SIEMENS

Application Example • 06/2016

Synchronizing recipes via a SIMATIC HMI Panel

Basic Panels, Comfort Panels, WinCC V13 SP1

https://support.industry.siemens.com/cs/ww/en/view/109736272

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete with regard to configuration, equipment or any contingencies. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for the correct operation of the described products. These Application Examples do not relieve you of the responsibility of safely and professionally using, installing, operating and servicing equipment. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time and without prior notice. If there are any deviations between the recommendations provided in this Application Example and other Siemens publications – e.g. Catalogs – the contents of the other documents shall have priority.

We do not accept any liability for the information contained in this document.
Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of fundamental contractual obligations ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.
Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens AG.

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 secure 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. With this in mind, Siemens' products and solutions are only part of such a concept.

It is the client's responsibility to prevent unauthorized access to his plants, systems, machines and networks. Systems, machines and components should only be connected with the company's network or the Internet, when and insofar as this is required and the appropriate protective measures (for example, use of firewalls and network segmentation) have been taken.

In addition, the recommendations by Siemens regarding the respective protective measures have to be observed. For more information on industrial security, visit http://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development in order to make them even more secure. Siemens explicitly recommends to carry out updates as soon as the respective updates are available and always only to use the current product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

In order to always be informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at http://www.siemens.com/industrialsecurity.

Table of Contents

1 Task

1.1 Overview

Introduction

Recipes are used in many production facilities.

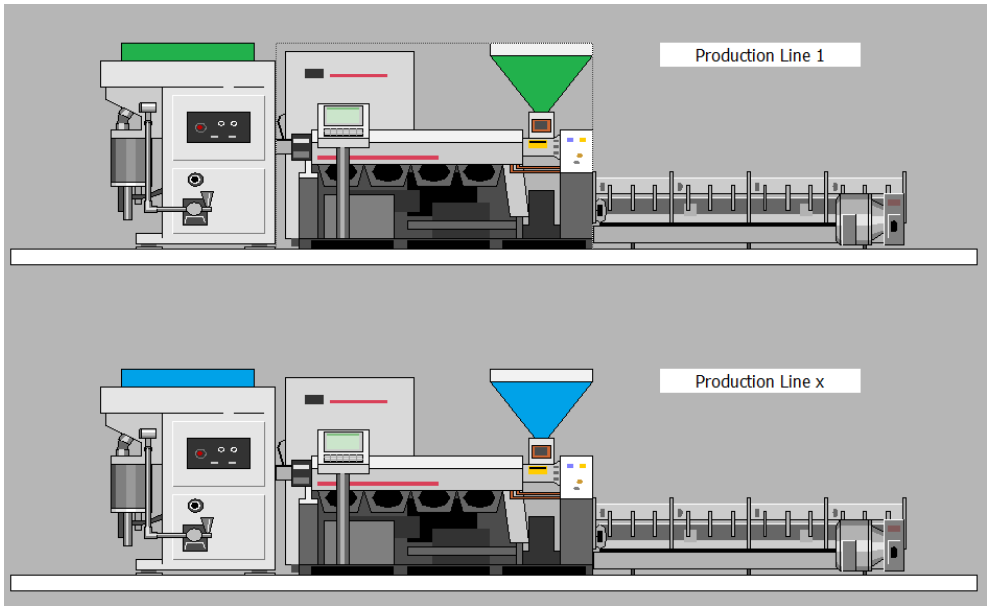
If in a plant, multiple production lines are operated, all of them using the same recipes, then changes to the recipes need to be done at all stations subsequently.

Overview of the automation job

The figure below provides an overview of the automation job.

There are several production lines which are identical in terms of their structure and scope of functions.

Figure 1-1



Every station is supposed to...

- work independently of each other
- be controlled separately via an HMI operating panel.
- use the same recipes and data records which, however, can be selected independently.

If at a station, the data records of a recipe are changed, the changes should be applied by all related stations.

1.2 Problem description

Synchronizing recipe data

The core task is to align (synchronize) changed data records on multiple panels.

Aligning the data records shall not be done manually, but automatically.

An automatic alignment is supposed to ensure that at all stations the same data records are stored.

Requirements of the automation job

Table 1-1

Problem description	Explanation
Easy handling.	
It should be possible to use all panels that support the “Recipe” system function.	
The changed data shall not be passed on from panel to panel via an USB stick.	
The changed data shall be passed on from panel to panel via an existing network connection.	
A centralized data storage of the recipe data shall be possible.	The recipe data shall be stored on and read from an “Office PC” within the network.
The time at which the data records are synchronized shall be freely selectable.	

2 Solution

2.1 Overview

There are several solutions to implement the job. Depending on the local conditions and the hardware you use, choose solution 1 or solution 2. As an extension, you can combine both solutions.

Selection help, Example 1

Example 1 can be used, if....

- You do not have any possibility to save the recipe data on a PC via the PROFINET network.
- You do not use panels that support scripts.
- It is sufficient that only **one** data record at a time can be cached.
- It is sufficient that only **one** data record at a time can be updated on the other panels. If several data records need to be adjusted, this has to be done one after another.

Selection help, Example 2

Example 2 can be used, if....

- You have the possibility to save the recipe data on a PC via the PROFINET network.
- The recipe data are to be managed centrally (processing data records via exported Excel files).
- You use panels that support scripts.
- You want to update all data records of a panel at once.

Comparison of both variants.

Summary of the properties of both examples.

Table 2-1

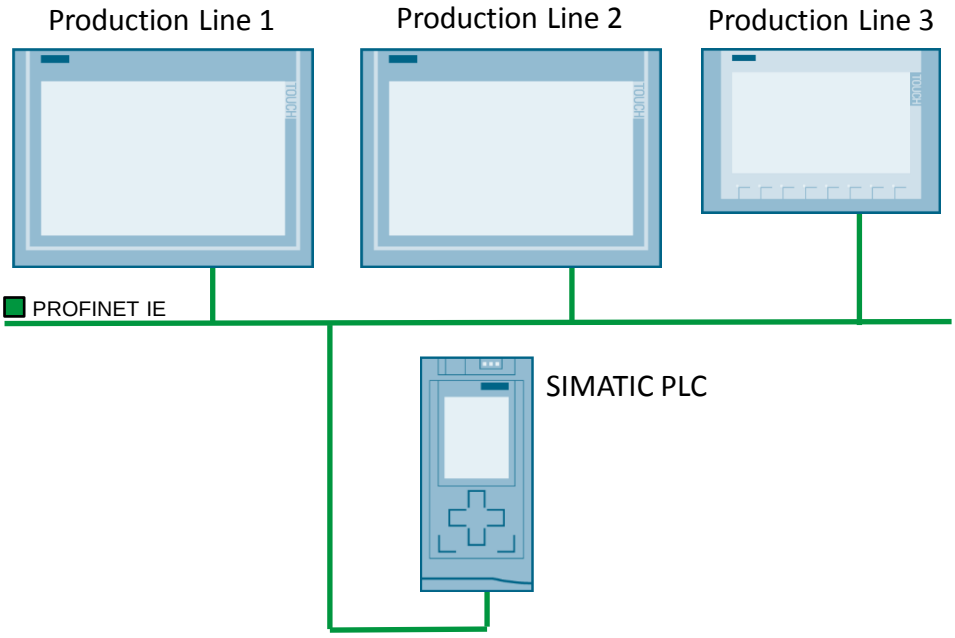
No.	Example 1	Example 2
1.	**Supported panels** All SIMATIC HMI panels that support the “Recipe” system function.	**Supported panels** All SIMATIC HMI panels that support the “Recipe” and the “**Scripts**” system function.
2.	**No** external PC required to save and back up the recipe data.	External PC **required** to save and back up the recipe data.
3.	Caching of only **one** data record possible.	Caching of **all** data records possible.
4.	Only **one** data record at a time can be cached and then transmitted to the other panels.	**All** data records at once can be read and then transmitted to **all** other panels.
5.	**No** external processing of recipe data possible.	External processing of recipe data via existing Excel file **possible**.

