

Quick start guide

Dynamically query for generating report

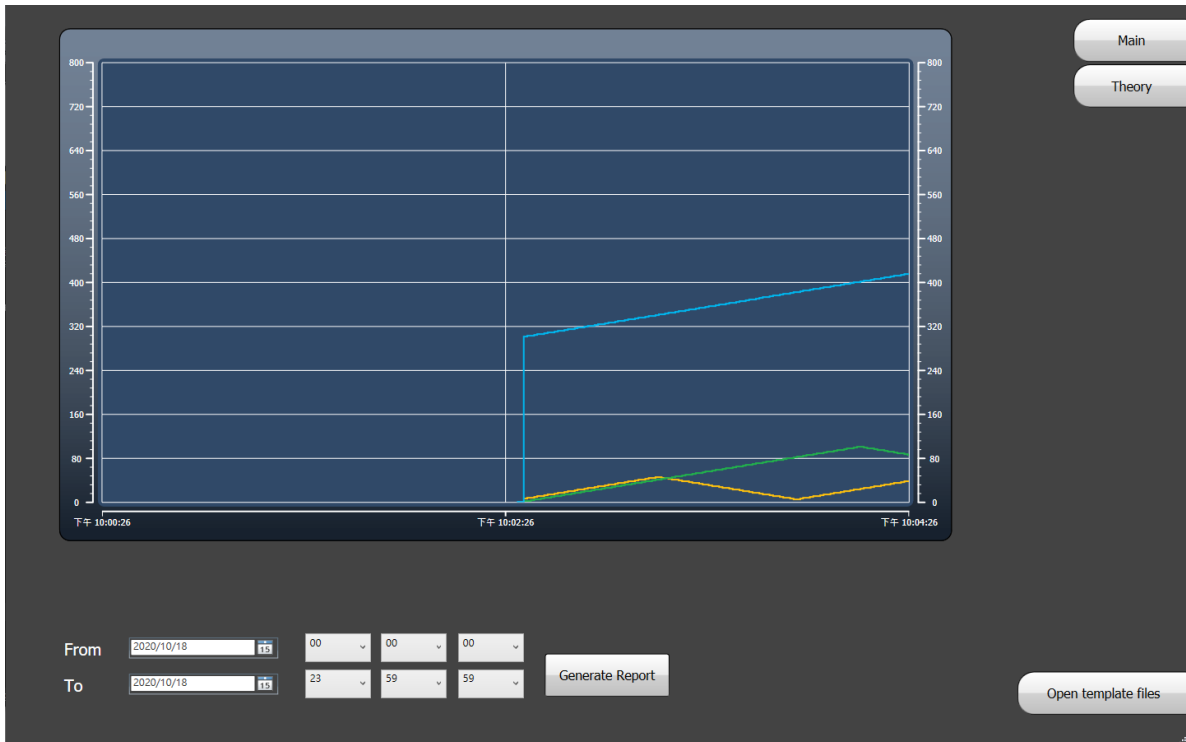
SER0013 - iX sample dynamic report template



1 Function and area of use

This document provides guidelines when working with the report.

This document explains the workaround for changing the database query script (SQL script) in run time. Although iX Developer provides a flexible way for report design in design time, it is not enough for some kinds of applications. The most situation is the end-user wants to select the date range for generating a report in run time, although they can do it in the excel with the generated report. It is annoying that an end-user manipulates data in MS Excel daily.



From	To		
2020-10-18 00:00:00	2020-10-18 23:59:59		
Date	Temperature	Humidity	CO2
2020/10/18			0.00
2020/10/18	6.00	1.00	301.00
2020/10/18	7.00	2.00	302.00
2020/10/18	8.00	3.00	303.00
2020/10/18	9.00	4.00	304.00
2020/10/18	10.00	5.00	305.00
2020/10/18	11.00	6.00	306.00
2020/10/18	12.00	7.00	307.00
2020/10/18	13.00	8.00	308.00
2020/10/18	14.00	9.00	309.00
2020/10/18	15.00	10.00	310.00
2020/10/18	16.00	11.00	311.00
2020/10/18	17.00	12.00	312.00
2020/10/18	18.00	13.00	313.00
2020/10/18	19.00	14.00	314.00
2020/10/18	20.00	15.00	315.00
2020/10/18	21.00	16.00	316.00

2 About this document

This quick start document should not be considered as a complete manual. It is an aid to be able to startup a normal application quickly and easily.

Copyright © Beijer Electronics, 2020

This documentation (below referred to as 'the material') is the property of Beijer Electronics. The holder or user has a non-exclusive right to use the material. The holder is not allowed to distribute the material to anyone outside his/her organization except in cases where the material is part of a system that is supplied by the holder to his/her customer. The material may only be used with products or software supplied by Beijer Electronics. Beijer Electronics assumes no responsibility for any defects in the material, or for any consequences that might arise from the use of the material. It is the responsibility of the holder to ensure that any systems, for whatever applications, which is based on or includes the material (whether in its entirety or in parts), meets the expected properties or functional requirements. Beijer Electronics has no obligation to supply the holder with updated versions.

Use the following hardware, software, drivers and utilities in order to obtain a stable application:

In this document we have used following software and hardware

- iX Developer 2.4 SP4 / SP5
- PC and C2 Series. (There is another sample for X2 series.)

