP Table from Web” and then use the following fields: “Country,” “Scale,” “Subject Descriptor,” “Units,” “Year,” “GDP.”



Creating Pages and Objects

Once the procedures for accessing data from different sources have been completed, we can then start creating project screen pages and their object contents. Our project will be composed of six pages in total.

On the first page, we will insert a Chart object to visualize data from the different Data Loggers which we configured in Movicon.NExT beforehand and stored in the SQL database configured in the Azure Cloud.



The Chart object's visualization mode and graphical customization can be done using the "Format" menu available in the graphical object's control panel. Note that the previously configured relationship between the DataLogger 3 and Data Logger 4 Tag fields can be seen on the left side of the Chart.

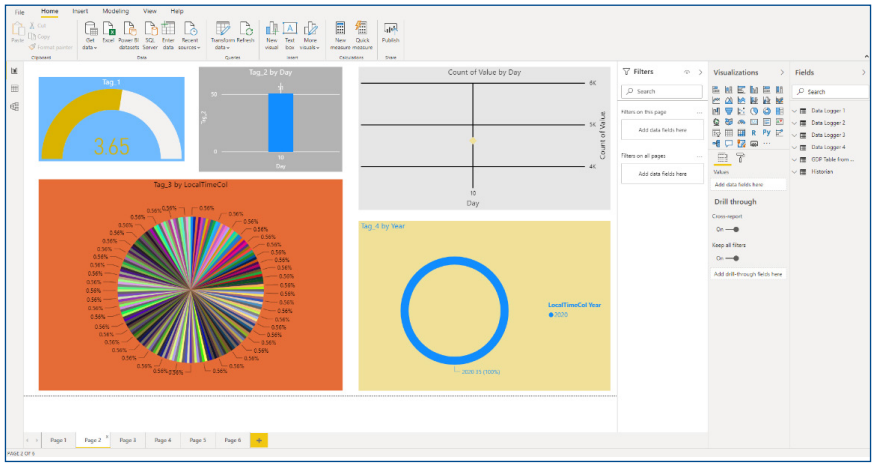
On the second page, we will insert different visuals, which will allow the various data contained within the Data Loggers and Historians in the Movicon.NExT project to be graphically displayed differently.

We will place the following objects in different positions on the page:

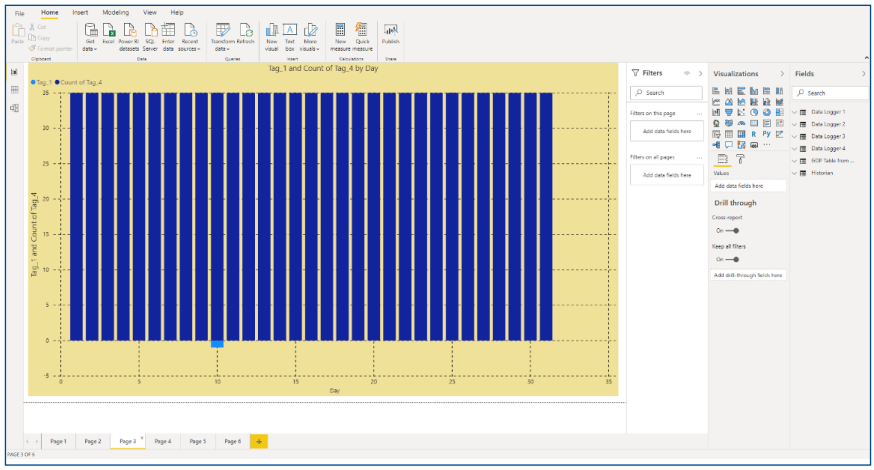
Example:

- “Indicators/Gauge Chart” to display Tag 1 data.
- “Column Chart” to display Tag 2 data.
- “Two Variable Chart” to display historical data based on the different available time frames.
- “Pie Chart” to display the contents of the “Data Logger 3” table.
- “Doughnut Chart” to display the contents of the “Data Logger 4” table.