2.2 Application Examples

2.2.1 Application Example 1

The following figure schematically displays the key components of the solution:

Figure 2-1



Setup

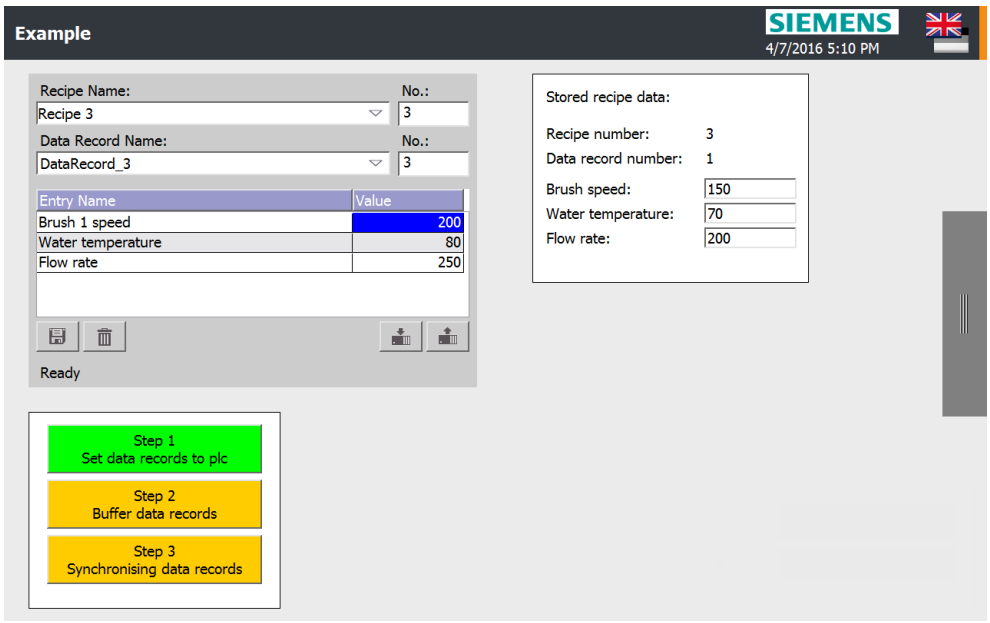
The attached configuration considers a plant with three production lines, each of them operated via a SIMATIC HMI operating panel.

- Two TP1200 Comfort Panel
- One KTP900 Basic Panel
- CPU 1500

All participants are connected to each other via PROFINET.

Description of core functionality, Example 1

Figure 2-2



The core task is to synchronize changed data records on multiple panels.

For this, a panel of a plant area is declared as “master”. Only on this “master panel” may changes be made to the data records of the recipes. All other panels (clients) synchronize their data records with this “master panel”.

Note

From a technical viewpoint, it would also be possible to allow changes to the data records from each station. In order to provide a better overview and transparency, this will be left out in this example.

For this, the data record of a recipe to be synchronized has to be selected at the “master panel” and cached in the CPU.

Subsequently, the cached data records can be transmitted to the other panels.

The values of the cached data record are maintained until a new value has been transmitted to the CPU.

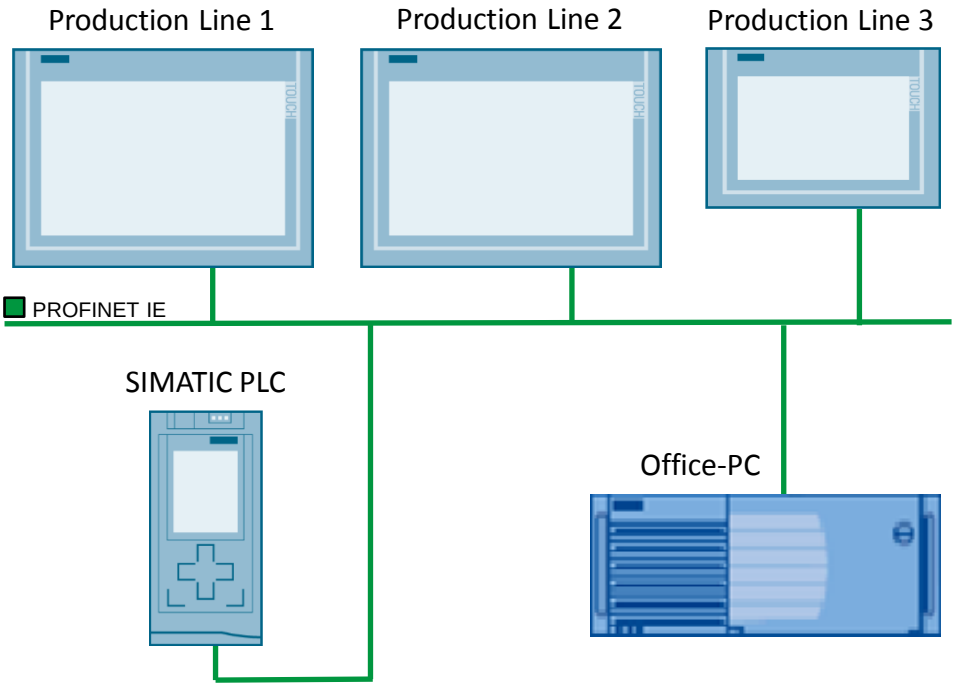
Note

Only one data record at a time can be selected and cached. If several data records are to be adjusted and transmitted to the other panels, it has to be done one after another.

This does not require a prior selection of a recipe or data record at the “client panels”.

2.2.2 Application Example 2

The following figure schematically displays the key components of the solution:
Figure 2-3



Setup

The attached configuration considers a plant with three production lines, each of them operated via a SIMATIC HMI operating panel.

- two TP1200 Comfort Panel
- one TP900 Comfort Panel
- S7-1500
- The recipe data are saved on an “Office PC”.

All participants are connected to each other via PROFINET.

Scope

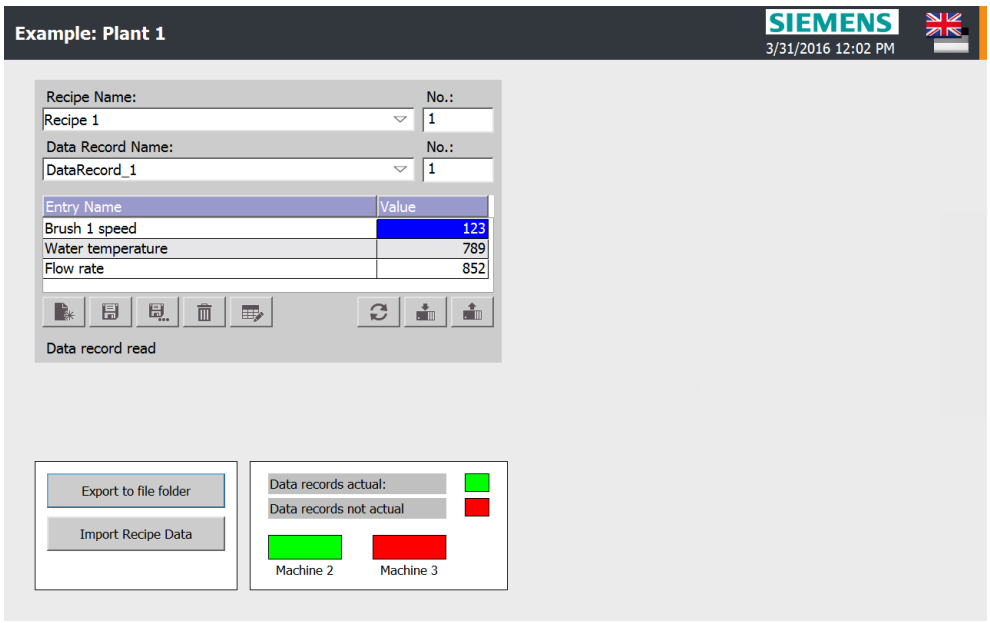
This application does not include a description of

- how to integrate a PC and a SIMATIC HMI operating panel into a network. For this, see the following entry “Integrating HMI operating panels into TCP/IP networks“ \3\.
- how to create a recipe.
- the hardware used.

Basic knowledge of these topics is required.

Description of core functionality, Example 2

Figure 2-4



The core functionality is to synchronize changed data records of a recipe on multiple panels.

For this, a panel of a plant area is declared as “master”. Only on this “master panel” may changes be made to the data records. All other panels (clients) synchronize their data records with this “master panel”.

Note

From a technical viewpoint, it would also be possible to allow changes to the data records from each station. In order to provide a better overview and transparency, this will be left out in this example.

Via the system function “ExportDataRecords”, all existing data records from the “master panel” are exported into an “Excel.csv” file. For the other panels to be able to access this Excel file, the Excel file needs to be saved on an “Office PC”.

Via the system function “ImportDataRecords”, the other panels (clients) access this “Office PC” and the exported Excel file directly via the network.

Notes

- All data records are usually exported. With regards to the example, all data records are imported at once via a button. It would also be possible to import singe data records.
- With “export or import” it is not required to first select a recipe or a data record at the “master panel” or at the “client panels”.

2.3 Hardware and software components

2.3.1 Validity

This application is valid for:

- STEP 7 as of V13 SP1
- WinCC Comfort V13 SP1 or higher

2.3.2 Components used

The application was created using the following components:

Hardware components

Table 2-2

Component	Qty	Article number	Note
SIMATIC HMI TP1200 COMFORT	2 ..	6AV2124-0MC01-0AX0	
SIMATIC HMI TP900 COMFORT	1 ..	6AV2124-0JC01-0AX0	
SIMATIC HMI, KTP900 BASIC	1 ..	6AV2123-2JB03-0AX0	
CPU 1516-3 PN/DP	1 ..	6ES7516-3AN01-0AB0	

Software components

Table 2-3

Component	Qty	Article number	Note
SIMATIC WinCC Comfort V13 SP1	1 ..	6AV2101-0AA03-0AA5	--
SIMATIC STEP 7 V13 SP1 Professional	1 ..	6ES7822-1AA03-0YA7	--

Example files and projects

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

Table 2-4

Component	Note
109736272_CODE_v10.zip	The zip file contains two configurations. • Solution via area pointer. • Solution via import/export.
109736272_DOKU_v10_d.pdf	This document.