For further information refer to

- ReferenceManual.pdf (Can be found in iX installation folder, the default path is C:\Program Files (x86)\Beijer Electronics AB\iX Developer 2.40 SP4\Bin\Resources\Help\ReferenceManual.pdf)
- [NPOI 2.5.1 - NuGet Gallery](#)
- [NPOI Project Site](#)
- [Getting Started with NPOI](#)
- [Beijer Electronics knowledge database, HelpOnline](#)

This document and other quick start documents can be obtained from our homepage. Please use the address support.europe@beijerelectronics.com for feedback.

3 Table of Contents

- 1 Function and area of use.....2
- 2 About this document.....3
- 3 Table of Contents.....4
- 4 Concept.....5
 - 4.1 *Preparing for the concept*5
- 5 Migration 13
- 6 About Beijer Electronics 14
 - 6.1 *Contact us* 14
 - Global offices and distributors* 14

4 Concept

This workaround is based on the Reports function of iX Developer. A report is generated according to the custom template in MS Excel format. If we want to change a query condition in run time, you just need to change it in the template file before generating a report. In other words, the Reports function always generates a report with a reading template process.

Further, you can also create a whole new template, and the report function will generate a report according to it regardless of the old one that was designed in design time.

4.1 Preparing for the concept

The preparations are as follows:

4.1.1 Download NPOI library and reference it:

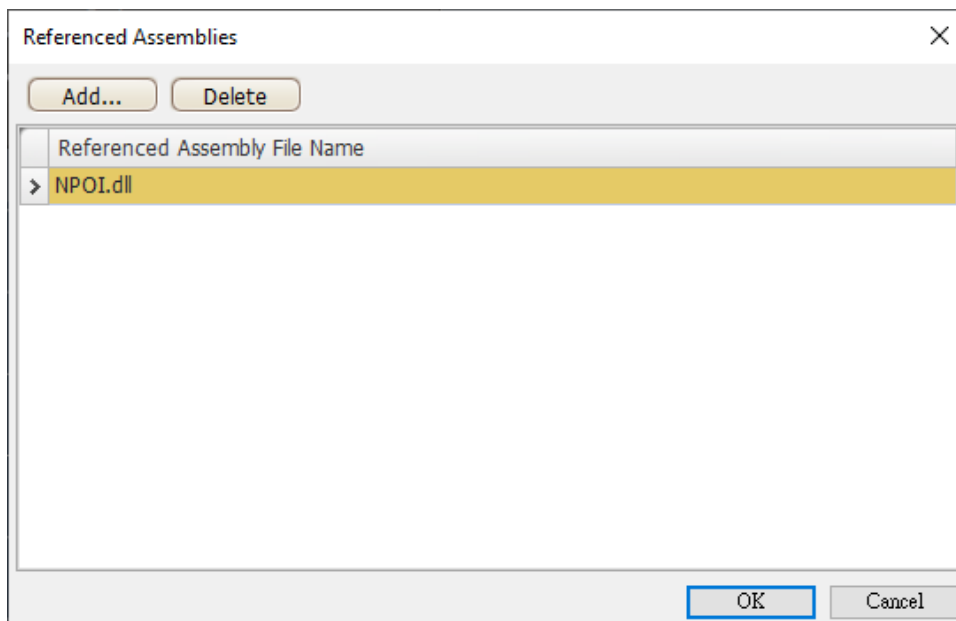
This sample uses the NPOI 2.5.1. You can download the NPOI from the NuGet via Visual Studio, or just copy the DLLs in the ReferencedAssembly folder (*{iX Project Path}\ReferencedAssembly*) from this sample.

For downloading NuGet package via CLI tool, please refer to [next chapter](#).

The DLLs need to be referenced:

