TIA Portal Openness: Introduction and Demo Application

TIA Portal V15.1

https://support.industry.siemens.com/cs/ww/en/view/108716692

Siemens Industry Online Support

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.
Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.
By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.
The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.
Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.
For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.
Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: https://www.siemens.com/industrialsecurity.

Table of Contents

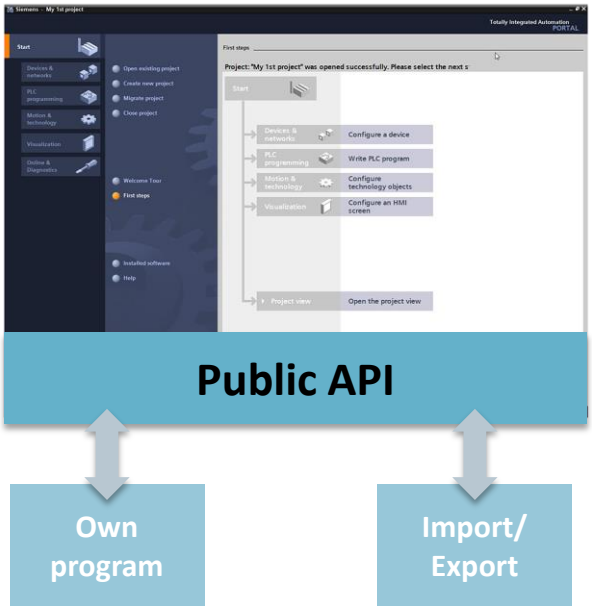
1 Task

Introduction

In STEP 7 or WinCC V15.1 TIA Portal Openness is included in the delivery of STEP°7 or WinCC in TIA Portal. This enables you to program the applications which automate the engineering in TIA Portal.

Overview of the automation task

Figure 1-1



Scenarios

- Based on a text database, project texts are to be complied program-controlled automatically and applied in the TIA Portal project. This way, you can quickly use standardized texts in new projects.
- The visualization is to be created automatically with the help of exported PLC data.
- Project statistics or backups can be created automatically. With the help of your program you can check whether programming guidelines were met.
- Projects can be automatically compared with global libraries and if required, updated and compiled.
- The offline project can be automatically compared with the online projects in order to ensure plant consistency.
- With the help of a tool the entire project is to be created based on prefabricated project parts.

2 Solution

2.1 Overview

Advantages

The solution presented here, offers you the following advantages

- More efficiency by executing the task faster
- Accuracy by automating constant processes
- Short commissioning times through configuration setup
- Competitiveness by targeted use of resources

Delimitation

This application does not contain a description of:

- Basics on object-oriented programming
- Basics on programming environment, such as e.g. Microsoft Visual Studio
- Basics on TIA Portal configuration

Basic knowledge of these topics is assumed.

2.2 Hardware and software components

2.2.1 Validity

This application is valid for

- STEP 7 / WinCC TIA Portal V15.1

2.2.2 Components used

The application was created with the following components:

Software components

Table 2-1

Component	No.	Article number	Note
STEP 7 Professional V15.1	1	6ES7822-1AA05-0YA5	
WinCC V15.1	1	6AV2102-0AA05-0AA5	
TIA Portal Openness	1		Included in the delivery of STEP 7 or WinCC V15.1
Microsoft Visual Studio 2015	1		

Example files and projects

The following list includes all files and projects that are used in this example.

Table 2-2

Component	Note
108716692_TiaPortalOpennessDemo_V15_1.zip	Demo application for the use of TIA Portal Openness (see chapter 6)
108716692_StartOpenness_V15_1.zip	Introductory example for the use of TIA Portal Openness (see chapter 5)
108716692_TIA_Openness_GettingStartedAndDemo_V15_1_en.pdf	This document.

3 Mode of Operation

3.1 TIA Portal Openness

In STEP 7 or WinCC V15.1 TIA Portal Openness is available for free on the respective product DVDs. Prerequisite for the use is an installation of STEP 7 or WinCC V15.1.

TIA Portal Openness provides DLLs via which you can access the TIA Portal platform. These DLLs are based on .NET Framework 4.6.1.

3.1.1 Range of functions

Table 3-1

Device	Function	see	Further information/restrictions
Throughout the project	Opening TIA Portal	P. 23	
Throughout the project	Closing/separating TIA Portal	P. 24	
Throughout the project	Connecting TIA Portal process	P. 23	
Throughout the project	Opening project	P. 25	
Throughout the project	Saving project	P. 25	
Throughout the project	Closing project	P. 25	
Throughout the project	Numerating devices and subnets	P. 26	
Throughout the project	Opening network and topology editor	P. 26	
Throughout the project	Creating folder	P. 27	
Throughout the project	Deleting elements	P. 28	
Throughout the project	Exporting/importing project graphics	P. 28	
Libraries	Opening global library	P. 30	
Libraries	Closing global library	P. 30	
Libraries	Types: Update check of instances	P. 31	
HMI	Compiling hardware and/or software	P. 28	
HMI	Exporting screen as XML	P. 28	Restrictions: TIA Portal Openness help chap. 8.3.7.1 *Overview of exportable screen objects*
HMI	Importing screens from XML	P. 28	
HMI	Exporting of the screen	P. 28	Exception: Pop-up and slide-in

Device	Function	see	Further information/restrictions
	management as XML		windows
HMI	Importing elements of the screen management from XML	P. 28	Exception: Pop-up and slide-in windows
HMI	Exporting tags/tag tables as XML	P. 28	
HMI	Importing tags/tag tables from XML	P. 28	
HMI	Exporting connections as XML	P. 28	Non-integrated connections
HMI	Importing connections from XML	P. 28	Non-integrated connections
HMI	Exporting scripts as XML	P. 28	
HMI	Importing scripts from XML	P. 28	
HMI	Exporting cycles as XML	P. 28	
HMI	Importing cycles from XML	P. 28	
HMI	Exporting text and graphic lists as XML	P. 28	
HMI	Importing text and graphic lists from XML	P. 28	
PLC	Compiling hardware and/or software	P. 28	
PLC	Connecting/separating controller online	P. 31	
PLC	Online/offline comparison	P. 32	
PLC	Offline/offline comparison	P. 32	
PLC	Displaying blocks in the editor	P. 26	
PLC	Exporting blocks as XML	P. 28	
PLC	Importing blocks from XML	P. 28	Only with enabler file
PLC	Importing external sources	P. 32	
PLC	Generating blocks from sources.	P. 33	
PLC	Generating sources from blocks	P. 33	STL and SCL blocks
PLC	Displaying tag tables and data types in the editor	P. 26	
PLC	Exporting tag tables and data types as XML	P. 28	
PLC	Exporting tag tables and data types as XML	P. 28	
PLC	Exporting individual tags	P. 28	

3.1.2 Restrictions

- With TIA Portal Openness V15.1 you can only access projects and libraries with version V15.1. If required, upgrade your project or your library before using TIA Portal Openness.
- Compatibility between the TIA Portal Openness versions cannot be guaranteed. A new version may require changes in your program.