3 Fundamentals

3.1 Recipes and data records

Recipes

The term “recipe” is a “general term”, summarizing various single elements from which a product emerges, depending on the combination.

Example:
Various brake discs are to be produced on a milling machine. The recipe is given the name “Brake dics_front wheel”.

In order to make sure that the milling machine knows,

- which drills to use => element “**type**”.
- which rotational speed is needed => element “**rotation speed**”.
- if cooling is needed => element “**cooling**”.

these and, if necessary, further elements are entered into the recipe.

For each element, a “plc tag” is stored.

Data record

The term data record describes a set of values which specify a special part of a product. In this case a very special brake disc type.

For example, the data record is given the name “Brake disc type1”, “...type2” etc.

Each data record automatically contains the elements previously set in the recipe.

Example:

- “Brake disc type1”
  - **Type** drill 1
  - **Rotation speed** 800 rpm
  - **Cooling** yes
- “Brake disc type2”
  - Type drill 8
  - Rotation speed 500 rpm
  - Cooling no

By creating data records, the machine can later be re-parameterized more quickly.

3.2 Area pointer „Job mailbox“

Application example 1

The PLC can use the job mailbox to transfer jobs to the HMI device to trigger corresponding actions on the HMI device.

In Application Example 1, the job mailbox “69 - Read data record from PLC” is used to implement the job.

Data structure

The first word of the job mailbox contains the job number “69”. Depending on the job mailbox, up to three parameters can be transferred.

Figure 3-1

Word	Most significant byte	Least significant byte
n+0	0	Job number
n+1	Parameter 1	
n+2	Parameter 2	
n+3	Parameter 3	

Note
First, the parameters have to be entered into the job mailbox, followed by the job number.

Job mailbox

Display of parameters for job number “69 - Reading data record from PLC”.

Figure 3-2

69	Reading data record from PLC	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	0: Do not overwrite existing data record 1: Overwrite existing data record

3.3 Export/import of recipes

Application Example 2

In Application Example 2, the data records are first exported from the “master panel” and then imported from the “client panels”.

The following properties have to be considered here:

Specification

In the HMI configuration, three recipes are stored under the following name.

- Recipe 1
- Recipe 2
- Recipe 3

Note The specified names are language-dependent!

During the export of the data records, the language “English” has to be selected at the “master panel”. Information about this can be found here (Link).

Export

Parameter “**ExportDataRecords**”	
Recipe number/name	0
Data record number/name	0
File name	\\computer name\share name\record.csv
Overwrite	Yes
Output status message	On
Processing status	

With these settings, all data records are saved on the specified “Office PC”.

The full file name of the exported data records consists of the specified name under the parameter “file name” and the name of the recipe.

With regards to the example, the names of the exported data records are:

- record_Recipe 1.csv
- record_Recipe 2.csv
- record_Recipe 3.csv

Import

Parameter “**ImportDataRecords**“

File name	\\network drive\share name\ record_Recipe 1.csv
Data record number/name	0
Overwrite	Yes
Output status message	On
Processing status	

During the import of the data records, the system function “ImportDataRecords“ has to be called up for each data record.

With regards to the example, the “file name” is:

- \\network drive\share name\record_Recipe 1
- \\network drive\share name\record_Recipe 2
- \\network drive\share name\record_Recipe 3

4 Mode of operation

This chapter describes the mode of operation and the program of both application Examples.

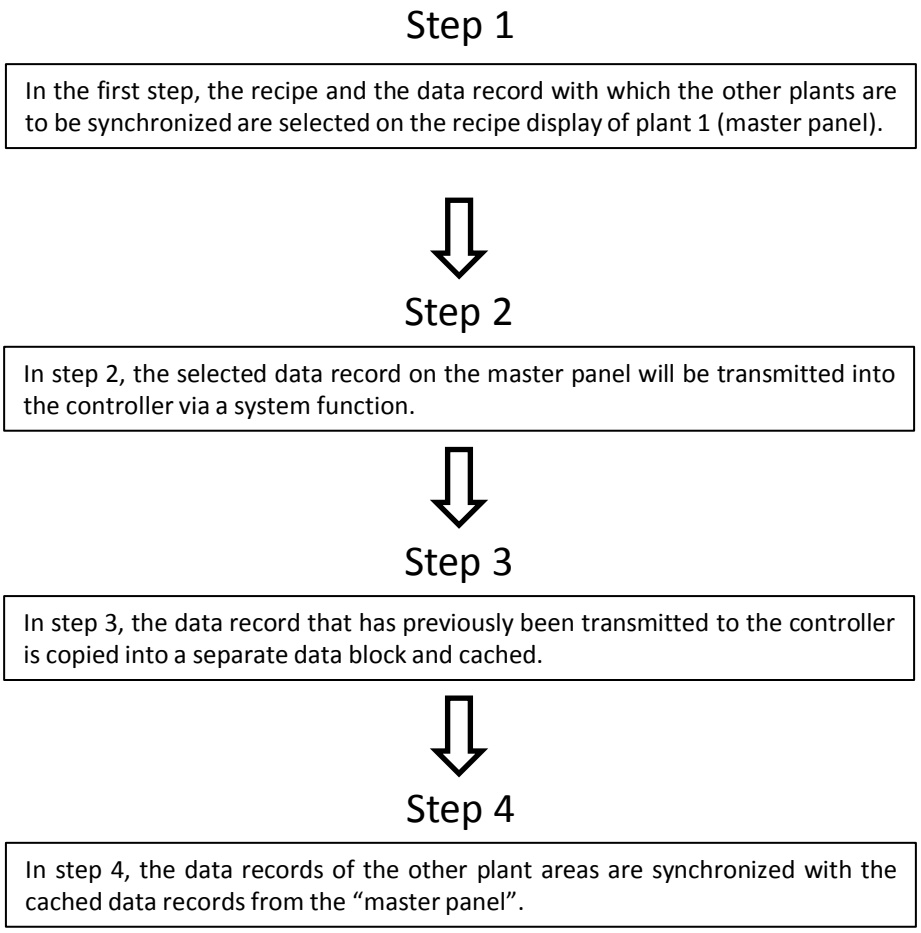
For a better understanding, it is helpful to have the supplied configurations already opened.

If recipe data are mentioned in this chapter, the following information are included:

- Recipe number
- data record number
- data records

4.1 Application Example 1

Figure 4-1



4.1.1 STEP 7 program overview - Example 1

Table 4-1

No.	Block	Description
1.	OB1 Main	Calling of program blocks
2.	FB91 “FB91_BufferingRecipeDataPlant1“	Via the “FB91”, the current recipe data are read and cached in the “DB100”.
3.	FB92 “FB92_SynRecipeDataPlant2“	Via the “FB92”, the data records of plant 2 are synchronized.
4.	FB93 “FB93_SynRecipeDataPlant3“	Via the “FB93”, the data records of plant 3 are synchronized.
Note The structure of FB92 and FB93 are identical. (For each plant area, a FB with this structure is used).		
5.	DB10 “DB10_HMI_AreaPointer_Plant1“	The “DB10” contains predefined area pointers for plant 1.
6.	DB11 “DB11_HMI_AreaPointer_Plant2“	The “DB11” contains predefined area pointers for plant 2.
7.	DB12 “DB12_HMI_AreaPointer_Plant3“	The “DB13” contains predefined area pointers for plant 3.
Note Every plant area uses the area pointer “job mailbox”. (For each plant area, a DB with this structure is used).		
8.	DB91 “DB91_InstanceDB_FB91“	Instance DB of the FB92.
9.	DB92 “DB92_InstanceDB_FB82“	Instance DB of the FB92.
10.	DB93 “DB93_InstanceDB_FB93“	Instance DB of the FB92.
11.	DB100 “DB100_RecipeBuffer“	The “DB100” serves to cache the recipe data.
12.	DB101 “DB101_HMI_Data“	The “DB101” contains the recipe data from plant 1.
13.	DB102 “DB102_HMI_Data“	The “DB102” contains the recipe data from plant 2.
14.	DB103 “DB103_HMI_Data“	The “DB103” contains the recipe data from plant 3.
Note The structure of the data blocks DB101, DB102 and DB103 are identical. (For each plant area, a DB is used).		

Program details

In the following, the functions of the corresponding block are described as well things that should be observed during the configuration.

FB91

Via the “FB91”, the recipe data provided by the “master panel” are copied and cached in a separate data block.

Table 4-2

No.	Description
1.	**Network 1** In network 1, the HMI button to cache the recipe data is evaluated.
2.	**Network 2 and network 3** In the networks, the recipe and data record number is read and cached.
3.	**Network 4** With “BLKMOV”, the data records are cached. The start address of the first data record and the total length of “words” to be copied are specified. The same structure of the two data blocks used facilitates the address assignment.

