HMI-Projecting style guide based on SIMATIC WinCC Unified.

TIA Portal

https://support.industry.siemens.com/cs/ww/en/view/8131867

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

1 Introduction

During configuring a SIMATIC WinCC, an HMI project engineer has the task to develop the application as readable and structured as possible. Each HMI project engineer applies their own strategy to realizing this task, e.g., naming or commenting scripts, screens and faceplates. Different HMI project engineers use different philosophies, therefore very different applications exist, which often can only be interpreted by the respective creator.

1.1 Goal

The rules and recommendations described in the following chapters will help you create a uniform HMI project which is maintainable and reusable. In case of multiple HMI project engineers working on the same project, it is recommended to apply a project wide terminology as well as an agreed upon projecting style. This allows you to detect and avoid errors at an early stage.

For the sake of maintainability and readability it is required, to follow a certain format. Optical effects have only a limited impact on the quality of software. It is more important to define rules, which support the developer as follows:

- Avoiding typos and inadvertent mistakes, which the compiler then misinterprets
  Objective: The compiler shall recognize as many errors as possible.
- Supporting the HMI project engineer diagnosing programming errors
  Objective: The Identifier indicates problems early.
- Standardization of applications and libraries
  Objective: The training shall be made easy and the reusability of the programcode / project elements shall be increased.
- Easy maintenance and simplification of further developments

Note The described rules and recommendations in this document are consistent and do not interfere with each other. The Numbers of the rules are based on the numbers of the rules in the STEP7-Styleguide. Same rules have same numbers. New rules have new numbers.

1.2 Advantages of a uniform projecting

- Uniform and consistent style
- Easy to read and understand
- Easy maintenance and increased reusability
- Easy and fast error recognition and correction
- Efficient cooperation of multiple programmers

1.3 Applicability

This document is applicable for projects and libraries in TIA Portal, which are configured with WinCC Unified.

1.4 Scope

This document doesn’t contain descriptions of:

- WinCC Unified configuration with TIA Portal
- Commissioning of WinCC Unified applications

Basic knowledge and experience in the mentioned topics above are the prerequisite to correctly interpreting and applying the given rules and recommendations.

1.5 Rule violations and other regulations

In customer projects the applicable regulations, customer or branch specific standards as well as technological regulations (e.g. Safety, Motion, …) are to be followed and take precedence over the style guide or parts thereof.

When combining both, customer regulations with regulations within this style guide, special care must be taken to maintain integrity and consistency of the rules.

A violation of any of the regulations must be justified and documented appropriately in the user program.

The customer provided rules and regulations must be documented appropriately.

2 Definitions

2.1 Rules/ Recommendations

The regulations in this document are either recommendations or rules:

- **Rules** are binding definitions and must be followed. They are essential for a reusable and performant programming. In exceptional cases rules may be violated. This must be justified and documented.

- **Recommendations** are regulations, which support the uniformity of the program code and serve as support and documentation. Recommendations should be followed in general. However, there are exceptions when such a recommendation may not be followed. Reasons for this may be a better efficiency or better readability.

2.2 Enumerating rules

For a unique rule identification, within categories rules and recommendations are identified with a prefix (2 characters) and are enumerated (3 digits).

In case a regulation is canceled its number will not be reassigned. In case more regulations become necessary, you may use the numbers between 901 and 999.

Table 2-1

Prefix	Category
ES	Engineering System: programming environment
GL	Globalization
NF	Nomenclature and formatting:
RU	Reusability
AL	Allocation: Referencing of objects
SE	Security
DA	Design and architecture:
PE	Performance

2.3 Performance

The performance of a WinCC application is defined by the loading time of the screen or the reaction time to events.

When mentioning a performance disadvantage, this means that the loading and reaction time can be reduced by using project engineering rules and an efficient way of configuration the WinCC application.

2.4 Identifier/ Naming

It is important to differentiate an identifier and a name. The name is part of the identifier, which describes the meaning of an identifier.

The identifier is assembled out of:

- prefix
- name
- suffix

2.5 Abbreviations

The following abbreviations are being used throughout this document:

Table 2-2

Abbreviation	Type
FP	Faceplate
GT	Graphic Type

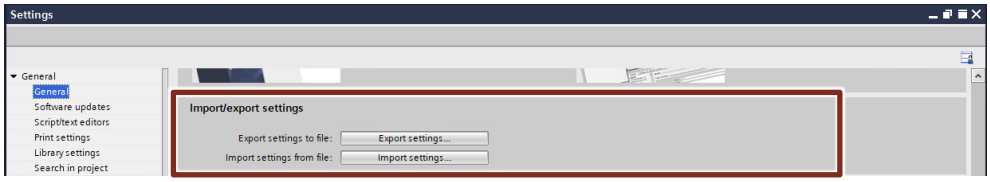
3 Settings in TIA Portal

In this chapter rules and recommendations for the initial setup of the programming environment are described.

