

The Siemens logo is displayed in a white rectangular box with a thin black border. The word "SIEMENS" is written in a bold, teal, sans-serif font. The background of the entire page is a blurred industrial factory setting with bright overhead lights and various pieces of machinery.

SIEMENS

OpennessScripter: Detailed Documentation

TIA Portal / Openness API

<https://support.industry.siemens.com/cs/ww/en/view/109742322>

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 under <https://www.siemens.com/cert>.

Table of Contents

Legal information	2
1 Introduction	5
2 Operation modes	5
2.1 Instances.....	5
2.2 Localization	5
2.3 User interface	5
2.4 Console.....	9
2.5 Integration in TIA Portal	13
3 Command set	14
3.1 Scope.....	14
3.2 Model and hierarchy	14
3.3 Syntax for the command set	17
3.4 Option parameters.....	18
3.5 Output	19
3.5.1 Advanced output.....	19
3.6 TIA Portal	20
3.6.1 Global libraries (user libraries)	20
3.7 Project.....	22
3.7.1 Project library	22
3.7.2 Project graphics.....	23
3.7.3 Networks	24
3.7.4 Topology	24
3.7.5 Transaction mode (exclusive access).....	24
3.8 PLC.....	25
3.8.1 User data types and system data types of a PLC	26
3.8.2 External sources of a PLC	27
3.8.3 Program blocks and data blocks of a PLC.....	27
3.8.4 System blocks of a PLC.....	28
3.8.5 F-blocks of a PLC	29
3.8.6 Blocks with know-how protection (KHP) of a PLC.....	30
3.8.7 Tags and user constants of a PLC	30
3.9 HMI	32
3.9.1 Screens of an HMI	32
3.9.2 Screen templates of an HMI.....	33
3.9.3 Global screen of an HMI	34
3.9.4 Graphic lists of an HMI.....	34
3.9.5 Permanent screen of an HMI	34
3.9.6 Text lists of an HMI	35
3.9.7 Tags of an HMI	35
3.9.8 VB scripts of an HMI	36
3.9.9 Connections of an HMI	37
3.9.10 Cycles of an HMI	38
3.10 Data types	39
3.11 Escape characters.....	39
3.12 Variables	40
3.13 Comments.....	40
3.14 Condition and negation.....	40
4 Run.....	41
4.1 Breakpoints	41
4.2 TIA Portal target version	42
5 Tips	43

6 Known problems44

7 Appendix.....45

 7.1 Service and support.....45

 7.2 Industry Mall46

 7.3 Links and literature46

 7.4 Change documentation.....46

1 Introduction

This documentation shows the details of the OpennessScripter. The OpennessScripter introduction can be found in the same entry with the ID 109742322:

<https://support.industry.siemens.com/cs/ww/en/view/109742322>

2 Operation modes

2.1 Instances

You can start multiple OpennessScripter instances. In a started OpennessScripter instance, you can open a script.

2.2 Localization

When starting the OpennessScripter, the tool is automatically set to the operating system language. It supports the following languages:

- German
- English

If you change the operating system language while running the OpennessScripter, you need to restart the OpennessScripter for the language to change. If no localization is available for the operating system language, English will be used automatically.

2.3 User interface

"File" menu

The "File" menu allows you to access the following commands:

Table 2-1

Command	Description	Keyboard shortcut
New	Creates a new script.	Ctrl+N
Open	Opens a dialog to open an existing script from the file system.	Ctrl+O
Close	Closes the current script and displays the start screen.	Ctrl+W
Save	Saves the current script.	Ctrl+S
Save as...	Opens a dialog to save the current script to a new storage location.	F12
Open script save location	Opens Windows Explorer with the current script's folder.	n/a
Open script in the default text editor	Opens the current script with the default text editor.	n/a
Exit	Exits the application.	Alt+F4

"Edit" menu

The "Edit" menu allows you to access the following commands:

Table 2-2

Command	Description	Keyboard shortcut
Undo	Undoes the last action in the Script Editor.	Ctrl+Z
Redo	Redoes the last undone action in the Script Editor.	Ctrl+Y
Cut	Cuts the selected text in the Script Editor and pastes it to the clipboard.	Ctrl+X
Copy	Copies the selected text in the Script Editor to the clipboard.	Ctrl+C
Paste	The text from the clipboard is pasted to the Script Editor.	Ctrl+V
Select all	Selects the entire text in the Script Editor.	Ctrl+A
Delete	Deletes the selected text in the Script Editor.	Del

"Run" menu

The "Run" menu allows you to access the following commands:

Table 2-3

Command	Description	Keyboard shortcut
Check	Checks the script for errors.	F6
Run	Starts script execution. Automatically includes a check.	F5
Run line by line	Starts running the script line by line. To run the next line, the user must then select the "Run" command.	F10
Pause	Requests a pause. The script stops at the earliest possible time.	Ctrl+Alt+Pause
Stop	Requests early termination. The script aborts at the earliest possible time.	Shift+F5
Toggle breakpoint	Inserts or deletes a breakpoint in the current line.	F9
Delete all breakpoints	Deletes all breakpoints.	Ctrl+Shift+F9
TIA Portal target version	Defines the TIA Portal target version for running the script.	n/a
Activate echo	Activates/deactivates echo. (The executed line is repeated in the output.)	n/a

"Window" menu

The "Window" menu allows you to access the following commands:

Table 2-4

Command	Description	Keyboard shortcut
Toolbar	Shows/hides the toolbar.	<i>n/a</i>
Recently used scripts	Shows/hides the "Recently used scripts" sidebar.	<i>n/a</i>
Output	Shows/hides the "Output" sidebar.	<i>n/a</i>
Always on top	Permanently moves the application to the foreground so that its visibility is permanently prioritized over TIA Portal.	<i>n/a</i>

"Settings" menu

The "Settings" menu allows you to access the following commands:

Table 2-5

Command	Description	Keyboard shortcut
Zoom in	Gradually increases the font size in the Script Editor and in the output (maximum: 600%).	Ctrl +
Zoom out	Gradually decreases the font size in the Script Editor and in the output (minimum: 75%).	Ctrl -
Reset zoom	Resets the font size in the Script Editor and in the output to 100%.	Ctrl+0
Create file association	Registers the .opns file extension for the current Windows user so that the script files can be opened in Windows Explorer by double-clicking or be executed by right-clicking.	<i>n/a</i>
Delete file association	Deletes the registration of the .opns file extension for the current Windows user.	<i>n/a</i>
Open "Local users and groups"	The Windows Management Console opens with the "Local Users and Groups" window. (In "Groups", the user can enter his details in "Siemens TIA Openness".)	<i>n/a</i>

"Help" menu

The "Help" menu allows you to access the following commands:

Table 2-6

Command	Description	Keyboard shortcut
Documentation	Opens the detailed documentation of the OpennessScripter.	F1
Online support	Opens the entry page at Siemens Industry Online Support.	<i>n/a</i>
Info	Opens a dialog with information about the OpennessScripter.	<i>n/a</i>

"X" menu

The "X" menu allows you to access the following commands:

Tabelle 2-7

Command	Description	Keyboard shortcut
Close	Closes the current script and displays the start screen.	Ctrl+W

Toolbar

Figure 2-1



The toolbar has the following commands:

Table 2-8

Command	Description	Keyboard shortcut
New	Creates a new script.	Ctrl+N
Open	Opens a dialog to open an existing script from the file system.	Ctrl+O
Save	Saves the current script.	Ctrl+S
Undo	Undoes the last action in the Script Editor.	Ctrl+Z
Redo	Redoes the last undone action in the Script Editor.	Ctrl+Y
Check	Checks the script for errors.	F6
Run	Starts script execution. Automatically includes a check.	F5
Pause	Requests a pause. The script stops at the earliest possible time.	Ctrl+Alt+Pause
Stop	Requests early termination. The script aborts at the earliest possible time.	Shift+F5
Toggle breakpoint	Inserts or deletes a breakpoint in the current line.	F9
TIA Portal target version	Defines the TIA Portal target version for running the script.	n/a
Zoom in	Gradually increases the font size in the Script Editor and in the output (maximum: 600%).	Ctrl +
Zoom out	Gradually decreases the font size in the Script Editor and in the output (minimum: 75%).	Ctrl -
Documentation	Opens the detailed documentation of the OpennessScripter.	F1

"Recently used scripts" sidebar

The recently used scripts are available in the "Recently used scripts" sidebar on the left. The X allows you to remove items. The pin allows you to pin a script.