3.2 "StartOpenness" application example

The "StartOpenness" program is to facilitate access to programming your first independent Openness application.

Some basic functions are already programmed in the program (e.g. starting TIA Portal) so that you can develop your own applications, based on this.

3.3 "DemoOpenness" application example

The "Demo_Openness" program includes many fully programmed functions of TIA Portal Openness. This is to give you a function overview as well as detailed programming help.

4 Creating a New TIA Portal Openness Application

4.1 TIA Portal V15.1

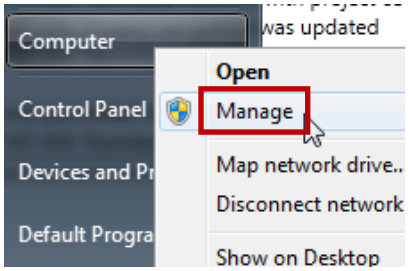
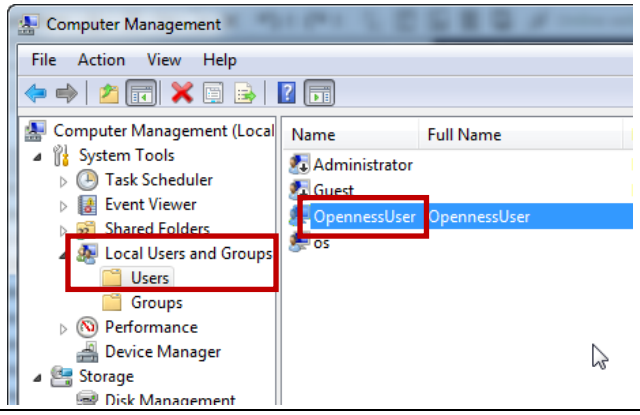
Install TIA Portal V15.1.

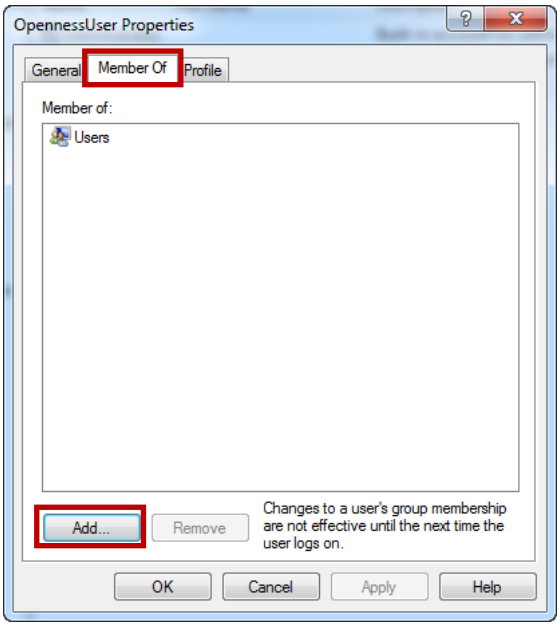
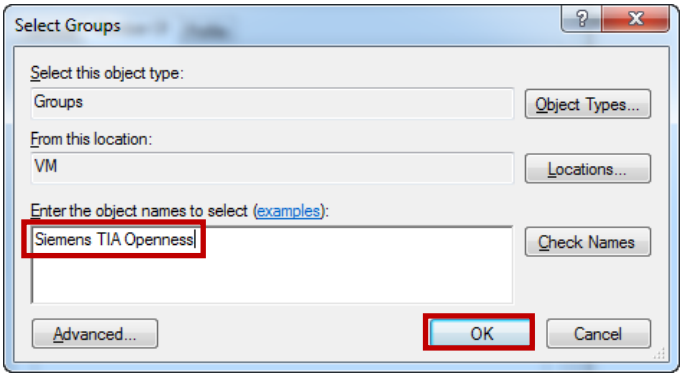
NOTE TIA Portal Openness is included in the delivery of STEP 7 V15.1 or WinCC V15.1 and is installed by default.

4.2 Managing user rights

In order to use or create a TIA Portal Openness application, the user has to be added to the "Siemens TIA Openness" user group.

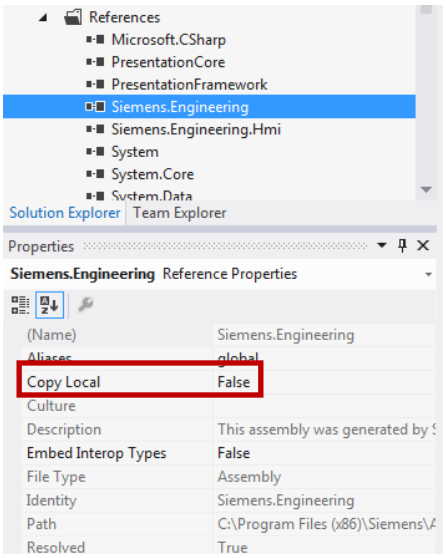
Table 4-1

No.	Action
1.	Right click "Computer" in the windows taskbar and select "Manage". 
2.	Open "Local Users and Groups and Groups > Users" and double click the user. 

No.	Action
3.	Go to the "Member Of" tab and click the "Add" button. 
4.	Enter "Siemens TIA Openness" and confirm by pressing OK. 
5.	Close the opened dialog boxes and log in again.

4.3 Creating a project

Table 4-2

No.	Action
1.	Create a new project (e.g. in Microsoft Visual Studio).
2.	Create the references for the Openness DLLs ((Siemens.Engineering.dll and Siemens.Engineering.HMI.dll). They are located in the TIA Portal installation directory in "... > Siemens > Automation > Portal V15_1 > PublicAPI > V15.1".
3.	Set the property "Copy Local" of the two DLLs to "False". 

4.4 Configuration file/AssemblyResolve

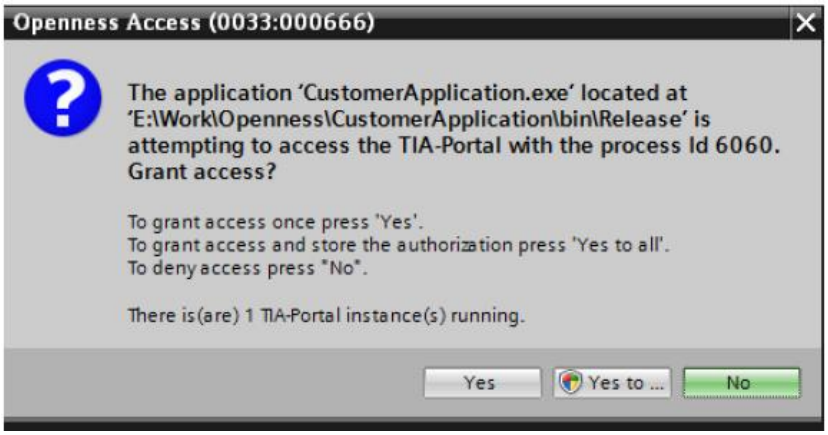
To detect the path of the Openness DLLs, you can either use a configuration file or the "AssemblyResolve" event.