Note

The rules and recommendations for the settings in TIA Portal listed here are stored in the TIA Portal Settings File (tps file). You can find the tps file as a separate download in this entry. To apply the settings, you can import the tps file into TIA Portal.

Figure 3-1



ES001 Rule: User Interface Language "English"

The User Interface Language is to be set to "English". In this way all newly created projects have the editing and reference language as well as all the system constants are set to English.

Justification: To have all system constants available in the same language the user interface language must be set to a common uniform language.

4 Globalization

This chapter describes the rules and recommendations for a global cooperation.

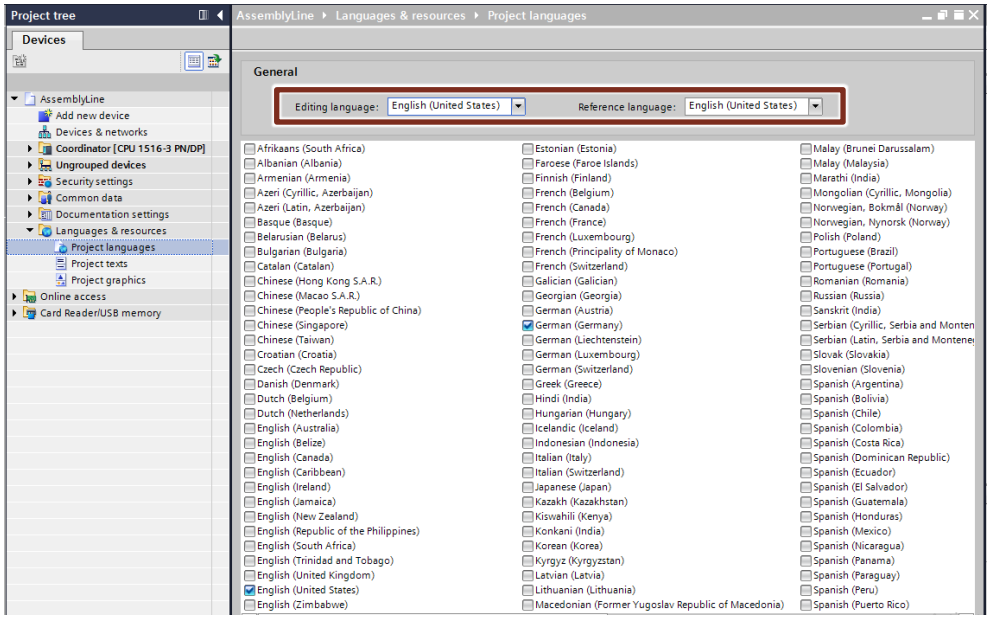
GL001 Rule: Use consistent language

The language used must be consistent in the PLC as well as in the HMI programming. This means, that English texts can only be found in the English language setting.

GL002 Rule: Set editing and reference language to "English (US)"

If not otherwise demanded by the customer, the language must be set to "English (United States)" for both the Editing and Reference language. The complete program including all comments must be created in English.

Figure 4-1

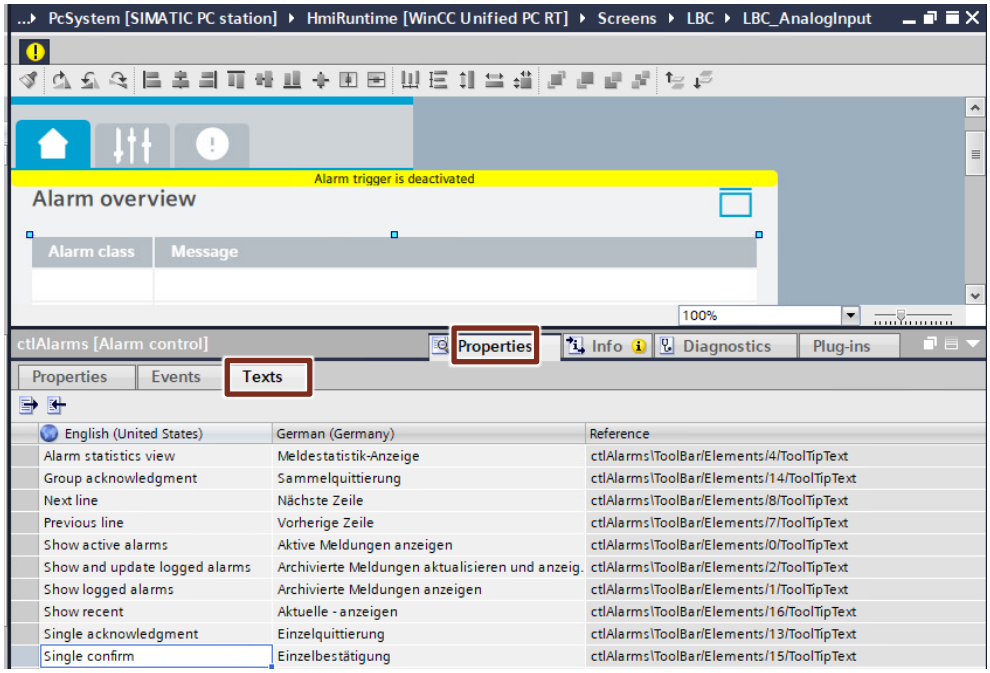


GL003 Rule: Supply texts in all project languages

All project texts must be provided at least in English as well as in all other used project languages.