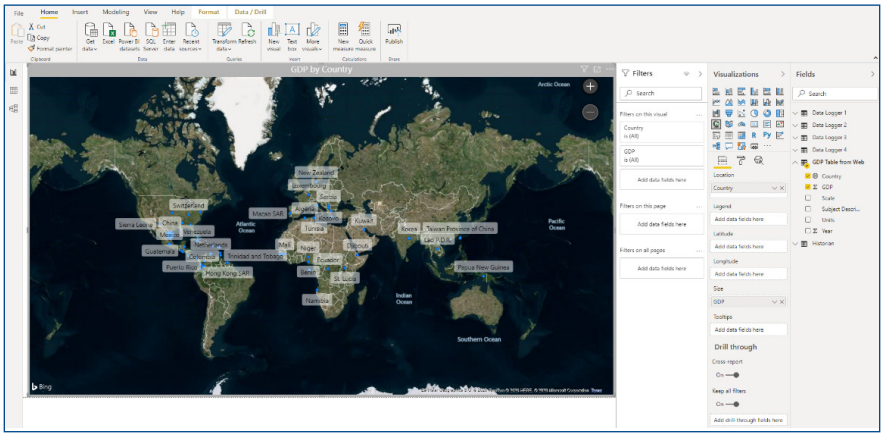
The “Format” menu is also available to customize the aspect and behavior of each of these objects.



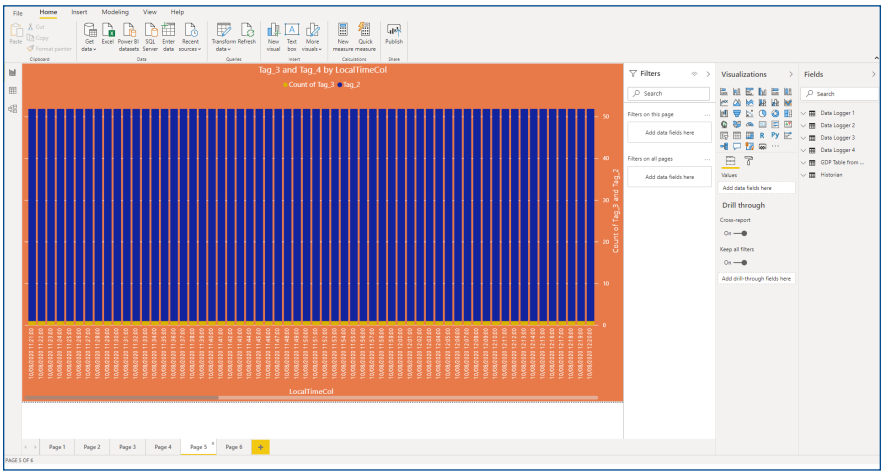
On the third page, we will insert a “Stack Histogram” visual, to display data from the “Data logger 1” and “Data logger 4” tables.



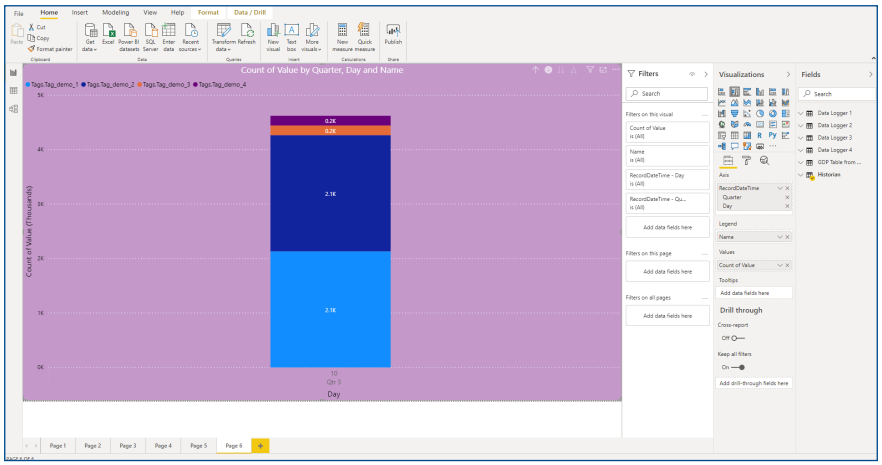
On the fourth page, we will insert the “Map” control to display data from the table configured as the web link.