The scripts are grouped:

"Pinned", "Today", "Last week" and "Older"

"Output" sidebar

The lower "Output" window displays the outputs of the check or script execution. It is continued with each check or execution. The "Output" window provides you with the following options:

- Trash bin icon: Empties the output.
- "Auto Scroll" option: Enables automatic scrolling of the output.
- Disk icon: Saves the output either to a text file (*.txt) or a XML file (*.xml).
- Filter selection: Various output lines are shown/hidden depending on the selected level (Debug, Information, Warning, Error, and Critical).

2.4 Console

Arguments

The OpennessScripter console provides you with automation options. In the command line, you can pass command arguments to the OpennessScripter:

- Switch for console mode (mandatory, must be the first argument):

```
/silent
```

- Script file to be run (mandatory):

```
/file:"D:\tmp\demoscript.opns"
```

- TIA Portal target version (optional):

```
- /target:"v13sp1"  
- /target:"v13sp2"  
- /target:"v14"  
- /target:"v14sp1"  
- /target:"v15"  
- /target:"v15.1"  
- /target:"v16"  
- /target:"v17"  
- /target:"v18"
```

If you do not specify a parameter, the system will automatically select a version. Chapter [4.2](#) explains how this is done.

- Storage location for the log file (optional) as text file or XML file:

```
/log:"output.txt"
```

or

```
/log:"output.xml"
```

If the file path does not exist, the mandatory directories will be created.

- Variable assignments (optional), example (the variable name is random):

```
projectPath="D:\demo\project\project.ap17"
```

Several assignments are possible that are made before the actual script. Except for option parameters, all data types are possible.

Variable assignments

Example of a console call with assignments for the `<exportFolder>` and `<projectPath>` variables that you can use in the script:

```
OpennessScripter.exe /silent /file:"D:\tmp\demoscript.opns"  
exportFolder="D:\tmp\" projectPath="D:\demo\project\project.ap17"
```

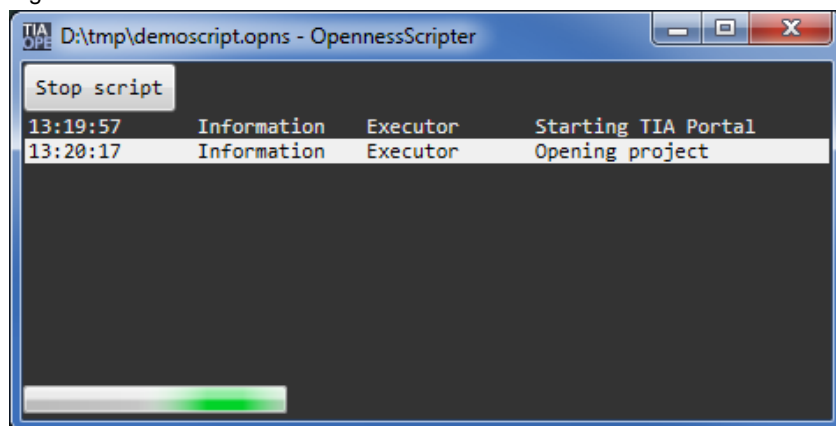
Example for Windows Batch

A Windows batch file for the command line can look as follows:

```
set exe="C:\Program Files (x86)\Siemens\Automation\OpennessScripter\  
OpennessScripter.exe"  
set script="D:\tmp\demoscript.opns"  
set log="D:\tmp\output.txt"  
echo "working..."  
start "Script is running" /wait %exe% /silent /file:%script%  
/log:%log%  
echo "done!"
```

When you run the Windows batch file, the OpennessScripter starts minimized in the Windows taskbar and stops at the end of script execution. Clicking the icon in the taskbar opens the user interface. The user interface allows you to both monitor progress and terminate script execution early.

Figure 2-2



If the log file is not created at the specified storage location, you should check the path. The file may have been saved to the OpennessScripter folder.

Note

The log file is not extended but overwritten.
Back up the log file or specify a new storage location.

Example for VB script (VBS)

A call in a VB script can look as follows:

```
Sub MyMethod()  
    StartScripter("D:\scripts\script.opns")  
End Sub  
  
Function StartScripter(script)  
    Dim application  
    application = "D:\OpennessScripter\OpennessScripter.exe"  
  
    Dim args  
    args = "/silent /file:" & """" & script & """"  
  
    Dim shellpath  
    shellpath = """" & application & """" & " " & args  
  
    Dim WshShell  
    Set WshShell = CreateObject("WScript.Shell")  
    WshShell.Run shellpath, 1, True  
End Function
```

Example for Visual Basic for Applications (Office macros, VBA)

A call in Visual Basic for Applications can look as follows:

```
Private Sub MyMethod()  
    StartScripter "D:\Scripts\Script.opns"  
End Sub  
  
Private Function StartScripter(ByVal script As String)  
    Dim application As String  
    application = "D:\OpennessScripter\OpennessScripter.exe"  
  
    Dim args As String  
    args = "/silent /file:" & """" & script & """"  
  
    Dim shellpath As String  
    shellpath = """" & application & """" & " " & args  
  
    Shell shellpath  
End Function
```

Example for Visual Basic .NET

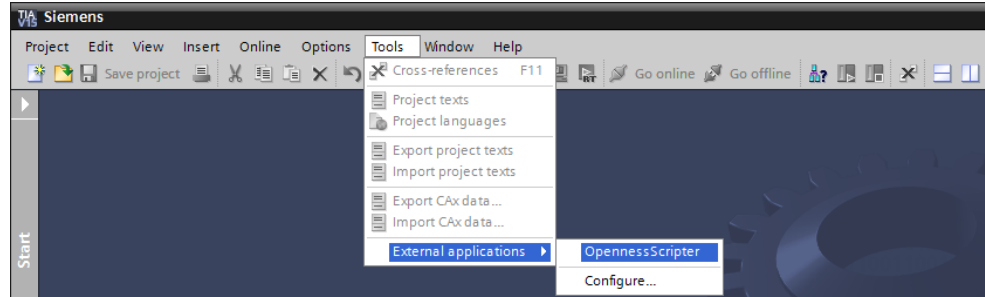
A call in Visual Basic .NET can look as follows:

```
Private Sub MyMethod()  
    StartScripter("D:\Scripts\Script.opns")  
End Sub  
  
Private Sub StartScripter(script As String)  
    Dim application As String  
    application = "D:\OpennessScripter\OpennessScripter.exe"  
  
    Dim args As String = "/silent /file:" & "\"" & script & "\""   
  
    Dim process As Process = Process.Start(application, args)  
    process.WaitForExit()  
End Sub
```