Note Texts and their translations can be easily managed in the Inspector window under "Properties" in the tab "Texts".

Figure 4-2



5 Nomenclature and Formatting

This chapter describes rules and recommendations for naming and writing of programs and comments.

NF001 Rule: Unique and consistent English identifiers

The name of an identifier (Blocks, variable, etc.) must be in English language (English – United States). The name describes the meaning of the identifier in the context of the source code and therefore promotes an understanding of the functionality and usage of the identifier.

- The chosen spelling of an identifier must be maintained in all blocks and PLC data types and shall be as short as possible.
- The same functional meaning of an identifier causes the same naming for the identifier. This applies to capitalization as well.
- Identifier names can be assembled out of multiple words; the order of the words has to be the same as in the spoken language.
- Is the identifier a name for an Array, then the name uses the plural. Non-countable nouns remain in their singular form ("data", "information", "content", "management").

Justification: A quick overview about the program and its inputs and outputs will be provided.

Note The names assigned by TIA Portal are place holders and need to be replaced by yours.

NF002 Rule: Use meaningful comments and properties

Comment and property fields shall be used and filled with meaningful comments and information. This includes

- Library types
- Comments in scripts
- PLC/ HMI data types and their variables
- HMI tag tables, HMI tags
- Text and graphic lists
- Library properties

Justification: Using this the user gets the most information and guidance in using the components, e.g. through tooltips.

NF003 Rule: Document developer information

Each screen and each faceplate contain a description header in the global definition area of dynamizations. The most important information about the faceplate development must be stored in the header.

User relevant information must be provided in the faceplate properties. This information is available to the user and can be read without opening the faceplate.

The following template for such description header contains the elements from the faceplate properties as well as the development relevant information, which don't need to be copied into the properties.

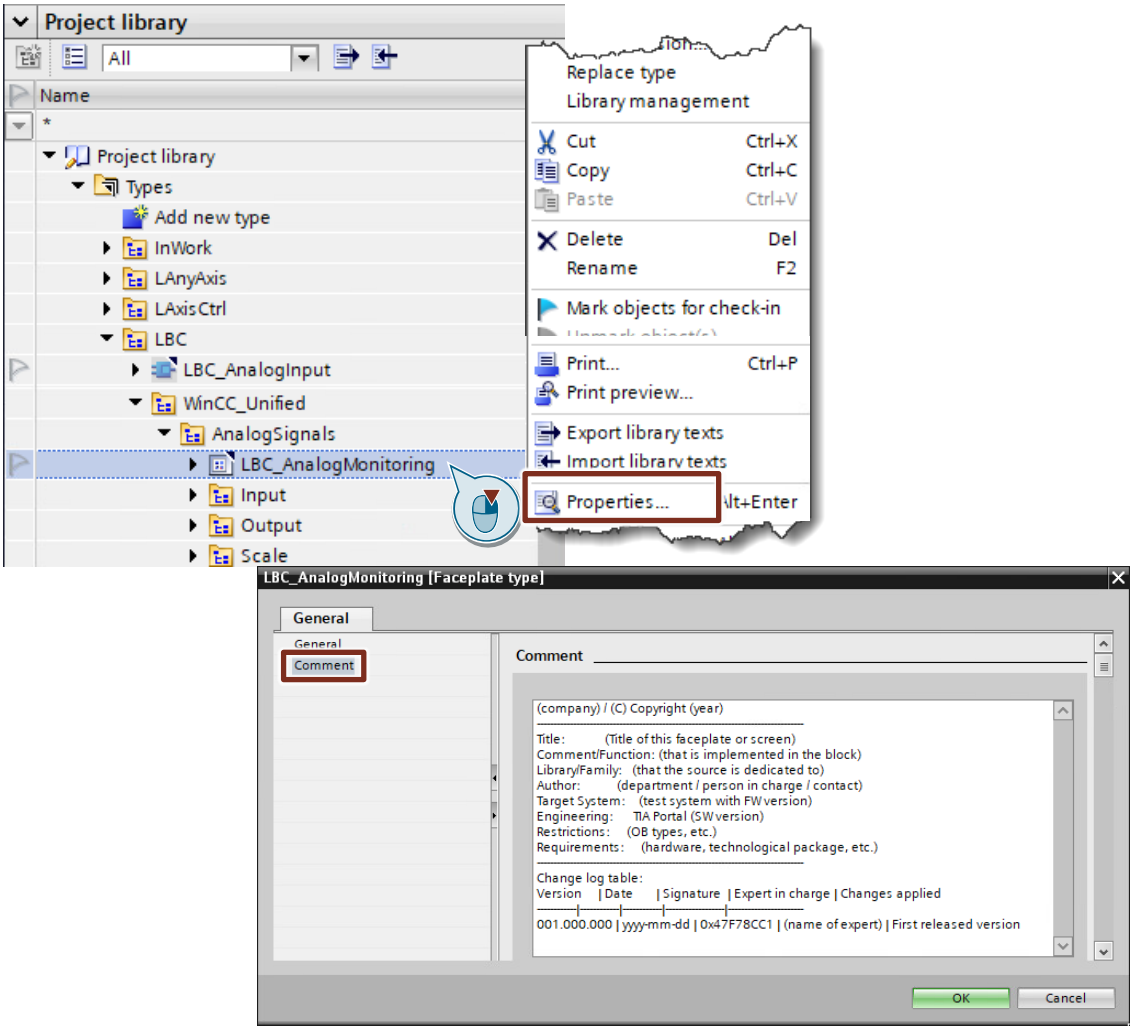
The description contains the following items:

- Company name / (C) Copyright (Year). All rights reserved. (Optional)
- Title/ Faceplate description
- Description of the functionality
- Name of the library (Optional)
- Department/ Author/ Contact
- Target system – WinCC Unified with firmware version (e.g. MTP1500 Unified Comfort V17.0.0.1)
- Engineering – TIA Portal with version at time of creation/ modification
- Limitations for usage
- Requirements (e.g. additional hardware)
- Additional information (Optional)
- Change log with version, date, author and change description. (Optional)

Template for a description header in the faceplate property

```
(company) / (C) Copyright (year)
--------------------------------------------------------------------------------
Title:            (Title of this faceplate or screen)
Comment/Function: (that is implemented in the block)
Library/Family:   (that the source is dedicated to)
Author:           (department / person in charge / contact)
Target System:    (test system with FW version)
Engineering:      TIA Portal (SW version)
Restrictions:     (OB types, etc.)
Requirements:     (hardware, technological package, etc.)
--------------------------------------------------------------------------------
Change log table:
Version     | Date       | Signature  | Expert in charge | Changes applied
------------|------------|------------|------------------|-----------------------
001.000.000 | yyyy-mm-dd | 0x47F78CC1 | (name of expert) | First released version
```

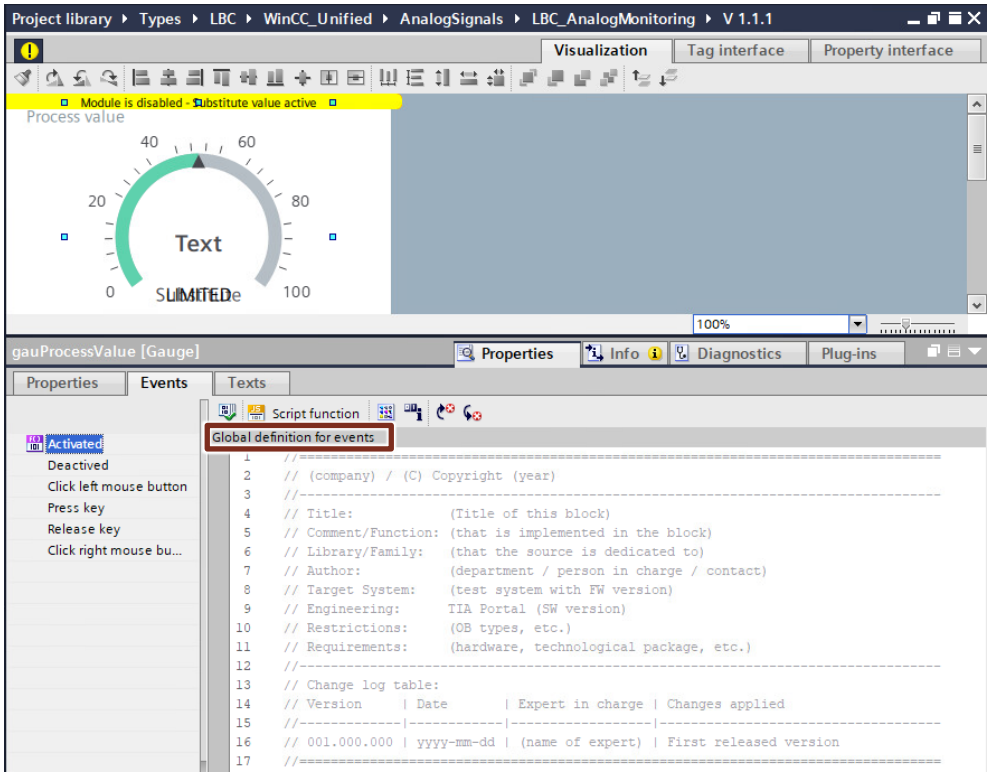
Figure 5-1



Template for a description header in JavaScript

```
REGION Description header
  //=============================================================================================
  // (company) / (C) Copyright (year)
  //---------------------------------------------------------------------------------------------
  // Title:            (Title of this block)
  // Comment/Function: (that is implemented in the block)
  // Library/Family:   (that the source is dedicated to)
  // Author:           (department / person in charge / contact)
  // Target System:    (test system with FW version)
  // Engineering:      TIA Portal (SW version)
  // Restrictions:     (OB types, etc.)
  // Requirements:     (hardware, technological package, etc.)
  //---------------------------------------------------------------------------------------------
  // Change log table:
  // Version     | Date       | Expert in charge | Changes applied
  //-------------|------------|------------------|-----------------------------------
  // 001.000.000 | yyyy-mm-dd | (name of expert) | First released version
  //=============================================================================================