NPOI 2.5.1

- NPOI.dll

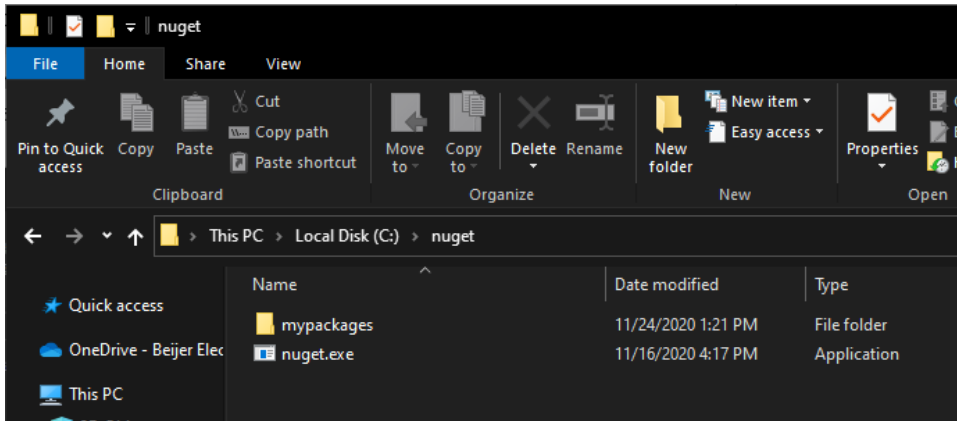


4.1.2 Download NPOI library via NuGet CLI

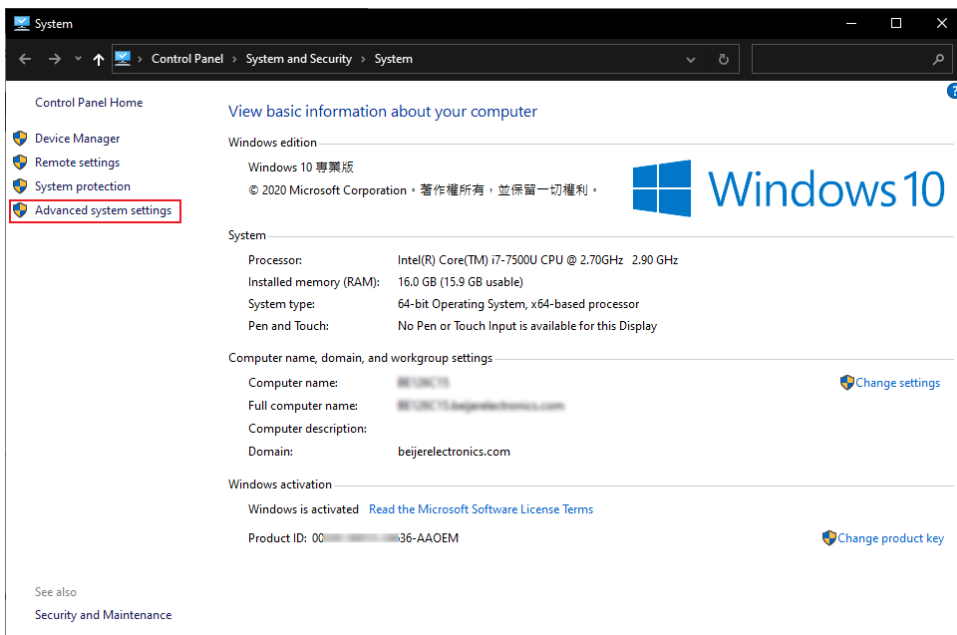
There are several ways to download a package from NuGet. For more information about it, please refer to [“Install NuGet client tools”](#). However, we only want to use the libraries (assemblies) in an iX Runtime application. It is better to get the package via nuget.exe CLI, please refer to [“Manage packages using the nuget.exe CLI”](#).

The article mentions that “Install the nuget.exe CLI by downloading it from [nuget.org](#), saving that .exe file to a suitable folder, and adding that folder to your PATH environment variable.”

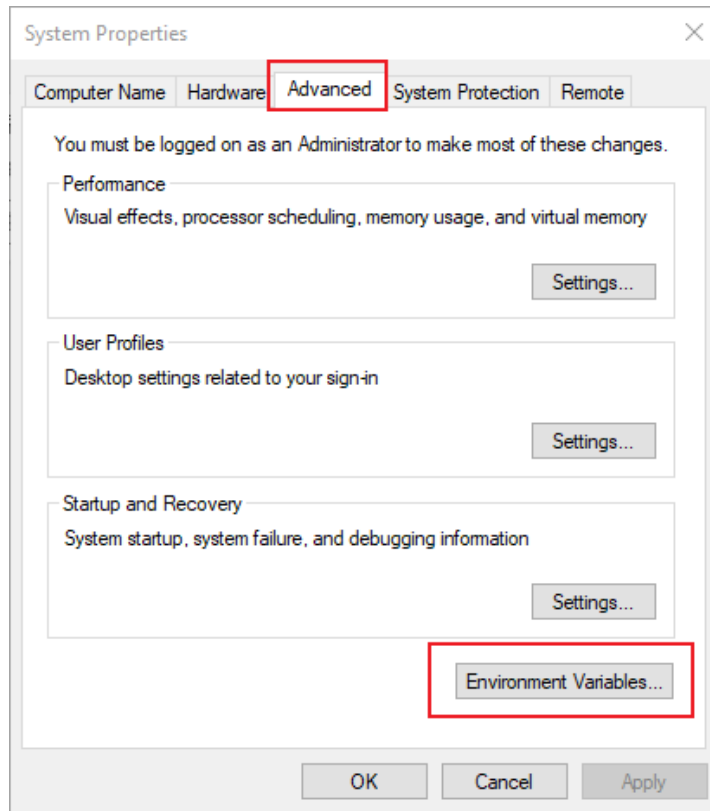
- After downloading nuget.exe CLI, put it to a suitable folder. For example, here we created a folder called nuget in C disk to demonstrate.