FB92

Via the “FB92”, the data records of plant 2 are synchronized. For this, the cached recipe data are read and copied into the preset data block with the data records of plant 2.

The job number “69” is required for the values to be transmitted directly into the specified recipe at the HMI operating panel.

“Simply” copying the recipe and data record number into the corresponding data area would also be possible. In that case, however, the data would have to be read into the recipe display manually at the corresponding “client panels”. This contradicts the automatic synchronization job.

Table 4-3

No.	Description
1.	**Network 1** In network 1, the HMI button to synchronize the recipe data is evaluated.

No.	Description
2.	**Network 2** With the “BLKMOV”, the cached data records are copied into the data block for the data records of plant 2. The start address of the first data record and the total length of “words” to be copied are specified. The same structure of the two data blocks used facilitates the address assignment.
3.	**Network 3-6** For the job number “69”, the corresponding values are assigned according to the area pointer. For details on this, see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** “Area pointer „Job mailbox“**Fehler! Verweisquelle konnte nicht gefunden werden.** **Note** Make sure that the job number “69” is called last.

FB93

Function and structure correspond with those of the “FB92”. Only the addresses for “plant 3” need to be adjusted.

To implement the job, an “FB” with this structure is used for every plant area.

DB10; DB11; DB12

The data blocks all have the same structure and contain all available area pointers.

To implement the job, a data block with this structure is used for every plant area.

DB100

The data block contains the recipe data cached by plant 1 (master panel).

DB101; DB102; DB103

The data blocks all have the same structure and contain the recipe data of the corresponding plant.

To implement the job, a data block with this structure is used for every plant area.

4.1.2 HMI configuration, Example 1

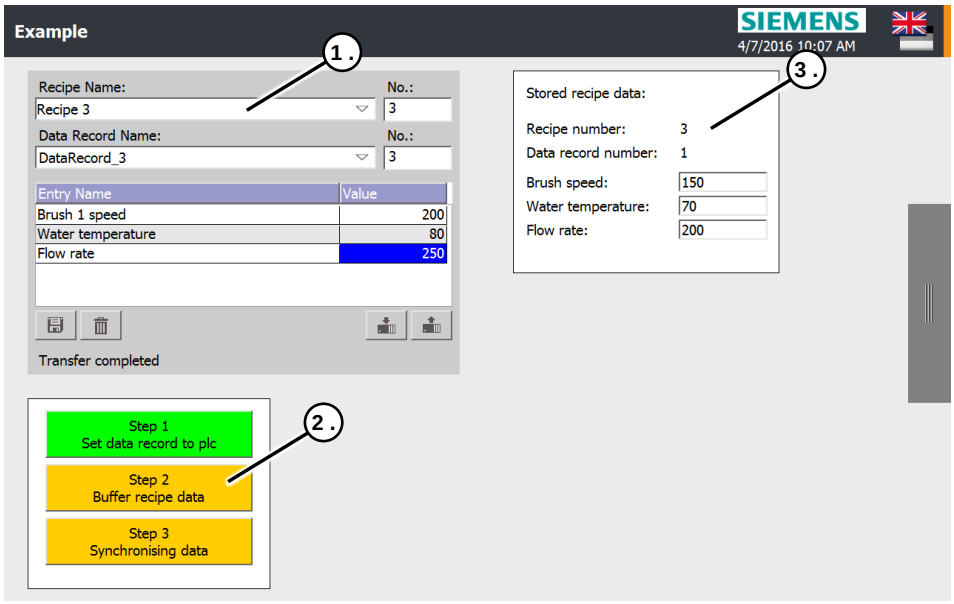
The most essential configuration steps are described below. You can adjust the configuration to your wishes.

Overview “master panel”

The “master panel” contains a “standard recipe display” (1) and three buttons via which the synchronization of the recipe data is triggered (2).

Additionally, the cached values are displayed for monitoring (3).

Figure 4-2



For a better overview, the buttons are provided with an animation. The steps are displayed in “green”, depending on whether they are selectable. The buttons are also interlocked. The evaluation for this is done via the “StepSequence” script.

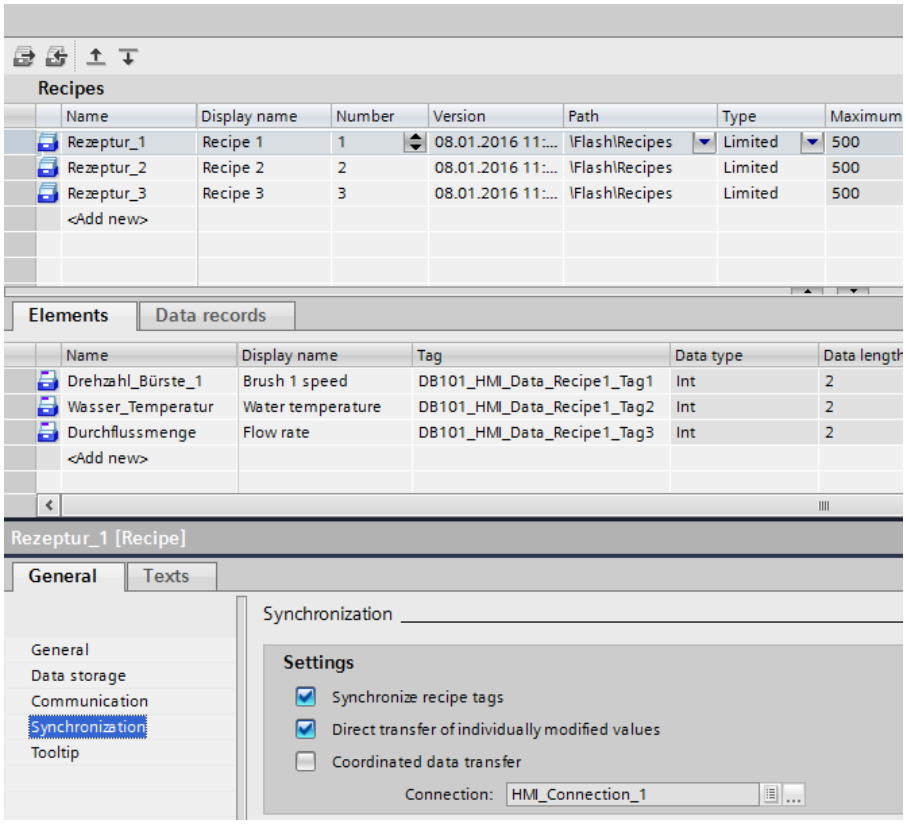
Notes

- The animation/interlocking of the buttons is not mandatory and may be implemented individually - depending on the requirements.
- Via the button “Step 1 Set data record to plc”, the system function “SetDataRecordToPLC” is executed. The same system function can also be executed via the icon in the recipe display.
  The icon in the recipe display is only used in this application, if adjustments are carried out

Recipe

The configuration contains three recipes which three data records each. Please refer to the only help or system manual on how to configure a recipe \4\.

Figure 4-3

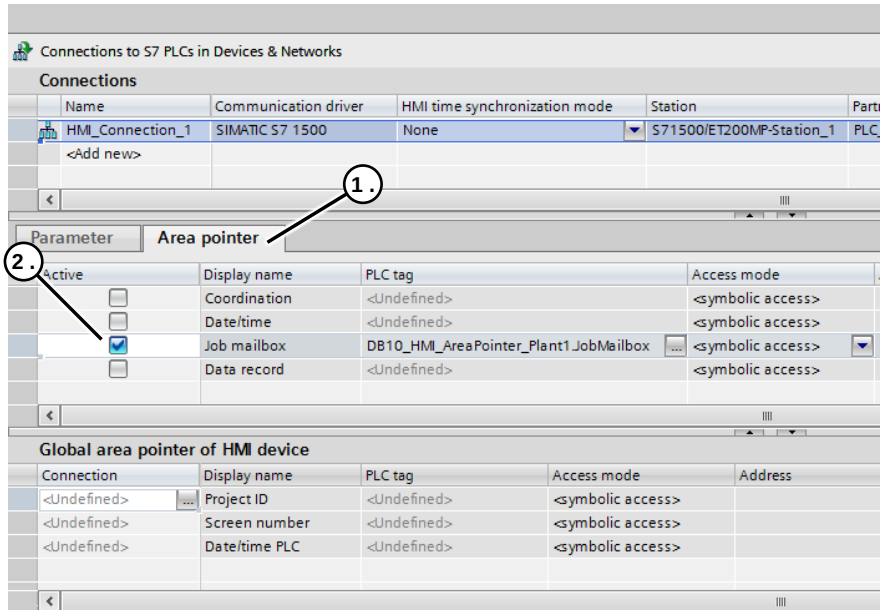


Area pointer

An already configured connection between the operating panel and the plc is required.

Table 4-4

No.	Description
1.	Open the menu “Connections” in the project navigation.
2.	Click on the “Area pointer” button (1).
3.	Activate the option box “Job mailbox” and assign the job mailbox the tag from the data block “DB10_HMI_AreaPointer_Plant1” (2). Further settings are not necessary at this point.