END_REGION
```

Figure 5-2



NF004 Rule: Comply with prefixes and structure for libraries

The identifier of a library has the prefix "L" and does not exceed a maximum length of 8 characters.

The identifier of a library starts with the prefix "L" and is followed by a maximum of 7 characters as the name (e.g. LGF, LCom). "L" stand for the word Library. After the library identifier an underscore (_) is used as a separator (e.g. LGF_).

The maximum length of an identifier for libraries (incl. prefix) is limited to 8 characters.

Justification: This limitation serves the purpose of assigning compact and short names.

Every element in the library carries the prefix.

All types and master copies contained in the library get the identifier of the library.

An element, which only demonstrates the use of the library, is not a library element in the sense of a standardized library, it is rather an example and therefore doesn't necessarily carry the library prefix.

Justification: With the prefix included in the identifier, naming collisions are being avoided.

Table 5-1

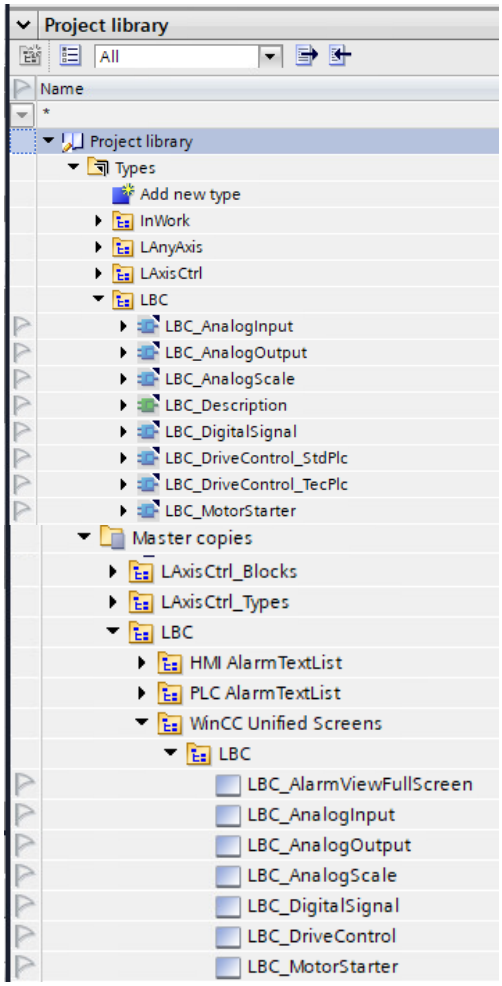
Type	Identifier according to style guide
Library, main folder of the library	LExample
PLC data type	LExample_type<Name>
HMI tag table	LExample_<Name>
HMI text and graphic lists	LExample_<Name>
Faceplate	LExample_<Name>
Graphic type	LExample_<Name>

Grouping within the library

All master copies and types shall be placed in a subfolder inside the library, which carries the library identifier as its folder name.

Justification: The subfolder supports the project harmonization efforts and allows grouping of multiple libraries within a project.

Figure 5-3



Note

This rule contains only information for the nomenclature of library elements. Additionally, it is recommend to follow the recommendations given and explained in detail in the library guideline available:

https://support.industry.siemens.com/cs/ww/en/view/109747503

NF005 Rule: Use PascalCasing for objects

Identifiers for TIA Portal objects are written using PascalCasing, such as:

- HMI tag tables
- HMI text and graphic lists
- Graphics
- Dynamic SVGs
- Screens
- Faceplates
- Global Scripts
- Scheduled tasks
- Parameter set types
- Logs
- Layers
- Custom Controls

The following rules apply for PascalCasing:

- The first character is a capital letter
- If an identifier is assembled out of multiple words, then the first character of each word is a capital letter.
- There are no separators (e.g. hyphen or underscore) used for the optical separation of the identifier. For structuring and specialization purposes the sparingly use of the underscore (not more than three) is permitted.