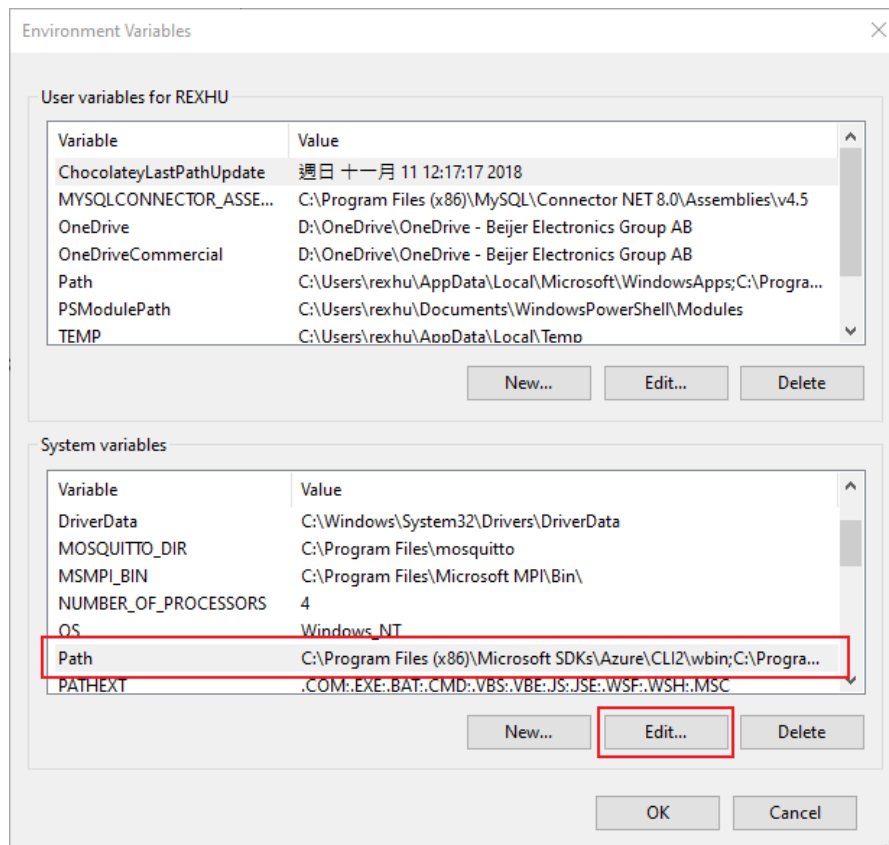
- Adding the folder to PATH environment variable
 1. Click “Advanced system settings”



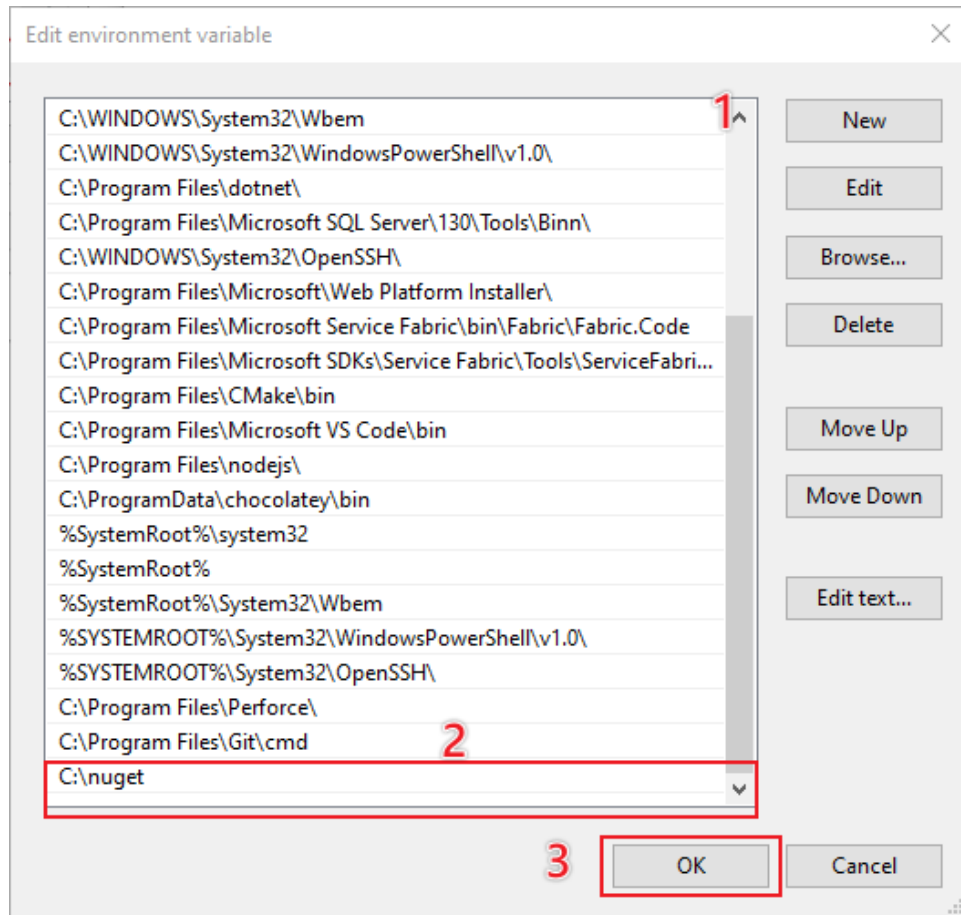
- 2. Click “Environment Variables” on “Advanced” tab of “System Properties” dialog.



- 3. Choose “Path” in “System variables” and click “Edit”.



- 4. Append the folder path to it.