No.	Description
4.	Menu display “Area pointer”. 

Buttons

On all three buttons, under...

- “Properties > Events > Release”, the script “StepSequence” is called.
- “Properties > Animation > Display”, the animation types “usability” and “Design” are used.

Table 4-5

No.	Description
1.	**Button “Step 1...”** Functions used: • “Properties > Events > Press” “RecipeViewSetDataRecordToPLC”
2.	**Button “Step 2...”** Functions used: • “Properties > Events > Press” “SetBit” • “Properties > Events > Release” “ResetBit” Via the “Bit”, the current data record is cached in the plc.

No.	Description
3.	**Button “Step 3...”** Functions used: • “Properties > Events > Press” “SetBit” • “Properties > Events > Release” “ResetBit” Via the “Bit”, the data records are synchronized in the plc.

Tags used

Open the folder “HMI tags” in the project navigation.

You can find the project relevant tags in the folder group “PanelDataSyn”.

Scripts used

Open the folder “Scripts” in the project navigation.

You can find the project relevant script “StepSequence” in the folder group “PanelDataSyn”.

The script serves to interlock the operation of the buttons for synchronizing the recipe data and also supports the animation for the button designs.

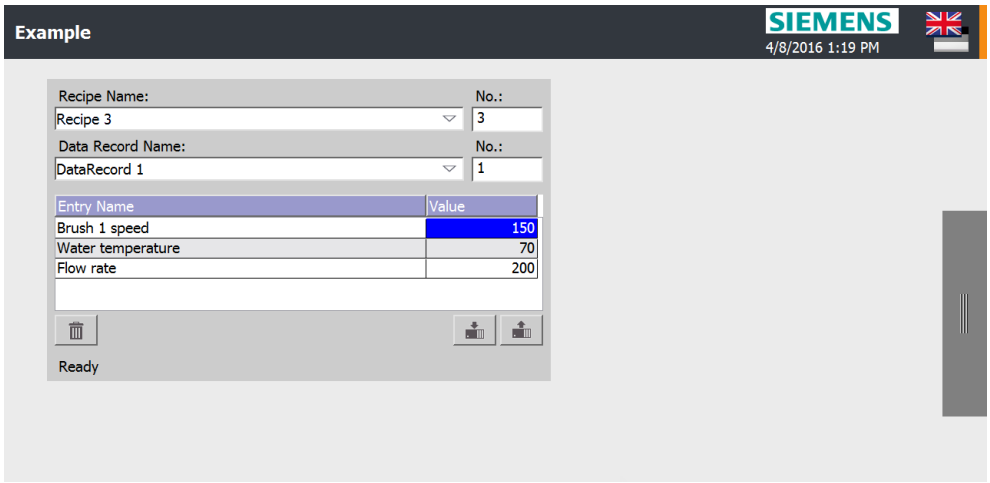
By pressing the button, the script is executed. The “Index No.” is transferred to the script via the parameter. The script is self-explaining.

Note
The interlocking does not necessarily have to be done via the script. It can also be done via the plc program, for example.

Overview “client panel”

The configuration of both “client panels” each contain one “Standard recipe display”.

Figure 4-4



Recipe

The configuration contains the same recipes as those from the “master panel”. Please refer to the only help or system manual on how to configure a recipe \4\.

Note

Make sure that the recipe data are identical with those of the “master panel”.

For this, copy the recipes from the master panel and paste them into the “client panels”. Alternatively, save the recipes of the “master panel” in a project library and then call it up in the “client project”.

After that, only the addresses need to be adjusted.

4.2 Application Example 2

General

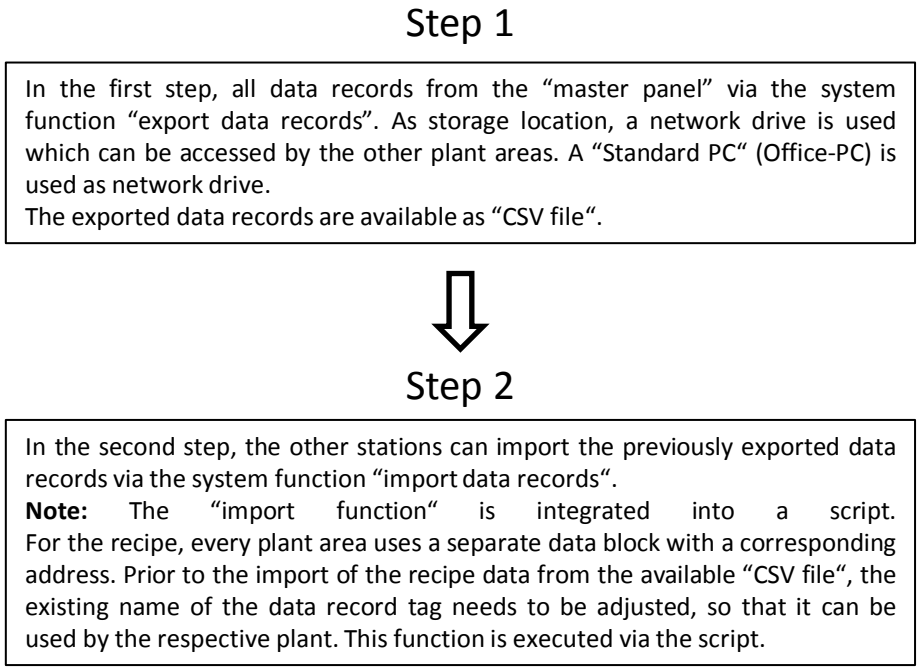
To highlight the function of Application Example 2, a simple and comprehensible application example has been chosen.

Alternatively, the following functions could be of interest.

- Language-independent import / export of data records.
  For this, the script “CopyReplaceData” for example would have to be adjusted accordingly (evaluation of which language is currently set in the panel).
- Import of data records for the “client panels” can be triggered directly via the “master panel”.

There are further options that would have to be technically implemented. To make this part of the manual as concise as possible, a description of alternative functions has been left out.

Figure 4-5



4.2.1 STEP 7 program overview - Example 2

Table 4-6

No.	Block	Description
1.	OB1 Main	Calling of program blocks
2.	FB90 “FB90_SyncRecipeHandling“	The “FB90” serves to evaluate, whether the data records at the other plants are up-to-date.
Note If the recipe data have been exported at the “mater panel”, a graphic indicates that the recipe data need to be synchronized at the other plants.		
3.	DB101 “DB101_HMI_Data“	The “DB101” contains the recipe data from plant 1.
4.	DB102 “DB102_HMI_Data“	The “DB102” contains the recipe data from plant 2.
5.	DB103 “DB103_HMI_Data“	The “DB103” contains the recipe data from plant 3.
Note The structure of the data blocks DB101, DB102 and DB103 are identical. (For a better overview, a DB is used for every plant area).		

Program details

In the following, the functions of the corresponding block are described as well what should be observed during the configuration.

FB90

The “FB90” serves to evaluate, whether the data records at the other plants are up-to-date.

If at the “master panel”, the “export data records” button is activated, this is subsequently used as “index” to indicate that at the other panels, the data records need to be updated.

Via a button on the HMI operating panel, every plant section can import the data records again.
By activating the button, a feedback is sent to the “master panel” confirming that the data records are synchronized again.

Note
It would also be possible to trigger the import of the data records directly from the “master panel”. In that case, it needs to be ensured that the panel is first logged into the network. See also the information on “login dialog” (Link).

Table 4-7

No.	Description
1.	**Network 1** In network 1, the HMI button to export the data records is evaluated.
2.	**Network 2** In network 2, it is evaluated whether the data records of plant 2 need to be synchronized. If the HMI button “Import data records” is activated at plant 2, the signal is reset.
3.	**Network 3** The function corresponds to the previous network 2. Only the addresses for plant 3 need to be adjusted.

DB101; DB102; DB103

The data blocks all have the same structure and contain the recipe data of the corresponding plant.

To implement the job, a data block with this structure is required for every plant area.