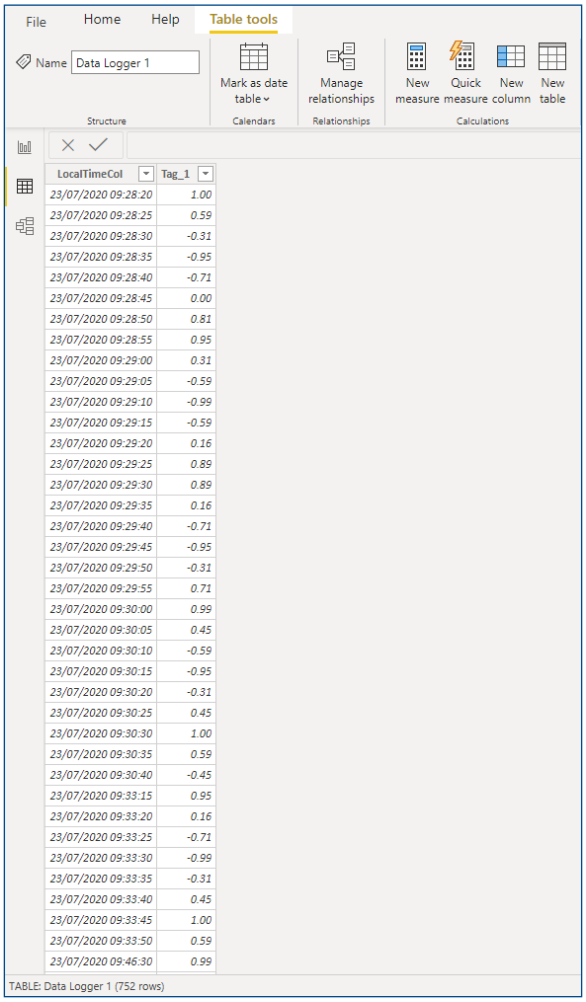
On the fifth page, we will insert a “Stack Histogram” visual, to display data into a last time zone from “Tag 3” and “Tag 4”



On the sixth and last page, we will insert another “Stack Histogram” control to display data from the Movicon.NExT project’s “Historian” along with two drill-through buttons to quickly navigate through detail within the inserted control.

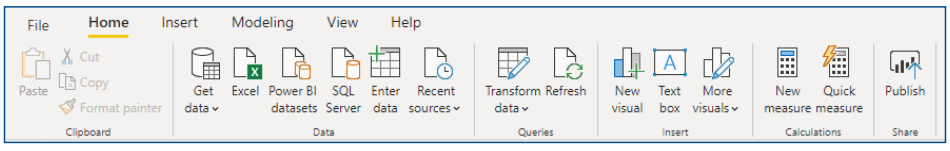


A table view will always be available for each visual control inserted on the page. This view can be accessed by selecting the icon available on the left of the Workspace.



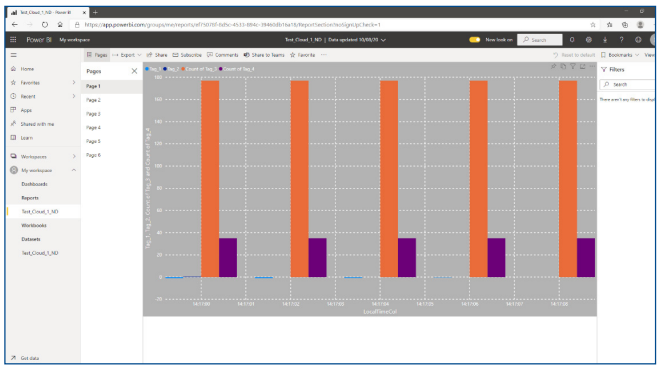
Publishing the Power BI project to the Web

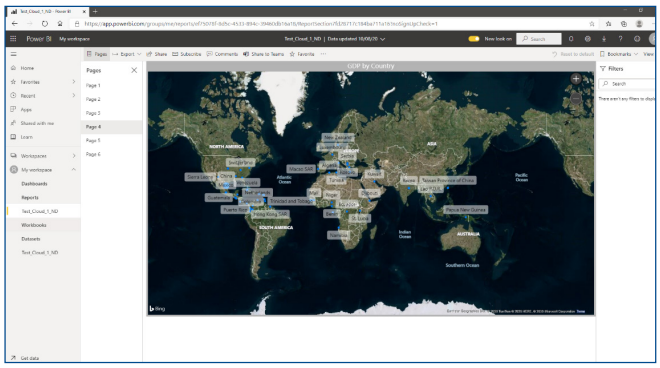
After having created the various project pages with the relative visual controls, we can now proceed with publishing the project we have created on the web.