Table 4-3

No.	Action
1.	**Configuration file** If you have selected a different path for the installation of WinCC or STEP 7 (TIA Portal), replace the default path in the configuration file by your installation path. Store the application configuration file in the same directory as the Openness application.
2.	**AssemblyResolve** In order to establish the connection to the TIA Portal, the "AssemblyResolve" method was used here. The installation path of the TIA Portal is read out of the registry so that the program can be used irrespective of the installation path.

4.5 Permitting access

Table 4-4

No.	Action
1.	When starting your application for the first time, the following security message will appear:  System manual (https://support.industry.siemens.com/cs/ww/en/view/109477163)
2.	Confirm the message with "Yes" in order to permit the access once. Confirm the message with "Yes to all" in order to always permit access to this application. Click "No" to deny the access.

Note If you are working with Microsoft Visual Studio, it can happen that you will receive the message, although you have already clicked "Yes to all". In order to avoid this, follow the instructions in entry \5\.

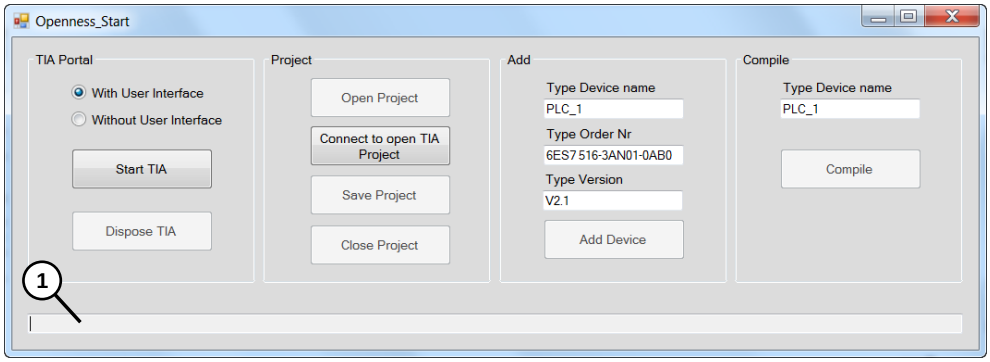
5 StartOpenness

5.1 Overview

The "StartOpenness" program is to facilitate access to programming your first independent Openness application.

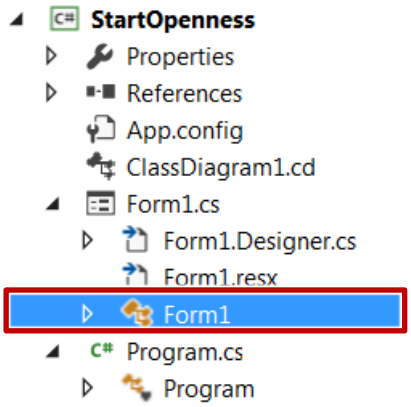
Note You find a ready compiled "exe"-file in the download "108716692_StartOpenness_V15_1.zip" with the path "…\StartOpenness\StartOpenness\bin\Release".

Figure 5-1



The status information is output via the text field (1).

Table 5-1

No.	Action
1.	Open the project "StartOpenness".
2.	Open the "Form1" class. 
3.	In order to establish the connection to the TIA Portal, the "AssemblyResolve" method was used here.

4.	Two tags were created:
	• "MyTiaPortal" of the "TiaPortal" type
	• "MyProject" of the "Project" type
	`public TiaPortal MyTiaPortal{get;set;}`
	`public Project MyProject{get;set;}`

5.2 "TIA Portal" group

Figure 5-2

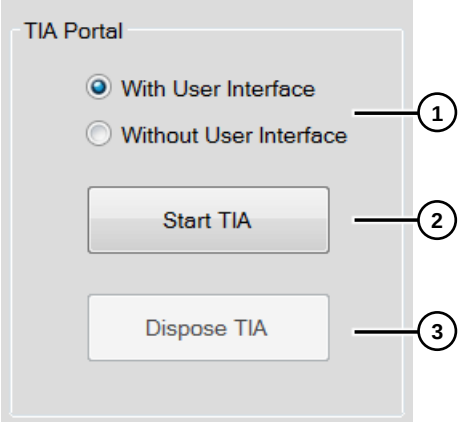


Table 5-2

No.	Description
1.	Select whether the TIA Portal is to be started with or without user interface via the option boxes "With User Interface" and "Without User Interface". In the "StartTIA()" method, the state of the option boxes is requested. `if (rdb_WithoutUI.Checked == true)`

2.	Click the "Start TIA" button in order to start the TIA Portal.
	The "StartTIA()" method is called.
	The "MyTiaPortal" tag is assigned a new TIA Portal instance.
	`if (rdb_WithoutUI.Checked == true)`
	`{`
	`  MyTiaPortal = new TiaPortal(TiaPortalMode.WithoutUserInterface);`
	`  txt_Status.Text = "TIA Portal started without user interface";`
	`  _tiaProcess = TiaPortal.GetProcesses()[0];`
	`}`
	`else`
	`{`
	`  MyTiaPortal = new TiaPortal(TiaPortalMode.WithUserInterface);`
	`  txt_Status.Text = "TIA Portal started with user interface";`
	`}`

3.	Click the "Dispose TIA" button in order to end the TIA Portal or the connection to the TIA Portal.
	The "DisposeTIA()" method is called.
	With "mytiaportal.Dispose()" the connection to the TIA Portal is terminated (if the TIA Portal was not started with the user interface) or the TIA Portal was terminated (if the TIA Portal was started without user interface).
	`private void DisposeTIA(object sender, EventArgs e)`
	`{`
	`  MyTiaPortal.Dispose();`
	`  txt_Status.Text = "TIA Portal disposed";`