4.2.2 HMI configuration, Example 2

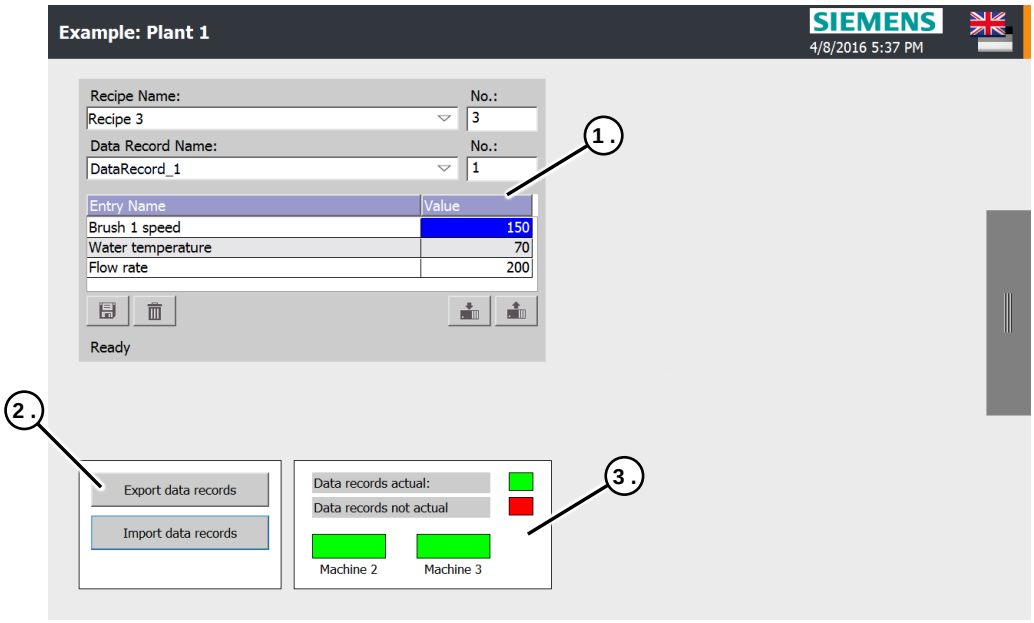
The most essential settings are described below. You can adjust the configuration to your wishes.

Overview “master panel”

The “master panel” contains a “standard recipe display” (1) and two buttons via which the data records can be exported and imported (2).

Via two graphic displays, the status of the other two plants is indicated. If these data records are not up-to-data, the status is shown in “red”. (3)

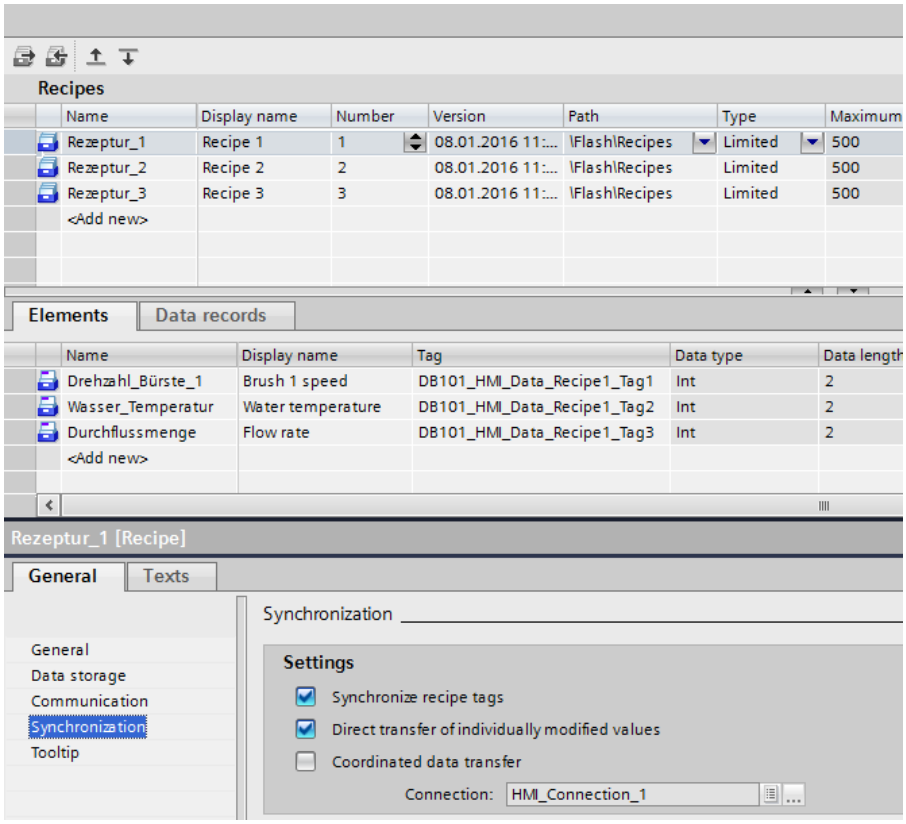
Figure 4-6



Recipe

The supplied configuration contains three recipes which three data records each. Please refer to the only help or system manual on how to configure a recipe \4\.

Figure 4-7

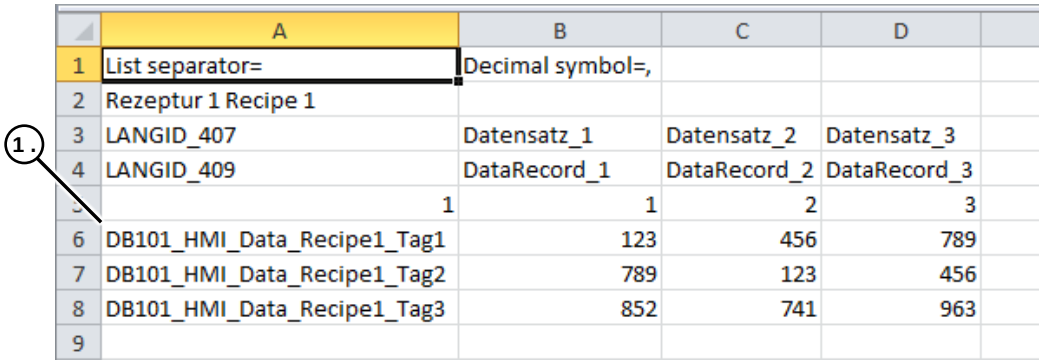


Tips for the recipe names

The “master panel” makes the data records available to the other plant areas. For this, the data records are exported into an “Excel.csv” file.

If you open the exported “Excel.csv”, the names of the individual record tags are displayed in “column A” (1). For comparison, take a look at the configured data records in the configuration.

Figure 4-8



The names of the “data record tags” need to be adjusted first, so that the “client panels” are able to import the “Excel.csv” file.

This renaming is automatically done via the “CopyReplaceData” script.

To keep the “renaming” of the “data records” technically simple, the names of the “data records” should not differ too much.

In this example, the names of the “data record tags” for each plant only differ in their prefixed “data block address”.

=> plant 1: DB101_xxxx

=> plant 2: DB102_xxxx

=> plant 3: DB103_xxxx

Via the script, it would also be possible to change “several” words. Technically, however, this would be more complex. Information on the script can be found here (Link).

A view of the “script parameters” can be found here (Link).

Buttons

Table 4-8

No.	Description
1.	**Button “Export data records”** Functions used: • “Properties > Events > Press” “SetBit” • “Properties > Events > Release” “ResetBit” The “Bit” serves to evaluate that the date records of the other plants need to be synchronized. • “Properties > Events > Press” “ExportDataRecords” **ExportDataRecords** File name \\ComputerName\Sharename\file-name.csv Data record number 0 (All)
2.	**Button “Import data records”** Functions used: • “Properties > Events > Press” “ImportDataRecords” For each data record to be imported, the system function “ImportDataRecords” needs to be called up. **ImportDataRecords** File name \\ComputerName\Sharename\file-name.csv Data record number 0 (All)

No.	Description
3.	**“Rectangles” (above the text machine...)** Functions used: • “Properties > Animation > Display“ “Design” The “Bit” is set via the plc - depending on whether the data of the other plant areas need to be synchronized.

Tags used

Open the folder “HMI tags” in the project navigation.

You can find the project relevant tags in the folder group “PanelDataSyn”.

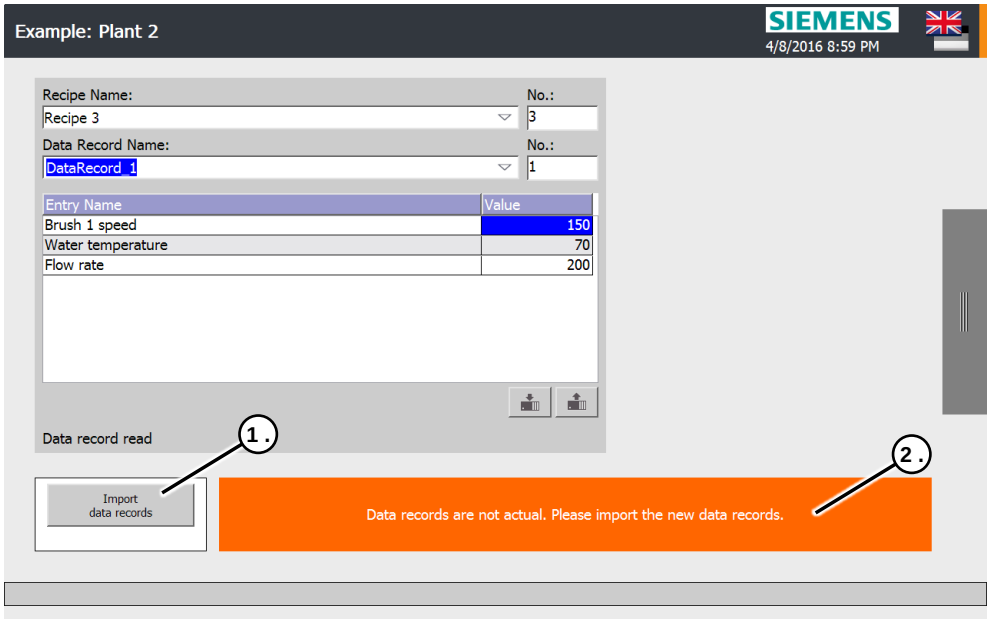
Overview “client panel”

The configuration of both “client panels” each contain one “Standard recipe display”.