Table 5-2

Sparingly	Excessive
GetAxisData_PosAxis GetAxisData_SpeedAxis GetAxisData_SyncAxis	Get_Axis_Data_Pos_Axis

NF006 Rule: Use camelCasing for code elements

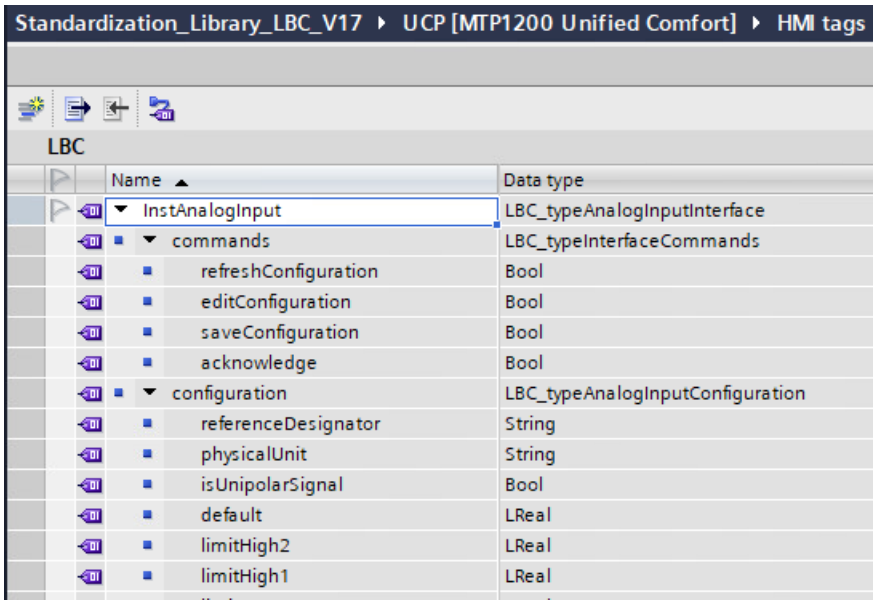
Identifiers for code elements are written using camelCasing, such as

- Variables (Exception: variables fetched from the PLC by drag&drop)
- PLC/ HMI data types
- Structures ("STRUCT")
- PLC/ HMI tags
- Parameter
- Internal HMI tags
- DataSet tags
- Faceplate Container
- Faceplate Tag Interface
- Faceplate Property Interface
- Graphics
- Graphic Container
- Library elements in screens and faceplates

The following rules apply for the camelCasing:

- The first character is a non-capitalized (small) letter
- If an identifier is assembled out of multiple words, then the first character of the following word is capitalized.
- The use of separators (e.g. hyphen or underscores) for the optical separation is not permitted.

Figure 5-4



NF007 Rule: Use prefixes

HMI system-wide variables with prefix "srv"

To distinguish system-wide HMI variables from external and session local variables, the prefixes defined in Table 5-3 shall be used.

Justification: this measure makes it easier for the HMI project engineer to distinguish between system-wide, session local and external variables

Instance data with prefix "inst"

Faceplate containers represent instances of faceplates and are named with the prefix "inst". Graphic containers represent instances of graphics and are named with the prefix "gfxInst".

Justification: With the prefixes it can be easier recognized if it is a typed element.

Graphic types and HMI data type with prefix "type"

A graphic type or HMI data type gets the prefix "type". The individual elements inside the HMI data type do not get a prefix.

Table 5-3

Prefix	Type
No prefix	External variables (Naming according to PLC style guide) Access possible from the outside
srv	HMI system-wide variables ➔ srvStationName
inst	Faceplate and graphic container ➔ instMotorInflow ➔ gfxInstPipeInflow
type	Graphic type and HMI data type Only the HMI data type gets the prefix, the elements do not get a prefix ➔ typeDiagnostic

Prefix of Screen- / Faceplate- Objects

Each object in the screen or faceplate has a prefix. See the following tables

Justification: In JavaScript code, it is easier to identify what kind of elements are accessed via script.

Overview table "Basic objects”

Table 5-4

Prefix	Type
Text box	txt
Polygon	pg
Graphic view	gfx
Circular arc	cirln
Line	ln
Elliptical arc	ellln
Rectangle	rec
Circle segment	cirs
Circle	cir
Ellipsen segment	ells
Ellipse	ell
Polyline	pgln

Overview "table Elements”

Table 5-5

Prefix	Type
IO field	io
Gauge	gau
Symbolic IO field	txtio
Clock	clk
List box	lb
Check box	cb
Button	btn
Radio button	rb
Switch	swi
Touch area	ta
Bar	bar
Slider	sld

Overview table "Controls”

Table 5-6

Prefix	Type
Alarm control	ctlAlarmView
System diagnostics control	ctlSysDiag
Trend control	ctlTrendView
Process control	ctlTabel
Web control	ctlBrowser
Media player	ctlMediaPlayer
Screen window	sw
Faceplate container	inst
Parameter set control	ctlParameterSet
Graphics IO field	gfxio
Date/Time field	dtio
User view	ctlUserView
Watch table	ctlWatchTable
Control	ctl
Groups	grp

Prefix for "Property Interface” of Faceplates
Each data type in the faceplate property interface has a prefix.