2.5 Integration in TIA Portal

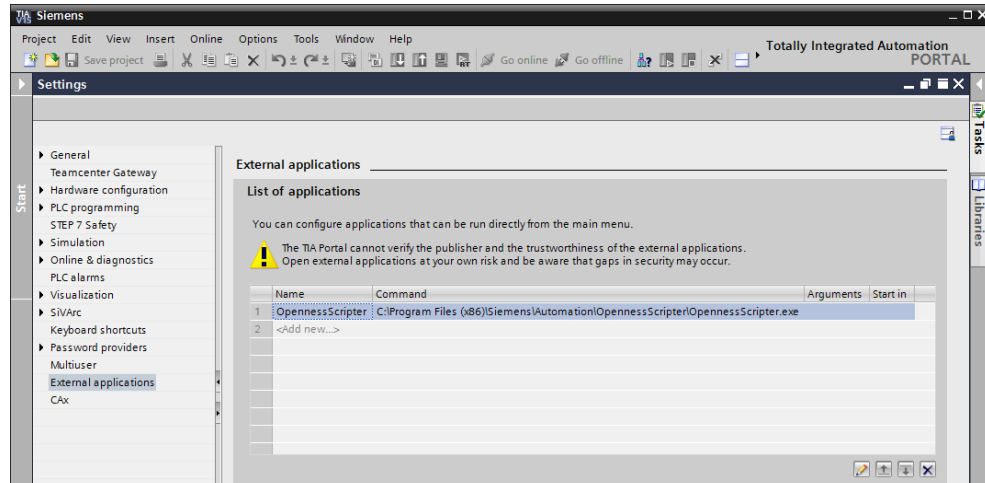
TIA Portal offers the possibility to integrate external applications in the menu bar since V15.

Figure 2-3



For example, you can link to the OpennessScripter. For an extended configuration see the previous chapter "Console" for calling a script.

Figure 2-4



3 Command set

3.1 Scope

The command set of the OpennessScripter includes **only the basic options** provided by the TIA Portal Openness API. High-level language functionality is not available.

The following Openness objects are currently not supported in the OpennessScripter:

- Hardware objects / hardware configuration
- Startdrive (SINAMICS)
- WinCC Unified
- WinCC Runtime Professional
- Groups within external sources
- Technological objects
- Software units

The same rules apply as for the TIA Portal Openness API (see System Manual, [3](#)); e.g., assigning names when importing or exporting.

NOTE

If you need more, we provide a **topic page "TIA Portal Openness"** [4\ Links and literature](#) at the Siemens Industry Online Support with further application examples.

3.2 Model and hierarchy

The script language has a hierarchical structure. Compiling, for example, a controller (PLC) requires that a connection to TIA Portal be established, a project be open, and the controller be selected.

Figure 3-1

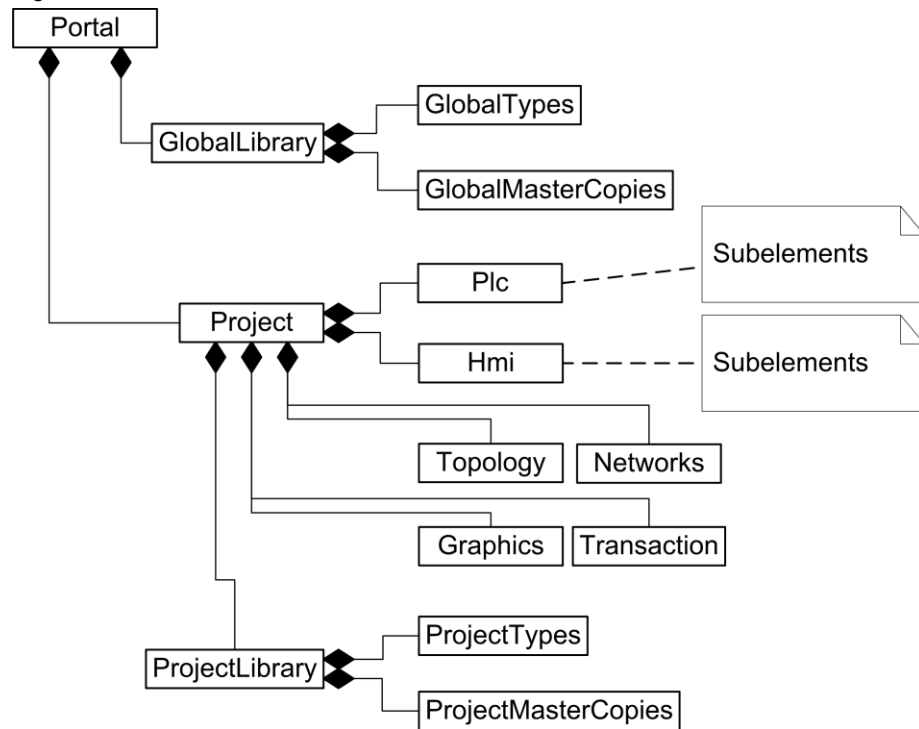


Figure 3-2

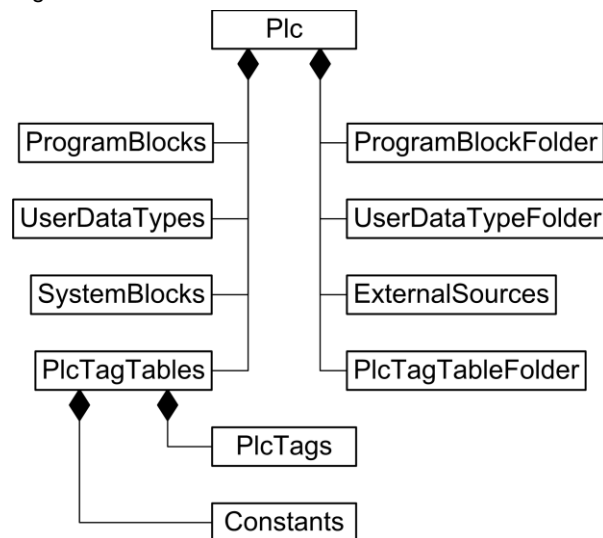
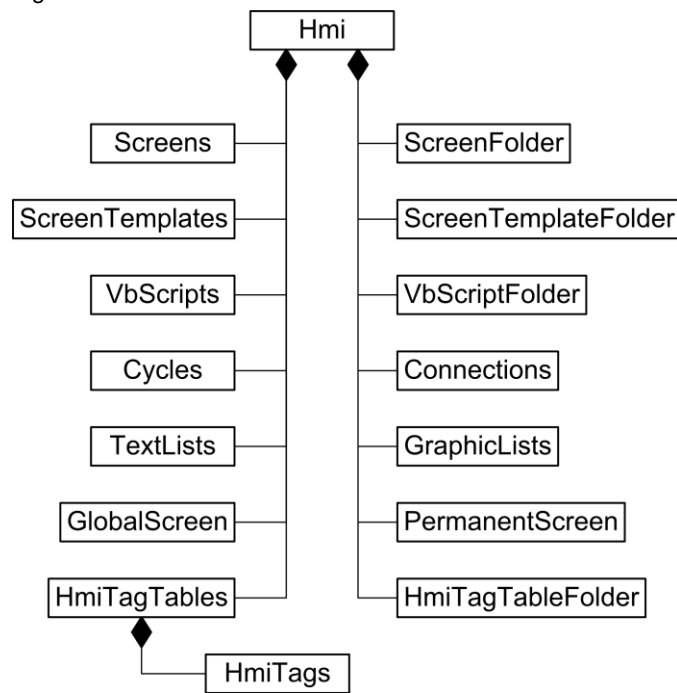


Figure 3-3



3.3 Syntax for the command set

The following notation is defined for the documentation of the commands.

Table 3-1

Symbol	Term	Meaning
[]	Brackets	This is an optional parameter. It can be omitted.
	Vertical bar	These are options where you have to choose one.
...	Three consecutive dots	The parameter before this can be repeated. Repetitions are separated by a whitespace.