Furthermore, the data records can be imported via a button (1).

Via an animated text field, an alarm is displayed if the data records are not up-to-date (2).

Figure 4-9



Recipe

The configuration contains the same recipes as those from the “master panel”. Please refer to the only help or system manual on how to configure a recipe \4\.

Note

Make sure that the recipe data are identical with those of the “master panel”.

For this, copy the recipes from the master panel and paste them into the “client panel”. Alternatively, save the recipes of the “master panel” in a project library and then call it up in the “client project”.

After that, adjust the plant-specific addresses.

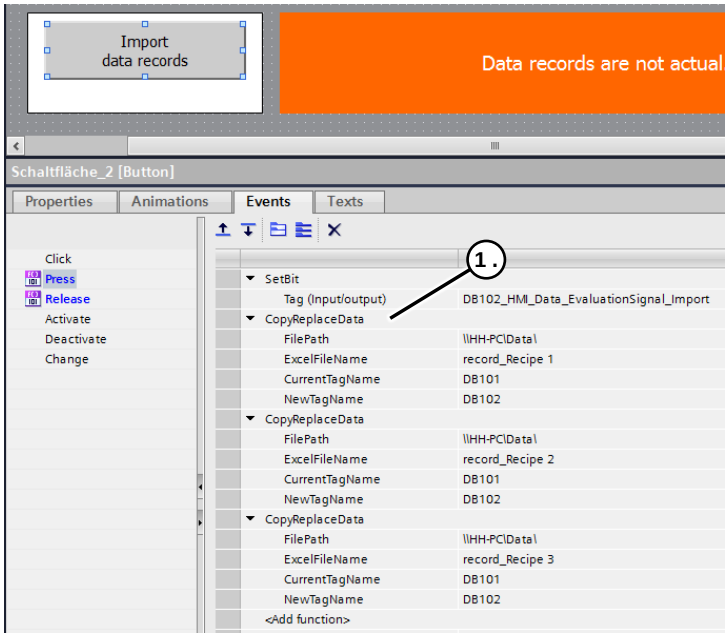
Button

Table 4-9

No.	Description
1.	**Button “Import data records”** Functions used: • “Properties > Events > Press” “SetBit” • “Properties > Events > Release” “ResetBit” The “Bit” serves to evaluate that the date records of the plant have been synchronized. • “Properties > Events > Press” “CopyReplaceData“ For each data record to be imported, the system function “CopyReplaceData” needs to be called up. Information on the function of the scripts can be found here (Link)
2.	**Text field** Functions used: • “Properties > Animation > Display“ “Visibility” The “Bit” is set in the plc - depending on whether the data of the plant need to be synchronized.

Information on the button “Import data records”.

Figure 4-10



Via the button, the script “CopyReplaceData” is called up (1).

The script has 4 parameters

- FilePath
  - Specifying the location where the “Excel.csv” file is stored.
    \\ComputerName\ShareName\
- ExcelFileName
  - The name of the “Excel.csv” file that is to be imported.
    record_Recipe 1
- CurrentTagName
  - The tag name that is to be exchanged.
    DB101
- NewTagName
  - New tag name
    DB102

Note

The tag names for the data records have been selected so as to easily adjust the names.

With regards to the application, only the name element “DB101” (an element of the data record tag name of the “master panel”) needs to be replaced with the intended data record tag name of plant 2. See also the following information on this (Link).

For plant 3, instead of “DB102” the name “DB103” is inserted under the parameter “CurrentTagName”. All other parameters do not need to be adjusted.

Tags used

Open the folder “HMI tags” in the project navigation.

You can find the project relevant tags in the folder group “PanelDataSyn”.

Script “CopyReplaceData”

The names of the “data record tags” need to be adjusted first, so that the “client panels” are able to import the “Excel.csv” file.

This renaming is automatically done via the “CopyReplaceData” script.

Table 4-10

No.	Description
1.	**Mode of operation** Among others, the script has the following two parameters - Current name that is to be exchanged. - New name that is to replace the current name. • In the specified “Excel.csv” file, the script searches for the specified name which is to be replaced and then replaces it with the new name. • With the new names of the data record tags, the “Excel.csv” file is temporarily saved on the disk drive. • Via the system function “ImportDataRecords”, the data records are imported from the temporarily available “Excel.csv”. • Upon completion of the import function, the temporarily available “Excel.csv” file will be automatically deleted.

No.	Description
2.	Program details **Row 12** Declaration of tags. **Row 17** Putting together the file path. IN parameter: FilePath (Storage path). ExcelFileName (name of Excel file). **Row 22, 23** Access to Windows operating system (PC) -> read and write. **Row 28, 29** Opening the exported “Excel.csv” file (1=read access) Opening the exported “Excel_tmp_csv” file (2=write access) **Row 34** The values from the “Excel.csv” file are row by row copied into the “Excel_tmp.csv” file. The copying is performed until the last row in the “Excel.csv” file has been reached. **Row 39** Reading out the current row in the “Excel.csv” file. **Row 44** Replacing the current name with the new name. **Row 49** Writing the new name into the “Excel_tmp.csv” file. **Row 54** Via the command “Wend”, the reading of the individual “Excel rows” is repeated until the command “EOF” (row 34) has the value “True”. The last row in the “Excel.csv” file has been reached. **Row 59, 60** Closing the “Excel.csv” file. Closing the “Excel_tmp.csv” file. **Row 65, 66** Releasing the memory space again. Releasing the memory space again. **Row 75** Executing the system function “ImportDataRecords”. As “Excel source”, the “**Excel_tmp.csv**” file is used. **Row 79 - 80** There is no feedback on when the import of the data records is finished. To make sure that all the individual data records are imported successively, the “counter” is used. **Note** Depending on how many data records are read in,

No.	Description
	the counter value needs to be increased. (Can only be calculated). **Row 86** Access to the Windows operating system (PC). **Row 91** The “Excel_tmp.csv” file used for caching can be deleted again. **Row 96** Releasing the memory space again.

4.3 Archiving to a network drive

For more information on this topic, see the application “Integrating HMI Operator Panels in TCP/IP Networks” under the entry-ID 92346478.

The application describes, how to integrate an HMI operator panel into an existing office / automation network.

Furthermore, terms like \\ComputerName and \ShareName etc. are also explained, as well as what needs to be considered.

5 Operating the application

5.1 General

Overview and description of the user interface

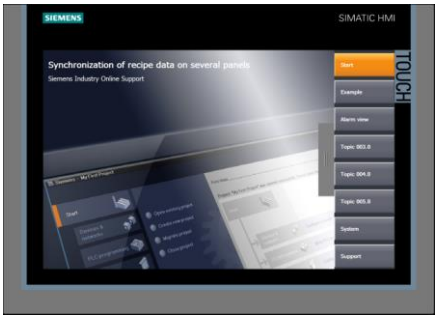
For both application examples, operating the interface is nearly identical as well as intuitive.

Via the image navigation at the side, you get to the application example as well as to the message window and the control panel (RT exit).

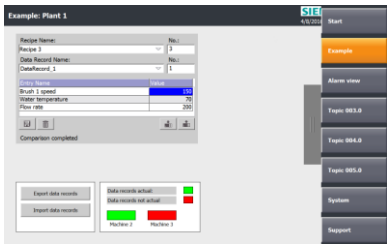
The buttons “Topic 003.0 to Topic 005.0” are freely available.