Justification: this facilitates the wiring from the outside.

Table 5-7

Prefix	Type
Color	color
Resource list	list
Configuration string	cfgStr
64-bi integer	int
Authorization	auth
Boolean	bool
Floating point number	real
Unsigned 64-bit integer	uint

NF009 Rule: Limit the character set for identifiers

For all object and code identifiers the Latin alphabet (a-z, A-Z) and the Arabic numerals (0-9) as well as the underscore (_) are to be used exclusively.

Table 5-8

Correct naming	Incorrect naming
tempMaxLength	temporary Variable 1

NF010 Recommendation: Limit the length of identifiers

The overall length of an identifier incl. prefix, suffix or library identifier shall not exceed 24 characters.

Justification: To avoid typing errors in the script, it is useful not to use long identifiers.

NF011 Recommendation: Use one abbreviation per identifier only

Multiple abbreviations shall not be used directly one after the other to realize the best possible readability. To reduce the amount of used characters in an identifier recommended abbreviations are listed in Table 5-9.

This table only contains the most commonly used abbreviations. The spelling of the abbreviations must follow the rules for the particular use and needs to be adopted accordingly (capitalization).

Table 5-9

Abbrev.	Type
Min	Minimum
Max	Maximum
Act	Actual, Current
Next	Next value
Prev	Previous value
Avg	Average
Sum	Total sum
Diff	Difference
Cnt	Count
Len	Length
Pos	Position
Ris	Rising edge
Fal	Falling edge
Old	Old value (e. g. for edge detection)
Sim	Simulated
Dir	Direction
Err	Error
Warn	Warning
Cmd	Command
Addr	Address

6 Reusability

This chapter describes the rules and recommendations applicable to ensure the multiple use of program elements.

RU002 Rule: Version entirely with libraries

Assign versions entirely. This means, that every change in a faceplate must be documented and the assigned version must be maintained. Every change in the assigned version must be documented in their respective locations, such as the description header of a faceplate.

When using a library and block types the block version is managed by TIA Portal. In this case it is not necessary to manually maintain the version in the block properties. The change log remains untouched by this fact.

An upgrade of the library to the latest TIA Portal version does not require a change in the block and is therefore not a new version.

Version numbers and their use

The first released version always starts with 1.0.0 (refer to Table 6-1).

The first digit describes the left most number.

The third digit in software versioning indicates changes, which have no effect on function or documentation, such as pure bug fixing.

Extension to the functionality the second digit will be incremented, and the third digit reset.

With a new major release, containing new functionality and incompatible changes to the previous version, increase the first digit and resets the second and third digit.

Each digit has a valid range between 0 and 999.

FP = Faceplate

GF = Graphic Type

Table 6-1

Library	FP1	FP2	GF1	GF2	Comment
1.0.0	1.0.0	1.0.0	1.0.0		Released version
1.0.1	1.0.1	1.0.0	1.0.0		Bug fix in FP1
1.0.2	1.0.1	1.0.1	1.0.0		Optimization of FP2
1.1.0	1.1.0	1.0.1	1.0.0		Extension to FP1
1.2.0	1.2.0	1.0.1	1.0.0		Extension to FP1
2.0.0	2.0.0	1.0.1	2.0.0		New and possibly incompatible function in FP1 and GF1
2.0.1	2.0.0	1.0.2	2.0.0		Bug fix in FP2
2.1.0 / 3.0.0	2.0.0	1.0.2	2.0.0	1.0.0	New function in GF2/ possibly larger new release

Note

This rule contains only generic information about versioning. A detailed explanation about the automatic versioning of library elements is provided in the library guideline:

https://support.industry.siemens.com/cs/ww/en/view/109747503

RU003 Rule: Keep only released types in released projects

Finalized projects contain only typified library elements, which are not in status "in test":

- Blocks (only functions and function blocks)
- PLC data types
- Faceplates

RU007 Recommendation: Project independently from hardware

To guarantee compatibility between the different systems it is recommended to project hardware independent.

The smallest panel has 7 inches. Therefore, it is recommended not to make faceplates larger than 800 x 480 pixels.

RU008 Recommendation: Use templates

Using templates, you can achieve a uniform basis for all programmers. The functionality provided by the templates can be considered validated and reduce the development times dramatically.

Another positive aspect is the easier usability and the improved perception due to the uniform basis.