Example 1

The following documentation

```
open Portal [WithoutUserInterface|WithUserInterface]
```

means that you can omit the optional parameter or choose between the two values. You can use one of the variants in the script:

```
open Portal
open Portal WithoutUserInterface
open Portal WithUserInterface
```

Example 2

The following documentation

```
select Graphics ["name" ...]
```

means that you can specify none, one, or more names. The names serve as filters. If no names are specified, all objects will be selected. You can use one of the variants in the script:

```
select Graphics
select Graphics "Graphic_1"
select Graphics "Graphic_1" "Graphic_2" "Graphic_3"
```

3.4 Option parameters

If you do not specify the option parameters, defaults will be used.

Example:

When opening TIA Portal using `open Portal` without specifying the "TiaPortalMode" option parameter, the default, `WithUserInterface`, will automatically be used.

TiaPortalMode

Table 3-2

Option	Description
<code>WithoutUserInterface</code>	Starts TIA Portal without the user interface.
<code>WithUserInterface</code>	<i>Default</i> Starts TIA Portal with the user interface.

CompilerOptions

Table 3-3

Option	Description
<code>Software</code>	Compiles only the software.
<code>Hardware</code>	Compiles only the hardware configuration.
<code>SoftAndHardware</code>	<i>Default</i> Compiles the software and the hardware configuration.

ExportOptions

Table 3-4

Option	Description
<code>None</code>	Exports only modified data or data that differs from the default and all the data required for the import.
<code>WithDefaults</code>	Like "None". Additionally exports the defaults.
<code>WithReadOnly</code>	Like "None". Additionally exports read-only values.
<code>WithDefaultsAndReadOnly</code>	<i>Default</i> Combines the "WithDefaults" and "WithReadOnly" options.

ImportOptions

Table 3-5

Option	Description
<code>None</code>	<i>Imports</i> without overwriting. An error is thrown.
<code>Override</code>	<i>Default</i> Imports with automatic overwriting.

3.5 Output

```
print "text"
```

Outputs the specified text.

```
print "name" ...
```

Outputs the specified names.

```
print "path"
```

Outputs the specified file path.

```
print /group/
```

Outputs the specified TIA Portal group.

```
print "number"
```

Outputs the specified version number.

3.5.1 Advanced output

You can output additional information, for example, to add to the execution progress when executing commands that generate no output.

The following data can be output:

- Variable
- Text
- Group
- Version (version number)
- File path
- Name (item name)
- Option parameter

```
print variable|text|group|version|file path|name|default
```

Outputs the specified content.

Example:

```
print "additional information"  
print <projectPath>
```

3.6 TIA Portal

```
open Portal [WithoutUserInterface|WithUserInterface]
```

Opens a new, empty TIA Portal instance.

```
connect Portal "projectPath"
```

Connects to an already running TIA Portal instance where the specified project (or local session) is open. The project path must at least contain the file extension (e. g. only ".ap17" or ".als17") but can also be a project name with file extension or an absolute path. The project must already be upgraded for connecting.

```
disconnect Portal
```

Disconnects the connection to the opened or connected TIA Portal.

3.6.1 Global libraries (user libraries)

These commands require an opened or connected TIA Portal.

```
open GlobalLibrary "libraryPath"
```

Opens the specified global library in read-only mode from the specified file path.

```
close GlobalLibrary
```

Closes the opened global library.

```
select GlobalMasterCopies ["name" ...] [/group/]
```

Selects master copies from the global library based on the specified names or GUIDs and/or group. If no parameters are given, all external sources will be selected.

```
print GlobalMasterCopies
```

Outputs the selected master copies of the global library.

```
copy GlobalMasterCopies  
ProgramBlockFolder|UserDataTypesFolder|PlcTagTableFolder [/group/]
```

Copies the selected master copies of the global library to the specified PLC target and the specified group. If the target exists, the object to copy will be renamed by the name rules of TIA Portal and inserted. This command is not supported in TIA Portal Openness V13 SP1.

```
copy GlobalMasterCopies  
ScreenFolder|ScreenTemplateFolder|VbScriptFolder|HmiTagTableFolder  
[/group/]
```

Copies the selected master copies of the global library to the specified HMI target and the specified group. If the target exists, the object to copy will be renamed by the name rules of TIA Portal and inserted. This command is not supported in TIA Portal Openness V13 SP1.

3 Command set

```
copy GlobalMasterCopies ProjectLibraryFolder [/group/]
```

Copies the selected master copies of the global library to the project library and the specified group. This command is not supported in TIA Portal Openness V13 SP1.

```
select GlobalTypes ["name" ...] [/group/]
```

Selects types from the global library based on the specified names and/or group. If no parameters are given, all types will be selected.

```
print GlobalTypes
```

Outputs the selected types of the global library.

```
instantiate GlobalTypes Screen|VbScript ["number"] [/group/]
```

Instantiates the selected types of the global library as the specified HMI type. If an instance exists the origin group will be kept.

```
instantiate GlobalTypes ProgramBlock|UserDataTypes ["number"]  
[/group/]
```

Instantiates the selected types of the global library as the specified PLC type. If an instance exists the origin group will be kept.

3.7 Project

These commands require an opened or connected TIA Portal.

```
open Project "projectPath"
```

Opens the specified project (or local session since V17). The project (e.g. *.ap17) will be upgraded automatically. The local session (e.g. *.als17) must already be upgraded for opening.

```
save Project
```

Saves the opened project (or local session).

```
close Project
```

Closes the opened project (or local session). No automatic saving!

3.7.1 Project library

These commands require an opened project.

```
update ProjectLibrary
```

Updates the project library based on the previously opened global library. For information about opening a global library, see the "Global libraries" chapter. Unused type versions are kept. If an instance exists the origin group will be kept.

```
select ProjectMasterCopies ["name" ...] [/group/]
```

Selects master copies from the project library based on the specified names and/or group. If no parameters are given, all master copies will be selected.

```
print ProjectMasterCopies
```

Outputs the selected master copies of the project library.

```
copy ProjectMasterCopies  
ProgramBlockFolder|UserDataTypesFolder|PlcTagTableFolder [/group/]
```

Copies the selected master copies of the project library to the specified PLC target and the specified group. If the target exists the object to copy will be renamed by the name rules of TIA Portal and inserted.

```
copy ProjectMasterCopies  
ScreenFolder|ScreenTemplateFolder|VbScriptFolder|HmiTagTableFolder  
[/group/]
```

Copies the selected master copies of the project library to the specified HMI target and the specified group. If the target exists the object to copy will be renamed by the name rules of TIA Portal and inserted.

3 Command set

```
copy ProjectMasterCopies ProjectLibraryFolder [/group/]
```

Copies the selected master copies of the project library to the project library and the specified group.

```
delete ProjectMasterCopies
```

Deletes the selected master copies of the project library.

```
select ProjectTypes ["name" ...] [/group/]
```

Selects types from the project library based on the specified names or GUIDs and/or group. If no parameters are given, all types will be selected.

```
print ProjectTypes
```

Outputs the selected types of the project library.

```
instantiate ProjectTypes Screen|VbScript|ProgramBlock|UserDataTypes ["number"] [/group/]
```

Instantiates the selected types of the project library as the specified HMI type. If an instance exists, the origin group will be kept.

```
instantiate ProjectTypes ProgramBlock|UserDataTypes ["number"] [/group/]
```

Instantiates the selected types of the project library as the specified PLC type. If an instance exists, the origin group will be kept.

```
delete ProjectTypes
```

Deletes the selected types from the project library. Deleting is canceled if other instances or master copies have to be deleted.

3.7.2 Project graphics

These commands require an opened project.

```
import Graphics "filePath" [None|Override]
```

Imports a new project graphic from the specified file path or all XML files (without subfolders) from the specified folder path.

```
select Graphics ["name" ...]
```

Selects project graphics based on the specified names. If no parameters are given, all project graphics will be selected.

```
print Graphics
```

Outputs the selected project graphics.

```
export Graphics "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected project graphics to the specified directory. If a corresponding subfolder (graphic files) exists, it will be deleted! If the target files already exist before exporting, they will be deleted.