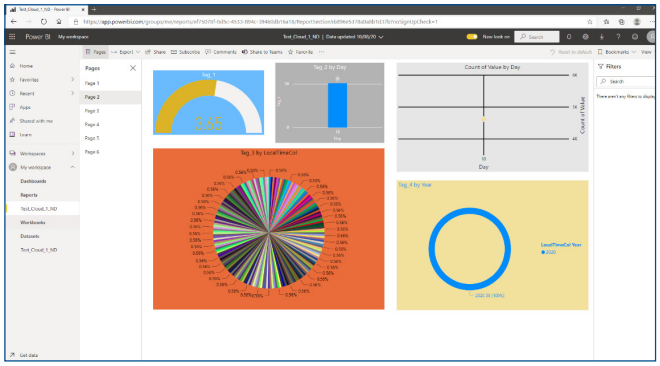


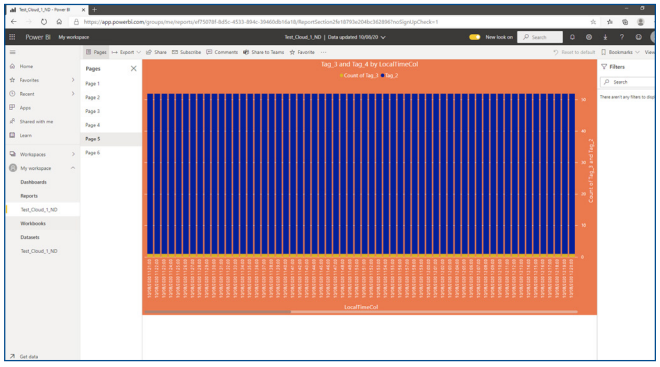
The project will be available for use by connecting to the chosen Workspace in a few seconds.

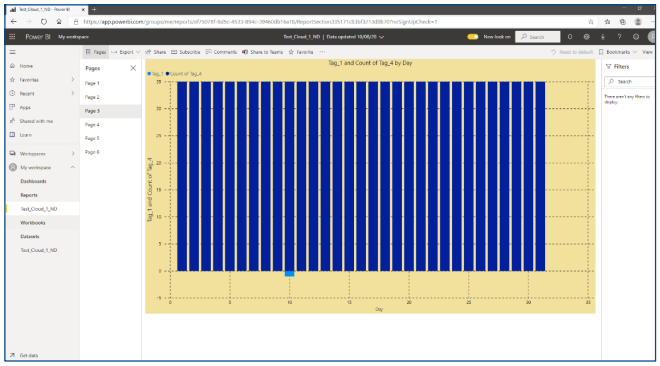
As you can see below, all the pages we created beforehand, and their relative data are displayed in the Workspace.

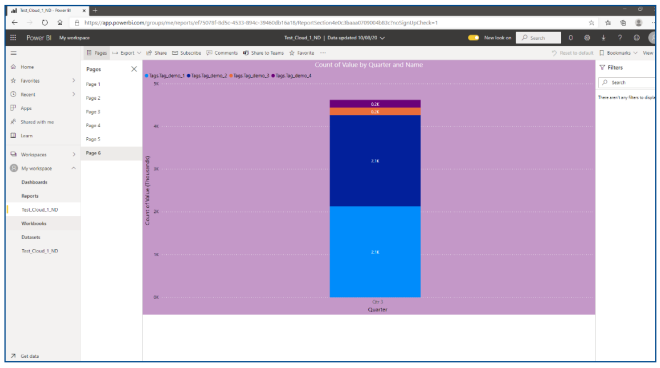












At this point, we can start using the various built-in features provided by the Power BI solution to create custom filters in each dashboard, save what is being visualized in pdf format, share the dashboards with other colleagues using the Team application or visualize data in local dashboards in PDF format.