You can find more information about this in the application example "HMI design with the HMI Template Suite"
https://support.industry.siemens.com/cs/ww/en/view/91174767

RU009 Rule: Use Graphic types

Use of graphic types instead of non-typed graphics. Graphic lists are an exception.
Justification: Graphic types can be used in faceplates and screens.
Non-typed graphics can only be used on screens.

RU010 Rule: Follow the JavaScript Styleguide

To increase the readability and reusability of JavaScript code, the rules of the JavaScript Styleguide should be followed.

For more information about the JavaScript styleguide, see the following link.

https://support.industry.siemens.com/cs/ww/en/view/109758536

7 Design guidelines/ architecture

This chapter describes the applicable rules and recommendations for program design and program architecture.

DA001 Rule: Structure and group a project/ library

Split your program into logical units. The system provides several different means for this matter.

- Group related screens or faceplates belonging together into a group or folder

DA009 Rule: Keep used elements only

The released project must contain only elements, which are being used.

Examples for violations:

- Unused screens
- Unused variables
- Never executed program code
- Commented out code
- PLC variables that are never accessed

Note Productive code, which may be used at a later point in time depending on an option, is not affected by this.

8 Performance

In this chapter the rules and recommendations are described, which support the development of performant user programs.

PE017 Rule: Visibility "false” for dynamized visibility

Set visibility to "false” for all elements for which visibility has been dynamized, to reduce loading times at first screen load.

Exception:

- Columns in "Alarm control"
- "Elements" in "Function bar"/"Toolbar" or "Information bar"/"Status bar" (Affects various controls such as the "Alarm control” or "Trend control")

If Visible is set to "false" here, the element is not loaded into the runtime. A dynamization would not work as a result.

Justification: Elements with visibility set to "true” are rendered on screen load.

PE018 Rule: Don’t use cyclic triggers

Scripts should only be executed when needed to avoid unnecessary resource consumption.

9 Cheat sheet

Figure 9-1

Tag table	PLC tag	User constant
	lightBarrierLeft	**MAX_BELTS**
Prefix	--	--
Casing	camelCasing	UPPER_CASING

Programming styleguide based on SIMATIC WinCC Unified in TIA Portal

Object		Prefix	Casing
Project	**AssemblyLine**	--	PascalCasing
Library	**LCom**	"L"	PascalCasing
HMI-Variabletables; Text- and graphiclists; graphics; dynamic SVGs; screens; CWCs	**Valves**		PascalCasing
Graphic types	**typeValve**	type	camelCasing
system-wide HMI Variables	**serverStationNr**	server	camelCasing
Text box	**txtTitle**	txt	camelCasing
Polygon		pg	camelCasing
Graphic view		gfx	camelCasing
Cicular arc		cirln	camelCasing
Line		ln	camelCasing
Elliptical arc		ellln	camelCasing
Rectangle		rec	camelCasing
Circle		cir	camelCasing
Ellipse		ell	camelCasing
Polyline		pgln	camelCasing
IO field	**ioProducedPieces**	io	camelCasing
Button	**btnDeleteEntries**	btn	camelCasing

Figure 9-2

- Unique, meaningful identifiers in English
- Only the characters a-z, A-Z, 0-9 and _
- Maximum 24 characters per identifier
- Array: axesData [0..#MAX] of type…
- Library: Name max. 8 chars; prefix "LExample_"

Usual abbreviations (maximum one per identifier)

Min / Max	Minimum / Maximum
Act	Actual, Current
Next / Prev	Next / Previous value
Avg	Average
Sum	Total sum
Diff	Difference
Cnt	Count
Len	Length
Pos	Position
Ris / Fal	Rising / falling edge
Old	Old value
Sim	Simulated
Dir	Direction
Err / Warn	Error / Warning
Cmd	Command
Addr	Address

10 Appendix

10.1 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

10.2 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

10.3 Links and Literature

Table 10-1

	Topic
\1\	Siemens Industry Online Support http://support.industry.siemens.com/
\2\	Download page of this entry https://support.industry.siemens.com/cs/ww/en/view/81318674
\3\	Standardization guideline https://support.industry.siemens.com/cs/ww/en/view/109756737
\4\	Library guideline https://support.industry.siemens.com/cs/ww/en/view/109747503
\5\	Provide user defined documentation https://support.industry.siemens.com/cs/ww/en/view/109755202/114872699275
\6\	SIMATIC S7-1200/ S7-1500 Compare list for programming languages https://support.industry.siemens.com/cs/ww/en/view/86630375
\7\	Library of General Functions (LGF) for SIMATIC STEP 7 (TIA Portal) and SIMATIC S7-1200/ S7-1500 https://support.industry.siemens.com/cs/ww/en/view/109479728
\8\	SIMATIC WinCC Unified - Tips and Tricks for Scripting (JavaScript) https://support.industry.siemens.com/cs/ww/en/view/109758536

10.4 History

Table 10-2

Version	Date	Changes
V1.0	03/2023	First Release