5.3 "Project" group

Figure 5-3

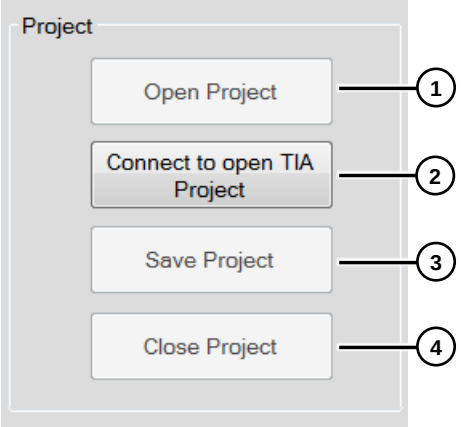


Table 5-3

No.	Description

1.	Click the "Open Project" button in order to open a new project. Select the project you would like to open in the file browser.
	The "SearchProject()" method is called.
	The file browser is opened. If a project has been selected the "OpenProject()" method is called.
	`private void SearchProject(object sender, EventArgs e)`
	`{`
	`  OpenFileDialog fileSearch = new OpenFileDialog();`
	`  fileSearch.Filter = "*.ap14|*.ap14";`
	`  fileSearch.RestoreDirectory = true;`
	`  fileSearch.ShowDialog();`
	The "MyProject" project tag is initialized via "MyTiaPortal.Projects.Open(new FileInfo(ProjectPath))".
	`MyProject = MyTiaPortal.Projects.Open(new FileInfo(ProjectPath));`

2.	Click the "Connect to Open TIA Project" button in order to connect to an open TIA Project.
	`private void btn_ConnectTIA(object sender, EventArgs e)`
	`{`
	`  btn_Connect.Enabled = false;`
	`  IList<TiaPortalProcess> processes = TiaPortal.GetProcesses();`
	`  switch (processes.Count)`
	`  {`
	`  case 1:`
	`  _tiaProcess = processes[0];`
	`  MyTiaPortal = _tiaProcess.Attach();`

3.	Click the "Save Project" button to save the project.
	The "SaveProject()" method is called.
	The project is saved via "MyProject.Save()".
	`private void SaveProject(object sender, EventArgs e)`
	`{`
	`  MyProject.Save();`
	`  txt_Status.Text = "Project saved";`
	`}`

No.	Description

4.	Click the "Close Project" button to close the project.
	The "CloseProject()" method is called.
	The project is closed via "MyProject.Close()".
	`private void CloseProject(object sender, EventArgs e)`
	`{`
	`  MyProject.Close();`
	`  txt_Status.Text = "Project closed";`

5.4 "Add" group

Figure 5-4

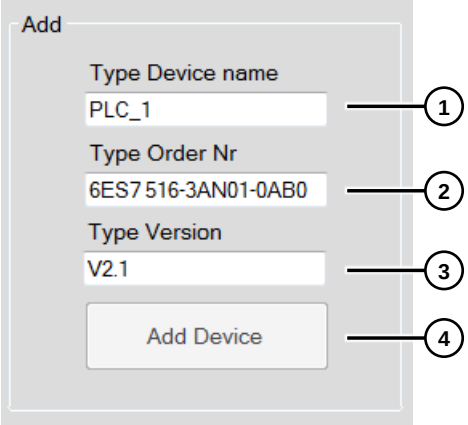


Tabelle 5-4

Nr.	Beschreibung
1.	Enter in the text field the name of the device you would like to add.
2.	Enter in the text field the order number of the device you would like to add.
3.	Enter in the text field the version of the device you would like to add.

4.	Click the "Add Device" button to add the device.
	The "btn_AddHW_Click()" is called.
	If no device with the entered name is found, the device is added and output in the status field.
	`Device deviceName = MyProject.Devices.CreateWithItem(MLFB, name, devname);`
	`txt_Status.Text = "Add Device Name: " + name + " with Order Number: "`
	`  + txt OrderNo.Text + " and Firmware Version: " + txt Version.Text;`

5.5 "Compile" group

Figure 5-5

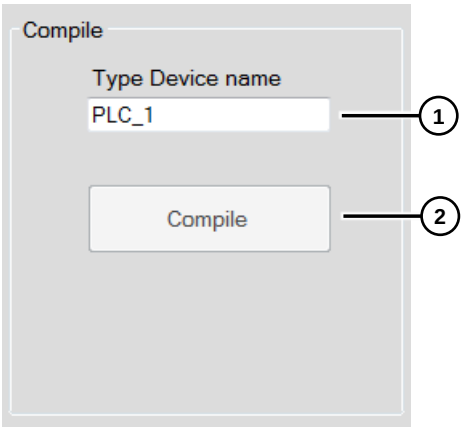


Table 5-5

No.	Description

1.	Enter in the text field the name of the controller or the operator panel you would like to compile.
	In the method Compile() the name of all devices and their DeviceItems are compared with the entered names.
	`string devname = txt_Device.Text;`
	`bool found = false;`
	
	`foreach (Device device in MyProject.Devices)`
	`{`
	`  DeviceItemComposition deviceItemAggregation = device.DeviceItems;`
	`  foreach (DeviceItem deviceItem in deviceItemAggregation)`
	`  {`
	`  if (deviceItem.Name == devname || device.Name == devname)`

2.	Click the "Compile" button in order to compile the device.
	The "Compile()" method is called.
	If a controller or an operator panel with this name is found, the device is compiled. The result is saved in the "result" tag of the "CompilerResult" type and output via the status field.
	`PlcSoftware controllerTarget = softwareContainer.Software as PlcSoftware;`
	`if (controllerTarget != null)`
	`{`
	`  found = true;`
	`  ICompilable compiler = controllerTarget.GetService<ICompilable>();`
	
	`  CompilerResult result = compiler.Compile();`

6 TIA Openness Demo

6.1 General

The "TIAPortalOpennessDemo" application consists of two projects.

- TiaOpennessHelper
  Some, especially the recurring methods were realized in the "TiaOpennessHelper". With the help of a reference to "TiaOpennessHelper.dll", you can use these methods in your own applications.
- TIAPortalOpennessDemo
  The demo application is to give you an overview of the functions that can be used with TIA Portal Openness. Due to the multitude of fully programmed examples, the application also provides detailed help for programming.

Below, you will find two directories which are to make handling of the application easier. If you are looking for a specific function or a specific application, you can use one of the following directories.

In order to comprehend the description start the "TIAPortalOpennessDemo.exe" program and open the project with Microsoft Visual Studio.

Note You find a ready compiled "exe"-file in the download "108716692_TiaPortalOpennessDemo_V15_1.zip" with the path "…\TiaPortalOpennessDemo_V15\TIAPortalOpennessDemo\TIAPortalOpennessDemo\bin\Release".