- Downloading the NPOI package

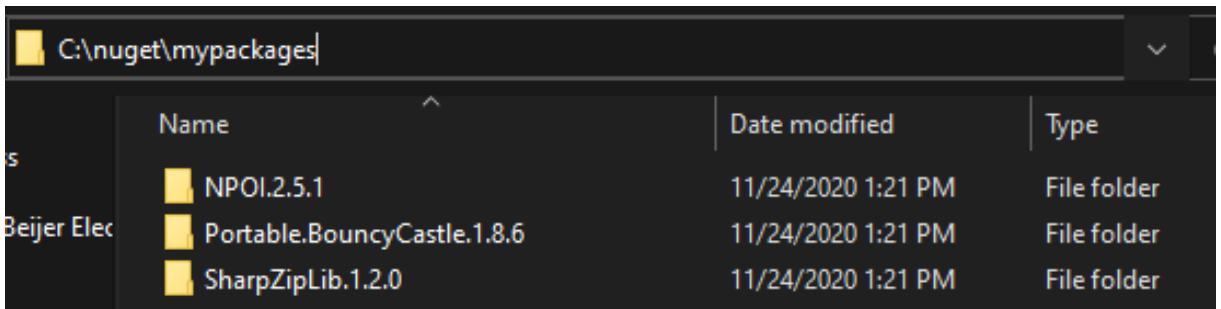
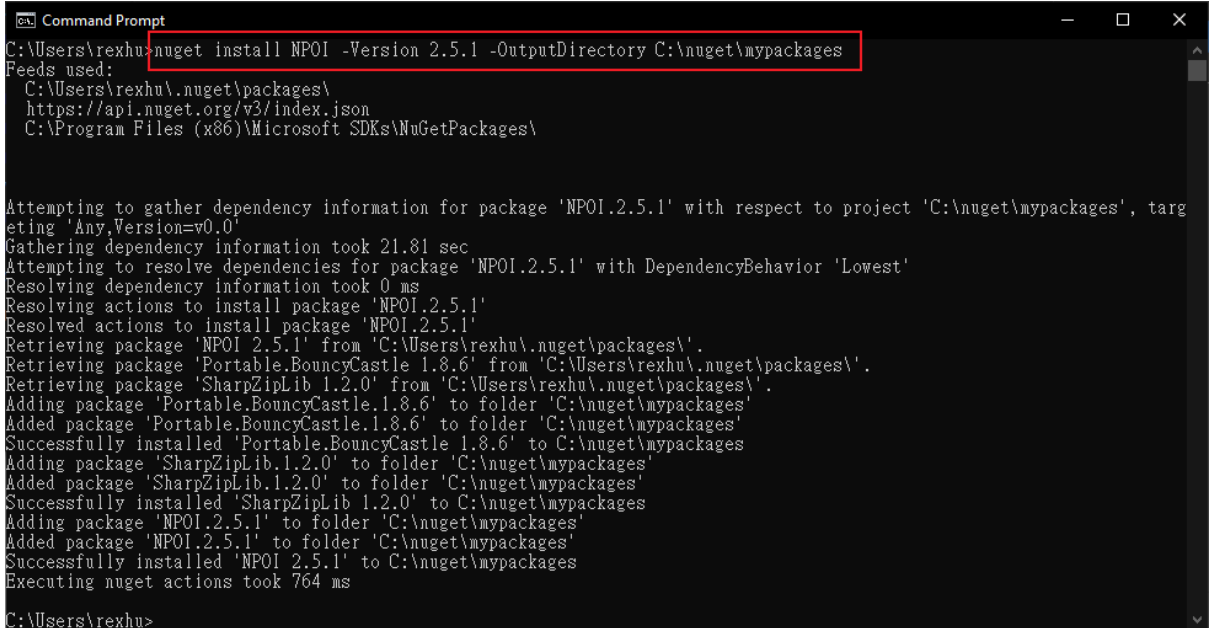
Next step, we can use the “install” command to download the package.

```
nuget install <packageID> -Version <version> -OutputDirectory <path>
```

For example, we created a folder called mypackages under C:\nuget\. And then, we execute the install command.

```
command prompt(cmd)>nuget install NPOI -Version 2.5.1 -OutputDirectory C:\nuget\mypackages
```