```
delete Graphics
```

Deletes the selected project graphics.

3.7.3 Networks

These commands require an opened project.

```
open Networks
```

Opens the network view in TIA Portal.

3.7.4 Topology

These commands require an opened project.

```
open Topology
```

Opens the topology view in TIA Portal.

3.7.5 Transaction mode (exclusive access)

These commands require an opened project.

Note

A transaction assures exclusive access in TIA Portal to actions of the OpennessScripter.

As long as the transaction is active, TIA Portal cannot be used manually. This transaction mode is indicated by a dialog. During an active transaction, project data is only changed temporarily. You need to apply the temporary changes using the appropriate command.

```
start Transaction
```

Starts transaction mode in TIA Portal. Before each start of a transaction, all devices must have the "Offline" status.

```
commit Transaction
```

Ends transaction mode and applies all changes.

```
rollback Transaction
```

Ends transaction mode and undoes all changes.

3.8 PLC

These commands require an opened project.

```
select Plc "name"
```

Selects a PLC based on the specified name.

```
export Plc "directoryPath"
```

Exports a hierarchical list of all items of the selected PLC as text file to the specified directory. If the target file already exists before exporting, it will be deleted.

```
compile Plc [Software|Hardware|SoftAndHardware]
```

Compiles the selected PLC with the specified options. For F-CPU's only the option `Software` is possible until V16. No safety password may be set (all versions)!

```
open Plc
```

Opens the device view in TIA Portal for the selected PLC.

```
identify Plc
```

Queries the selected PLC's online status and outputs it.

```
compare Plc  
ProjectLibrary|GlobalLibrary|GlobalMasterCopies|ProjectMasterCopies|  
Online
```

Compares the selected PLC to the specified counterpart. The global library, the global master copy or the master copy from the project library must be selected before this step. For information about opening a global library, see the "Global libraries" chapter.

```
connect Plc
```

Goes online with the selected PLC. See also chapter "Known problems".

```
disconnect Plc
```

Goes offline with the selected PLC.

```
update Plc ProjectLibrary
```

Updates the selected PLC based on the project library. Unused type versions are kept. If an instance exists the origin group will be kept.

```
update Plc GlobalLibrary
```

Updates the selected PLC based on the previously opened global library. For information about opening a global library, see the "Global libraries" chapter. Unused type versions are kept. If an instance exists the origin group will be kept.

3.8.1 User data types and system data types of a PLC

These commands require a selected PLC.

```
create UserDataFolder /group/
```

Creates a new group based on the specified structure.

```
delete UserDataFolder /group/
```

Deletes a group based on the specified structure.

```
import UserDataTypes "filePath" [None|Override] [/group/]
```

Imports a new user data type from the specified file path or all XML files (without subfolders) from the specified folder path. Since TIA Portal V15.1 software unit attributes, missing referenced objects and structural changes are ignored during import, so that the import sequence is no longer relevant.

```
select UserDataTypes ["name" ...] [/group/]
```

Selects user data types and system data types based on the specified names and/or group. If no parameters are given, all user data types and system data types will be selected.

```
print UserDataTypes
```

Outputs the selected user data types and system data types.

```
export UserDataTypes "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected user data types to the specified directory. Only consistent (compiled) data types are exportable. It is not possible to export system data types, therefore they will be skipped automatically. If the target files already exist before exporting, they will be deleted.

```
delete UserDataTypes
```

Deletes the selected user data types and system data types.

```
compile UserDataTypes
```

Compiles the selected user data types and system data types.

```
sourcegen UserDataTypes "directoryPath"
```

Generates source files from the selected user data types and saves them to the specified directory. It is not possible to generate source files from system data types, therefore they will be skipped automatically. If the target files already exist before generating the sources, they will be deleted.

```
open UserDataTypes
```

Opens the selected user data types and system data types in the TIA Portal Editor.

3.8.2 External sources of a PLC

These commands require a selected PLC.

```
import ExternalSources "filePath"
```

Imports a new external source (only AWL/STL, SCL, DB and UDT) from the specified file path or all files (without subfolders) from the specified folder path.

```
select ExternalSources ["name" ...]
```

Selects external sources based on the specified names. If no parameters are given, all external sources will be selected.

```
print ExternalSources
```

Outputs the selected external sources.

```
blockgen ExternalSources
```

Generates appropriate blocks/user data types from the selected external sources. If a same-named block/user data type exists it will be automatically overwritten. If the external source contains errors, a block or user data type is still created and remains.

```
delete ExternalSources
```

Deletes the selected external sources.

3.8.3 Program blocks and data blocks of a PLC

These commands require a selected PLC.

```
create ProgramBlockFolder /group/
```

Creates a new group based on the specified structure.

```
delete ProgramBlockFolder /group/
```

Deletes a group based on the specified structure.

```
import ProgramBlocks "filePath" [None|Override] [/group/]
```

Imports a new program block (only FB, FC and OB in the languages AWL/STL, FUP/FBD, KOP/LAD and GRAPH) or a new data block (DB) from the specified file path or all XML files (without subfolders) from the specified folder path. Since TIA Portal V15 program blocks can be imported in the language SCL, too. Since TIA Portal V15.1 software unit attributes, missing referenced objects and structural changes are ignored during import, so that the import sequence is no longer relevant.

```
select ProgramBlocks ["name" ...] [/group/]
```

Selects program blocks and data blocks based on the specified names and/or group. If no parameters are given, all program blocks and data blocks will be selected.

```
print ProgramBlocks
```

Outputs the selected program blocks and data blocks.

```
export ProgramBlocks "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected program blocks (only FB, FC and OB in the languages AWL/STL, FUP/FBD, KOP/LAD, GRAPH and SCL) and data blocks (DB) to the specified directory. Only consistent (compiled) blocks are exportable. For SCL blocks in TIA Portal lower V15, only the block interface is exported (see `sourcegen`). It is not possible to export ProDiag blocks, therefore they will be skipped automatically. If the target files already exist before exporting, they will be deleted.

```
delete ProgramBlocks
```

Deletes the selected program blocks and data blocks.

```
compile ProgramBlocks
```

Compiles the selected program blocks and data blocks.

```
sourcegen ProgramBlocks "directoryPath"
```

Generates source files (interface and, if applicable, code) from the selected program blocks (only AWL/STL and SCL) and data blocks (DB) and saves them to the specified directory. It is not possible to generate source files from ProDiag blocks, therefore they will be skipped automatically. If the target files already exist before generating the sources, they will be deleted.

```
open ProgramBlocks
```

Opens the selected program blocks and data blocks in the TIA Portal Editor.

3.8.4 System blocks of a PLC

These commands require a selected PLC.

```
import SystemBlocks "filePath" [None|Override] [/group/]
```

Imports a new system block (only FB, FC, OB and DB in the languages AWL/STL, FUP/FBD, KOP/LAD and GRAPH) from the specified file path or all XML files (without subfolders) from the specified folder path. Since TIA Portal V15 system blocks can be imported in the language SCL, too. Since TIA Portal V15.1 software unit attributes, missing referenced objects and structural changes are ignored during import, so that the import sequence is no longer relevant.

```
select SystemBlocks ["name" ...] [/group/]
```

Selects system blocks based on the specified names and/or group. If no parameters are given, all system blocks will be selected.

```
print SystemBlocks
```

Outputs the selected system blocks.