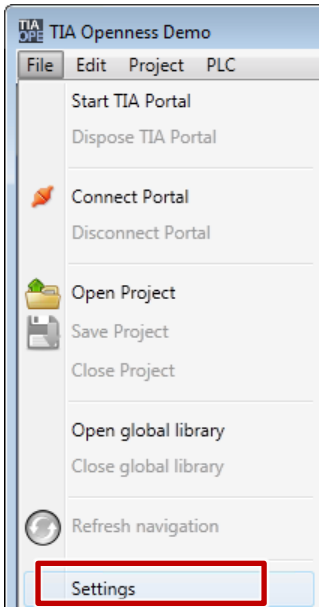
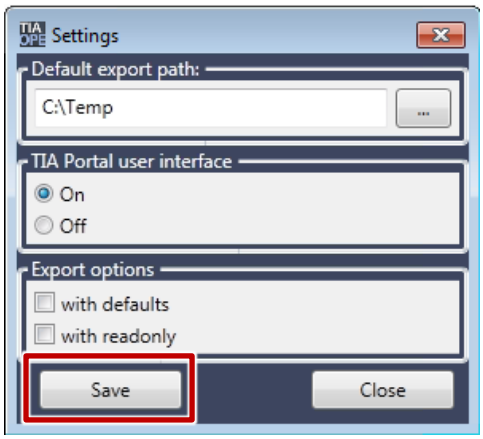
6.2 Application examples

6.3 Methods

6.4 Description of the application

6.4.1 General settings

Table 6-1

No.	Action
1.	Click the „File->Settings“.  
2.	Select whether the TIA Portal is to be started with or without user interface.
3.	Select the export settings.
4.	Select the default path for the export and import files.
5.	Click the “Save” button to save the settings.

6.4.2 Flexible AssemblyResolve

Open the “General.cs” class of the “TiaOpennessHelper” project.

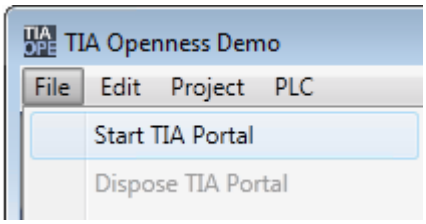
In order for you to be able to use the application on any PC with the TIA Portal, the TIA Portal installation path is read out from the registry and is thus passed on flexibly.

6.4.3 TIA Portal functions

Starting the TIA Portal

You can open a new TIA Portal instance via “File > Start TIA Portal”. Depending on your settings the TIA Portal is started with or without user interface.

Figure 6-1



Open the “OpenTiaPortalCommand_Executed” method in the “MainWindowViewModel.cs” class.

The “tiaPortal” tag from the “TiaPortal” type is initialized. With the parameter “TiaPortalMode.WithoutUserInterface” or “TiaPortalMode.WithUserInterface” you specify whether the TIA Portal is started with or without user interface.

Connecting with the active TIA Portal instance

Via the “File->Connect Portal” the active TIA Portal instances are displayed.

Figure 6-2

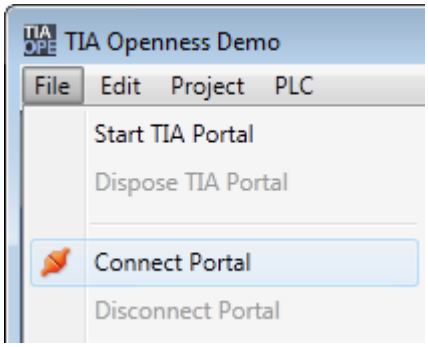
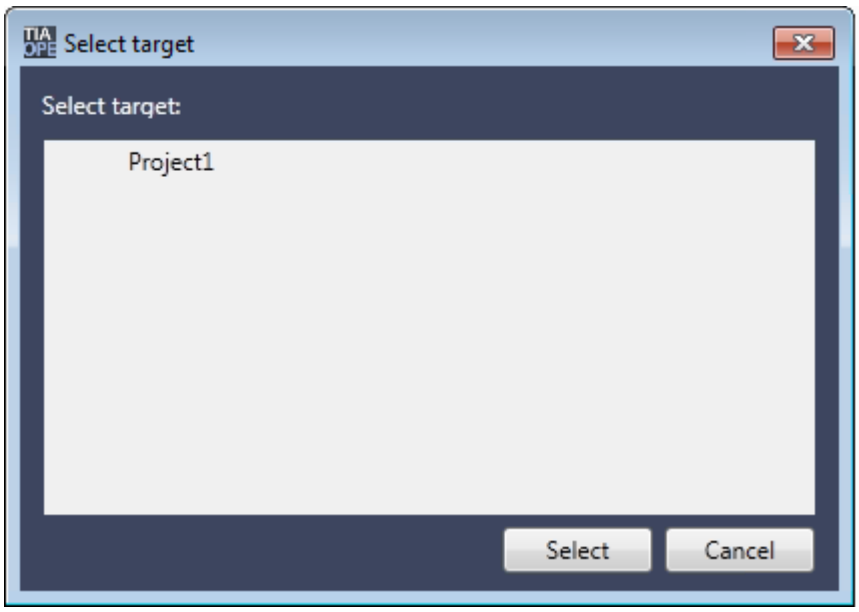


Figure 6-3



By clicking the instance, the demo application connects with this instance.

Open the “CreateInstanceTreeView” method in the “MainWindowViewModel” class.

For this purpose the “TiaPortal.GetProcesses()” method is used.

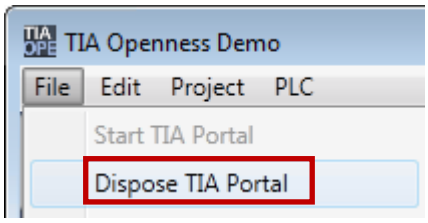
In the “ShowConnectCommand_Executed” method, the connection to the selected instance is established.

For this purpose, the TIA Portal tag is assigned a value via “process.Attach()”.

Separating from TIA Portal

You can end the connection to the TIA Portal instance again via “File > Dispose TIA Portal”. If you have started the TIA Portal instance without user interface, the instance is terminated. Otherwise the connection between your application and the TIA Portal instance is terminated.

Figure 6-4



Open the “DisposeTiaPortalCommand_Executed” method in the “MainWindowViewModel.cs” class.

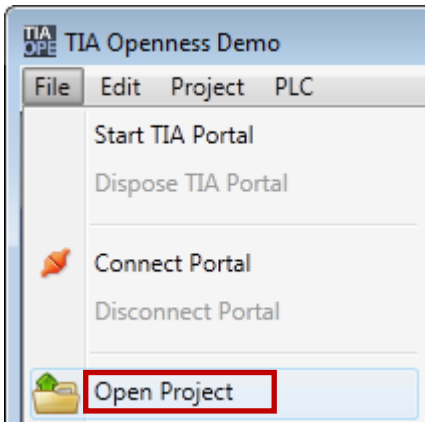
The connection is separated via the “Dispose()” method of the TIA Portal tag.

6.4.4 Project functions

Opening/saving/closing project

If no project is opened yet, you can open a project via “File > Open project”.