Power BI Project Sharing

You must have a Power BI Pro license to be able to share and visualize the dashboards you create. This will make access to them available in different ways.

Workspace Collaboration

Different Workspaces can be created in Power BI and shared with a team. In this way, the properties and dashboard management, reports, data sets and work folders can all be shared together with members of the team.

The Workspaces offer roles to which certain authorizations can be associated. The use of these authorizations and roles will determine who can manage the Workspace, modify, or distribute its contents or simply visualize them.

To collaborate with team members, the Workspaces are preferable to having a My Workspace as their content can be co-owned. The user and the whole team can easily update or allow access to other users. The My Workspace is more suitable for the individual user for occasional or personal content.

When should you allow access to the dashboards? The answer to this depends on many situations.

- When colleagues need to keep the dashboard updated or access its content in the workspace, you can add them to the Workspace as members or collaborators.
- When colleagues only need to visualize the Workspace content, you can add them as viewers.
- When colleagues simply need to visualize the dashboard and not all the Workspace's content, you can share the dashboard directly with them.
- When the dashboard is part of a set of dashboards and reports which need to be distributed among many colleagues, the best thing you can do is publish an app.

Collaborate with Microsoft Teams

You can foster data-driven collaboration across your organization by implementing the Power BI reports and the Power BI paginated reports in Microsoft Teams. The Power BI service has a button for sharing reports in Teams. You can add separate Power BI tabs for each individual report and assign each tab the name of the report or any other name.

When you add a tab for a Power BI report to Microsoft Teams, Teams automatically creates a chat on the tab for the report. This will enable all users of that Microsoft Teams channel to view and discuss the report with each other in the chat.

Sharing Dashboards and Reports

The Power BI server offers another way of sharing dashboards and reports directly. This is the one offered directly by the Power BI service which allows you to publish the report in your My Workspace or in another Workspace.

Power BI Pro license is also needed in this case by those who wish to publish reports in their Workspace and by those who wish to use it. When you share a dashboard or report, the recipients can view and interact with it. You can also give them authorization to edit it, make a copy and share it with their colleagues. They will see the same data that is visible to the owner in the dashboard or in the report, and they will have access to all the datasets of the underlying data.

You can also share with external users outside the organization who will be allowed to display the dashboard or report and interact with it, but not share it with others. These external users will also need to have Power BI Pro license.

Using the Power BI apps to Share

The Power BI apps are available from the appropriate app stores for iOS and Android mobile devices. These apps can be used to visualize reports and dashboards shared in Teams. You can also add notations to a pane, report or visual and then share them with anyone by email.

Snapshots of panes, reports or visuals can also be shared. The recipients will see exactly what was captured when sending the email. The email will also contain a link to the dashboard or report. You will also be able to send snapshots of panes to anyone and not just to colleagues of the same email domain.

Pane snapshots from the Power BI app can also be shared with those using Windows 10 devices but not annotated.

Embedding Reports in Secure Portals or Public Websites

Power BI reports can be embedded in portals or websites where users can visualize them. Reports can be securely embedded for external users using the Embed option in SharePoint Online and the Power BI service's Embed option.

Regardless of which option has been chosen, Power BI enforces all the authorization and security procedures before allowing users to view the content.

Embedding in SharePoint Online functions with the Power BI web-side for SharePoint Online. It offers a Single Sign-On experience with control over how the report is embedded.

The Embed option functions with any portal or website that supports embedded content using a URL or iFrame.

Publishing to Public Websites

You can publish Power BI reports anywhere on the Internet by embedding visualizations in blog posts, websites, social media, and other online communications by using the Publish to Web function.

Anyone on the Internet can view the reports and there is no control over who can view what has been posted. You do not need to have a Power BI license. The web publishing feature is only available for reports that you can edit.

You cannot publish reports to the web when they are being shared with other users or included in an app.

Contact Us
www.emerson.com/contactus