```
export SystemBlocks "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected system blocks (only FB, FC, OB and DB in the languages AWL/STL, FUP/FBD, KOP/LAD, GRAPH and SCL) to the specified directory. Only consistent (compiled) blocks are exportable. For SCL blocks in TIA Portal lower V15, only the block interface is exported (see [sourcegen](#)). It is not possible to export system blocks generated by TIA Portal, therefore they will be skipped automatically. If the target files already exist before exporting, they will be deleted.

```
delete SystemBlocks
```

Deletes the selected system blocks.

```
compile SystemBlocks
```

Compiles the selected system blocks.

```
sourcegen SystemBlocks "directoryPath"
```

Generates source files (interface and, if applicable, code) from the selected system blocks (only AWL/STL, SCL and DB) and saves them to the specified directory. It is not possible to generate source files from system blocks generated by TIA Portal, therefore they will be skipped automatically. If the target files already exist before generating the sources, they will be deleted.

```
open SystemBlocks
```

Opens the selected system blocks in the TIA Portal Editor.

3.8.5 F-blocks of a PLC

These commands require a selected PLC.

In the script, F-blocks are treated like program blocks, data blocks and system blocks (see previous chapters).

In TIA Portal lower V15.1, F-blocks can only be exported but not imported.

F-system blocks cannot be imported/exported.

3.8.6 Blocks with know-how protection (KHP) of a PLC

These commands require a selected PLC.

In the script, know-how protected blocks are treated like program blocks, data blocks and system blocks (see previous chapters).

Only the block interface would be exported for know-how protected blocks, afterwards an import is no longer possible. There are no know-how protected XML files. Source files cannot be generated.

Therefore, know-how protected blocks are skipped automatically during export and source generation.

3.8.7 Tags and user constants of a PLC

These commands require a selected PLC.

```
import Constants "tagTableName" "filePath" [None|Override]
```

Imports a new user constant from the specified file path or all XML files (without subfolders) from the specified folder path into a specific tag table.

```
select Constants ["name" ...]
```

Selects user constants based on the specified names. If no parameters are given, all user constants will be selected.

```
print Constants
```

Outputs the selected user constants.

```
export Constants "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected user constants to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete Constants
```

Deletes the selected user constants.

```
import PlcTags "tagTableName" "filePath" [None|Override]
```

Imports a new tag from the specified file path or all XML files (without subfolders) from the specified folder path into a specific tag table.

```
select PlcTags ["name" ...]
```

Selects tags based on the specified names. If no parameters are given, all tags will be selected.

```
print PlcTags
```

Outputs the selected tags.

3 Command set

```
export PlcTags "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected tags to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete PlcTags
```

Deletes the selected tags.

```
create PlcTagTableFolder /group/
```

Creates a new group based on the specified structure.

```
delete PlcTagTableFolder /group/
```

Deletes a group based on the specified structure.

```
import PlcTagTables "filePath" [None|Override] [/group/]
```

Imports a new tag table from the specified file path or all XML files (without subfolders) from the specified folder path.

```
select PlcTagTables ["name" ...] [/group/]
```

Selects tag tables based on the specified names and/or group. If no parameters are given, all tag tables will be selected.

```
print PlcTagTables
```

Outputs the selected tag tables.

```
export PlcTagTables "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected tag tables to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete PlcTagTables
```

Deletes the selected tag tables.

3.9 HMI

These commands require an opened project.

```
select Hmi "name"
```

Selects an HMI based on the specified name.

```
export Hmi "directoryPath"
```

Exports a hierarchical list of all items of the selected HMI as text file to the specified directory. If the target file already exists before exporting, it will be deleted.

```
compile Hmi [Software|Hardware|SoftAndHardware]
```

Compiles the selected HMI with the specified options. TIA Portal Openness currently does not support the compile of HMI hardware.

```
open Hmi
```

Opens the device view in TIA Portal for the selected HMI.

```
update Hmi ProjectLibrary
```

Updates the selected HMI based on the project library. Unused type versions are kept. If an instance exists, the origin group will be kept.

```
update Hmi GlobalLibrary
```

Updates the selected HMI based on the previously opened global library. For information about opening a global library, see the "Global libraries" chapter. Unused type versions are kept. If an instance exists, the origin group will be kept.

3.9.1 Screens of an HMI

These commands require a selected HMI.

```
create ScreenFolder /group/
```

Creates a new group based on the specified structure.

```
delete ScreenFolder /group/
```

Deletes a group based on the specified structure.

```
import Screens "filePath" [None|Override] [/group/]
```

Imports a new screen from the specified file path or all XML files (without subfolders) from the specified folder path. The screen numbers must be unique and must not yet exist, else TIA Portal errors will occur.

```
select Screens ["name" ...] [/group/]
```

Selects screens based on the specified names and/or group. If no parameters are given, all screens will be selected.

```
print Screens
```

Outputs the selected screens.

```
export Screens "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected screens to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete Screens
```

Deletes the selected screens.

3.9.2 Screen templates of an HMI

These commands require a selected HMI.

```
create ScreenTemplateFolder /group/
```

Creates a new group based on the specified structure.

```
delete ScreenTemplateFolder /group/
```

Deletes a group based on the specified structure.

```
import ScreenTemplates "filePath" [None|Override] [/group/]
```

Imports a new screen template from the specified file path or all XML files (without subfolders) from the specified folder path.

```
select ScreenTemplates ["name" ...] [/group/]
```

Selects screen templates based on the specified names and/or group. If no parameters are given, all screen templates will be selected.

```
print ScreenTemplates
```

Outputs the selected screen templates.

```
export ScreenTemplates "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected screen templates to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete ScreenTemplates
```

Deletes the selected screen templates.

3.9.3 Global screen of an HMI

These commands require a selected HMI.

```
import GlobalScreen "filePath" [None|Override]
```

Imports the global screen from the specified file path.

```
export GlobalScreen "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the global screen to the specified directory. If the target files already exist before exporting, they will be deleted. The exported file will always have the name "globalscreen.xml".

3.9.4 Graphic lists of an HMI

These commands require a selected HMI.

```
import GraphicLists "filePath" [None|Override]
```

Imports a new graphic list from the specified file path or all XML files (without subfolders) from the specified folder path.

```
select GraphicLists ["name" ...]
```

Selects graphic lists based on the specified names. If no parameters are given, all graphic lists will be selected.

```
print GraphicLists
```

Outputs the selected graphic lists.

```
export GraphicLists "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected graphic lists to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete GraphicLists
```

Deletes the selected graphic lists.

3.9.5 Permanent screen of an HMI

These commands require a selected HMI.

```
import PermanentScreen "filePath" [None|Override]
```

Imports the permanent screen from the specified file path.

```
export PermanentScreen "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the permanent screen to the specified directory. If the target files already exist before exporting, they will be deleted. The exported file will always have the name "permanentscreen.xml".

3.9.6 Text lists of an HMI

These commands require a selected HMI.

```
import TextLists "filePath" [None|Override]
```

Imports a new text list from the specified file path or all XML files (without subfolders) from the specified folder path.

```
select TextLists ["name" ...]
```

Selects text lists based on the specified names. If no parameters are given, all text lists will be selected.

```
print TextLists
```

Outputs the selected text lists.

```
export TextLists "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected text lists to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete TextLists
```

Deletes the selected text lists.

3.9.7 Tags of an HMI

These commands require a selected HMI.

```
import HmiTags "tagTableName" "filePath" [None|Override]
```

Imports a new tag from the specified file path or all XML files (without subfolders) from the specified folder path into a specific tag table.

```
select HmiTags ["name" ...]
```

Selects tags based on the specified names. If no parameters are given, all tags will be selected.

```
print HmiTags
```

Outputs the selected tags.

```
export HmiTags "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected tags to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete HmiTags
```

Deletes the selected tags.

```
create HmiTagTableFolder /group/
```

Creates a new group based on the specified structure.

```
delete HmiTagTableFolder /group/
```

Deletes a group based on the specified structure.

```
import HmiTagTables "filePath" [None|Override] [/group/]
```

Imports a new tag table from the specified file path or all XML files (without subfolders) from the specified folder path.

```
select HmiTagTables ["name" ...] [/group/]
```

Selects tag tables based on the specified names and/or group. If no parameters are given, all tag tables will be selected.

```
print HmiTagTables
```

Outputs the selected tag tables.

```
export HmiTagTables "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected tag tables to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete HmiTagTables
```

Deletes the selected tag tables.