Figure 6-5

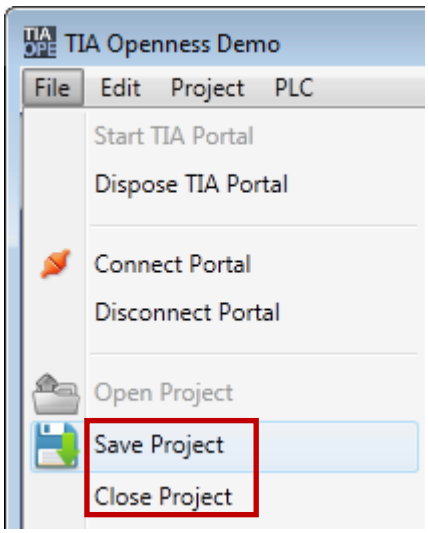


Open the “OpenProjectCommand_Executed” method in the “MainWindowViewModel.cs” class.

For this purpose, a tag of the “Project” type is initialized with the .Projects.Open(new FileInfo(filename)) method of the TIA Portal tag.

If a TIA Portal project is opened, you can save or close the project via “File > Save project” or “File > Close project”.

Figure 6-6



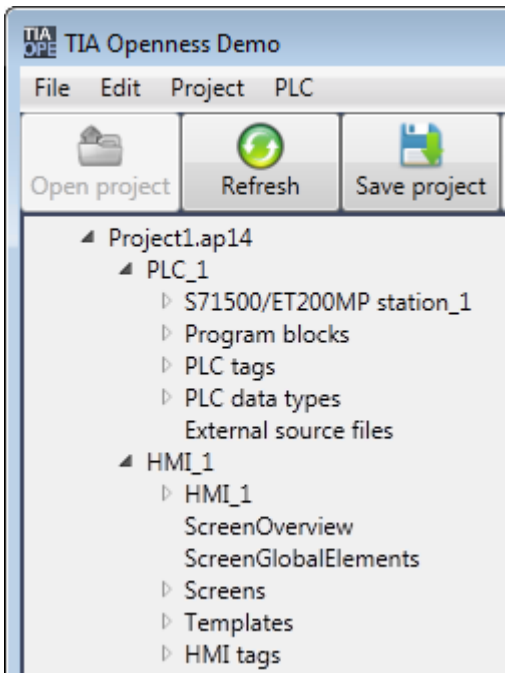
Open the method “SaveProjectCommand_Executed” or “CloseProjectCommand_Executed” in the “MainWindowViewModel.cs” class.

Via the method “Project.Save()” or “Project.Close()” of the project tag, the project is saved or closed.

Reading out devices

On the left side of the application, the devices and their elements are shown, similar to a project tree in the TIA Portal.

Figure 6-7



For this purpose, all “Devices” and their “DeviceItems” included in the project are read out.

Open the class “General.cs” in the “TiaOpennessHelper” project.

In this class the project contents are read out and provided as lists.

In the “MainWindowViewModel.cs” class of the “TIAPortalOpennessDemo” project this list is accessed in the “CreateDeviceTreeViewItem ()” method.

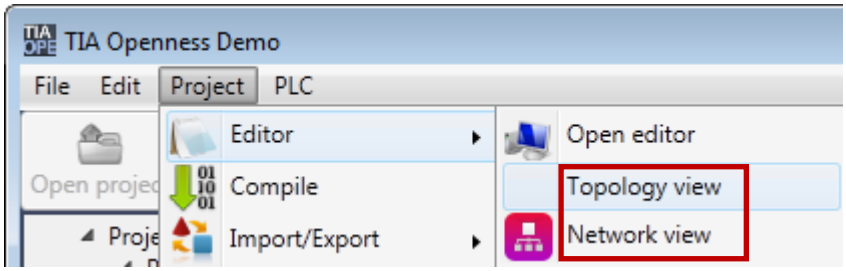
When listing the elements, please note the following:

- Many elements can be located in user-defined folders. The nesting depth is not necessarily known here. For this reason the functions should be recursive.

Opening editors

You can start the topology and the network editor in the user interface of the TIA Portals with “Project > Editor”.

Figure 6-8

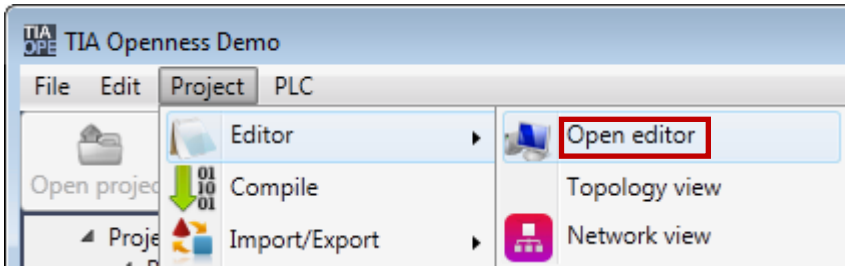


Open the method “OpenTopologyViewCommand_Executed” in the “MainWindowViewModel.cs” class.

You can start the topology and the network editor via “Project.ShowHwEditor(Siemens.Engineering.HW.View.Topology)” or “Project.ShowHwEditor(Siemens.Engineering.HW.View.Network)”

You can open other editors (e.g. block editor) via the “Project > Editor > Open editor” menu if the focus is on a respective object.

Figure 6-9



Open the “OpenEditorCommand_Executed” method in the “MainWindowViewModel.cs” class.

For this purpose, the “ShowInEditor()” method is called on a respective object (e.g. a block).

Creating folder

Depending on your selection in the project tree of the TIA Portal Openness application, you can create new folders (e.g. block folder).

Figure 6-10

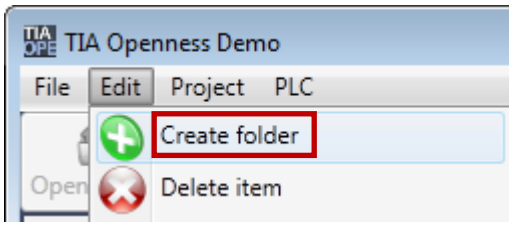
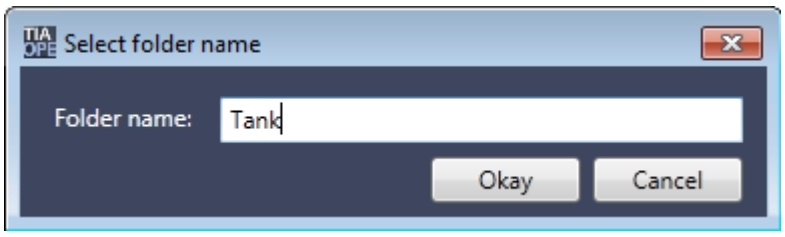


Figure 6-11



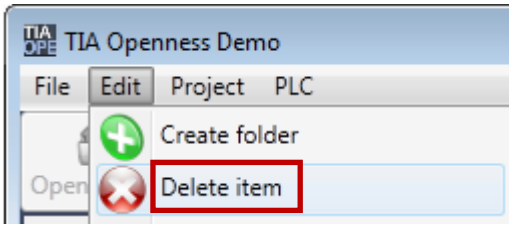
Open the “CreateFolderCreateCommand_Executed” method in the “MainWindowViewModel.cs” class.