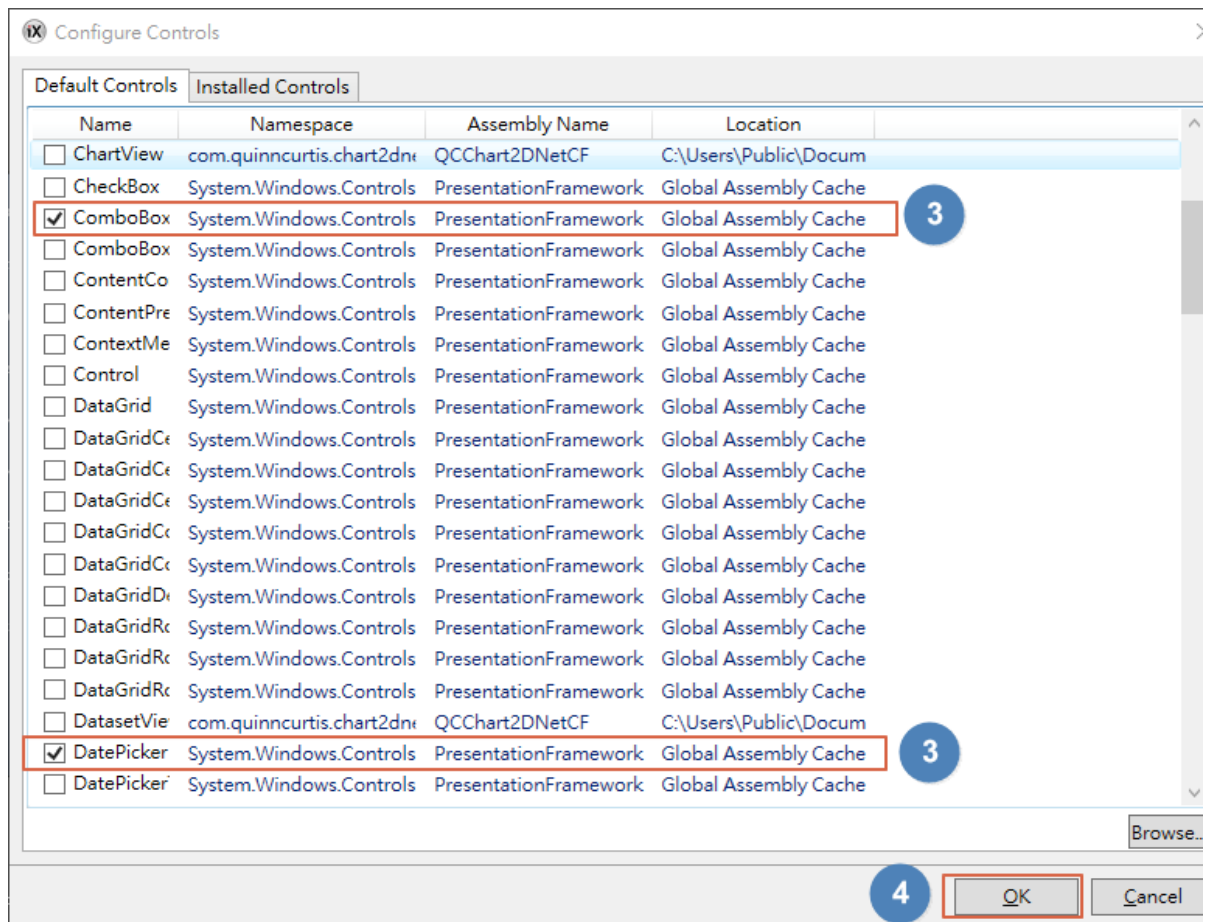
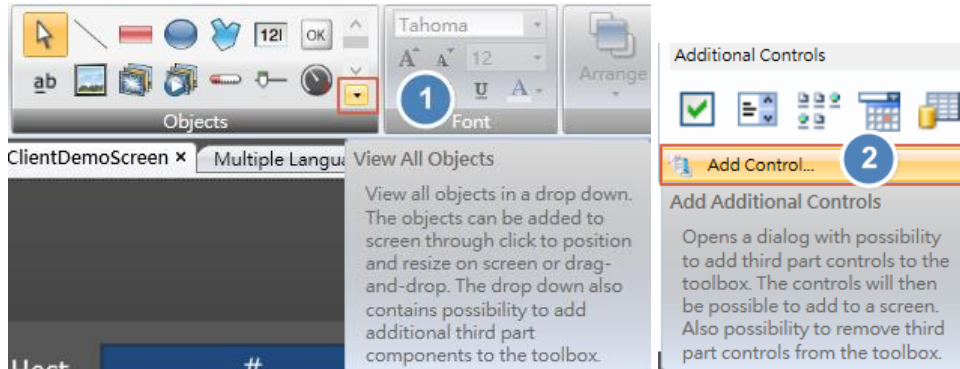
Then, you got the library and dependencies.



4.1.3 Add additional controls:

Reference the WPF DatePicker control and ComboBox control via “Add Control” button.
 The following description shows how to add additional controls.

“Home” → “View All Objects” of “Object” → “Add Control” of “Additional Controls” → check controls as desired → “OK”