Figure 5-1

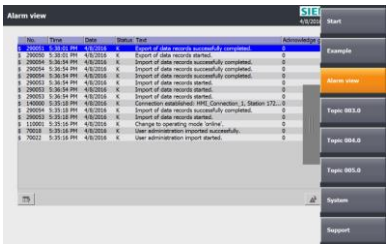
Start page



Example



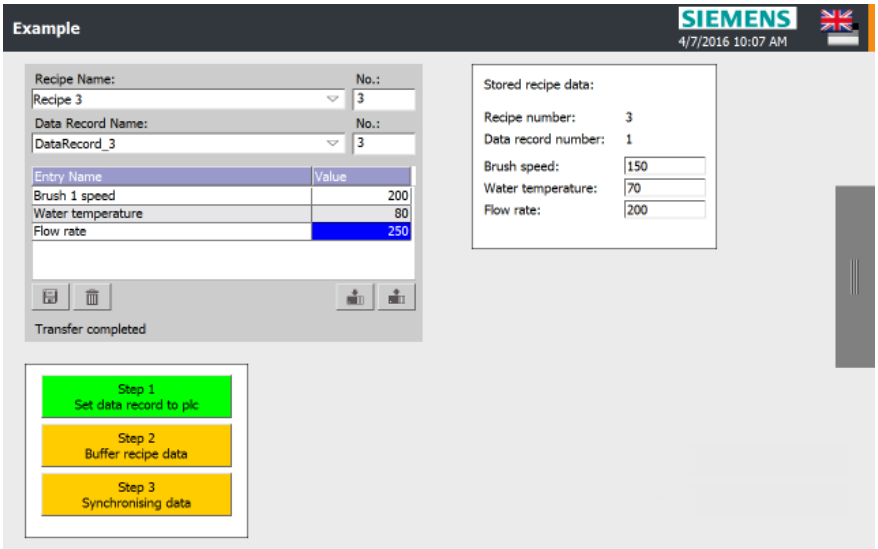
Message

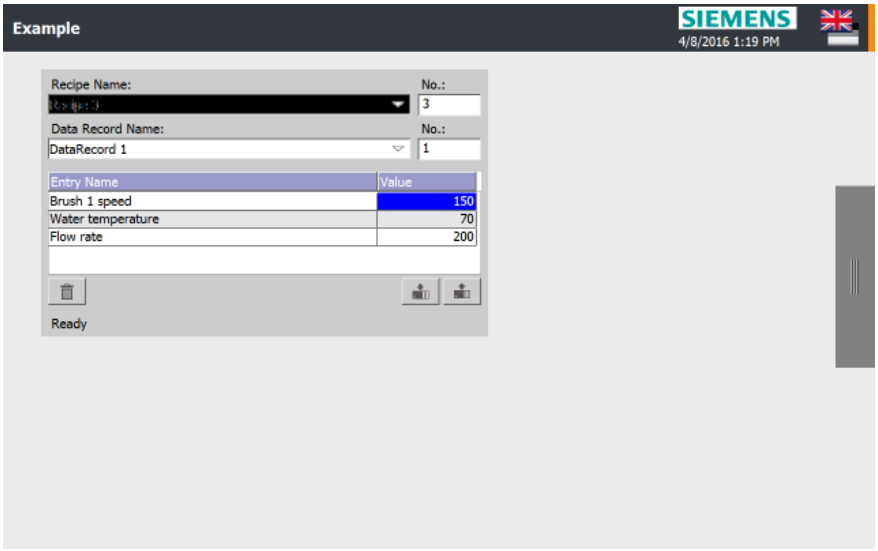


System

5.2 Application Example 1

Table 5-1

No.	Action
1.	**Operating the “master panel”** • Via the recipe display, the recipes and the data records can be selected as usual. • Contrary to the “client panels”, the values for the data record parameters can be adjusted. **Caching data record** • Select the recipe and the data record that is to be cached. • Via the “Step 1” button, the selected data record is written into the plc. • Via the “Step 2” button, the data record that has previously been transmitted to the plc is cached. For monitoring, the values are displayed to the right of the recipe display. • Via the “Step 3” button, the cached value is transmitted to the corresponding plants. This does **not** require that at the corresponding plants, the page with the recipe display is open. **Tip** If the selected data record has not been modified, you can use the “icon” in the recipe display for transmitting the data record into the plc. **Limitation** Only existing data records should be edited whereas new data records should not be added. Although a newly created data record can be transmitted just the same, its name cannot be adopted. It would have to be adapted manually at the corresponding plants. 

No.	Action
2.	**Operating the “client panels”** • Via the recipe display, the recipes and the data records can be selected as usual. • Contrary to the “master panel”, the values for the data record parameters **cannot** be adjusted. The aim is to be able to apply changes to only one place in order to avoid inconsistencies. 

5.3 Application Example 2

General

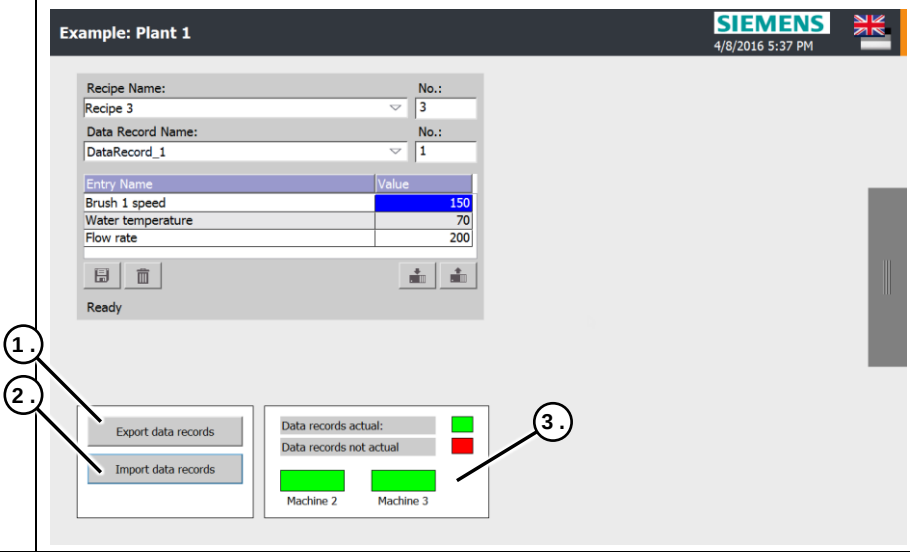
Operating the Application Example 2 is language-dependent.

In order to read in the data record names from the “client panels”, the export of the data records from the “master panel” needs to be performed in the “English” language.

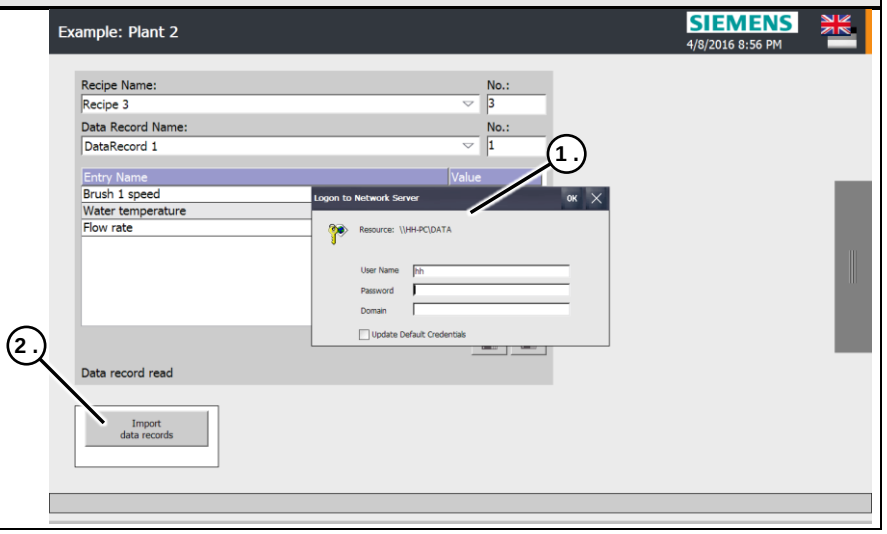
Technically, it would be possible to make the export “language-independent”. However, this would increase the configuration effort accordingly.

Table 5-2

No.	Action
1.	**Operating the “master panel”** • Via the recipe display, the recipes and the data records can be selected as usual. • Contrary to the “client panels”, the values for the data record parameters can be adjusted. **Exporting data records** • Select the button “export data records” (1) to export all data records. **Notes** - Prior to the export, the language at the “master panel” needs to be set to “English”. - If a data record has been modified, it needs to be saved prior to the export. - In order to be able to import the data, there has to be a connection to the “Office PC” and the login to the network had to be successful. If you have not logged into the network yet, a login dialog appears after activating the button, via which you can log into the network (see next table section “login dialog”). **Importing data records** • Select the button “import data records” (2) to import all data records. **Note** In order to be able to import the data, there has to be a connection to the “Office PC” and the login to the network had to be successful.

No.	Action
	Status displays • The status displays (3) show, whether the data records of the other machines are synchronous with those of the “master panel”. 
3.	**Operating the “client panels”** • Via the recipe display, the recipes and the data records can be selected as usual. • Contrary to the “master panel”, the values for the data record parameters **cannot** be adjusted. The aim is to be able to apply changes to only one place in order to avoid inconsistencies. **Login dialog (1)** • If the login data for the network access are **not** permanently stored on the operating panel (Control panel > network ID), the login dialog appears upon activating the “Import data records” button. **Importing data records (2)** • Select the button “import data records” to import all data records. Note In order to be able to import the data, there has to be a connection to the “Office PC” and the login to the network had to be successful.

5.3 Application Example 2

No.	Action
	

6 References

Table 6-1

	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/109736272
\3\	Integrating HMI Operator Panels in TCP/IP Networks https://support.industry.siemens.com/cs/ww/en/view/92346478
\4\	System manual “WinCC Advanced V13.0 SP1” https://support.industry.siemens.com/cs/ww/en/view/109091876

7 History

Table 7-1

Version	Date	Modifications
V1.0	06/2016	First version