A new folder is created via the “Groups.Create(FolderName)” method of the selected object. In order to do this, it is checked beforehand, whether a folder can be created in the selected object.

Deleting elements

Depending on your selection in the project tree of the TIA Portal Openness application, you can delete folders.

Figure 6-12



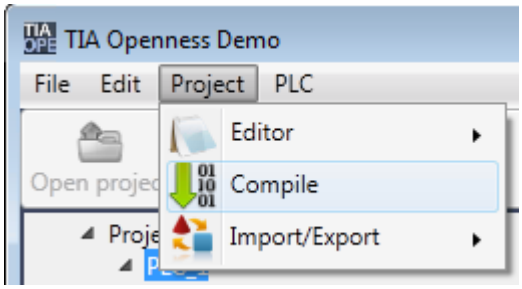
You find the “DeleteCommand_Executed” method in the “MainWindowViewModel.cs” class.

You can delete it via the “Delete()” method of the respective element.

Compiling elements

The device can be compiled with TIA Portal Openness.

Figure 6-13



Open the class “Compile.cs” in the “TiaOpennessHelper” project.

The call of this method can be found in the “MainWindowViewModel.cs” class in the “Compiler” region.

You can start compiling via the “Compile()” method.

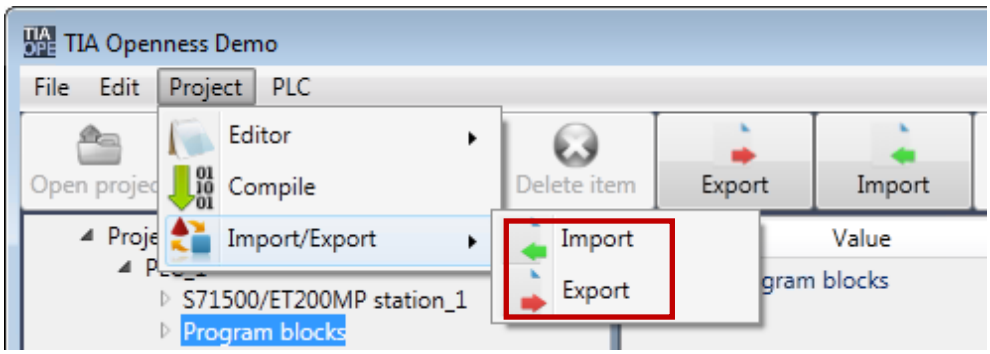
6.4.5 Importing/exporting objects

The following objects can be exported and imported as XML file.

- Project graphics
- Blocks (apart from SCL), PLC tag tables and data types
- Screens, screen management, tags, (non-integrated) connections, scripts, cycles, text and graphics lists

For this purpose, select “Project > Import/Export” from the menu.

Figure 6-14



Note You can only export consistent blocks. Compile the blocks before exporting.

In the “TiaOpennessHelper” project the classes “Export.cs” and “Import.cs” are included.

An element is imported or exported with the “Import(file, importOption)” or “Export(path, exportOption) method.

With the ImportOptions you can select whether a file is to be overwritten (“Override”) automatically or whether the import is to be cancelled with an error message (“None”), if the object already exists.

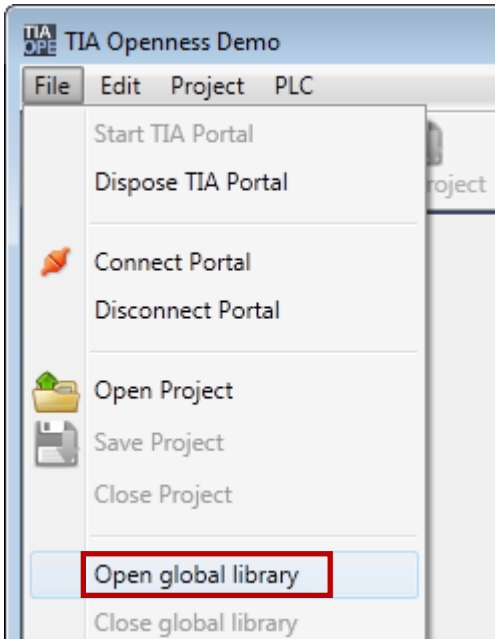
With the ExportOptions it is specified which data is to be exported. With “ExportOptions.None”, all settings that were changed are exported. With “ExportOptions.WithDefaults” additional default values are exported. With “ExportOptions.WithReadOnly” additional write-protected values are exported.

6.4.6 Libraries

Opening/closing global libraries

If the TIA Portal is opened you can open the global library via the “File > Open global library” menu.

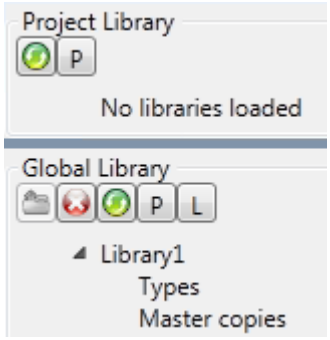
Figure 6-15



Note The global library is opened for access via Openness. It is not displayed in the user interface of the TIA Portal.

With the “Libraries” button on the right side of the demo application, the project library and global library that has been opened with Openness is shown.

Figure 6-16



Open the “OpenGlobalLibraryCommand_Executed” method in the “MainWindowViewModel.cs” class.

For this purpose, a tag of the “GlobalLibrary” type is initialized with the “GlobalLibraries.Open(ilename)” method of the TIA Portal tag.

If a global library is opened and selected in the application, you can close the global library with “File > Close global Library”.

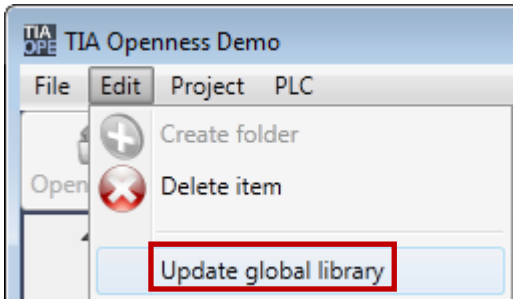
You find the “CloseGlobalLibraryCommand_Executed” function in the “MainWindowViewModel.cs” class.

The library is closed with the “Close()” method of the library tag.

Update check

With the “Edit > Update global library” you can perform an update check.

Figure 6-17



The result of the check is displayed in the status bar of the application.

Open the “UpdateGlobalLibraryCommand_Executed” method in the “MainWindowViewModel.cs” class.

The “UpdateCheck” method of the global library provides the update event as tag from the “UpdateCheckResult” tag.