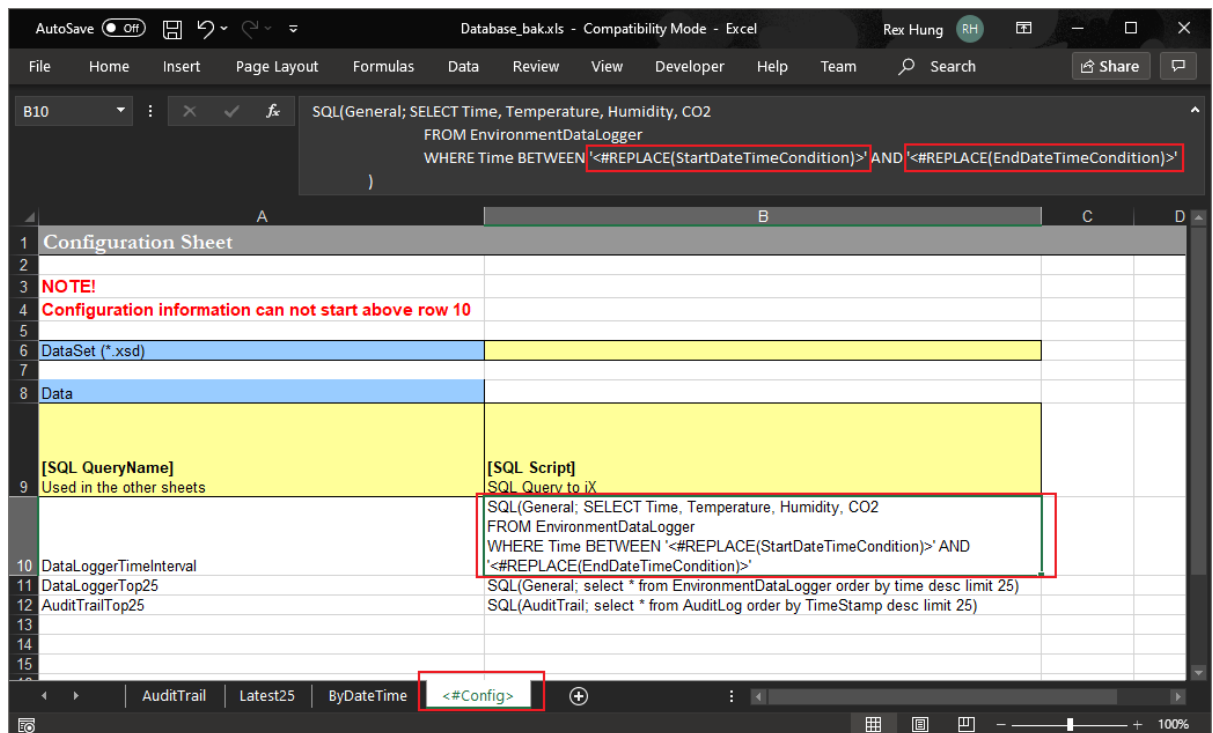
4.1.4 Define a dynamic report template with Reports of the iX Developer as usual:

This is for creating a path of the template that the iX report generating function refers to in run time. Actually, the template file will be replaced when the query script changed. Therefore, it can be a blank excel file if you will change the query script the first time generating a report.

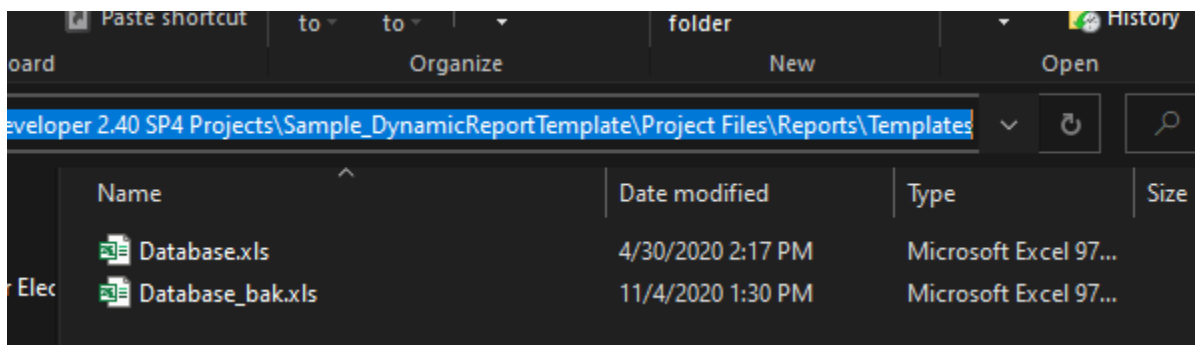
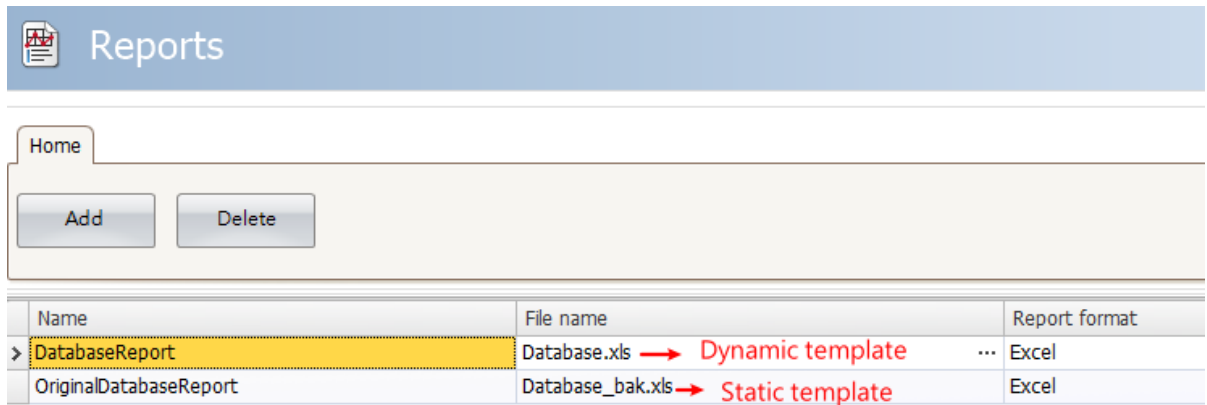
4.1.5 Define a static report template:

This template is the real template that includes the replacement indicators.