3.9.8 VB scripts of an HMI

These commands require a selected HMI.

```
create VbScriptFolder /group/
```

Creates a new group based on the specified structure.

```
delete VbScriptFolder /group/
```

Deletes a group based on the specified structure.

3 Command set

```
import VbScripts "filePath" [None|Override] [/group/]
```

Imports a new VB script from the specified file path or all XML files (without subfolders) from the specified folder path.

```
select VbScripts ["name" ...] [/group/]
```

Selects VB scripts based on the specified names and/or group. If no parameters are given, all VB scripts will be selected.

```
print VbScripts
```

Outputs the selected VB scripts.

```
export VbScripts "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected VB scripts to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete VbScripts
```

Outputs the selected VB scripts.

3.9.9 Connections of an HMI

These commands require a selected HMI.

```
import Connections "filePath" [None|Override]
```

Imports a new connection from the specified file path or all XML files (without subfolders) from the specified folder path. The connection must be unique across the project and must not yet exist, else TIA Portal errors will occur.

```
select Connections ["name" ...]
```

Selects connections based on the specified names. If no parameters are given, all connections will be selected.

```
print Connections
```

Outputs the selected connections.

```
export Connections "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected connections to the specified directory. Exporting integrated connections is not possible. If the target files already exist before exporting, they will be deleted.

```
delete Connections
```

Deletes the selected connections.

3.9.10 Cycles of an HMI

These commands require a selected HMI.

```
import Cycles "filePath" [None|Override]
```

Imports a new cycle from the specified file path or all XML files (without subfolders) from the specified folder path.

```
select Cycles ["name" ...]
```

Selects cycles based on the specified names. If no parameters are given, all cycles will be selected.

```
print Cycles
```

Outputs the selected cycles.

```
export Cycles "directoryPath"  
[None|WithDefaults|WithReadOnly|WithDefaultsAndReadOnly]
```

Exports the selected cycles to the specified directory. If the target files already exist before exporting, they will be deleted.

```
delete Cycles
```

Deletes the selected cycles.

3.10 Data types

There are different data types that are automatically managed by the meta language.

Table 3-6

Data type	Description	Example
Text ("text")	Any string in straight quotation marks	"Hello World"
Group ("group")	Folder name from TIA Portal. Notation starts and ends with a slash	/group1/sub1/
Version ("version")	Version number of a library type	"1.3.2"
File path ("filePath")	Folder or file path in the file system (e.g. single or mass import)	"D:\sources\ or "D:\sources\block.xml"
Folder path ("directoryPath")	Folder path in the file system	"D:\sources\"
Name ("name")	Item name	"Programblock_1"
Option parameter	Option passed to a command as a parameter.	WithUserInterface

NOTE

In file paths only backslashes ('\') are allowed directory separator chars. Slashes (/) are no valid chars in a file path.

3.11 Escape characters

Quotation mark

If you need a quotation mark " within a text it will be escaped with a backslash \".

Example:

The text "The variable "projectPath" is not assigned." has to be outputted.

The script looks like this:

```
print "The variable \"projectPath\" is not assigned."
```

Slash

If you need a slash / within a group it will be escaped with a backslash \/.

Example:

The program block group "Sensors / Actors" must be created.

The script looks like this:

```
create ProgramBlockFolder /Sensors \/ Actors/
```

3.12 Variables

When writing the script, you can use variables (placeholders) to define, for example, a file path at the beginning of the script and use it multiple times later in the script. Enclose variables in angle brackets `< >`. You can freely choose a variable name. You can save the following data in variables:

- Text
- Group
- Version (version number)
- File path
- Name (item name)
- Option parameter

The value of a variable is set as follows:

```
<variable name> = text|group|version|file path|name|default
```

Example with use:

```
<projectPath> = "D:\demo\project\project.ap17"
open Project <projectPath>
```

3.13 Comments

You can provide the scripts with comments to increase readability and document individual tasks. Prefix comments with a diamond. These are line-by-line comments:

```
# comment text
```

You can also add a comment to the end of the line. Example:

```
open Project <projectPath> # opens the project
```

3.14 Condition and negation

You can apply conditions to the variables to have different script branches executed depending on the variable content.

Notation:

```
if [no] <variable name>
    # code that is executed if the condition is met
[else
    # code that is executed if the condition is not met
]
end if
```

Both negation using the word `no` and the `else` branch are optional. If the variable is not assigned or contains the value `"0"` or `"false"`, the condition is not met; in all other cases, the condition is met.

Example:

```
if no <projectPath>
    print "The project path was not assigned."
end if
```

4 Run

4.1 Breakpoints

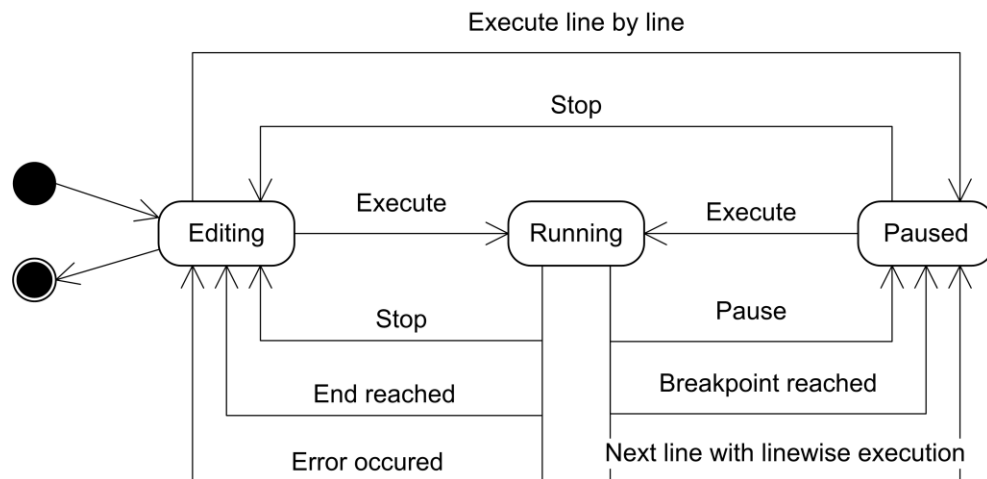
Breakpoints are marks within the script. They allow you to halt execution at the marked line. Execution does not continue until you manually continue it. Enabled breakpoints are visualized by a circle filled with dark red color to the left of the line number.

When using breakpoints, please note the following:

- Breakpoints have no effect in blank lines and comment lines.
- Breakpoints are not saved.
- You cannot enable breakpoints during execution.
- Breakpoints are tied to the line number and do not move automatically.
- You cannot enable breakpoints by clicking. You have to select the line and enable/disable the breakpoints using "Run" > "Toggle breakpoint" or the "F9" function key.

To obtain more information during the script check or execution, set the output filter to "Debug" or activate the echo option ("Run" menu).

Figure 4-1



4.2 TIA Portal target version

In the OpennessScripter, you can set the TIA Portal target version for running a script in both the user interface and in the console.

Only the following TIA Portal Openness API versions are supported:

- V13 SP1 / V13 SP2
- V14 / V14 SP1
- V15
- V15.1
- V16
- V17
- V18 and higher

From version V18 a compatibility mode for future versions is available, these are automatically recognized and displayed during installation process, when the longtime-stable TIA Portal V18 Openness DLL is found.

When starting the OpennessScripter, the latest TIA Portal version installed on the Windows operating system is selected by default. If no version is installed V18 will be selected.

When opening a script, a check is performed whether the script accesses V13 SP1 projects, V13 SP2 projects, V14 projects, V14 SP1 projects, V15 projects, V15.1 projects, V16 projects, or V17 projects. The strings `.ap13` or `.al13` for V13 SP1 and V13 SP2 or the strings `.ap14`, `.als14`, or `.al14` for V14 and V14 SP1 or the strings `.ap15`, `.als15`, or `.al15` for V15 or the strings `.ap15_1`, `.als15_1`, or `.al15_1` for V15.1 or the strings `.ap16`, `.als16`, or `.al16` for V16 or the strings `.ap17`, `.als17`, or `.al17` for V17 are checked for occurrence.

- If only V13 strings occur and TIA Portal V13 SP1 is installed, V13 SP1 will be selected as target version, else V13 SP2.
- If only V14 strings occur and TIA Portal V14 is installed, V14 will be selected as target version, else V14 SP1.
- If only V15 strings occur, V15 will be selected as target version.
- If only V15.1 strings occur, V15.1 will be selected as target version.
- If only V16 strings occur, V16 will be selected as target version.
- If only V17 strings occur, V17 will be selected as target version.
- If only V18 strings occur, V18 will be selected as target version.
- If strings from several versions occur, the previous selection will be kept.

The stated automated selection is only executed while opening a script. If you modify the script you must change the target version by own, especially if you start with an empty template.

The script commands are independent of the TIA Portal target version. The scripts are compatible.

When running the script, a check is performed to determine whether the TIA Portal target version is installed on the Windows operating system. If the target version was not found, script execution is aborted with the following message: "TIA Portal Openness is not installed." You can select a different target version and restart script execution.

During execution, a script can only use one target version. You can start the script one after the other for different target versions.

5 Tips

- If you have problems during script check or execution you should set the output filter to "Debug" or activate the "Echo" option ("Run" menu) to see all information.
- If you cannot save a file you should check the write protection and write access for this script file.
- External changes in a script file are not monitored. If the file is modified, e.g. by an external text editor, it will be overwritten by the OpennessScripter when saving.
- The scripts are saved and read in UTF-8 encoding.
- There are no commands possible besides the specified command set. You cannot extend the command set. If you need additional tasks you should "switch" the process: A batch file (see "Console" operation mode) calls several scripts one after the other. Between the scripts you can use other tools, e.g. for manipulating XML files.
- You don't have to check a script manually before execution. When executing the check will be performed automatically.
- The installation path can be read out from the Windows Registry:
`HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Siemens\Automation\OpennessScripter`

6 Known problems

- Relative file paths and directory paths are currently not supported. Please use absolute paths with drive letter at the beginning.
- If you use variables (placeholders) in commands with several parameters, you currently will have to specify all parameters for this command.
Example: You want to pass a variable for names when selecting program blocks. But the command `select ProgramBlocks` has 2 optional parameters. Therefore, you must set the second parameter explicitly in this case (slash for main group) and cannot use a variable:
`select ProgramBlocks <blockNames> /`
The issue is that the variable `<blockNames>` can be a list. For parameters, which are no lists, you can use several variables, e.g.:
`export ProgramBlocks <path> <option>`
- For going online with a PLC via `connect Plc` you currently must pre-configure the online connection in TIA Portal, by going online manually once.
Then you can do an online-offline comparison:
`compare Plc Online`
A download or upload is not possible in the OpennessScripter, because the configuration is too complex for a script. (Lots of options in the load dialogs must be handled.)
- If you don't enter the OpennessScripter in the TIA Portal Openness whitelist but allow the access at each execution manually it may result in unexpected behavior in certain circumstances.
- The option `Override` for import commands will only work if the object to be overwritten is stored in the same group or for tags and constants in the same tag table.
- If no TIA Portal Openness installation could be determined, please perform a repair installation of TIA Portal.

7 Appendix

7.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:

support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:

siemens.com/SupportRequest

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:

support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:

support.industry.siemens.com/cs/ww/en/sc/2067

7.2 Industry Mall



The Siemens Industry Mall is the platform on which the entire Siemens Industry product portfolio is accessible. From the selection of products to the order and the delivery tracking, the Industry Mall enables the complete purchasing processing – directly and independently of time and location:

mall.industry.siemens.com

7.3 Links and literature

Table 7-1

	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page of the entry https://support.industry.siemens.com/cs/ww/en/view/109742322
\3\	TIA Portal Openness System Manual (API for automation of engineering workflows) https://support.industry.siemens.com/cs/ww/en/view/109798533 (V17)
\4\	Topic page: TIA Portal Openness: Automation of engineering workflows https://support.industry.siemens.com/cs/ww/en/view/109792902

7.4 Change documentation

Table 7-2

Version	Date	Modifications
V1.0.0	08/2016	<ul style="list-style-type: none"> First version
V1.1.0	09/2016	<ul style="list-style-type: none"> Chapter on breakpoints Settings "Local users and groups" Run "Activate echo" File "Open script in the default text editor" Console "Variable assignments" Name or GUID for types in libraries Parameter for commands omitted Import of file path and folder path Selecting elements at the group level Command Set – Scope Help "Online support" Run, toolbar and console "TIA Portal target version" Chapter on the TIA Portal target version

Version	Date	Modifications
V1.1.1	01/2017	<ul style="list-style-type: none"> • Entry ID • Command set chapter structured • Supported languages for generating source files and exporting/importing • Command set – F-blocks • Chapter on known problems
V1.1.2	03/2017	<ul style="list-style-type: none"> • Examples for TIA Portal V14 • Notes on transaction mode • Notes on commands with global libraries
V1.1.2	04/2017	<ul style="list-style-type: none"> • Revision of the document
V1.1.2	05/2017	<ul style="list-style-type: none"> • Notes on integrated connections • Notes on not upgraded projects • Command set – Scope • Support for TIA Portal V14 SP1 and V13 SP2 and notes • Omitted parameters for commands • Notes on user libraries • Notes on user constants
V1.1.2	06/2017	<ul style="list-style-type: none"> • Copyediting and translation
V1.1.3	01/2018	<ul style="list-style-type: none"> • Support for TIA Portal V15 and notes • Notes on system data types • Notes on data blocks • Overwriting while generating from external sources • Right-click on a script file for executing • Chapter on escape characters • Chapter on integration in TIA Portal
V1.1.4	10/2018	<ul style="list-style-type: none"> • Support for TIA Portal V15.1 and notes • Command set – F-blocks • Command set – Know-how protected blocks • Note on software units
V1.1.5	02/2020	<ul style="list-style-type: none"> • Support for TIA Portal V16 and notes • Notes on exporting and generating sources • Notes on connecting to an opened project • Notes on selecting objects • Notes on calling from VBS and VBA • Note on file paths • Notes on compiling of HMI • Note on master copies in global libraries
V1.1.6	10/2021	<ul style="list-style-type: none"> • Support for TIA Portal V17 and notes • Support for PLC and HMI as part of a PC station • Support for PLC as part of an NCU/CNC (Sinumerik) • Open local sessions (TIA Portal project server / Multiuser) • Save log as XML file • Notes on opening/upgrading projects • Notes on calling from VB.NET • Notes on know-how protection, ProDiag, system blocks and system data types • Notes on importing and generating blocks and data types

Version	Date	Modifications
		<ul style="list-style-type: none">Revision of the scope (chapter 3.1), syntax for the command set (chapter 3.3), tips (chapter 5), and known problems (chapter 6)
V1.1.7	02/2023	<ul style="list-style-type: none">Support for TIA Portal V18 + higher and notes