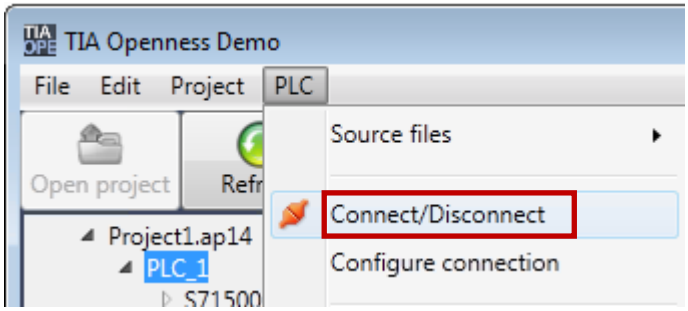
With the help of the “UpdateCheckMode.ReportOutOfDateAndUpToDate” parameter, identical and non-identical types are output. With the “UpdateCheckMode.ReportOutOfDateOnly” parameter, only the non-identical types are output.

6.4.7 PLC

Online connection to controller

You can establish an online connection to a connected controller. Select “PLC > Connect/Disconnect” in the menu.

Figure 6-18



Open the “ConnectPlcCommand_Executed” method in the “MainWindowViewModel.cs” class

A tag of the “OnlineProvider” type is created. With the “GoOnline()” method, an online connection is established.

Prerequisite is, that a connection between TIA Portal and controller has already been configured (e.g. by downloading the configuration). Otherwise an error message will be output in the status line of the application.

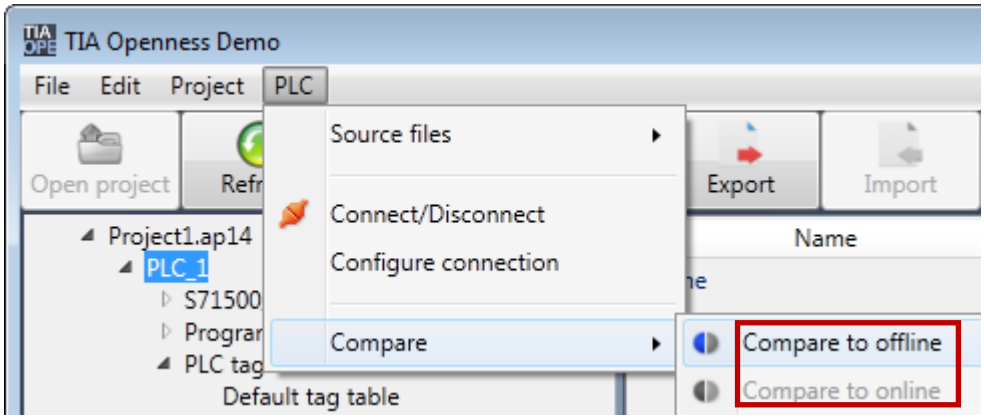
With the “GoOffline()” method the connection is separated again.

Comparing controllers

- Online/offline

You can compare the selected controller with the online connected controller with “PLC > Compare > Compare to online”.

Figure 6-19



The result of the comparison is displayed in the status bar of the application.

Open the “CompareOnlineCommand_Executed” method in the “MainWindowViewModel.cs” class.

The “CompareToOnline()” method of the control tag compares the controllers and provides the result of the comparison of the “CompareResult” type as return value.

- Online/offline

With “PLC > Compare > Compare to offline” you can compare a controller in the project with a controller of a global library. Open the global library and select the controller. Select the controller you want to compare in the project tree of the application. The result of the comparison is displayed in the status bar of the application.

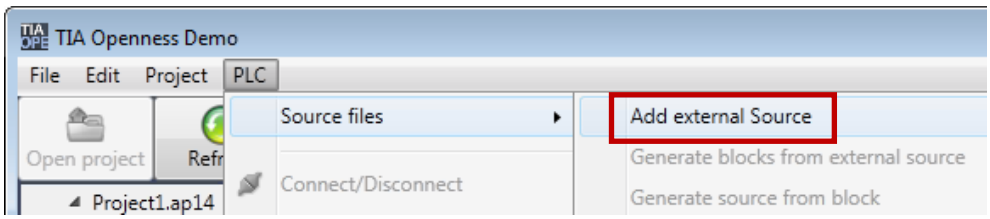
Open the “CompareOfflineCommand_Executed” method in the “MainWindowViewModel.cs” class.

The “CompareTo(Object)” method compares the controllers and provides the result of the comparison of the “CompareResult” type as return value.

Integrating external sources

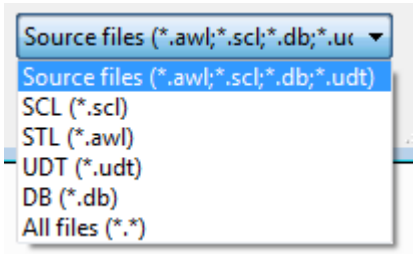
When the “External source” folder is selected you can integrate an external source via “PLC > Source files > Add external Source”.

Figure 6-20



With the filter settings, you can select whether STL or SCL sources are to be displayed.

Figure 6-21



Open the “AddExternalSourceCommand_Executed” method in the “MainWindowViewModel.cs” class.

The source is adopted in the TIA Portal project with “CreateFromFile(sourcename, filename)”.

Generating block from source

When a source is selected you can generate a block from the source with “PLC > Source files > Generate blocks from external source”.

Open the “GenerateBlockfromSourceCommand_Executed” method in the “MainWindowViewModel.cs” class.

A block is generated with “GenerateBlockFromSource()”.

Generating source from block

You can generate a source file from a SCL or an STL block with “PLC > Source files > Generate source from block”.

Specify the storage path via the file browser.

Open the “GeneratesourcefromBlockCommand_Executed” method in the “MainWindowViewModel.cs” class.

A source is generated with the “GenerateSourceFromBlocks(blocklist, filepath)”.

7 Related literature

Table 7-1

	Topic
\1\	Siemens Industry Online Support http://support.industry.siemens.com
\2\	Download page of the entry https://support.industry.siemens.com/cs/ww/en/view/108716692
\3\	Support Request http://www.siemens.com/automation/support-request
\4\	Why does your TIA Portal Openness application not work as expected? http://support.industry.siemens.com/cs/ww/en/view/109251656
\5\	Why, when using a TIA Portal Openness application, do you get the error message “Cannot connect to TIA Portal”? http://support.industry.siemens.com/cs/ww/en/view/109038214
\6\	System manual https://support.industry.siemens.com/cs/ww/en/view/109477163

8 History

Table 8-1

Version	Date	Modifications
V1.0	02/2015	First version
V1.0	09/2015	Fixed minor problems
V1.1	12/2016	Version for TIA Portal V14
V1.2	05/2017	Version for TIA Portal V14 SP1
V1.3	02/2018	Version for TIA Portal V15
V1.4	05/2019	Version for TIA Portal V15.1