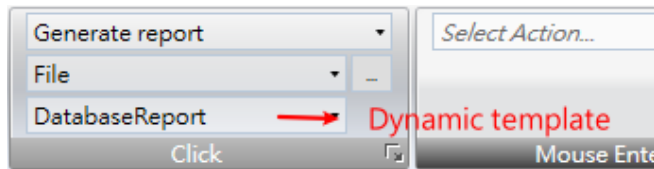
The indicators will be replaced with the real condition strings. In this sample, the indicators are `<#REPLACE(StartDateTimeCondition)>` and `<#REPLACE(EndDateTimeCondition)>`. You can add this static template by using Reports of the iX Developer as usual, or you can just put it at any place in the “Project Files” folder. And then, you need to modify the “StaticTemplateFilePath” variable in the ReportTemplateUtility script module according to the real file path you designed.



The below pictures show that the dynamic and static templates were both added via iX Developer Reports.



Generate Report action uses the dynamic template, DatabaseReport, defined in Reports.



5 Migration

This sample is a guideline for dynamically changing query conditions/a report template. You need to define replacement indicators in your static report template. And then, you need to write a method in a script module like *UpdateDateTimeQueryCondition* in the *ReportTemplateUtility* script module to replace the indicator. This sample demonstrates how to replace the date&time range conditions. When a user selects the time range from UI, the condition tags that present start time and end time will both be updated with SQL Datetime string.

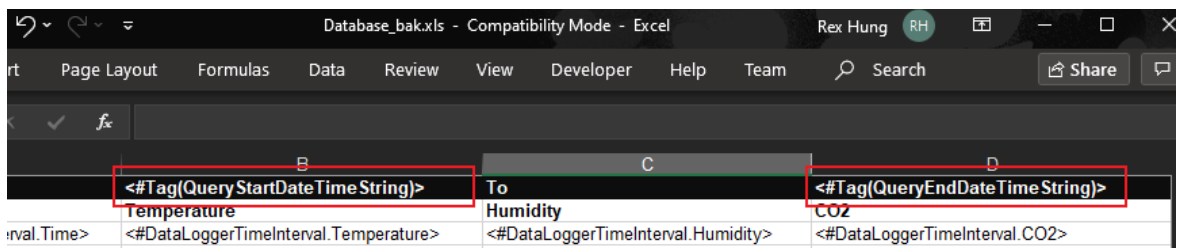
The following steps are for the user who wants to generate a report according to the time range during the run time. For the other user, it needs to modify the SQL string according to the requirements:

1. Copy the *ReportTemplateUtility* class to your project.
 - Update *StaticTemplateFilePath* variable: As stated above, it has to modify the path of the static report template according to your design. Or you can remove this variable and call the *ChangeDateTimeCondition* method with a path string directly.
 - Update *UpdateDateTimeQueryCondition* method or add methods: this step would need to be executed when the SQL script conditions are not the date&time range. a.) Define your replacement indicators. b.) Write a method to build a SQL script by replacing the indicators. This SQL script would be executed in the report generating process.
2. Add *QueryStartDateTime* tag and *QueryEndDateTime* tag:

- Set the Data Type to DATETIME

<i>QueryStartDateTime</i>	DATETIME	ReadWrite	DEFAULT		
<i>QueryEndDateTime</i>	DATETIME	ReadWrite	DEFAULT		

- Hoop event handlers to the ValueChange event: call the *ChangeDateTimeCondition* method in the handlers.
3. Add *QueryStartDateTimeString* tag and *QueryEndDateTimeString* tag for being used in the excel report template:



- Set the Data Type to STRING
- | | | | | | |
|---------------------------------|--------|-----------|---------|--|--|
| <i>QueryStartDateTimeString</i> | STRING | ReadWrite | DEFAULT | | |
| <i>QueryEndDateTimeString</i> | STRING | ReadWrite | DEFAULT | | |
- Update the value of *QueryStartDateTimeString* after *QueryStartDateTime* value changed. In this sample, it is performed in the ValueChange event handler.
 - Update the value of *QueryEndDateTimeString* after *QueryEndDateTime* value changed. In this sample, it is performed in the ValueChange event handler.
4. Design a screen layout that lets a user trigger updating *QueryStartDateTime* tag and *QueryEndDateTime* tag processes. This sample uses DatePicker and ComboBox control to perform it.

6 About Beijer Electronics

Beijer Electronics is a multinational, cross-industry innovator that connects people and technologies to optimize processes for business-critical applications. Our offer includes operator communication, automation solutions, digitalization, display solutions and support. As experts in user-friendly software, hardware

and services for the Industrial Internet of Things, we empower you to meet your challenges through leading-edge solutions.

Beijer Electronics is a Beijer Group company. Beijer Group has a sale over 1.6 billion SEK in 2019 and is listed on the NASDAQ OMX Nordic Stockholm Small Cap list under the ticker BELE. www.beijergroup.com

China

Shanghai

NORWAY

Drammen

TAIWAN

Taipei

DENMARK

Roskilde

SOUTH KOREA

Seoul

TURKEY

Istanbul

FRANCE

Paris

SWEDEN

Göteborg
Malmö
Stockholm

UNITED KINGDOM

Nottingham

GERMANY

Nürtingen

USA

Salt Lake City

6.1 Contact us

[Global offices and distributors](#)